MobiCentraList: Software Keyboard with Predictive List for Mobile Device

Georges Badr[1(✉)], Antoine Ghorra[2], and Kabalan Chaccour[1]

[1] TICKET LAB, Université Antonine, Hadat-Baabda, Lebanon
{georges.badr,kabalan.chaccour}@ua.edu.lb
[2] Université Antonine, Hadat-Baabda, Lebanon
antoine.ghorra1@gmail.com

Abstract. Software keyboards were designed to provide accessibility to mobile users as well as to people with motor disability. Text entry is henceforth made possible on portable devices such as mobile phones, tablets, pads. Despite their obvious utility, these keyboards present major drawbacks in terms of speed of acquisition and induced fatigue comparatively to conventional physical keyboards. Optimization efforts have showed efficacy by adding prediction lists and dictionaries. Other researches have considered the effect of the position of characters and the prediction list relatively to the time of acquisition and performance. Those researches were designed for computer software keyboards. In this paper, the position of the prediction list is investigated on a mobile device. The "MobiCentraList" is the mobile version of a previous computer software keyboard "Centralist" which was developed for this purpose. The position effect of the prediction list is studied and compared to natural software keyboards.

Keywords: Software keyboard · Prediction list · Text entry speed · Dictionary · Mobile device

1 Introduction

Nowadays digital texting have become an extremely important activity in our daily life. Beyond writing on hard papers as part of office activities, digital writing is considerably increasing and changing the way we communicate. As a matter of fact, verbal communication which was previously a privilege is leaving more room for the written communication using digital means through the use of short messaging systems (SMS) or even multimedia messaging systems (MMS). This amendment, which uses the digital text entry, is also accompanied by a significant development of portable computing devices. Desktop computers are currently replaced by mobile devices (mobile phone, touch pad, etc.). For portability and mobility reasons, these computing devices are becoming smaller and generally are not equipped with keyboards for digital entry. Hence, alternative methods of text input to replace the standard keyboard has become an insisting need. The software keyboard is presented as an intuitive and natural way to replace the physical keyboard. The user interacts with the system using a pointing device like a mouse, a stylus or even a finger.

M. Kurosu (Ed.): HCI 2016, Part II, LNCS 9732, pp. 268–277, 2016.
DOI: 10.1007/978-3-319-39516-6_25

However, experiments have shown that these onscreen keyboards were less efficient than their physical counterparts (45 words per minute for QWERTY standard keyboard [1], 20 words per minute for the software keyboard with the same layout of the characters [2] and 5 words per minute for a person with a disability entering text on a software keyboard [3]).

The current research examines the performance of a software keyboard developed on a mobile device. The developed keyboard is optimized in terms of character arrangement and prediction list position to increase the efficiency of digital data acquisition. We developed a mobile version of a previous computer software keyboard "CentraList". Performance indicators are also compared to a natural software keyboard that was also developed for this purpose to validate the previous results.

The paper is divided into four sections. A literature review on software keyboards is compiled in the first section. A state of the art on keyboards associated with a predictive list is detailed. In the second section, a new software keyboard model coupled to a prediction list is proposed for mobile devices. The third section presents the evaluation of our model in terms of performance, extraction and analysis of results. The fourth section concludes the paper.

2 Related Works

For portability and mobility reasons, the use of desktop computers follows an inclined road. As a result, physical standard keyboards will no longer exist. The most intuitive alternative to replace theses keyboards is to develop a software keyboard. In terms of input means, the user can hold a pointer, stylus or even use his fingers to enter text. Controversially, experiments carried out by MacKenzie [1, 2] showed the effectiveness of the results achieved by the physical keyboards compared to those obtained on software keyboards. These results were confirmed by Le Pévédic [3] for people with reduced mobility. The prestigious QWERTY or AZERTY arrangements were first introduced to slow down the text entry speed when typing on old typewriters. In fact, the speed caused the character bars of the machines to overlap. This constraint disappeared with the introduction of software keyboards. In this intuitive data entry application, text entry speed must be accelerated. To speed up character entry; optimizations must be made on the software keyboard. The literature reviews many optimizations techniques. These are either changing the shape of the keyboard or coupling it with a prediction system. Mixed techniques are also found.

Some keyboards use a list of fixed words like UKO-II [4]. Other systems, display the prediction list next to the last inserted character like PO-Box [5]. FASTY [6] pops up the list near the writing cursor in order to reduce the pointer's distance. These lists could be used with pointing devices [5, 7], or with a scanning system [8]. These lists are coupled to a word (or text) prediction system. The reader may find a survey concerning on text prediction systems with word lists in [9] (Fig. 1).

The main goal of adding a list of predicted words is to reduce the number of keystrokes and the distance of the cursor. It also aims to increase the speed of writing. But according to [10], the text entry rate is still not optimized regardless of the increase

in the keystroke saving rate. In all cases, the user has to move his cursor between the keyboard and the list. Moreover, in cases like PO-BOX, the list is covering an important part of the keyboard. This may charge additional cognitive effort on the user. Those cognitive and perceptual efforts result in tiredness of the user and make him loose his focus on both the keyboard and the list.

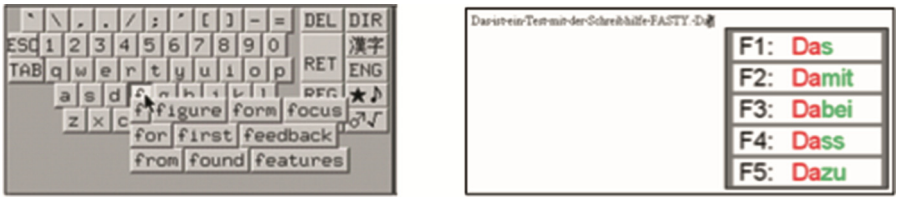

Fig. 1. On the left, PO-BOX interface: when selecting a character, the list appears directly on it. On the right, FASTY interface, the list appears next to the cursor.

In this context, the most intuitive or natural character configuration is the alphabetical order. While this arrangement does not reduce the distance and the cursor movement, it has the advantage of being known by everyone. In [11], Norman and Fisher distributed the letters on several lines (Fig. 2A). Based on their experiments, alphabetical keyboards are not more efficient than QWERTY keyboards. MacKenzie [1] proposes to reduce the keyboard in two lines containing 13 letters each (Fig. 2B). Their experiments do not show any difference in the text entry rate. Neither MacKenzie nor Norman advises the alphabetical arrangement of keyboards. The text entry rate remains lower than with an ordinary keyboard because of the great distance between characters.

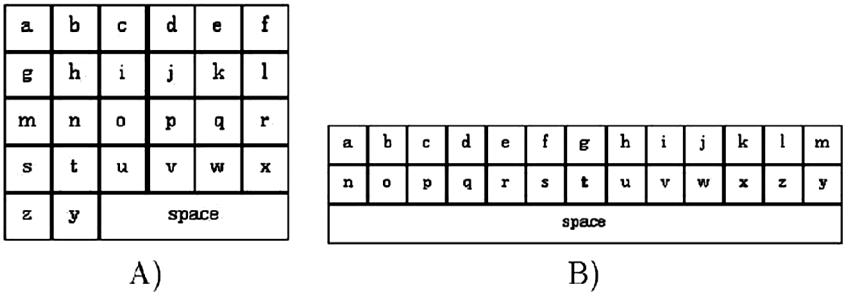

Fig. 2. Software keyboards with alphabetic arrangement

Far from the natural and intuitive solutions, we can find the methods that apply optimization algorithms. These methods compute a set of opportunities and keep only the best. The first system designed with such a method is the Getschow keyboard in [12]. The keyboard uses the construction algorithm (also named “greedy algorithm”) and the frequency of occurrence of letters in the English language. The algorithm rearranges the character buttons to minimize average distances between them. Other proposed systems such as the GAG in [13] are based on genetic algorithms to generate a new keyboard

design with an arrangement of characters that theoretically increase the speed of text entry. The proposed system would theoretically increase the speed of entering text with a stylus almost 50 % compared with QWERTY keyboards and soft AZERTY [13]. Others like Metropolis and Hooke [14] apply the laws of physics and thermodynamics to bi-gram tables to reduce the distances between the most frequently consecutive letters.

As simple as it may seem, creating a new keyboard arrangement would make the user lose his focus and eye gaze from both the keyboard and the list. This would dramatically slow the text entry rate and speed, and increase the tiredness of the user.

In the mobile world, systems couple their keyboard with a prediction algorithm in order to optimize text entry. Word completion or a list of words is presented to the user, but interaction is still the same as on desktop computers and thus results are the same (Fig. 3).

Fig. 3. Integration of list of predicted words on different mobile platforms

The present paper focuses on this problem of interaction. A mobile software keyboard model allied with a list of predicted words is proposed. The MobiCentraList is a mobile version of the previously designed software keyboard "Centralist" [15].

The previous study showed improvements in terms of speed and use of the list, as well as improvements in reducing the distance travelled by the cursor. Thus, the main objective was to validate the results obtained with CentraList on the desktop with MobiCentraList developed for mobile.

3 MobiCentralist

MobiCentraList is the new name of the mobile version of the CentraList. The latter was originally designed to place the prediction list in the center of attention of the user. The advantage is to keep the attention of the user in one place without hiding the keyboard characters. As shown in Fig. 4, letters were located around the prediction list in a square-like shape. CentraList keyboard was developed for desktop computers. MobiCentraList inherits the same properties of the desktop version. It aims to reduce the user's tiredness by minimizing the distance between the list and keys.

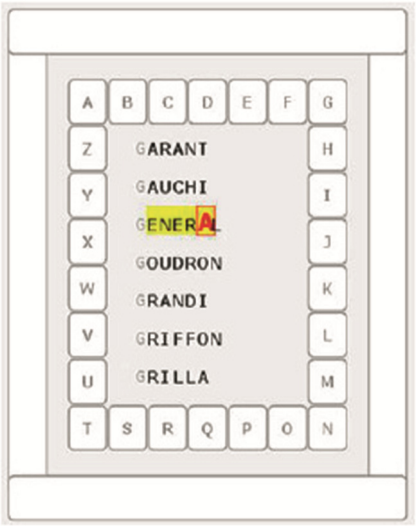

Fig. 4. CentraList desktop interface

3.1 Design

As shown in Fig. 5, Character keys are distributed alphabetically so the user will never lose the reference. Letters surround the list of predicted words, in a square like shape. The keyboard has 4 space character buttons, each on a side of the keyboard, as it is the most selected character. This configuration allows the user to select the nearest space to the last input letter.

Fig. 5. MobiCentraList mobile interface

3.2 Experiments

To prove our hypothesis, MobiCentraList should be evaluated. The main idea is to validate the results of CentraList on a different platform.

Participants. Eight people (six men and two women) participated in this experiment. They were all volunteers aged between 21 and 43. The participants were all regular users of mobile and desktop QWERTY keyboards. They do not use the prediction list.

Apparatus and Equipment. The program was developed with JAVA for mobile using Android Studio 4.0.1 platform running on a Toshiba laptop with Windows 10. The program was then deployed on a 7 inches Android Tablet of the brand "ELEMENTS".

Evaluation. MobiCentraList is compared to a standard alphabetical keyboard with a list of prediction words (cf. Fig. 6). Both systems are limited with the 26 characters of the Latin alphabet and the space. They both have the same interactive list designed in [16] and use the same prediction algorithm based on the lexicographical tree. Same dictionary is also used by both systems. Participants had two exercises. They had to copy 40 words, with lengths greater than 5 letters, on both keyboards. The words were the same for the two exercises. The order of the exercises was counterbalanced. The first subject starts his first exercise with MobiCentraList then switches to the standard alphabetical keyboard. The second participant starts with the alphabetical keyboard then MobiCentraList and so on.

Fig. 6. Alphabetical keyboard interface.

Procedure. Participants were asked to copy as quickly as possible and with the least possible errors the list of 40 words using one finger only. At the end of each typed word, the user pressed the space bar to validate his entry and move to the next word and the list is re-initialized to its original state. Experimentations data were collected in an "XML" file when typing. These data are the words entered, the characters clicked, the coordinates (x, y) of the characters, the time taken to click the character and time of each word.

Results. The following variables were computed: the input speed (CPS), the distance between two consecutive strokes, the error rate, the key stroke per character (KSPC), and the hit rate (HR).

CPS. The input speed or character per second (CPS) is the number of characters entered per unit of time (e.g. second). This speed is obtained by dividing the number of characters (N) by the time. The entry time is usually taken between the first character to enter and the last character. In this case, N − 1 is the total number of character.

$$\text{CPS} = \frac{\text{Number of characters entered} - 1}{\text{time}} \text{(Error! Bookmark not defined.)}$$

The analysis of the results (Fig. 7) shows that the users have gained speed while typing on MobiCentraList compared with the alphabetic keyboard. In fact, the number of characters clicked per one second has increased.

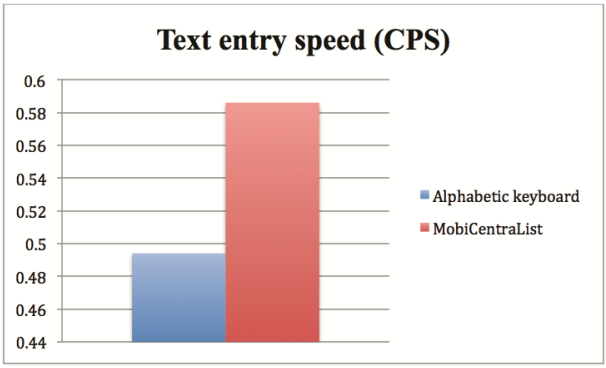

Fig. 7. Text entry speed for both keyboards

On average, CPS has increased from 0.494 with alphabetic keyboard to 0.586 with MobiCentraList showing an average gain of 18.6 %.

KSPC and Hit Rate. KSPC is the number of operations required to enter a word or character. If no list is present, each character requires one click on the keyboard (KSPC is equal to one). When the list is present, and the user selects the word from it, the number of operations needed is reduced, thus, KSPC is less than one.

$$\text{KSPC} = \frac{\text{Number of characters entered}}{\text{Total number of characters}} \text{(Error! Bookmark not defined.)}$$

Hit Rate represents the percentage of times the user selects his word from the prediction list. The results of this experiment showed that MobiCentraList has drawn user's attention to the list and thus he is using the list further. In fact, the use of the prediction list has increased by 8.3 % when copying words with the new system.

The frequent use of the list is shown by the decrease of the KSPC's value. This means that the user requires fewer operations to enter a word. The result shows a significant reduction in the average number of operations needed to enter a character from 0.831 actions on the alphabetic keyboard to 0.756 actions on MobiCentraList (a decrease of 9.9 %). In comparison, a standard software keyboard requires an action by character (Fig. 8).

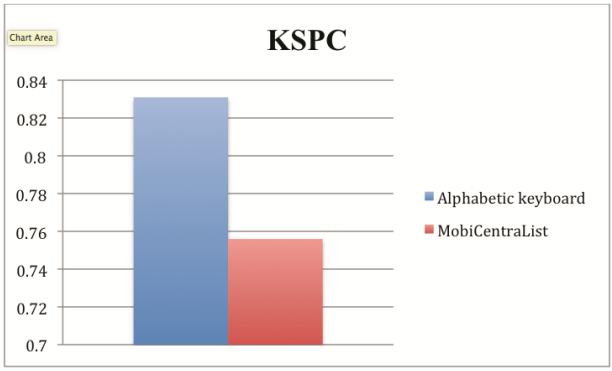

Fig. 8. KSPC for both keyboards

Distance between Two Strokes. This measures the distance of the path travelled by the finger between letters themselves and between letters and the prediction list. As far as the distance is reduced, the user's fatigue is reduced, thus, the performance of the user is increased.

Our hypothesis was that when placing the prediction list in the center of the keyboard, this would lead to a reduction in the distance between two strokes. The results of the experiments confirmed that hypothesis. Thus, we obtained a reduction in term of distance the user has to do between two clicks. In fact, the pointer (finger in our case) travelled an average distance of 1638.6 pixels with the classical keyboard and an average distance of 1484.97 pixels on MobiCentraList. This will give the user a benefit of about 9.3 %.

Error Rate. The error rate is the percentage of typos made when entering a number of words. Whenever a mistake is made, the system records it. During the exercises, error was recorded when the current character differed from the expected character. In average, the error rate was less than 3 % for both keyboards. (1.9 % for the classical keyboard and 1.8 % with MobiCentralist).

4 Conclusion

In this article, we presented a software keyboard for mobile operating systems with a list of predictions (MobiCentraList). This paper was to confirm the results obtained with the same design implemented on a desktop computer (CentraList). The keyboard is constituted with the 26 alphabetic letters positioned around a list of prediction. By this

arrangement, the user may focus his eye-gaze on the center of the keyboard, and select the word from the list once it appears. By this, he will improve his text entry speed and reduce the number of operations he needs to enter a word, thus reduce his fatigue. We also evaluated the system and compared it with a classical alphabetical keyboard. The results showed a benefit for the users in terms of speed and reduction of the distance. Those results confirmed the ones we obtained using the desktop version.

References

1. Mackenzie, I.S., Zhang, X.S.: The design and evaluation of a high performance soft keyboard. In: Proceedings of the ACM Conference on Human Factors in Computing Systems. ACM Press, New York, pp. 25–31 (1999)
2. Mackenzie, I.S., Zhang, X.S., Soukoureff, R.W.: Text entry using soft keyboards. Behav. Inf. Technol. **18**, 235–244 (1999)
3. Le Pévédic, B.: Prédiction morphosyntaxique évolutive dans un système d'aide à la saisie de textes pour les personnes handicapées physiques. Rapport de thèse, Université de Nantes, France (1997)
4. Harbusch, K., Hasan, S., Hoffmann, H., Kühn, M., Schüler, B.: Domain-specific disambiguation for typing with ambiguous keyboards. In: Proceedings of the 2003 EACL Workshop on Language Modeling for Text Entry Methods. ACL Workshops. Association for Computational Linguistics, Morristown, NJ, pp. 67–74 (2003)
5. Masui, T.: An efficient text input method for pen-based computers. In: Proceedings of the ACM Conference on Human Factors in Computing Systems, CHI 1998 (1998)
6. Beck, C., Seisenbacher, G., Edelmayer, G., Zagler, W.L.: First user test results with the predictive typing system FASTY. In: Miesenberger, K., Klaus, J., Zagler, W.L., Burger, D. (eds.) ICCHP 2004. LNCS, vol. 3118, pp. 813–819. Springer, Heidelberg (2004)
7. Keystrokes on-screen keyboard, user manual. http://www.assistiveware.com/files/KeyStrokesXmanualA4.pdf
8. Wandmacher, T., Antoine, J.Y., Poirier, F., Départe, J.P.: SIBYLLE, an assistive communication system adapting to the context and its user. ACM Trans. Access. Comput. **1**(1), 1–30 (2008)
9. Garay-Vitoria, N., Abascal, J.: Text prediction systems: a survey. Univ. Access Inf. Soc. **4**(3), 188–203 (2006). Springer-Verlag
10. Newell, A.F., Booth, L., Beattie, W.: Predictive text entry with PAL and children with learning difficulties. Br. J. Educ. Technol. **22**(1), 23–40 (1991)
11. Norman, D.A., Fisher, D.: Why alphabetic keyboards are not easy to use: keyboard layout doesn't much matter. Hum. Fact. **24**, 509–519 (1982)
12. Getschow, C.O., Rosen, M.J., Goodenough-Trepagnie, C.: A systematic approach to design of a minimum distance alphabetical keyboard. In: Proceedings of Rehabiliation Engineering Society of North America ñ RESNA 9th Annual (1986)
13. Raynal, M.: Claviers GAG: claviers logiciels optimisés pour la saisie de texte au stylet. In: Proceedings of the 18th International Conference of the Association Francophone d'Interaction Homme-Machine. ACM (2006)
14. Zahi, S., Hunter, M., Smith, B.A.: The Metropolis Keyboard - an exploration of quantitative techniques for virtual keyboard design. In: Dans Actes de The 13th Annual ACM Symposium on User Interface Software and Technology (UIST), pp. 119–218, San Diego, California (2000)

15. Badr, G., Raynal, M.: Centralist. In: European Conference for the Advancement of Assistive Technology in Europ, Maastricht, 31/08/2011–02/09/2011. IOS Press, pp. 944–951, September 2011
16. Badr, G., Raynal, M.: WordTree: results of a word prediction system presented thanks to a tree. In: Stephanidis, C. (ed.) UAHCI 2009, Part III. LNCS, vol. 5616, pp. 463–471. Springer, Heidelberg (2009)