Comparing a Scanning Ambiguous Keyboard to the On-screen QWERTY Keyboard

Chris T. Waddington
University of Central Lancashire
Preston UK PR1 2HE
chris.waddington123@gmail.com

I. Scott MacKenzie
York University
Toronto Canada M3J 1P3
mack@cse.yorku.cz

Janet C. Read
University of Central Lancashire
Preston UK PR1 2HE
jcread@uclan.ac.uk

Matthew Horton
University of Central Lancashire
Preston UK PR1 2HE
mplhorton@uclan.ac.uk

This paper explores text entry on a scanning ambiguous keyboard (SAK) and the Windows on-screen keyboard (OSK) operating in scanning mode. The SPACEBAR was used for physical input with both keyboards. Testing involved 12 participants entering five phrases of text with each keyboard. On entry speed, the means were 5.06 wpm for the SAK and 2.67 wpm for the OSK, thus revealing a significant speed advantage for the SAK. However, the character-level error rate of 13.3% for the SAK was significantly higher than the error rate of 2.4% for the OSK. On subjective preference, 7 of 12 participants preferred the Windows OSK over the SAK, citing familiarity with the QWERTY layout as the most common reason. However, participants appreciated the efficiency of the SAK keyboard. A limitation of the results is the small amount of text entered.

Scanning ambiguous keyboard (SAK), Windows on-screen keyboard (OSK), text entry, accessible interfaces
.

1. INTRODUCTION

Inputting text with a keyboard is an everyday task for humans, be it through a standard keyboard on a desktop or laptop computer, or on a touch-based keyboard on a tablet or mobile phone. Whilst alternative keyboard layouts, such as Dvorak, have been proposed, none has challenged QWERTY as the default keyboard standard.

Most people enter text using physical or soft keyboards and make full use of their hands during input. Accessible interfaces are for users with physical disabilities. Keyboard entry in an accessible interface requires a pointing action, be this with a mouse, head pointer, or eye gaze. With this default mode of entry, the design of the keyboard is more critical for users with less than full motor capacity (Shih et al., 2011). A number of disabilities have more limited options in terms of input, including amyotrophic lateral sclerosis (ALS) (a motor-neurone disease) where users may gaze at boards displaying letters in groups, with input tracked using an eye tracker (Revell, 2017). Eye tracking itself is considered useful for individuals with particular diseases that strongly limit motor abilities (Porta, 2015).

One approach for users with limited movement is to use scanning keyboards or keyboards that use prediction to minimise effort. The redundancy of the English language is well suited to predictive systems.

In this paper we first look at the evolution of keyboard design, and then examine the efficiency and effectiveness of two keyboards designed primarily for users with impaired motor function. A user study with able-bodied users was conducted, as is common practice in initial investigations of accessible interfaces. Comparisons are made between two scanning keyboards: an optimised scanning ambiguous keyboard with four keys (SAK) and the Windows on-screen keyboard with a QWERTY layout (OSK).

2. BACKGROUND

The QWERTY keyboard (seen in Figure 1) is ubiquitous but it is also inefficient.

Figure 1: Standard QWERTY keyboard

There is debate about the motivation in its design, with Baker (2010) suggesting the design was intended to keep typing speeds down, stating that Sholes "rejigged the letters on his machine". David (1985) claimed the design was in response to a problem, where keys "clash and jam if struck in rapid succession" (p. 333). The QWERTY keyboard's design is "not optimal from a physiological standpoint" (Rempel, 2008, p. 338) and, in addition, the keys are not grouped logically

(Ciobanu, 2014). Anderson et al. (2009) commented that alternative keyboards "offer ergonomic benefits over the conventional, single plane QWERTY keyboard designs" (p. 35), although these alternative keyboards lack widespread acceptance.

A variation on QWERTY is the split keyboard (Figure 2) which evolved from Kroemer's work in the 1960s (Rempel, 2008). A variation on the split keyboard is the half-QWERTY keyboard which was designed to facilitate the transfer of two-handed typing skill to one hand (Matias et al., 1993).
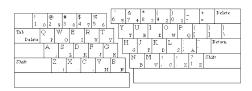

Figure 2: Half-QWERTY keyboard layout

The best-known alternative to QWERTY is the Dvorak keyboard, patented in 1936 by August Dvorak and William Dealey (Dell'Amico et al., 2009; Purewal, 2015). Dvorak's keyboard emphasised improving typing speed by placing the most common letters in the home row (Tenebaum, 1996; Dell'Amico et al., 2009).

2.1 Optimized Keyboards

The Dvorak keyboard used a low level of optimisation to speed up text entry. Once stylus-based or touch-based "soft keyboards" became possible, there was a flurry of designs that optimised according to language models. The OPTI keyboard, designed using a Fitts-digram model (Figure 3), was shown to be about 25% faster than a QWERTY layout after a few hours of practice with stylus entry (MacKenzie & Zhang, 1999).

Figure 3: The OPTI keyboard layout

The FITALY keyboard is another optimised keyboard, designed such that more frequent letters were located in the centre of the keyboard, thus improving input speed. The ATOMIK keyboard was similarly created and used hexagonal keys to assist in entry (Zhai et al., 2002).

Keyboards optimised in this way are ideal for users who operate keyboards with one finger or a stylus but not for users for whom effort of movement is a limiting factor.

2.2 Assistive Keyboards

A number of keyboards have been specifically designed to improve accessibility, particularly for individuals who are unable to use their hands due to a physical disability, such as cerebral palsy (CP) or locked-in syndrome (Ashtiani & MacKenzie, 2010; MacKenzie & Ashtiani, 2011).

Individuals with motor disabilities find using a standard keyboard with a pointing device, such as a mouse, difficult (Pouplin et al., 2014). Poláček et al. (2011) note that physically disabled users are often unable to achieve high speed text entry due to their motor constraints and this creates a deficit in communication experiences: They cannot keep up in a conversation and they get frustrated.

Several input devices, such as alternative keyboards, electronic pointing devices, or touch screens, are commonly used by individuals with disabilities (Conderman, 2015). Particular stimuli can also be used for operating a device. Examples include eye tracking, physical buttons, joysticks, and EEG signals (Poláček et al., 2011).

Assistive keyboards and related tools use additional methods to meet the needs of persons with physical disabilities. These methods include humming (Poláček et al., 2011) or a brain-computer interface (BCI), as discussed by Millán et al. (2010), Thompson et al. (2009), and Mayaud et al. (2013). Additionally, assistive keyboards were designed to suit people with a particular disability. For example, Microsoft's *MouseKeys* is for users who struggle or are unable to use a mouse (Shih et al., 2011), and *GazeSpeak* is an application designed for individuals suffering from ALS (Revell, 2017). The latter uses eye movement to convert text into speech.

2.2.1 On-screen keyboard

One commercially available keyboard for users with disabilities is the Microsoft on-screen keyboard (OSK). See Figure 4. There are two sources of input, a mouse (for non-disabled users) and a switch (where scanned input is used). In addition, Evans and Blenkhorn (2003) mention that "the system produces keystrokes that need to be routed to the application" (p. 222).

Using scanning and switch input, the OSK uses the SPACEBAR for physical input. Scanning proceeds row by row at a default rate of one scan step per second. When the row bearing the desired letter is highlighted, the user presses the SPACEBAR. Scanning then enters the row in groups of keys, depending on the row. With the next press of SPACEBAR, scanning enters the group and proceeds key by key. A final press of SPACEBAR

selects the key and sends the corresponding letter or command (e.g., BKSP) to the currently active application. When a letter (or BKSP) is selected, scanning restarts, row by row.

Figure 4: Windows OSK

2.2.2 Scanning ambiguous keyboard

A research alternative is the scanning ambiguous keyboard (SAK) (MacKenzie, 2009; MacKenzie & Felzer, 2010). The SAK uses a predictive text model and combines the most demanding requirement of scanning (input using one key or switch) with the most appealing feature of an ambiguous keyboard (one key press per letter).[1] The default implementation includes four soft keys and uses the SPACEBAR for input. See Figure 5. Scanning proceeds in a cyclic fashion from key to key. Letters are group alphabetically across the first three keys.

Figure 5: SAK implementation

To enter a word, the user presses the SPACEBAR when the key bearing the desired letter is highlighted. As letters are entered, a collection of candidate words is presented in a word region (bottom of Figure 5). After the last letter in a word is entered, the user presses the SPACEBAR when the fourth key is highlighted; scanning then enters the word region. When the desired word is highlighted, the user presses the SPACEBAR again to select the word. A SPACE is automatically appended. Errors are corrected by a long press on the SPACEBAR,

---

[1] The best known ambiguous keyboard is the telephone keypad which contains three or four letters on each key. A disambiguating method, commonly known as *T9*, uses a language model to allow the entry of text using one keypress per letter.

which removes either the last keycode (if in the middle of a word) or the last word (if between words).

A novel feature of SAK is multiple-letter-selection, whereby multiple selections can be performed in a single scanning interval if two or more consecutive letters in a word are on the same key. For example, all the letters in *limp* appear on the second key (see Figure 5). Thus, input is possible simply by pressing the SPACEBAR four times in succession when the second key is highlighted. Importantly, the scan timer restarts when the SPACEBAR is pressed, thus providing additional time for additional selections.

With SAK, the candidate word list is organised in two parts, with the first presenting complete words matching the current keycode sequence (ordered by frequency in the language) and the second presenting extended words where the keycode sequence is treated as a word stem. SAK's use of word completion reduces the number of scan steps required to provide a performance gain (Anson et al., 2006). For common words, the user need only enter part of the word; when the full word appears in the word region, it can be selected.

An early experiment with the prototype SAK showed entry speed improved over five blocks, from 3.98 wpm to 5.11 wpm (MacKenzie, 2009; MacKenzie & Felzer, 2010). One participant did extended trials and achieved a rate of 9.28 wpm on the 10th block. Error rates were low at 0.96%.

3. METHOD

Given the familiar layout of the Windows OSK and the potential improvement in performance of the SAK, a study was designed to compare the two choices of keyboard.

3.1 Participants

Twelve able-bodied participants located at a local university campus in the UK took part in the experiment. This sample size is common for keyboard comparison experiments (e.g., MacKenzie, 2009; Bi et al., 2012).

3.2 Apparatus

Testing was done on desktop computers running the *Windows 10* operating system. The OSK is part of the operating system. The SAK was installed via an online link, which included experiment software with built-in test phrases, data collection, etc.[2] Phrases were selected at random for input. This was done automatically for the SAK condition and manually for the OSK condition. In the latter case,

---

[2] http://www.yorku.ca/mack/ExperimentSoftware/

the phrases were presented in Microsoft *Notepad* and participants entered the transcribed phrase on the line below the presented phrase. Entry was timed manually using a stopwatch. Error rates were computed after the experiment via a script that compared the presented phrases with the transcribed phrases.

The scanning interval of the SAK remained the same throughout at 700 ms, which is in contrast to other experiments with this keyboard, where the scanning speed increased during the experiment (MacKenzie, 2009).

3.3 Procedure

Participants were first explained the experiment goals. Then, a demonstration was given on the first keyboard tested. Following this, the participant entered a set of five phrases. This procedure was then repeated for the second keyboard. Half the participants first used the SAK, half first used the OSK. At the end of testing, each participant was asked which keyboard they preferred and why.

3.4 Design

The experiment was a 2 x 5 within-subjects design with the following independent variables and levels:

    Keyboard: OSK, SAK

    Phrase: 1, 2, 3, 4, 5

The dependent variables were entry speed (wpm) and error rate (%). The total number of trials was 12 Participants x 2 Keyboards x 5 Phrases = 120.

4. RESULTS

The main results of interest are the performance measures for entry speed and accuracy (viz. error rates). Learning effects over the five phrases of input are also of interest as are participants' subjective impressions of the two keyboards.

4.1 Entry Speed

The grand mean for entry speed was 3.86 wpm. The SAK mean was 5.06 wpm, which was about 89% higher than the mean of 2.67 wpm for OSK. See Figure 6. The difference was statistically significant ($F_{1,10} = 70.8$, $p < .0001$).

These results suggest a considerable performance advantage for the SAK. As well, participants improved more with SAK, as seen in Figure 7 which shows the progression of entry speed over the five phrases of input. From the first to the fifth phrase, the improvement was 22% for SAK, with a fifth-phrase mean of 5.68 wpm. For OSK, the first-to-fifth improvement was a modest 5%, with a fifth phrase mean of 2.68 wpm. The effect of phrase on entry speed was statistically significant ($F_{4,40} =$

3.18, $p < .05$). However, the keyboard by phrase interaction effect was not statistically significant ($F_{4,40} = 2.44$, $p > .05$).
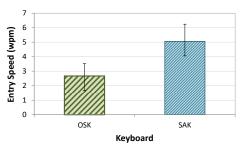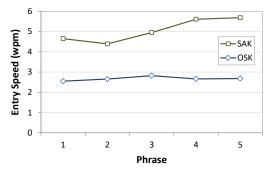
Figure 6: Entry speed (wpm) by keyboard

Figure 7: Entry speed (wpm) by keyboard and phrase

4.2 Error Rate

Error rates were computed using the minimum-string distance metric which reflects the character-level difference between the presented phrase and the transcribed phrase. The grand mean for error rate was 7.86%. The results were quite different between keyboards, with the results more favourable for the OSK (2.4%) than for the SAK (13.3%). See Figure 8. The difference was statistically significant ($F_{1,10} = 11.7$, $p < .01$).
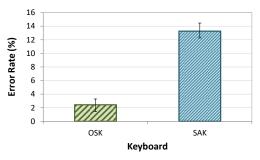
Figure 8: Error rate (%) by keyboard

The high error rate for SAK was likely due to the choice of 700 ms for the scanning interval. The initial evaluation of SAK noted in Section 2.2.2 used a scanning interval of 1100 ms in the first block as a means to help ease participants into the operation of SAK. The scanning interval was reduced by 100 ms on each subsequent block, to 700 ms on the fifth block. Using 700 ms throughout

testing in the present study was perhaps a poor choice and a likely reason for the high error rate.

More errors on the OSK tended to appear in the first phrase, where the error rate was higher than 5%. There was less consistency with the accuracy on the SAK. The fewest errors when inputting text with the SAK were made on the first phrase, but then the error rate peaked at its highest point on the next phrase. These trends are seen in Figure 9 which shows the progression of error rates by phrase. However, statistical significance did not appear for the phrase effect ($F_{4,40} = 1.10$, $p > .05$) nor for the keyboard by phrase interaction effect ($F_{4,40} = 1.45$, $p > .05$).
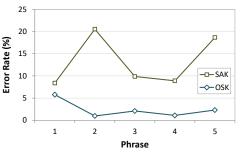

Figure 9: Error rate (%) by keyboard and phrase

4.3 Keyboard Preference

Overall, the results for subjective preference were relatively close, with 7 of 12 participants preferring the Windows OSK to the SAK. Participants preferring the OSK cited reasons such as "familiar QWERTY layout", "easier to understand", "more efficient for selection", and "errors easier to correct". Participants preferring the SAK cited reasons such as "more efficient", "alphabetic layout", and "easier to use".

4.4 Discussion and Limitations

One reason for the higher entry speed with SAK is multiple-letter-selection, as described in Section 2.2.2. With the Windows OSK, however, the user can only input one letter in each scanning sequence.

As well as being slower, the Windows OSK saw a lack of improvement over the five phrases, unlike the SAK. Entry speed with the OSK generally stayed under 3 wpm throughout testing. The layout of the Windows OSK was a QWERTY layout. Ciobanu (2014) highlighted several drawbacks of the QWERTY layout. One that is relevant here is that the standard QWERTY layout was designed to slow down the typing speed, although the nature of the Windows OSK – entering just one letter per scan sequence – seemed to have a bigger impact on the results.

Certain participants were better able to understand the keyboards even before the initial demonstration was completed, whilst others needed extra help. It must be noted that none of the participants were expert users, especially with the SAK (although this keyboard produced the faster entry speed).

A few participants could input phrases with the SAK in under a minute, but this did not happen with any of the participants on the OSK. One option to improve the efficiency of scanning keyboards is to deviate from the QWERTY layout and position common letters near the beginning of the scanning sequence (MacKenzie, 2012). However, non-QWERTY or non-alphabetic letter arrangements require additional time to learn.

The are several limitations to the research described herein. Only five phrases of text were input for each keyboard; thus, the potential for learning was limited. The scanning interval for testing the OSK remained at the default value of 1000 ms. Thus, the OSK had an inherent speed disadvantage compared to the SAK, which had a scanning interval of 700 ms.[3] This is one possible reason for both the higher speed and the lower accuracy with the SAK. However, it should be noted that, irrespective of the scanning interval, more keystrokes and more scan steps are required for entering characters on the OSK.

5. CONCLUSION

A text entry experiment was carried out with 12 participants with a series of interesting results. Text entry speed was higher for the scanning ambiguous keyboard (SAK), but the Windows on-screen keyboard (OSK) proved more accurate.

It appears that with continued practice, the SAK could be a good choice for individuals with motor disabilities wanting to achieve good input speed.

Future work will involve a lengthier study and will include a case study with individuals with motor disabilities.

6. REFERENCES

Anderson, A. M., Mirka, G. A., Joines, S. M. B., & Kaber, D. B. (2009) Analysis of alternative keyboards using learning curves. *Human Factors, 51*, 35-45.

Anson, D., Moist, P., Przywara, M., Wells, H., Saylor, H., & Maxime, H. (2006) The effects of word completion and word prediction on typing rates using on-screen keyboards. *Assistive Technology, 18*, 146-154.

---

[3] The alert reader will recognize scanning interval as a confounding variable, since it varied across the levels of the independent variable keyboard (high for the OSK, low for the SAK).

Ashtiani, B., & MacKenzie, I. S. (2010) BlinkWrite2: An improved text entry method using eye blinks. *Proc ETRA 2010*, pp. 339-345. ACM, New York.

Baker, N. (2010) Why do we all use Qwerty keyboards? Available at: http://www.bbc.co.uk/news/technology-10925456 (Accessed 18/7/2016).

Bi, X., Smith, B. A., & Zhai, S. (2012) Multilingual touchscreen keyboard design and optimization. *Human-Computer Interaction, 27*, 352-382.

Ciobanu, O. (2014) An evaluation of the ergonomics of the QWERTY keyboards. *Applied Mechanics and Materials, 657*, 1051-1055.

Conderman, G. (2015) Assistive technologies: A lifeline for learning. *Kappa Delta Pi Record, 51*, 173-178.

David, P. A. (1985) Clio and the economics of QWERTY. *The American Economic Review, 72*, 332-337.

Dell'Amico, M., Díaz, J. C. D., Iori, M., & Montanari, R. (2009) The single-finger keyboard layout problem. *Computers & Operations Research, 36*, 3002-3012.

Evans, G., & Blenkhorn, P. (2003) Architectures of assistive software applications for Windows-based computers. *J Network and Computer Applications, 26*, 213-228.

MacKenzie, I. S. (2009) The one-key challenge: Searching for a fast one-key text entry method, *Proc ASSETS 2009*, pp. 91-98. ACM, New York.

MacKenzie, I. S. (2012) Modeling text input for single-switch scanning, *Proc ICCHP 2012*, pp. 423-430. Springer, Berlin.

MacKenzie, I. S., & Ashtiani, B. (2011) BlinkWrite: Efficient text entry using eye blinks. *UAIS, 10*, 69-80.

MacKenzie, I. S., & Felzer, T. (2010) SAK: Scanning ambiguous keyboard for efficient one-key text entry, *ACM Trans Computer-Human Interaction, 17*, 11:1-11:39.

MacKenzie, I. S., & Zhang, S. X. (1999) The design and evaluation of a high performance soft keyboard. *Proc CHI '99*, pp. 25-31. ACM, New York.

MacKenzie, I. S., Zhang, S. X., & Soukoreff, R. W. (1999) Text entry using soft keyboards. *Behaviour & Information Technology, 18*, 235-244.

Matias, E., MacKenzie, I. S., & Buxton, W. (1993) Half-QWERTY: A one-handed keyboard facilitating skill transfer from QWERTY. *Proc INTERCHI '93*, pp. 88-94. ACM, New York.

Mayaud, L., Filipe, S., Pétégnief, L., Rochecouste, O., & Congedo, M. (2013) Robust brain-computer interface for virtual keyboard (RoBIK): Project results. *IRBM, 34*, 131.

Millán, J. D. R., Rupp, R., Müller-Putz, G. R., Murray-Smith, R., Giugliemma, C., Tangermann, M., Vidaurre, C., Cincotti, F., Kübler, A., Leeb, R., Neuper, C., Müller, K.-R., & Mattia, D. (2010) Combining brain-computer interfaces and assistive technologies: State-of-the-art and challenges. *Frontiers in Neuroscience, 4*, 161.

Poláček, O., Míkovec, Z., Sporka, A. J., & Slavik, P. (2011) Humsher: A predictive keyboard operated by humming. *Proc ASSETS 2011*, pp. 75-82, ACM, New York.

Porta, M. (2015) A study on text entry methods based on eye gestures. *J Assistive Technologies, 9*, 48.

Pouplin, S., Robertson, J., Antoine, J., Blanchet, A., Loup Kahloun, J., Volle, P., Bouteille, J., Lofaso, F., & Bensmail, D. (2014) Effect of dynamic keyboard and word-prediction systems on text input speed in persons with functional tetraplegia. *J Rehabilitation Research & Development, 51*, 467-479.

Purewal, S. J. (2015) Why should you try an alternative keyboard layout. Available at: http://www.cnet.com/uk/how-to/why-you-should-try-an-alternative-keyboard-layout/ (Accessed 25/7/2016).

Rempel, D. (2008) The split keyboard: An ergonomics success story. *Human Factors, 50*, 385-392.

Revell, T. (2017) Microsoft app helps people with ALS speak using their eyes*.* Available at: https://www.newscientist.com/article/2121579-microsoft-app-helps-people-with-als-speak-using-just-their-eyes (Accessed 18/2/2017).

Shih, C.-T., Shih, C.-H., & Luo, C.-H. (2011) Development of a computer assistive input device through a commercial numerical keyboard by position coding technology for people with disabilities. *Disability & Rehabilitation: Assistive Technology, 6*, 169-175.

Tenebaum, D. (1996) Dvorak keyboards: The typist's long-lost friend. *Technology Review, 99*, 21.

Thompson, D. E., Baker, J. J., Sarnacki, W. A., & Huggins, J. E. (2009) Plug-and-play brain-computer interface keyboard performance*. Proc NER 2009,* pp. 433. IEEE, New York.

Zhai, S., Hunter, M., & Smith, B. A. (2002) Performance optimization of virtual keyboards. *Human-Computer Interaction, 17*, 229-269.