Exploring an Ambiguous Technique for Eyes-Free Mobile Text Entry

Dylan Gaines
Michigan Technological University
Houghton, MI, USA
dcgaines@mtu.edu

ABSTRACT

Mobile text entry has become an increasingly important part of many peoples' daily lives. While most input occurs through individual letters being tapped on a virtual QWERTY keyboard, this does not have to be the case. We explore how well users are able to learn an ambiguous keyboard that is modeled after a standard QWERTY layout but does not require users to tap specific keys. We show that this keyboard is a plausible text entry technique for users with little or no vision, with users achieving 19.09 Words per Minute (WPM) and 2.08% Character Error Rate after 8 hours of practice.

Author Keywords

Accessibility, blind, visually impaired, text entry

INTRODUCTION

For persons with little or no vision, accurate text entry on a mobile device can be a daunting task. Some solutions, such as Apple's VoiceOver, are location-dependent, but have additional feedback and require multiple touch events for confirmation [2]. Others remove location altogether in favor of chorded Braille entry methods. These include BrailleType [5], Perkinput [1], TypeInBraille [3], and BrailleTouch [6]. However, statistics released in 2009 by the National Federation of the Blind show that less than 10 percent of legally blind Americans can read Braille [4].

Vertanen [7] proposed a method that removed the need to know Braille patterns in favor of an ambiguous model based on the number of fingers in a tap. Taps ranging from one to five fingers signified groups of characters that could be based on a variety of mappings. With Tap123, we extend this work by creating an input method using this theory that generates mappings from a QWERTY keyboard layout instead of Braille.

ASSETS 2018, October 22–24, 2018, Galway, Ireland

ACM ISBN 978-1-4503-5650-3/18/10.
https://doi.org/10.1145/3234695.3240991

System Description

The Tap123 keyboard accepts taps of between one and three fingers, which corresponds to the row of a QWERTY keyboard that the intended letter is in. Unlike in [7], the side of the screen that the average of the fingers falls on is used to determine the side of the keyboard. A left swipe can be used to backspace a tap, while a right swipe sends the tap sequence to the VelociTap [8] decoder for recognition and inserts a space. If a left swipe is performed immediately after decoder recognition, it will instead delete the entire word. For example, to type the word 'ran', the user would tap first with one finger on the left side, then with two fingers on the left side, and finally with three fingers on the right side before swiping to the right to begin decoder recognition.

As in [7], the VelociTap decoder is configured to treat each combination of finger count and side as a key with multiple possible letters. The decoder is configured to return a list of the 6 most likely words, which we will refer to as the N-Best list, based on the taps entered and the left context. If the first word returned is not the word the user intended, swiping up or down on the keyboard will iterate forwards or backwards through N-best list, respectively.

With the absence of visual feedback, Tap123 instead provides audio feedback to convey information to the user. When any tap is performed, the keyboard speaks the letters that correspond to that tap. Upon recognition or an up or down swipe, the keyboard speaks the word that has been selected. If a word or tap is deleted, the keyboard will inform the user what they deleted. A long press (600ms or longer) will queue the keyboard to read both the prompt and the contents of the entry text field.

USER STUDY

Our experiments consisted of a series of sessions designed to introduce participants to the entry method slowly and allow them to learn the techniques. Though all 4 participants were sighted, the device on which entry occurred was obscured from the participants' views during all text entry tasks. In each session, participants were asked to type given phrases using Tap123. These phrases were read to the participants using Android's native Text-to-Speech. Participants completed 8 sessions, each totaling about 40 minutes of text entry broken into about 10-

Session	Sample Reference Text
1	leaves
2	are you there
3	should be fine
4	he says he has some ideas
6	so you're ignoring me
8	this is the crew

Table 1: Sample prompts by session.

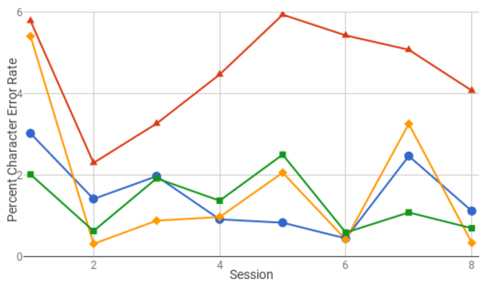

Figure 1: Participants' CER by session.

minute segments. Participants were asked to enter text bimanually with the device flat on the table.

The first few sessions were designed to build slowly and introduce the participant to the entry method. During the first session, the prompts given to participants consisted of single words ordered in small sets designed to obtain complete coverage of all available characters. The second session progressed to simple phrases with at most 4 words and a maximum word length of 6 characters. The phrases given to participants in the second session were also pruned to remove any that would require use of the N-Best list. In the third session, participants still entered simple phrases, but they were introduced to the N-Best list feature. For example, in the context of the Session 3 sample prompt shown in Table 1, the tap sequence for 'fine' initially recognizes the word 'done'. There were no restrictions on the prompts that participants entered in all subsequent sessions.

Results

We measure text entry quality using 3 metrics. First we use CER, measured as the minimum number of insertions, deletions, and substitutions required to obtain the reference text, divided by the number of characters in the reference text, multiplied by 100%. Participants had an average 4.05% CER in Session 1 and progressed to a 2.08% CER across the final 3 sessions. Figure 1 shows that most participants obtained much better accuracy after initially struggling in Session 1.

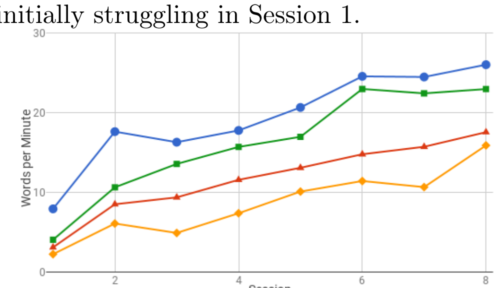

Figure 2: Participants' entry rates by session.

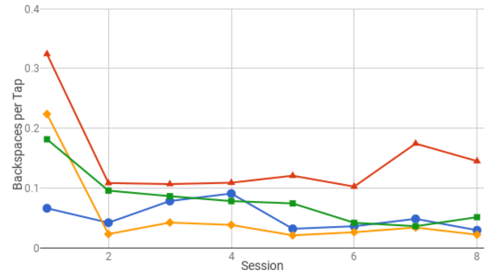

Figure 3: Participants' BPT by session.

Next, we measure Entry Rate using a standard of 5 characters per word (including a space) and time measured from the start of the first tap to the closing of the keyboard. Displayed in Figure 2, participants averaged 4.32 WPM in Session 1, and achieved an average 19.09 WPM across the final 3 sessions.

Finally, we measure Backspaces per Tap (BPT) by dividing the total number of taps deleted by the total number of taps entered. BPT allows us to measure the accuracy of participants' initial taps instead of the accuracy of the final entered text. Participants averaged .199 BPT in Session 1, but as shown in Figure 3, remained relatively stable at .068 average BPT for the remainder of the sessions.

We excluded in total 25 prompts that participants expressed difficulty hearing or spelling during entry, since in a real application the user will know what they are trying to type. Excluded prompts frequently included proper nouns, such as in 'did you mean oxley'.

CONCLUSIONS

Though all participants were sighted, our experiments demonstrate that Tap123 is a feasible eyes-free text entry method. The entry rate of 4.32 WPM in the first session is highly competitive with the entry rates found by Oliveira *et al.* for both VoiceOver (2.11 WPM) and BrailleType (1.45 WPM) [5]. However, the final three sessions' entry rate of 19.09 WPM is only slightly over half the 38.02 WPM achieved using two-handed entry on Perkinput [1]. Both the error rates from Session 1 (4.05%) and the final 3 sessions (2.08%) are below those of VoiceOver (14.12%) and BrailleType (8.91%) [5].

As future work, we will improve the keyboard in response to participant comments. We will give users the ability to delete an entire word at a time even after they have moved on, improve the gesture to close the keyboard, and use VelociTap's probabilistic decoding to recognize words that may not match the tap sequence exactly, but are similar and fit the context of previously entered text. Tap123 is being reviewed by three accessibility experts, all of whom are blind or low-vision, and we will improve Tap123 based on their recommendations. We expect these changes to improve both the entry and error rates of text entered using Tap123. We will then run additional trials with participants who are visually impaired.

REFERENCES

1. Shiri Azenkot, Jacob O. Wobbrock, Sanjana Prasain, and Richard E. Ladner. 2012. Input Finger Detection for Nonvisual Touch Screen Text Entry in Perkinput. In *Proceedings of Graphics Interface 2012*. 121–129.

2. Matthew Bonner, Jeremy Brudvik, Gregory Abowd, and W. Keith Edwards. 2010. No-Look Notes: Accessible Eyes-Free Multi-Touch Text Entry. In *Proceedings of IEEE Pervasive Computing*. 409–426.

3. Sergio Mascetti, Cristian Bernareggi, and Matteo Belotti. 2012. TypeInBraille: Quick Eyes-Free Typing on Smartphones. In *Proceedings of the 13th International Conference on Computers Helping People with Special Needs*. 615–622.

4. National Federation of the Blind Jernigan Institute. 2009. The Braille Literacy Crisis in America. (2009).

5. Joao Oliveira, Tiago Guerreiro, Hugo Nicolau, Joaquim Jorge, and Daniel Goncalves. 2011. BrailleType: Unleashing Braille over Touch Screen Mobile Phones. In *Proceedings of the IFIP TC International Conference on Human-computer Interaction*. 100–107.

6. Caleb Southern, James Clawson, Brian Frey, Gregory Abowd, and Mario Romero. 2012. BrailleTouch: Mobile Touchscreen Text Entry for the Visually Impaired. In *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services Companion*. 155–156.

7. Keith Vertanen. 2016. Counting Fingers: Eyes-Free Text Entry without Touch Location. In *CHI '16: Extended Abstracts of the the ACM Conference on Human Factors in Computing Systems*.

8. Keith Vertanen, Haythem Memmi, Justin Emge, Shyam Reyal, and Per Ola Kristensson. 2015. VelociTap: Investigating Fast Mobile Text Entry Using Sentence-Based Decoding of Touchscreen Keyboard Input. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. 659–668.