

Ambiguous Keyboards and Scanning: the Relevance of the Cell Selection Phase

Julio Miró-Borrás, Pablo Bernabeu-Soler, Raul Llinares, and Jorge Igual

Department of Communications
Universidad Politécnica de Valencia, Plaza Ferrandiz-Carbonell s/n,
03801 Alcoy, Spain
{jmirobo, pbernabe, rllinares, jigual}@dcom.upv.es

Abstract. This paper focuses on the relevance of the cell selection phase in the overall performance of a text entry system based on scanning and with an ambiguous keyboard. Most of the layouts are designed trying only to minimize the ambiguity of the keyboard, and taking into consideration only the disambiguation process when entering text. Nevertheless, the number of scan cycles necessary for selecting the cells has great importance in the overall performance. As we show, the performance depends on the number of cells and the linguistic model used in the cell selection phase.

Keywords: Text Entry, Scanning, Ambiguous Keyboards, Disambiguation.

1 Introduction

Keyboards with fewer keys¹ than characters (ambiguous keyboards) can be used instead of scan matrices in text entry systems based on scanning in order to increase the text entry rate. Nevertheless, most proposals try to create layouts that minimize the ambiguity of the keyboard based only on the disambiguation process. This is the standard way when considering physical keyboards such as the ones used in mobile phones or small devices. MacKenzie *et al.* [1] compared most of the ambiguous keyboards proposed by other researchers in terms of KSPC (keystrokes per character). Although some of these keyboards work with scanning, they were all treated the same way: direct access. An ambiguous keyboard with indirect access and character-level disambiguation is presented in [2], [3]. Leshner *et al.* [4] proposed several ambiguous keyboards that minimize the ambiguity using what is called the confusability matrix. All these proposals consider only the disambiguation process, without analyzing the effect of scanning the different cells on the overall performance.

This paper describes the research carried out with a single-switch text entry system using an ambiguous virtual keyboard and a word-level disambiguation process. Some of the current ambiguous layouts proposed by researchers are analyzed in our scanning system concluding that the best physical ambiguous layouts are not the best layouts when are used with scanning.

¹ We use cells instead of keys when talking about virtual keyboards accessed through scanning.

The remainder of this paper is organized as follows. In the next section we explain the methods employed in the research, specifically the mathematical model and the layouts to be analyzed. Next, the results are presented and analyzed and finally, some conclusions are drawn.

2 Methods

The research is done using the proposal in [3]. It consists of a scanning system with ambiguous keyboard and a word-level disambiguation method. Instead of using a separate key for space and next function, it uses a combination of automatic and inverse scanning, and short and long keypresses as follows. The operation of this proposal is composed of two phases. The first one is the cell selection phase. Using direct scanning, the user presses the switch when the desired cell is highlighted. This is done for all the characters in a word except the last one. For the last letter, the user presses and keeps the switch pressed, indicating to the system that the word is finished. Then, the disambiguation phase starts, presenting the suggested words one after another using inverse scanning. When the desired word is shown, the user releases the switch and the word followed by a space is entered into the system.

In this study, we used the same word-frequency list derived from the British National Corpus and described by Silverberg *et al.* [5], with the 9022 most common words in English. In the first phase, we obtained the probabilities of all characters for each position in a word, considering all the words in the list. Then, for each position, we obtained the key probabilities considering all letters in that key. In the second phase, we used the 9022 word list.

In a scanning environment, both the action of pressing the switch as well as the cursor movement are traditionally considered as keystrokes. A parameter that comprises all kinds of keystrokes is n_W or average weighted number of keystrokes per word (Equation 1), where $n_W(w)$ is the weighted number of keystrokes in word w , and $p(w)$ its probability. Additionally, w_S , w_C , and w_L are the weights for the scan cycles, the short and the long presses of the switch respectively, and n_S , n_C and n_L are the average values of the respective number of keystrokes.

$$n_W = \sum_w n_W(w) \cdot p(w) = w_S \cdot n_S + w_C \cdot n_C + w_L \cdot n_L. \quad (1)$$

When calculating the keystrokes values, n_L and n_S keep constant, so the best layouts are those that minimize the number of scan cycles. When considering physical layout performance, the main parameter is KSPC [1] or keystrokes per character (Equation 2), where K_W is the number of keystrokes needed for entering word w , F_W is the frequency of w , and C_W is the number of letters in w .

$$KSPC = \frac{\sum (K_w \times F_w)}{\sum (C_w \times F_w)}. \quad (2)$$

In our research we evaluate the layouts of other keyboards presenting the lowest value of KSPC for a fixed number of keys. These are: L2K, UKO-II, TMK4, L6K, Mobile Phone, and QP10213.

3 Results

We present in this section the number of scan cycles for the best physical layouts with 2, 3, 4, 6, 8 and 9 keys, i.e., those that present the smaller value of KSPC for the predictive method of disambiguation. The values for the number of short and long keystrokes for all layouts are: $n_{C1}=3.50$, $n_{L1}=1$, $n_{C2}=1$ and $n_{L2}=0$ in both phases.

In Table 1 we present the number of scan cycles in phase 1 (n_{S1}), phase 2 (n_{S2}) and total (n_{ST}) for the known layouts. The last column shows the KSPC. In Table 2 we present the values obtained for other layouts with alphabetical arrangement of letters with better performance when used with our scanning system. The value of KSPC is also shown in the same table.

Table 1. Analysis of the best current keyboards.

Layout	Keys	n_{S1}	n_{S2}	n_{ST}	KSPC
QP10213	9	10.82143	0.02337	10.84480	1.00431
Phone	8	10.41355	0.03479	10.44835	1.00641
L6K	6	6.11822	0.16165	6.27987	1.02979
TM4K	4	5.22093	0.27777	5.49869	1.05118
UKO II	3	3.74514	0.54090	4.28605	1.09967
L2K	2	2.06310	2.96924	5.03235	1.54712

4 Discussion

The value of KSPC for physical keyboards working in predictive disambiguation mode, decreases generally when increasing the number of keys. The reason is that the ambiguity decreases when less characters share the same cell. As we can see in Table 1, the best physical layout is QP10213 with a KSPC of 1.00431. Nevertheless the number of scan cycles is 10.84, much greater than UKO-II, which needs only 4.29 scan cycles with only 3 keys. KSPC for UKO-II is 1.09967, i.e., it requires 8.7% more keystrokes than QP10213 when considering the disambiguation phase, but saves 153.0% scan cycles.

The data in Table 1 shows that the keyboards with smaller KSPC are not the best keyboards when using scanning. In Table 2, we propose other keyboards with an alphabetical ordering of letters on the keys. They permit us to compare pairs of keyboards with the same number of keys. As we can see, all keyboards in Table 2 except AK2 present larger values of KSPC than their equivalents in Table 1. And what is more interesting, all keyboards present a smaller number of scan cycles. The highest differences are found in the 8-key devices: the AK8 needs 5.1% more keystrokes than the Phone keyboard, but it saves 51.8% scan cycles. Moreover, AK3 needs 15.1% more keystrokes than UKO-II, and saves 15.1% scan cycles. Notice that

L2K presents a smaller number of scan cycles than AK2 but with a greater KSPC. In general, we can state that with higher number of keys, the savings in keystrokes are smaller than with keyboards with 2 or 3 keys, but the savings in scan cycles are just the opposite. This is because with more cells, the average number of scan cycles for selecting them increases, while the ambiguity is smaller.

Table 2. Proposal of several alternative keyboards (AK) with a smaller number of scan cycles.

Layout	Keys	Layout	n_{s1}	n_{s2}	n_{sT}	KSPC
AK9	9	a-i, jk, lm, no, pq, rst, uv, wx, yz	5.35468	0.28484	5.63952	1.05249
AK8	8	a-i, jk, lm, no, pq, r-v, wx, yz	4.70206	0.32906	5.03112	1.06063
AK6	6	a-i, j-l, m-o, p-t, u-w, x-z	4.46755	0.34948	4.81703	1.06440
AK4	4	a-j, k-n, o-v, w-z	3.37419	0.63304	4.00723	1.11665
AK3	3	a-g, h-t, u-z	2.16769	1.59956	3.76724	1.29474
AK2	2	a-l, m-z	2.13221	2.93368	5.06589	1.54057

Traditional studies with ambiguous keyboards to be used with scanning try to get the best layouts for the disambiguation phase, and many times with a large number of keys [5]. Nevertheless, as we have shown here, it is more efficient to consider the cell selection phase in the design of the layout. The relevance of the cell selection phase in the total number of scan cycles depends on the linguistic model being used. The results presented in this work are obtained using a 1-gram model. With a fixed scanning pattern, the differences would be even greater, and with more elaborated prediction systems, the differences would be smaller. Only in the case of perfect prediction in phase 1, will the layouts with smaller KSPC have the best performance when using scanning.

Acknowledgments. This study has been partly funded by Cátedra Telefónica - UPV.

References

1. MacKenzie, I.S., Tanaka-Ishii, K.: Text entry using a small number of buttons. In: Text Entry Systems: Mobility, Accessibility, Universality, pp. 105--121 Morgan Kaufmann (2007)
2. Miró, J., Bernabeu, P.A.: Text entry system based on a minimal scan matrix for severely physically handicapped people. In: ICCHP 2008. LNCS, vol. 5105, pp. 1216--1219. Springer-Verlag (2008)
3. Miró-Borrás, J., Bernabeu-Soler, P.: Text Entry in the E-Commerce Age: Two Proposals for the Severely Handicapped. *J. theor. appl. electron. commer. res.* 4, 101--112 (2009)
4. Leshner, G.W., Moulton, B.J., Higginbotham, D.J.: Optimal Character Arrangements for Ambiguous Keyboards. *IEEE Trans. Rehab. Eng.* 6, 415--423 (1998)
5. Silfverberg, M., MacKenzie, I.S., Korhonen, P.: Predicting text entry speed on mobile phones. In: SIGCHI Conference on Human Factors in Computing Systems, pp. 9--16. (2000)