The Impact of Number of Predictions on User Performance in a Dwell Keyboard

JIBAN ADHIKARY, Michigan Technological University, USA
MAX ISOM, Michigan Technological University, USA
KEITH VERTANEN, Michigan Technological University, USA

Word predictions in a text entry interface can help accelerate a user's input. This may especially be true for users who have a slow input rate due to some form of motor-impairment. The choice of how many word predictions to offer in a text entry interface is an important design decision. In this work, we offered different number of word predictions in a keyboard where able-bodied users had to dwell on a key for one second to click it. We found participants' text entry rate did not improve with increasing number of predictions.

CCS Concepts: • **Human-centered computing** → **Text input**; **Accessibility technologies**.

ACM Reference Format:
Jiban Adhikary, Max Isom, and Keith Vertanen. 2022. The Impact of Number of Predictions on User Performance in a Dwell Keyboard. In *TEXT2030: MobileHCI'22 Workshop on Shaping Text Entry Research in 2030, October 1, 2022, Vancouver, Canada.* ACM, New York, NY, USA, 6 pages.

1 INTRODUCTION

One important design decision in a text interface is how many word predictions we should provide in the interface. Providing word predictions is particularly useful for people who have some form of motor- or speech-impairments and use an Augmentative and Alternative Communication (AAC) device to communicate. The rate at which these users enter text is very low, often less than 10 words-per-minute [5–7, 13, 16, 19]. Word predictions can help them accelerate their input. The intuitive choice of the number of word predictions in an interface might be the number of prediction slots the interface can fit. But previous work [2, 3, 18, 20, 22] has shown increasing number of prediction slots does not necessarily improve performance. With large number of slots, there is an associated cognitive cost while looking for the desired text. Also with increasing number of slots, we have smaller slot targets which can make selecting a slot difficult. In this paper, we examine the text entry performance of a keyboard with varying numbers of word predictions in a large online user study. While existing work tested multiple word predictions, the effect of multiple word predictions is unknown with keyboards where it requires dwelling on a key for a certain amount of time to activate it. In our work, the users had to dwell over a key for 1000 ms using their mouse pointer to click it. Our intention was to simulate the slow-input of a dwell keyboard with eye-gaze input.

2 RELATED WORK

Our choice of 1000 ms dwell-time was based on previous work [10]. Past work has examined both static and dynamic dwell times. Stampe and Reingold [17] used a dwell time of 750 ms in eye typing. Majaranta et al. [9, 10] used a dwell time of 400 ms and 900 ms. Paulus and Remijn [12] found users preferred a dwell time of 600 ms for selecting objects. Mott et al. [11] investigated adjusting a key's dwell time dynamically based on the likelihood

TEXT2030, October 1, 2022, Vancouver, Canada

of the key being selected next. Other work [8, 14, 15] also investigated adjustable dwell times with able-bodied users.

Our dwell keyboard differs from previous work in that we did not use an eye tracker. Instead, we substituted dwelling on a key using a mouse pointer. We did this for two reasons. First, it allowed us to compare performance using a large number of participants in a between-subject experimental design conducted online. Second, we felt dwelling via mouse would not materially impact the end-user utility of number of word prediction slots. For example, Hansen et al. [4] found users spent similar amount of time selecting a key via eye gaze and via a mouse for a given dwell time.

3 USER STUDY

To examine how many prediction slots provide the best performance for a text entry interface designed for AAC users, we conducted a user study using a web-based dwell keyboard.

Participants. We recruited 219 Amazon Mechanical Turk workers. Workers copied a set of phrases using our dwell keyboard. We instructed workers to enter text "quickly and accurately". Workers wrote phrases written by people with ALS for voice banking purposes [1]. We paid each worker $3.50. Workers took about 20 minutes to complete the study.

Design. Our web-based keyboard interface was similar to a touchscreen mobile keyboard but with a configurable number of prediction slots above the keyboard (Figure 1). The keyboard contained the letters A–Z, a spacebar, a backspace button, and a done button which moved to the next phrase. There were 2–9 prediction slots above the keyboard and a given user only saw the same number of slots in their session. The leftmost slot was dedicated to showing the literal text, i.e. the series of letters nearest to each tap for a word thus far. Before entering any letters for a word, the literal slot displayed an additional word prediction. The predictions were shown above the keyboard in real-time. We used the VelociTap [21] decoder hosted on a remote server to provide the word suggestions. The dwell keyboard application running in the user's web browser requested predictions from the server at the start of a dwell on a key. This allowed us to provide predictions in most cases instantly to users when they completed a dwell.

A user had to dwell on a button (i.e. a key or a prediction slot) using their mouse for one second (1000 ms) to select it. When a user hovered on a button, it was highlighted with a red rectangle and a green shade started filling the button. If the user stayed inside the button for a second, a click sound was played and the character or word corresponding to the button was added to the text field above the keyboard. If a prediction slot was most probable according to the decoder, we highlighted it in purple. In this situation, if the user entered space, it would select the highlighted prediction slot. We did this to imitate the iPhone keyboard. In other cases, space would select the literal slot.

Procedure. We conducted a between-subject user study with the number of prediction slots as the independent variable. There were eight conditions: two slots through nine slots. After entering each phrase, the participants saw the error rate and entry rate on that phrase. Each participant entered 32 phrases in total with the first two being practice phrases. We did not analyze the practice phrases. A participant never saw the same phrase twice.

We logged participants' interaction with the keyboard, including: what key they were hovering on, which prediction slot they used, how many times they used backspaces, and how much time it took to get a response from the remote server hosting the decoder. We measured the average server response time for each participant. We replaced participants in which more than 5% of their responses were more than 1000 ms. Such delays may have resulted in participants having to wait for predictions to populate; we wanted to approximate performance of a local prediction engine as would be commonly the case for an actual AAC device.

Due to a bug, in some cases the keyboard automatically selected a second word prediction immediately if a user selected a prediction and then moved the mouse pointer to an area above the keyboard. To identify phrases where

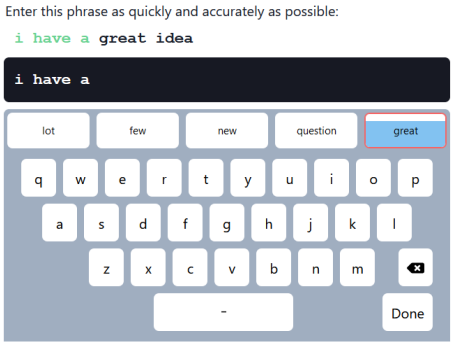

Fig. 1. The dwell keyboard interface. The user is dwelling on the "great" slot. The red rectangle indicates that the mouse pointer is currently hovering on this slot and a blue color fills the button as the dwell progresses.

Slots	WPM	CER	KS
2	9.1 ± 0.7	0.75 ± 0.40	32.0 ± 5.4
3	10.2 ± 0.9	1.56 ± 0.80	45.2 ± 4.1
4	9.7 ± 1.0	2.10 ± 1.11	44.0 ± 6.4
5	10.7 ± 0.7	0.56 ± 0.34	47.5 ± 4.9
6	10.8 ± 1.1	0.83 ± 0.32	51.0 ± 4.8
7	11.0 ± 1.0	1.27 ± 0.98	50.5 ± 4.8
8	10.3 ± 1.0	1.21 ± 1.18	47.9 ± 6.1
9	10.5 ± 1.0	1.18 ± 0.74	50.0 ± 6.4

Table 1. Entry rate (WPM), character error rate (CER), and keystroke savings (KS) result from the study. ± values denote 95% user-wise bootstrap confidence intervals.

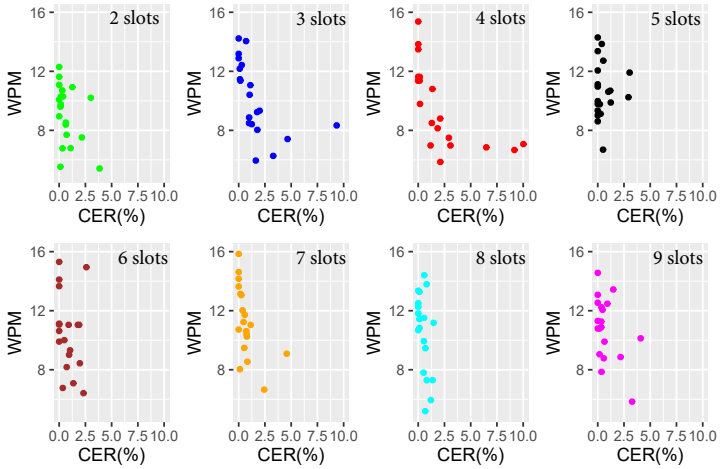

Fig. 2. Entry rate in words per minute (WPM) versus character error rate (CER) for each condition. Each dot represents one participant.

this issue occurred, we checked if there were two prediction slot selections within less than one second (as this would not be feasible with a one second dwell). We determined the number of phrases where a participant had the issue. If a worker had the issue in more than five phrases, we replaced that worker. Otherwise, we removed the problematic phrases from the corresponding worker's data. After removing 59 problematic workers and 80 problematic phrases, we ended up having 160 users and 4,720 clean phrases where the bug did not occur. For each number of slots from two to nine, we had 20 participants in the final set.

Results. Table 1 shows the input performance in the eight different conditions. We calculated entry rate using words per minute (WPM). We defined a word as consisting of five characters including spaces. Participants entered text at about 9–11 WPM using keyboards with differing number of slots. A one-way ANOVA revealed this difference was not statistically significant ($F_{7,152} = 1.15$, $p = 0.33$).

We calculated the keystroke savings (KS) on each phrase. KS measures the percentage reduction in key presses made compared to letter-by-letter input of a given phrase without the use of word predictions. Keystroke savings was highest with 6-slots at 51% but after that it flattened with increasing number of slots. Despite somewhat higher KS for the conditions with more slots, the corresponding entry rates were similar suggesting the cognitive cost of checking the additional slots was washing out any gains resulting from reducing the keystrokes required. However, it should be noted that for some AAC users, each input action can be tiring, so lowering the number of actions required even without an increase in entry rate can be advantageous.

We calculated error rate using character error rate or CER. CER is the number of character insertions, substitutions, and deletions required to transform user text into the reference text. Error rates were low in all conditions and ranged between 0.56%–2.10%. A one-way ANOVA test was not significant ($F_{7,152} = 1.01$, $p = 0.42$).

Table 1 suggests entry rate with 4-slots was less than 3-slots. Figure 2 shows entry rate versus character error rate plot for each slot. We can see eight 4-slot users had an average entry rate lower than 8 WPM and five had a CER of more than 2.5%. Perhaps, more correction led to slow entry rate for 4-slot users.

We also calculated the backspaces-per-output-character with all slots. Since it took participants at least one second to click a button, participants could stop dwelling on a button if it was less than a second and they were dwelling on the wrong button. Despite this, participants still sometimes made mistakes and selected an incorrect button. Average backspaces-per-output-character was low in all conditions at roughly four backspaces per 100 characters.

Given our interface's dwell-based interaction, we were interested to know how much time participants spent on each key. For each phrase, we calculated the elapsed time between when the participant started dwelling on the first letter of a phrase and when they dwelled on the done button. We divided the elapsed time by the total number of taps including tapping the done button to calculate the spent time per tap. Participants' average time spent per tap was 2,298 ms, with a minimum of 1,609 ms, and a maximum of 5,273 ms.

4 DISCUSSION AND LIMITATIONS

We conducted a crowdsourced user study with able-bodied but artificially rate-limited users to investigate the number of word predictions to offer in a keyboard style interface. We tested a range of prediction slots from two to nine but did not find any particular choice that provides a significant performance improvement. Except with only two slots, participants entered text at 10–11 words-per-minute and had keystroke savings of 44–51%. With increasing number of slots, performance flattened after 5 slots. This is similar to the finding in Vertanen et al. [20] in which participants typed on a smartwatch. While we had hoped that in cases where users typed slowly (as is the case for many AAC users) additional word predictions would offer improved performance, our study results suggest otherwise.

We used a static dwell-time of 1000 ms for the crowdsourced user study. As we discussed, past work has also investigated dynamic dwell time with eye-gaze input. But since we used the mouse pointer to have the participants dwell on the key, we did not use a dynamic dwell time. Our static dwell time was long. Future work might investigate the use of different and shorter dwell times. Our technique of dwelling using a mouse pointer was different from a traditional eye-gaze keyboard. Traditional eye-gaze would involve an actual eye or head tracker. Also able-bodied participants may differ from actual AAC users. Therefore, our study needs further validation with a more realistic input device and actual rate-limited users.

5 CONTRIBUTION

Our work helps keyboard designers decide the number of word predictions to include in a text entry interface. In particular, we focus on users who have a slow input speed due to a speech- or motor-impairment, which is

relevant to the theme of the workshop. We think our empirical data on prediction efficacy tested with a large number of users will also be useful to text entry interface designers in general.

6 AUTHOR BIOGRAPHIES

Jiban Adhikary. Jiban Adhikary completed his PhD in Computer Science from Michigan Technological University in 2022. His research interest lies in the intersection of natural language processing and human-computer interaction. His research focuses on investigating techniques to accelerate text input of users with speech- or motor-impairments.

Max Isom. Max Isom graduated from Michigan Technological University in 2021 with a BS in Computer Science. He is interested in research around human-computer interactions, especially how to improve the quality of life for users with disabilities. Max is currently a software engineer at a startup called Seam.

Keith Vertanen. Keith Vertanen is an Associate Professor at Michigan Technological University. He specializes in designing intelligent interactive systems that leverage uncertain input technologies. This includes input via speech, on touchscreens, in mid-air, and via eye-gaze. A particular focus of his research is on systems that enhance the capabilities of users with permanent or situationally-induced disabilities.

REFERENCES

[1] John M. Costello. 2014. Message Banking, Voice Banking and Legacy Messages. *Boston Children's Hospital, Boston, MA* (2014).

[2] John J. Darragh, Ian H. Witten, and Mark L. James. 1990. The Reactive Keyboard: A Predictive Typing Aid. *Computer* 23, 11 (1990), 41–49.

[3] Luís Garcia, Luis de Oliveira, and David de Matos. 2014. Word and Sentence Prediction: Using the Best of the Two Worlds to Assist AAC Users. *Technology and Disability* 26, 2-3 (2014), 79–91.

[4] John Paulin Hansen, Anders Sewerin Johansen, Dan Witzner Hansen, Kenji Ito, and Satoru Mashino. 2003. Command Without a Click: Dwell Time Typing by Mouse and Gaze Selections. In *Interact*, Vol. 3. Citeseer, 121–128.

[5] Jeffery Higginbotham, Howard Shane, Susanne Russell, and Kevin Caves. 2007. Access to AAC: Present, Past, and Future. *Augmentative and alternative communication* 23, 3 (2007), 243–257.

[6] Heidi Horstmann Koester and Richard Callaghan Simpson. 2014. Method for Enhancing Text Entry Rate with Single-switch Scanning. *Journal of Rehabilitation Research and Development* 51, 6 (2014), 995.

[7] Simon Levine, John Gauger, Lisa Bowers, and Karen Khan. 1986. A Comparison of Mouthstick and Morse Code Text Inputs. *Augmentative and Alternative Communication* 2, 2 (1986), 51–55.

[8] Päivi Majaranta, Ulla-Kaija Ahola, and Oleg Špakov. 2009. Fast Gaze Typing with an Adjustable Dwell Time. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 357–360.

[9] Päivi Majaranta, I. Scott MacKenzie, Anne Aula, and Kari-Jouko Räihä. 2003. Auditory and Visual Feedback during Eye Typing. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA) *(CHI EA '03)*. Association for Computing Machinery, New York, NY, USA, 766–767. https://doi.org/10.1145/765891.765979

[10] Päivi Majaranta, I Scott MacKenzie, Anne Aula, and Kari-Jouko Räihä. 2006. Effects of Feedback and Dwell Time on Eye Typing Speed and Accuracy. *Universal Access in the Information Society* 5, 2 (2006), 199–208.

[11] Martez E Mott, Shane Williams, Jacob O Wobbrock, and Meredith Ringel Morris. 2017. Improving Dwell-based Gaze Typing with Dynamic, Cascading Dwell Times. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 2558–2570.

[12] Yesaya Tommy Paulus and Gerard Bastiaan Remijn. 2021. Usability of Various Dwell Times for Eye-gaze-based Object Selection with Eye Tracking. *Displays* 67 (2021), 101997. https://doi.org/10.1016/j.displa.2021.101997

[13] Ondrej Polacek, Zdenek Mikovec, Adam J Sporka, and Pavel Slavík. 2011. Humsher: A Predictive Keyboard Operated by Humming. In *The Proceedings of the 13th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, 75–82.

[14] Kari-Jouko Räihä. 2015. Life in the Fast Lane: Effect of Language and Calibration Accuracy on the Speed of Text Entry by Gaze. In *IFIP Conference on Human-Computer Interaction*. Springer, 402–417.

[15] Kari-Jouko Räihä and Saila Ovaska. 2012. An Exploratory Study of Eye Typing Fundamentals: Dwell Time, Text Entry Rate, Errors, and Workload. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 3001–3010.

[16] Richard Simpson, Heidi Koester, and Ed LoPresti. 2006. Evaluation Of An Adaptive Row/Column Scanning System. *Technology and disability* 18, 3 (2006), 127–138.

[17] Dave M. Stampe and Eyal M. Reingold. 1995. Selection By Looking: A Novel Computer Interface And Its Application To Psychological Research. In *Eye Movement Research*, John M. Findlay, Robin Walker, and Robert W. Kentridge (Eds.). Studies in Visual Information Processing, Vol. 6. North-Holland, 467–478. https://doi.org/10.1016/S0926-907X(05)80039-X

[18] Keith Trnka, John McCaw, Debra Yarrington, Kathleen F McCoy, and Christopher Pennington. 2008. Word Prediction and Communication Rate in AAC. *Telehealth and Assistive Technologies (Telehealth/AT)* (2008), 19–24.

[19] Keith Trnka, John McCaw, Debra Yarrington, Kathleen F. McCoy, and Christopher Pennington. 2009. User Interaction with Word Prediction: The Effects of Prediction Quality. *ACM Transactions on Accessible Computing* 1 (February 2009), 17:1–17:34. Issue 3.

[20] Keith Vertanen, Dylan Gaines, Crystal Fletcher, Alex M. Stanage, Robbie Watling, and Per Ola Kristensson. 2019. VelociWatch: Designing and Evaluating a Virtual Keyboard for the Input of Challenging Text. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3290605.3300821

[21] 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 SIGCHI Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*. ACM, New York, NY, USA, 659–668. https://doi.org/10.1145/2702123.2702135

[22] Tonio Wandmacher, Jean-Yves Antoine, Franck Poirier, and Jean-Paul Départe. 2008. SIBYLLE, An Assistive Communication System Adapting to the Context and Its User. *ACM Transactions on Accessible Computing* 1, 1, Article 6 (2008), 6:1–6:30 pages. Issue 1.