Proposal of Word Prediction Method for Gaze Swipe Text Entry

Hisanori Fujii(✉) and Ryo Katsuma

Osaka Metropolitan University, Sakai, Osaka 599-8531, Japan
sb22651s@st.omu.ac.jp, katsuma@omu.ac.jp

Abstract. Gaze swipe, one of the text entry methods that do not require physical interaction with a device, involves sequentially tracing the characters of the desired word on a displayed keyboard using the user's gaze. This approach holds the promise of enabling quick word entry. However, previous gaze swipe method presents an issue where the time required for typing increases with longer words, consequently burdening the user. As a solution, there is a demand for an input word prediction method that presents the desired word for typing before the user completes tracing all the characters with their gaze. Thus, this paper proposes a gaze swipe typing method incorporating word prediction functionality. Through experimentation, the base gaze swipe typing method's typing speed was measured at 8.30 words per minute, with an accuracy rate of 91% for the top 3 estimated words. Additionally, for the proposed method, when 70% of the gaze path length was typed, the accuracy rate for the top 3 predicted words was confirmed to be 63%.

Keywords: Gaze typing · Word prediction · Text entry

1 Introduction

The popular method for text entry on computers has typically been through keyboards, while in recent years, mobile devices like smartphones have introduced touch screen keyboards as well as alternative entry methods such as toggle typing and swipe typing. Most of these methods share the characteristic of typing using hands and are not suitable for users with dirty hands, hand injuries, or physical disabilities that make it difficult to handle key input devices. Gaze typing, which is an input method that does not use the hands, is expected to facilitate communication with others via a computer in various situations.

Gaze typing involves utilizing devices like eye trackers to convert the user's gaze information into input signals. The typical gaze-based text entry involves placing a virtual keyboard on the screen and repeatedly gazing steadily at the desired characters to type in text. While this method ensures stability, it comes with the drawback of consuming time for text entry. The necessity to strictly gaze at characters one by one adds a time-consuming aspect to the operation,

A. Zaslavsky et al. (Eds.): MobiQuitous 2023, LNICST 594, pp. 204–216, 2024.
https://doi.org/10.1007/978-3-031-63992-0_13

becoming a bottleneck in quick text entry. The fixed amount of time it takes to gaze at a character and make a decision is called **dwell time**, and keyboard text entry using dwell time is called **dwell typing**. Dwell typing requires the dwell time to be set to a sufficiently longer duration. The shorter the dwell time, the more likely it is that a character will be entered unintentionally. Because of these characteristics, dwell typing is considered to have a limit to the typing speed. In order to facilitate text entry, it is necessary to establish other methods that allow text to be entered in a short time with fewer errors.

Gaze swipe is expected to be a fast, gaze-based text entry method. In gaze swipe, users track the characters of the intended word using their gaze. By matching the resultant, the path of the user's gaze (hereafter referred to as **gaze path**) with the paths of pre-established words in a dictionary, the typing word is estimated. This method does not require the user to strictly gaze at each character as dwell typing, but rather, the input is accomplished by moving the user's gaze in a flowing way along the string of characters that make up the word. Therefore, instead of dwell typing, which required strictly typing one character at a time, gaze swipe is expected to enable faster typing. In particular, research has been conducted on English words, and this paper also covers them.

There is one thing to keep in mind when implementing gaze swipe. The computation required to evaluate the gaze path against words in a dictionary is complex, and it is necessary to reduce the computational load. A previous study of gaze swipe is EyeSwipe [4] by Kurauchi et al. EyeSwipe uses gaze gestures on the first key of a word to signal the start of word, and on the last key of a word to signal the end of word. This method determines the first and last letters of the word to be entered, reducing the number of candidate words, increasing accuracy and reducing the amount of computation. However, there is a problem with this method. It is impossible to estimate a word unless the user tracks the last letter of the word with the user's gaze. This makes text entry time-consuming and increases the burden on the user. Therefore, a method that enables text entry without tracking all the letters that make up a word is required.

In this paper, we propose a new gaze swipe method with a mechanism for predicting English words during the typing process. Although word prediction methods that predict the most likely word to be typed next based on the context of the immediately preceding typing string are possible, in this study, the goal is to predict words without dynamic information other than during the typing of a single word as a preliminary step. On a PC, typing one character at a time on the keyboard is easy to predict because the input is determined one character at a time, but gaze swipe, which uses path matching, is difficult because the character string is not determined in the middle of typing. The proposed method enables prediction by matching partial paths of words in a dictionary and ranks predicted words using the frequency of occurrence of English words on the Web.

2 Related Work

Gaze-based text input methods can be divided into three main categories: methods that improve dwell typing [7,8], methods that use characteristic interfaces

[1,10–12], and gaze swipe [4,9]. Table 1 describes the characteristics of each gaze-based text input.

Table 1. The characteristics of each gaze-based text entry

	typing speed	familiarity	robustness	ease of implementing prediction function
characteristic interfaces [1,10–12]	○	×	○	○
dwell typing' [7,8]	×	◎	×	○
gaze swipe [4,9]	○	○	○	×

Approaches that use characteristic interfaces are expected to optimize typing speed and accuracy and reduce the burden of user's entry. However, there is a problem of time-consuming learning, such as memorizing key layouts, and it would be better to use key layouts that many users are familiar with. This is one of the reasons why many gaze-based studies use a virtual QWERTY keyboard. QWERTY keyboard-based dwell typing including its advanced versions (denoted as dwell typing' in Table 1) are expected to be easy to master, the presence of at least some dwell time limits the improvement in typing speed. The study by Kristensson et al. [3] shows the potential of gaze swipe. They hypothesized a perfect interface in which typing word is completed by looking near the key on the virtual keyboard corresponding to the letter of the word and conducted simulations using this interface. After 40 min of practice, the subjects' average typing speed reached 46 wpm, showing that there is room for improvement in gaze swipe and high promise for research to make gaze swipe speed up. Therefore, this study aims to further improve usability by introducing word prediction for gaze swipe, which has a large room for growth. By design, widely used word prediction mechanisms cannot be simply applied to gaze swipe, and there is no research on predictive entry for gaze swipe. This study challenges that issue.

Conventional input prediction methods are designed for typing a sentence or word one character at a time from the front, and present candidates by performing a forward matching from the letters the user is typing. Existing "studies" on gaze gestures and dwell typing realizes the input prediction by this kind of prediction method [5,6]. On the other hand, dwell-free typing like gaze swipe is not a form of typing one letter at a time, so the letters required for forward matching is indeterminate, making it impossible to predict a word. For example, if the user tries to type "admini", the gaze path may pass through "asdfgbnmk-ijnji" in that order. Since it is difficult to accurately determine the user's input letters from this path, it is impossible to extract the letters necessary for forward matching. The proposed method can efficiently select candidates of the prefixes for forward matching and uses these prefixes to estimate words. Specifically, the proposed method selects possible prefixes in typing by using a chronologically ordered letters in the neighborhood of the gaze path, and then efficiently selects prefixes for prediction by assigning a preference to them. The proposed word prediction method then performs a forward matching from the prefixes that are most likely to be in the typing state, in that order.

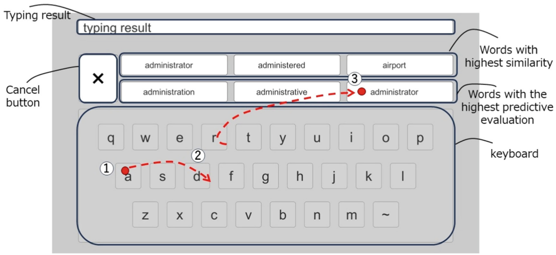

Fig. 1. Overview of the proposed system

3 Proposed System

The interface of the system targeted by this research is shown in Fig. 1. As shown in Fig. 1, the system has a virtual keyboard with 27 keys, including 26 alphabetic characters in a QWERTY layout and an "∼" for consecutive letters. The reason for using the key "∼" is described later in Sect. 4.1. It is assumed that the user types a word composed only of alphabetical characters. A non-contact eye tracker is attached to the monitor, and the user moves their gaze on the screen to type words. The movement of their gaze is measured by the eye tracker. The eye tracker can acquire the coordinates of both eyes' gaze on the monitor and the time at which the gaze is captured.

Describes the word input procedure. First, word input is initiated by gazing at the first key of the word for dwell time. Subsequently, the user traces each letter of the word using their gaze. While the user continues to move user's eyes on the virtual keyboard for typing, the system successively estimates the most similar words and additionally predicts the words to be typed by input prediction. These words are then listed and displayed in descending order of their respective evaluations. When the desired word is displayed, the user selects it to finish inputting the word. The input up to this point is regarded as a single word typing, and sentences can be composed by repeating this process.

4 Method

4.1 Base Gaze Swipe Typing Method

Acquisition of Gaze Path. Eye tracker can acquire the coordinates of the gaze in real time. The user tracks the letters of a word in sequence with their gaze as input for gaze swipe, and the coordinate time-series data of the gaze path at that time is denoted by $\{\boldsymbol{a}_1, \boldsymbol{a}_2, ..., \boldsymbol{a}_t, ..., \boldsymbol{a}_{t_{\mathrm{end}}}\}$. Currently, the gaze coordinates obtained from a non-contact eye tracker are too noisy for using as a gaze path. This is because light from sunlight or illumination reflects off the pupil or glasses, adversely affecting accuracy. It is necessary to format the gaze path data from the perspective of removing noise and accurately determining the degree of similarity to each word. First, average filtering is performed to remove noise, as in the following Eq. (1).

$$\boldsymbol{a}'_t = \begin{cases} \boldsymbol{a}_t & t = 1, t_{\mathrm{end}} \\ \frac{\boldsymbol{a}_{t-1}+\boldsymbol{a}_t+\boldsymbol{a}_{t+1}}{3} & o.w \end{cases} \tag{1}$$

To accurately determine the degree of similarity, the gaze coordinates above a certain speed are removed from the gaze path. The formatted coordinates are then treated as the coordinate data of the gaze path.

Selection of Candidate Words. Gaze swipe requires efficient search of a large number of words in order to estimate which words are appropriate based on the gaze path. One of the most commonly used data structures for text is the trie, which is used as a dictionary in the proposed method. The coordinates of each key on the virtual keyboard are denoted as $\{\boldsymbol{\lambda}_{\mathrm{a}}, \boldsymbol{\lambda}_{\mathrm{b}}, ..., \boldsymbol{\lambda}_{\mathrm{z}}, \boldsymbol{\lambda}_{\sim}\}$. The initial letter of a typing word is denoted as c_{start}.

Let the set of words contained in the trie-structured dictionary be denoted as D. Since calculating the similarity for all words in dictionary D would be impractical in terms of computation time, it's necessary to select candidate words before calculating the similarity. In this paper, we refer to the selected words as **candidate words**. When selecting candidate words, letters present in the proximity of the gaze path are arranged in chronological order. Words that are similar to this sequence are chosen. Let the coordinate of the gaze at time t be represented as $\boldsymbol{x_t}$ and the distance between $\boldsymbol{x_t}$ and the coordinates of each letter be calculated. Then, k elements are obtained in order of shortest distance and the corresponding characters $C_t = \{c_{(1,t)}, c_{(2,t)}, ..., c_{(k,t)}\}$ are obtained. The character chosen for time t is represented as $c'_t \in (C_t \cup \mathrm{NULL})$. Note that "NULL" represents an empty string. The sequence of characters selected at each time up to time t forms a character string, which can be expressed using the following Eq. (2).

$$w_t = c_{\mathrm{start}} + c'_1 + c'_2 + ... + c'_t \tag{2}$$

The + sign in Eq. (2) represents the concatenation of strings. It follows that the text string represented by Eq. (2) takes a form close to the prefix of the word to be typed. Therefore, among the strings expressed in Eq. (2), we call one corresponding to the prefix of a word the **candidate prefix** at time t. A candidate word obtained a string of characters that form a word among the candidate prefixes of the end time. Among the candidate prefixes at the ending time t_{end}, only those that can form words are obtained as candidate words.

Calculation of Similarity. With a gaze swipe, the words with the highest similarity to the gaze path are presented on the screen among the candidate words. The DTW (Dynamic Time Warping) method is used for similarity. DTW calculates the DTW distance, which represents the distance between two time series data, and the shorter the DTW distance, the higher the similarity between the two data. In the DTW method for gaze swipe, one data is the coordinates time series of the gaze path, and the other is the center coordinates of letters of a word, arranged in order (hereafter referred to as **ideal path**). For example, the ideal path for "nice" is $\{\boldsymbol{\lambda}_{\mathrm{n}}, \boldsymbol{\lambda}_{\mathrm{i}}, \boldsymbol{\lambda}_{\mathrm{c}}, \boldsymbol{\lambda}_{\mathrm{e}}\}$.

While the user is inputting a gaze path, the system generates candidate words and calculates the similarity between all of them and displays the top words in terms of similarity on the interface. The user selects the desired word from the list to complete the input.

There is one drawback to using the DTW method to calculate word similarity. Two words like "meet" and "met" that differ only in whether or not they contain consecutive letters have equal paths and thus equal similarity. To solve this problem, different paths are provided by mapping the " " key on the keyboard to consecutive letters. In this way, "meet" and "aaa" are treated as "me t" and "a a" on the keyboard, respectively.

4.2 Input Prediction

Input prediction is a feature that predicts what words will be entered while typing. It is highly likely that the word to be entered is somewhere among the derivatives of the candidate prefixes. Thus, the similarity between the gaze path during typing and part of the ideal path is expected to be high. Use statistical information to determine the words that are most likely to be typed among them. Let $s \in D$ be a word in the dictionary and $n(s)$ be the frequency of occurrence in the 1-gram model and let $n_w(u)$ be the frequency of the prefix u, expressed as the sum of the frequencies of the derivations. The probability of transition from a prefix u to its child node v in trie is defined as follows.

$$p(v|u) = \frac{n_w(v)}{n_w(u)} \tag{3}$$

The probability of transition from a prefix u to a word s whose letters matches the prefix is defined as follows.

$$p(s|u) = \frac{n(s)}{n_w(u)} \tag{4}$$

As a concrete example, Fig. 2 shows the probability distribution of five words {ad, adapt, adopt, advice, advise}. In Fig. 2, arrows indicate the probability of transition. A black node indicates the existence of a word terminating at that node, and the probability of transition to that word is indicated by an arrow returning to its own node. In gaze swipe, the above probability distribution is used as a measure for input prediction. First, we calculate the similarity of the candidate prefixes obtained from the base gaze swipe typing method. Then, up to m words with high transition probability are calculated among the words derived from each candidate prefix. It means that similarity is used as the first priority factor, and probability distribution is used as the second.

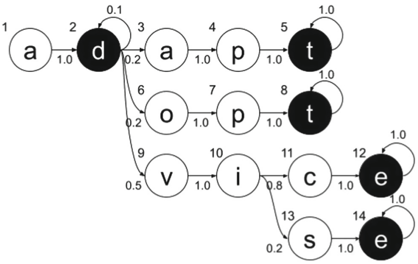

Fig. 2. Example of probability distribution

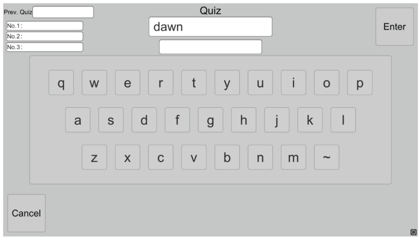

Fig. 3. Interfaces for experiment

5 Experiments

In this chapter, we describe two kinds of experiments. The first one is a comparison in which multiple subjects type words using both the proposed method and dwell typing interfaces. The purpose of this experiment is to evaluate the accuracy of word estimation and typing speed of the two methods. The second one evaluates the accuracy of the proposed method's input prediction under various conditions using the input data from the first one.

5.1 Experiment Environment

In the first experiment, we used the interface shown in Fig. 3, which implements the proposed method. This interface was implemented as a prototype of the proposed system to obtain the gaze path from the start to the end of typing, and is designed according to the proposed system, except that only the typing completion procedure differs slightly from the one shown in Fig. 1. The proposed system is designed to complete typing by selecting an estimated or predicted word. On the other hand, in the experimental interface, typing is completed by moving the user's gaze to the area in the upper right corner of the interface.

The interface randomly assigns a word from the dictionary to a question. After a word has been typed, the next question is presented, regardless of whether the typing was correct or incorrect. A non-contact eye tracker was attached to the monitor, and the user moved their gaze on the screen to type words, which was measured by the eye tracker. The eye tracker can acquire the coordinates of both gazes on the screen and the time at which the gazes were acquired. We used the Tobii Pro Nano eye tracker, which was a non-contact eye tracker. Its sampling rate was 60 Hz and was mounted on a 27-in. Full HD monitor.

5.2 Comparison Experiment

In the experiment, six subjects typed with both interface of the dwell typing and the proposed method. Calibration for each user is necessary to obtain highly accurate eye tracking information with a non-contact eye tracker. Even for a single user, the accuracy of calibration is not stable due to the fact that the distance and direction between the eye tracker and the user's eyes are not constant during the experiment and the light environment changes. Therefore, calibration was performed before the start of the experiment, as well as during the experiment.

Table 2. The typing speed, correct typing speed and top-i accuracy for each method

		typing speed [wpm]	correct typing speed [wpm]	top-i accuracy [%]
dwell typing		5.63	5.47	97.0
gaze swipe	top-1	8.30	6.92	83.4
	top2		7.41	89.3
	top3		7.56	91.0
	top4		7.67	92.4
	top5		7.69	92.6

The subjects were novice users with little experience using eye tracker. Before the data acquisition, the subjects watched a video explaining the typing procedure and practiced typing a few words as preparation. Each subject was given 6 sets of 5 min to type one word after another. Three sets consisted of dwell typing and the other three sets consisted of the proposed method. Subjects were asked to type as many correct answers as possible within a time limit.

The following is a description of the parameter values set for each typing method. In dwell typing, the dwell time, which is the time it takes for a letter to be typed, must be set. In this experiment, the dwell time was set to 0.4 s. The proposed method also uses dwell time for typing the first letter of a word, so we also set one to 0.4 s.

The list of dictionary words used in the proposed method consists of the 10,000 words with highest frequency of occurrence extracted by Kaufman from the Google Web Trillion Word Corpus [2]. The parameter k, the number of neighboring characters for selecting candidate words, was set to 3.

The evaluation items in this experiment were typing speed, correct typing speed, and top-i accuracy for both typing methods. Typing speed is the speed of typing regardless of correct or incorrect answers, and correct typing speed is the speed at which the correct word is typed. The top-i accuracy is the rate at which the correct word was among the top i estimated candidates. The top-i accuracy for the base gaze swipe was measured for word lengths. The top-i accuracy indicates the proportion of correctly estimated words among the top-i ranked candidate words. In addition, the typing time for both input methods was measured for word lengths.

Result for Top-i Accuracy in Both Methods. Dwell typing yielded 507 typing data, while gaze swipe yielded 747 ones. Table 2 shows the typing speed, correct typing speed and top-i accuracy for each method.

However, only the top-1 accuracy is shown for dwell typing. On the other hand, the proposed gaze swipe had the typing speed 1.47 times faster than that of dwell typing, with a lower accuracy but a superior correct typing speed. Therefore, the proposed gaze swipe demonstrated better performance compared to dwell typing.

Results for Word Length and Accuracy in Gaze Swipe. Figure 4 shows the relationship between word length and top-i accuracy in gaze swipe. The shorter the typing word, the more stable the top-i accuracy became when the number of nearest neighbor characters k was increased. On the other hand, the top-i accuracy for longer ones was not stable. One possible reason for this is that the number of long words input was small. In addition, the subjects did not fully remember the spelling of long words, which may have negatively affected the accuracy.

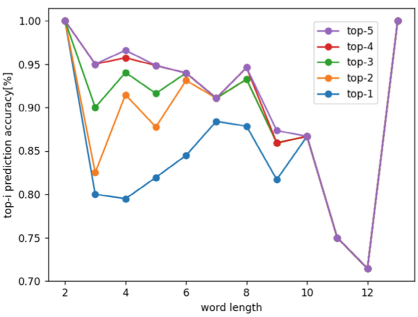

Fig. 4. The relationship between word length and top-i accuracy in gaze swipe

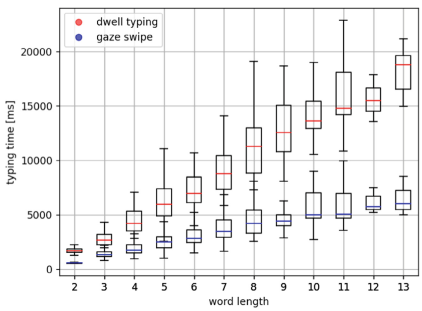

Fig. 5. The relationship between word length and typing time for both methods

Results for Word Length and Typing Time for both Methods. A box plot of the relationship between word length and the time it takes to typing word for both methods is shown in Fig. 5. However, since the start time of typing was not clearly known, the time measurement started when the first letter of the word was determined. Figure 5 shows that the time required to type a single word was shorter and the typing speed was faster with gaze swipe. Figure 5 also shows that both dwell typing and gaze swipe are close to proportional. A linear approximation yielded slopes of 1566.8 ms and 579.03 ms for dwell typing and gaze swipe, respectively. By comparison, the typing speed of gaze swipe is about 2.71 ($\simeq 1566.8/579.03$) times faster than that of dwell typing, regardless of the length of the word. This value is significantly different from the value of 1.47 obtained in the comparison of typing speeds for the entire experiment. The reason for this seems that the number of times the user canceled typing for gaze swipe because the desired word was not likely to be obtained during the deciding stage of the input. Therefore, if gaze swipe enables more reliable typing and requires fewer cancellations, the theoretical typing speed advantage of the eye-swiping method is approximately 2.71 times greater.

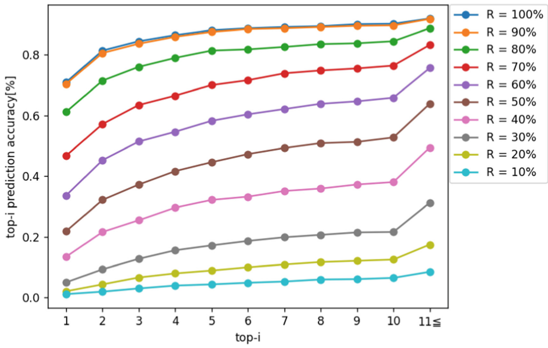

Fig. 6. Top-i accuracy for each split rate R

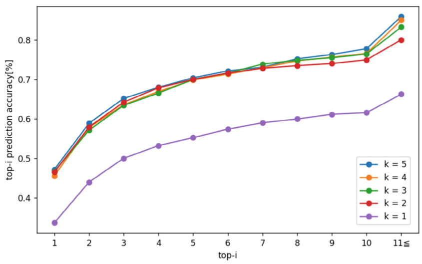

Fig. 7. Top-i accuracy for each number of nearest neighbor characters k

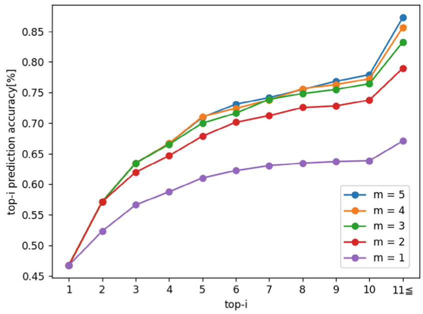

Fig. 8. Top-i accuracy for each number of derived words m

5.3 Analysis of Input Prediction

We analyzed the input prediction of the proposed method using the gaze paths obtained from the experiments in Sect. 5.2. Specifically, we calculated the predicted rank of the desired words and the top-i accuracy when predicting which words were going to be typing using a path that was clipped by $R\%$ from the beginning of the gaze path. The calculations were performed for various values of the parameters k, m, and R.

The measured top-i accuracy when the split rate R of the gaze path was varied from 10% to 100% is shown in Fig. 6. Here, $k = 3$ and $m = 3$. The $11 \leqq$ in the horizontal axis represents the 11th or later word in the word priority. Figure 6 shows that the higher the split rate R, the higher the top-i accuracy, and the split rate almost saturated at $R = 90\%$.

The measured top-i accuracy when number of nearest neighbor characters k was varied from 1 to 5 is shown in Fig. 7. Here, $m = 3$ and $R = 70\%$. Figure 7 shows that there was a large difference in the top-i accuracy between cases where the number of nearest neighbor characters k was 1 and other cases, but there was almost no difference in the top-i accuracy between cases where k was between 2 and 5. Therefore, once there was no difference in the accuracy rate, it is reasonable to use the smaller value of $k = 2$ or $k = 3$ in order to reduce the computational load.

The measured top-i accuracy when number of derived words m was varied from 1 to 5 is shown in Fig. 8. Here, $k = 3$ and $R = 70\%$. Figure 8 shows that the higher the number of derived words m, the higher the top-i accuracy. Therefore, it is reasonable to adopt $m = 5$, which has a high top-i accuracy, in the actual typing system. The top-3 accuracy was 0.63 when the parameters were set to reasonable values $k = 3$, $m = 5$ and 70% of the gaze paths were typed.

Considerations on the Framework of Input Prediction Methods. In this experiment, we examine an input prediction method for typing in response to gaze swipe using the frequency of occurrence of each word. The frequency of occurrence in a 1-gram model was used as statistical information to determine the derivation of candidate prefixes. That is very simple statistical information, and input prediction using only that is limited in its accuracy. Therefore, instead of a 1-gram model, more complex statistical information such as user input history and language corpora could be used to improve prediction accuracy. However, the use of complex statistics is a trade-off for computation time, since gaze swiping requires multiple forward matching searches.

6 Conclusion and Future Work

In this paper, we proposed a new typing method for gaze swipe with a mechanism for input prediction of English words to be typed during typing. The proposed method enabled input prediction by matching the prefix paths of words in a dictionary and ranking words by using their frequency of occurrence. Experimental results show that the proposed method is 1.47 times faster than dwell typing. When the system presented three words in response to typing that traced all of the letters of a word, percentage of correct word existence in the presented words was 91%. Furthermore, when the system presented three words for a gaze typing in which the user had typed 70% of the gaze path length, the accuracy was 63%, thus demonstrating the relationship between the input prediction accuracy and the ratio of the input gaze path length to the ideal path length. In this paper, as a preliminary step, we used the frequency of occurrence of words to rank the words to be predicted. In addition, we expect to improve the accuracy of prediction by taking into account various factors such as the context of the preceding and following words. The ranking of words to be predicted can be expressed by adjusting the probability distribution of word derivation, and this will be a subject for future work.

References

1. Bee, N., André, E.: Writing with your eye: a dwell time free writing system adapted to the nature of human eye gaze. In: Proceedings of the 4th IEEE Tutorial and Research Workshop on Perception and Interactive Technologies for Speech-Based Systems: Perception in Multimodal Dialogue Systems, pp. 111–122. Springer-Verlag, Kloster Irsee, Germany (2008). https://doi.org/10.1007/978-3-540-69369-7_13
2. Kaufman, J.: google-10000-English. https://github.com/first20hours/google-10000-english. Accessed 16 Aug 2023
3. Kristensson, P.O., Vertanen, K.: The potential of dwell-free eye-typing for fast assistive gaze communication. In: Proceedings of the Symposium on Eye Tracking Research and Applications, pp. 241–244. Association for Computing Machinery, Santa Barbara, California (2012). https://doi.org/10.1145/2168556.2168605
4. Kurauchi, A., Feng, W., Joshi, A., Morimoto, C., Betke, M.: EyeSwipe: dwell-free text entry using gaze paths. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, pp. 1952–1956. Association for Computing Machinery, San Jose, California, USA (2016). https://doi.org/10.1145/2858036.2858335
5. MacKenzie, I.S., Chen, J., Oniszczak, A.: Unipad: single stroke text entry with language-based acceleration. In: Proceedings of the 4th Nordic Conference on Human-Computer Interaction: Changing Roles, pp. 78–85. Association for Computing Machinery, Oslo, Norway (2006). https://doi.org/10.1145/1182475.1182484
6. MacKenzie, I.S., Zhang, X.: Eye typing using word and letter prediction and a fixation algorithm. In: Proceedings of the 2008 Symposium on Eye Tracking Research & Applications, pp. 55–58. Association for Computing Machinery, Savannah, Georgia (2008). https://doi.org/10.1145/1344471.1344484
7. Morimoto, C.H., Amir, A.: Context switching for fast key selection in text entry applications. In: Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications, pp. 271–274. Association for Computing Machinery, Austin, Texas (2010). https://doi.org/10.1145/1743666.1743730
8. Mott, M.E., Williams, S., Wobbrock, J.O., Morris, M.R.: Improving dwell-based gaze typing with dynamic, cascading dwell times. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, pp. 2558–2570. Association for Computing Machinery, Denver, Colorado, USA (2017). https://doi.org/10.1145/3025453.3025517
9. Pedrosa, D., Pimentel, M.D.G., Wright, A., Truong, K.N.: Filteryedping: design challenges and user performance of dwell-free eye typing. ACM Trans. Access. Comput. **6**(1) (2015). https://doi.org/10.1145/2724728

10. Rough, D., Vertanen, K., Kristensson, P.O.: An evaluation of dasher with a high-performance language model as a gaze communication method. In: Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces, pp. 169–176. Association for Computing Machinery, Como, Italy (2014). https://doi.org/10.1145/2598153.2598157
11. Ward, D.J., Blackwell, A.F., MacKay, D.J.C.: Dasher-a data entry interface using continuous gestures and language models. In: Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology, pp. 129–137. Association for Computing Machinery, San Diego, California, USA (2000). https://doi.org/10.1145/354401.354427
12. Wobbrock, J.O., Rubinstein, J., Sawyer, M.W., Duchowski, A.T.: Longitudinal evaluation of discrete consecutive gaze gestures for text entry. In: Proceedings of the 2008 Symposium on Eye Tracking Research & Applications, pp. 11–18. Association for Computing Machinery, Savannah, Georgia (2008). https://doi.org/10.1145/1344471.1344475