Predicting Users' Attention Breakpoints During Mobile Text Entry

Ioulia Simou
University of Patras
Rio, Greece
simo@ceid.upatras.gr

Andreas Komninos
University of Patras
Rio, Greece
akomninos@ceid.upatras.gr

ABSTRACT

During mobile text entry, users must shift their gaze from the keyboard to the entry area to check for errors. Many of these shifts are wasteful, since users have not committed any errors in the input stream. This waste might be reduced if, for example, on-keyboard visual feedback about the presence or absence of errors could be provided to users during text entry. However, constant feedback might result in visual clutter disruptive to the input process. We aim to address the challenge of predicting opportune moments (breakpoints) in user's attention to the keyboard to deliver such feedback, utilizing open gaze data collected from 30 participants while typing a) using the index finger of user's dominant hand and b) using both thumbs. We find that our model achieves promising results in predicting attention breakpoints under both typing conditions and, particularly, during text entry with both thumbs.

CCS CONCEPTS

• **Human-centered computing → Text input**; **Touch screens**; **Smartphones**.

KEYWORDS

Mobile text entry, smartphones, touchscreens, eye-tracking, attention management

ACM Reference Format:
Ioulia Simou and Andreas Komninos. 2023. Predicting Users' Attention Breakpoints During Mobile Text Entry. In *25th International Conference on Mobile Human-Computer Interaction (MobileHCI '23 Companion), September 26–29, 2023, Athens, Greece*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3565066.3608747

1 INTRODUCTION

Novel prototypes for mobile text entry are typically evaluated in laboratory settings, using transcription tasks, which require participants to quickly and accurately copy a set of phrases presented to them during the experiment. In these environments, it has been often noted that participants leave few uncorrected mistakes [4, 19]. For this reason, the Total Error Rate metric has been recommended as the only measure that captures both corrected and uncorrected errors, as well as the effort expended to make the necessary corrections [5]. One plausible explanation for the low uncorrected error rate is that users will frequently interrupt the process of selecting and entering characters (typing) to observe the text entered so far, therefore interleaving typing with error-checking activity. This behaviour inherits the problems associated with task interruptions. In related research, the interleaving of tasks has been found to require time to mentally prepare to assume the next task, as well as time to mentally prepare to resume the preceding task, which includes a high cognitive load as the state of the paused task has to be recalled into working memory [12].

Based on this information, it is natural to wonder how the interleaving of tasks associated with text entry (typing and checking) impacts user text entry speed, as many of the performed checks might actually be unnecessary - the user has indeed typed the intended letter sequence, but nevertheless still feels compelled to check, because mobile text entry is known to be difficult and unreliable. If it were indeed possible to somehow inform the user, while they are engaged in the typing task, that no errors have occurred, then this would alleviate the need to disrupt the typing task in order to assume the checking task, leading to less frequent context switches and therefore improved text entry performance. To provide this type of positive and reassuring feedback to users, a prototype design might imagine some form of persistent, always-on feedback in the user's field of vision (or using another modality). However, the danger in such an approach might be that the user would substitute the checking task, with a new checking task, i.e. often checking the feedback mechanism. Additionally, if the visual feedback mechanism is animated or uses vivid re-colouring to draw users' attention to it, it would succeed in redirecting users' attention to it, but frequent redirection would increase cognitive load and be just as (if not more) disruptive to the input process [9]. Instead, it might be better to provide users with reassurance periodically, e.g. every 2-3 characters, or even better, at *opportune* moments. Such moments might be, for example, at the end of composing a word. It would be even better if the feedback system could accurately identify or predict contexts in which it is *likely* that a user might feel the need to disrupt the typing task to perform some checking, and provide the feedback at that exact time.

We present an investigation on the feasibility of making such predictions, using open data from a mobile text entry experiment [13], involving gaze tracking and typing data. We demonstrate that it is possible to predict whether a user will engage in a checking task after entering a given character and these findings open up interesting pathways for future research which might improve support for text entry with mobile soft keyboards.

MobileHCI '23 Companion, September 26–29, 2023, Athens, Greece

ACM ISBN 978-1-4503-9924-1/23/09.
https://doi.org/10.1145/3565066.3608747

2 BACKGROUND

2.1 User behaviour during erroneous mobile text entry

Text entry is a complex process that consists of three main user-performed sub-tasks. First, some text has to be determined in the user's mind as the desired text to be entered. Fixing errors during mobile text entry is time and effort-intensive [6]. As such, users will slow down their rate of entry (writing speed) to avoid making more mistakes. Less is known about the strategies that users adopt to detect and fix errors. An analysis of typing data from a small cohort of users during a field study, ascertained that users might prefer to fix errors using backspaces, or short backspace sequences, rather than position the cursor prior to deletions [16]. This hints at users following an "on-the-spot" detection and correction strategy, which requires frequent shifting of attention between the keyboard and text entry area. A study of mobile text entry in lab settings, using gaze-tracking alongside other data capture methods, showed that there indeed exists a frequent allocation of attention between the keyboard and text entry area, during transcription tasks [13]. In the same work, authors demonstrated that error-checking is present even in the typing of phrases where users made no actual errors (approximately on average 2.6 gaze shifts to the text entry area, per phrase). Later work, using the same data released by [13], suggested that this number might actually be an underestimation of the actual amount of attention shifts taking place [15]. One disadvantage of the last two studies is that in the transcription task setting used in the experiment, the text entry area was immediately adjacent to the screen location where the stimulus phrase was displayed to users. Therefore, it isn't possible to discern between participant attention paid to the text entered, or the stimulus phrase (e.g. when the user forgot what to type next). Another recent gaze-tracking mobile text entry study which clearly distinguished between the areas of interest in the transcription task user interface, found that attention to the text entry area alone accounts for as much as 25% of the user's fixations during lab transcription tasks [18].

2.2 Task switching in text entry

In the context of text entry, it is acknowledged that the process of typing is a complex cognitive process, which relies heavily on memory (e.g. to retrieve appropriate words to form sequences) and fine motor control [10]. We can therefore consider text entry as consistent of two parallel, or interleaved sub-tasks: *determining and planning* the text to enter, and *executing* the appropriate motor functions to enter it. Recent literature views typing as a hierarchical control process [20] and it is argued that the digital equivalent of expression through writing (i.e. typing) is cognitively more challenging than its traditional counterpart (handwriting) [8].

There is, however, a third, less explicitly discussed sub-task in text entry, which is the need to check the text that has been entered for correctness. Failures may occur due to errors in the determining and planning sub-task (e.g. realizing that a better synonym for the currently composed word could have been used), or in the execution sub-task (e.g. inadvertently striking a proximal key to the intended one, resulting in a substitution error). Therefore, while checking the entered text is an important part of text entry, the checking action significantly interrupts the flow of the writing process. Mobile text entry particularly suffers from such disruptions to the writing process, primarily because touchscreens do not afford users the opportunities to touch-type (thus motor control cannot be fully automated or perfected), and also because typing in the wild involves unpredictable disruptions due to events in the external environment (e.g. movement, noise and so forth). Further, attempts to support text entry through methods like autocorrect, frequently result in inadvertent input which needs to be corrected by the user [2, 7]. These limitations make mobile text entry an inherently uncertain process, and therefore require that the task of writing is frequently interrupted by the task of checking.

A long strand of research investigating the impact of attention disruption and task switching, especially when working with computers, has found that interruptions are detrimental to task performance, mental state and affect, and ultimately performance. A reasonable overview of related literature on attention and disruption is provided in [12]. On the other hand, research on another source of disruptions in mobile use, namely notifications, has shown the benefit of delivering disruptive events at *opportune* moments, which may be predicted based on a range of contextual cues [22]. As discussed, typing is a mentally challenging task, and thus interruptions to it, according to Wicken's attention theory, are prone to be costlier since resumption of the task will be significantly more difficult [24]. Without support, users must continuously mentally balance the cost of disrupting the flow of their writing process to check for errors, against the cost of fixing any errors, which becomes greater if they are not immediately spotted. This decision-making process can therefore result in unnecessary (wasted) instances of checking. Improved support might enhance users' text entry speed and reduce frustration.

2.3 Summary

We hypothesized that a support mechanism to reduce unnecessary checking behaviour might provide users with some cues to continue writing (reassurance) to prevent shifts of focus away from the keyboard. Previous literature has attempted to support error awareness by providing visual cues within the text entered by the user, or on the keyboard itself, but only after a word has been completed [1, 17]. As explained in the preceding sections, ideally, these cues should be delivered at opportune moments, namely those when the user is about to decide on whether to look up or down. The goal of our paper is therefore to examine whether a prediction of these decision points is possible, using a machine learning method that should be lightweight and easy to integrate into the source code of a mobile keyboard.

3 MOTIVATION, MATERIALS AND METHOD

3.1 Motivation and Materials

The aim of this work was to develop a model for predicting the breakpoints in users' attention away from the keyboard, during text entry on mobile devices. To achieve that, gaze and touch data would need to be utilized. If such model could be developed, it would be possible to provide visual feedback about the presence or absence of errors to users, thus possibly decreasing the amount of users' gaze shifts and increasing their input efficiency. To illustrate the potential applications of the proposed model, imagine a user typing on a mobile device. Before an imminent gaze shift, as predicted by the model, multimodal feedback can be presented to the user in order to support the decision on whether to disrupt the flow of the input process, in order to check for errors. For example, keyboard graphical components could change colour according to the input stream state (e.g. green for no errors, red for errors detected), or a combination of haptic and audio feedback (e.g. short tones or vibrations) could indicate if there is a real need to disrupt the input process flow to check for errors. This concept has been demonstrated in [17] but in that paper, the feedback was continuously provided therefore potentially distracting the user in circumstances when they were not intending to make a decision on whether to disrupt their flow.

Building on previous work presented in [15], we utilized the open datasets released with Jiang et al.'s [13] paper to build the proposed predictive model. Gaze and touch data were collected while typing a) using the index finger of user's dominant hand and b) using both thumbs. As stated in Jiang et al.'s [13] paper, 30 users participated in the experiment, out of which 27 were right-handed. Each user conducted 20 unique trials per typing condition. For a more detailed description of the data collection process, readers can consult the original paper by Jiang et al. [13].

The gaze and typing datasets were obtained following the link provided in Jiang et al.'s paper [13] and the processing scripts via the GitHub repository provided by [15]. All data was analysed using a Python 3.10.2 environment, where the necessary modules for data analysis and machine learning (e.g. Pandas 1.4.2, Scikit-learn 1.1.1) have been installed.

We used the data related to both typing conditions. For one-finger typing, 15645 entries of touch data and 153860 entries of gaze data were recorded, while for two-thumbs typing there were 16843 entries and 121847 entries respectively.

As discovered in [15], the openly available datasets for the one-finger typing condition had not been cleaned or corrected (e.g. missing gaze data, invalid sentence ids, negative values in timestamp fields). The same observations emerged in the data collected during two-thumbs typing. We applied the pre-processing scripts provided by [15] to rectify erroneous values. Following pre-processing, the remaining data entries were, for one-finger typing, 13809 entries of touch data and 139179 entries of gaze data, while for two-thumbs typing, the sizes were 14793 and 111632 entries respectively.

3.2 Procedure

3.2.1 Upper and lower gaze labelling. In order to proceed with the extraction of the features to be used for training the predictive model, it is necessary to categorize the gaze data records into records concerning gazes at the display area of the input text (upper part of mobile), and gazes at the keyboard area (lower part of mobile). To achieve this, we follow the approaches for gaze shift detection provided in [15]. In this work, the author proposes two different approaches to gaze shift detection: the first method is based on a statically defined threshold that divides the screen into two parts - the upper (text entry area) and the lower (keyboard area) part. The second method adopts a more dynamic technique, relying on clustering algorithms to detect the change of the user's gaze position.

Applying those algorithms to the data collected during one-finger typing, 1993 gaze shifts were detected using the threshold-based method and 2719 gaze shifts using the cluster-based method. Likewise, during two-thumbs typing, 1637 gaze shifts were detected using the threshold-based method and 2321 gaze shifts using the cluster-based method.

To test which approach best approximated the true number of gaze shifts, we manually checked the total amount of gaze shifts for 6 of the 30 unique participants. Comparing the results of the two algorithms for these users, we found that the cluster-based algorithm approximated the true number better than the threshold-based algorithm, which supports the claim in [15]. Therefore, to proceed with the training of the predictive model we chose to exploit the labelling of the records into upper and lower gazes, as obtained by applying the cluster-based gaze shift detection algorithm.

3.2.2 Feature extraction. Our aim is to predict whether the next gaze event will be in the keyboard area (downwards gaze) or outside it (upwards gaze). To extract the features that will be used to train the predictive model, we processed the gaze data in combination with the corresponding keystroke data. Going through the gaze dataset on a per-user and per-trial basis, we decided upon the following features that might provide information about an imminent gaze shift. These features represent our attempt to model the likelihood of an error having occurred since the last check, and the cost of fixing an error in terms of corrective actions, but also wasted actions, which the user has to balance against speed gains accrued from not pausing the typing flow to check for errors.

- The last character that has been pressed, since some characters may have special meaning (e.g. word separators, phrase terminators or frequently mistyped or omitted characters);
- The number of characters typed since the last time space was pressed, since the longer a user types after beginning a new word, the higher the likelihood of errors;
- The number of lower gaze points that have elapsed since the last time an upper gaze occurred, since the more intently a user has focused on the keyboard the more likely it is that they would want to check their input;
- The time elapsed from the last upper gaze to the current gaze event, since the more time a user has focused on the keyboard the more likely it is that they would want to check what they've typed, and;
- The number of characters typed since the last upper gaze, since the more motoric effort a user has expended for typing, the more likely it is that they would want to check what they've typed.

To illustrate, imagine our intended classification target is the gaze event marked by a red circle in Fig. 1. This figure shows the vertical coordinates of gazes (grey dots) and those of typing events (red dots), along a timeline. The blue line shows the upper vertical coordinate limit of the keyboard area. The red dotted line is the midpoint between the top of the device screen and the start of the keyboard area. According to our feature descriptions, we can determine the related values. The last character pressed is 'S'. The number of lower gaze points elapsed since the last upper gaze is 6.

The time elapsed since the last upper gaze event is approximately 445ms. Finally, the number of characters typed since the last upper gaze event is 3 (*space*, 's', 's').

To extract these features, for each selected trial, we considered only those gaze events that have been recorded at times when the user had already started typing, i.e., had a "trialtime" value greater than or equal to the trialtime value of the first character press and less than the trialtime value of the last character press. Furthermore, if no cluster of upper gazes had preceded, i.e., at least one upper gaze event had not been recorded, the features associated with the upper gazes were given a default *NaN* value and were not taken into account for training the predictive model. Finally, we also excluded from the training process the cases of records that corresponded to consecutive upper gazes, during which no typing event has taken place. The resulting datasets were imbalanced.

Specifically, for the one-finger typing condition, 69134 (95.4%) gaze events were labeled as a lower gaze and 3371 (4.6%) gaze events as an upper gaze. For the two-thumbs typing condition, 41777 (89.7%) gaze events were labeled as a lower gaze and 4786 (10.3%) gaze events as an upper gaze. We note that checking behaviour is more prominent in the two-thumbs typing condition.

3.2.3 Model training. As shown, there are exactly two classes in the available data. Hence, we chose to use Support Vector Machine (SVM) for the development of our model, as it yields good results in binary classification problems. SVMs can run with good performance in terms of inference time and low power consumption on mobile devices [21]. For these reasons, we avoided a more complex deep neural network approach, which has been shown to take up to several seconds to provide inference [11].

For each typing condition an SVM was trained. Two approaches were followed during training and testing the models. Firstly, we adopted a k-fold cross validation approach ($k = 10$) to repeatedly train and test the model on randomly selected data from all users. This equates to a process where the model is trained on data from all users, and then tested for performance on instances belonging to more than one user. Considering a single user, this approach equates to a model that has been trained with some data from the user themselves, as well as data coming from other users. Second, we adopted a Leave-One-Person-Out cross validation approach. In this approach, we use data from 29 participants to train the model and test performance on the one remaining participant, cycling the process through all participants. This approach represents the starting user problem and assesses the model performance if the model knows nothing about the individual on whom it is about to be tested.

4 RESULTS

4.1 K-fold Cross Validation

As shown in Table 1, model performance is surprisingly good across all metrics. Due to the aforementioned class imbalance, we pay particular attention to the macro F1-score, especially since predicting correctly either label for the next gaze (on-keyboard or off-keyboard) is equally important. However, we present results for weighted metrics, as well as a range of related metrics, including precision, recall, accuracy and AUROC. Macro-F1 score for one-finger typing has a mean of $\bar{x} = 0.823(\sigma = 0.012)$, while this is slightly improved in two-thumbs typing with a mean of $\bar{x} = 0.892(\sigma = 0.007)$. On the other hand, macro recall is considerably lower in one-finger typing ($\bar{x} = 0.748, \sigma = 0.013$) compared to two-thumbs typing ($\bar{x} = 0.856, \sigma = 0.012$). In general, across metrics, the model exhibits improved performance in two-thumbs typing, which could be caused by the faster text entry speed and corrected error rate noted in the dataset paper [13]. We posit that events that may precede a gaze shift might be better differentiated in two-thumbs typing. For example, a pause in typing is more likely to indicate a pause in the input stream production in two-thumbs typing, whereas in one-finger typing, a pause may also be caused by reflection time, or time taken to locate and select a distant or not frequently used character on the keyboard.

4.2 Leave-One-Person-Out

Continuing to the second approach, model performance remains good, as shown in Table 2. Again, we present more related metrics in addition to the results for the macro F1-score. Macro-F1 score for one-finger typing has a mean of $\bar{x} = 0.801(\sigma = 0.077)$, while this is slightly improved in two-thumbs typing with a mean of $\bar{x} = 0.876(\sigma = 0.063)$. On the other hand, macro recall is considerably lower in one-finger typing ($\bar{x} = 0.734, \sigma = 0.079$) compared to two-thumbs typing ($\bar{x} = 0.842, \sigma = 0.072$). Across metrics, the model exhibits improved performance in two-thumbs typing, which corroborates the results of the first training approach. For this approach, we also show the ROC curves in Fig. 2. We observe that the results are, for all users, better than random, but for some users, the features and model built by them seem to work better than for others. This variability is more obvious in one-finger typing, while in the two-thumbs typing condition the ROC curves are more closely converged.

5 DISCUSSION

Building on previous work by [13, 15], we found that the prediction of breakpoints in users' attention utilizing touch and gaze data during text entry can be approximated with encouraging results by our proposed model. Usefully, the model seems to work well even in the case where no behavioural information is known about a particular user (Leave-One-Out). This fact might suggest that the cohort used in [13] exhibits very uniform behaviour, hence it is possible for the model to perform well even for a completely unknown user. However, the dispersion of ROC curves seems to negate such assumption. We note also that ROC curves are more dispersed in the one-finger typing condition and that model performance is overall better in the two-thumbs typing condition. We aren't certain why this phenomenon emerges, but we can hypothesise that given the age and profile characteristics of the cohort in [13], as well as the WPM and corrected error rates in that experiment, the participants might be more familiar with two-thumbs typing and display more consistent and thus predictable behaviour in this condition, compared to more hesitant, unfamiliar or otherwise non-predictable behaviour in one-finger typing.

In this paper, we selected SVM as a classifier that could be integrated in a mobile keyboard, as literature supports its suitability for binary classification, speed of inference on mobile devices, and low

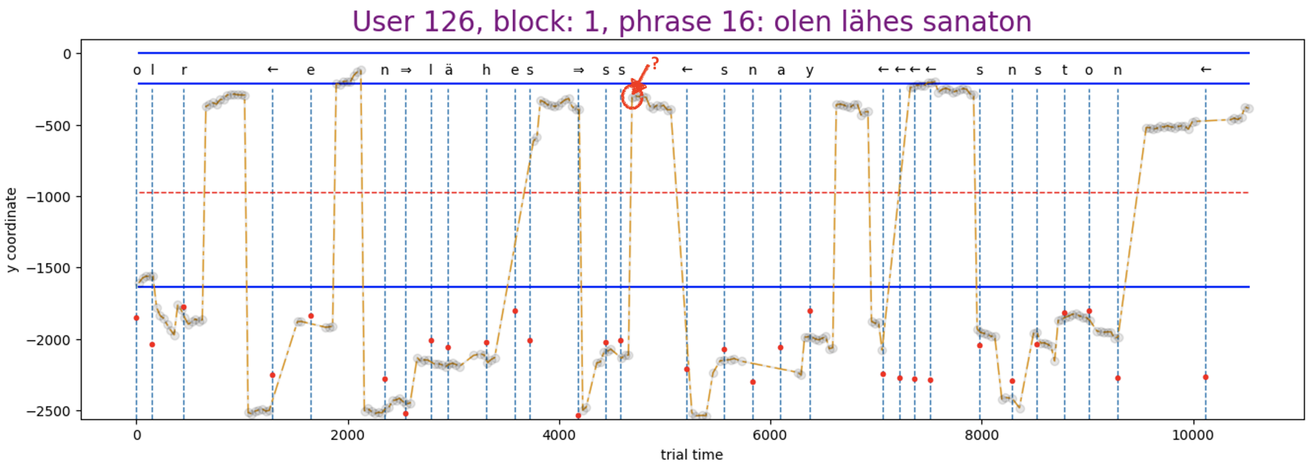


Figure 1: Timeline of typing and gaze events for a specific user.

Table 1: Metrics from 10-fold CV with One-finger and Two-thumbs typing

	One-finger typing						Two-thumbs typing					
	macro			weighted			macro			weighted		
	$\bar{x}$	σ	$95\%CI$	$\bar{x}$	σ	$95\%CI$	$\bar{x}$	σ	$95\%CI$	$\bar{x}$	σ	$95\%CI$
precision	0.98	0.007	0.004	0.976	0.002	0.001	0.94	0.008	0.005	0.963	0.002	0.001
recall	0.748	0.013	0.008	0.976	0.001	0.001	0.856	0.012	0.007	0.964	0.002	0.001
f1-score	0.823	0.012	0.008	0.972	0.002	0.001	0.892	0.007	0.004	0.962	0.002	0.001

	One-finger typing			Two-thumbs typing		
	$\bar{x}$	σ	$95\%CI$	$\bar{x}$	σ	$95\%CI$
accuracy	0.976	0.001	0.001	0.964	0.002	0.001
roc_auc	0.825	0.023	0.014	0.879	0.014	0.009
balanced_accuracy	0.748	0.013	0.008	0.856	0.012	0.007

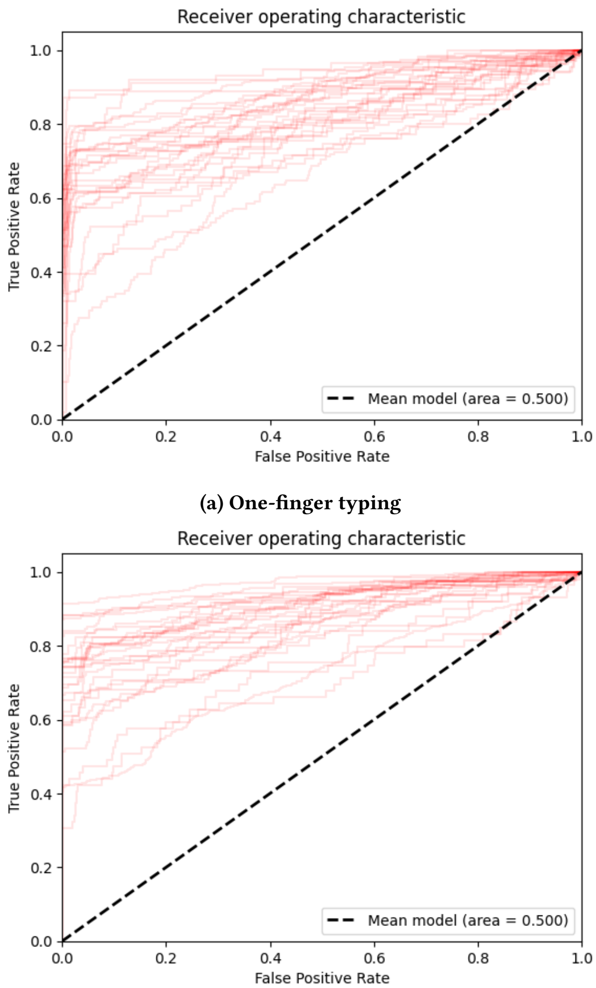
Table 2: Metrics from Leave-One-Person-Out with One-finger and Two-thumbs typing

	One-finger typing						Two-thumbs typing					
	macro			weighted			macro			weighted		
	$\bar{x}$	σ	$95\%CI$	$\bar{x}$	σ	$95\%CI$	$\bar{x}$	σ	$95\%CI$	$\bar{x}$	σ	$95\%CI$
precision	0.975	0.029	0.01	0.974	0.014	0.005	0.927	0.056	0.02	0.961	0.019	0.007
recall	0.734	0.079	0.028	0.974	0.014	0.005	0.842	0.072	0.026	0.962	0.019	0.007
f1-score	0.801	0.077	0.028	0.969	0.016	0.006	0.876	0.063	0.023	0.96	0.018	0.007

	One-finger typing			Two-thumbs typing		
	$\bar{x}$	σ	$95\%CI$	$\bar{x}$	σ	$95\%CI$
accuracy	0.974	0.014	0.005	0.962	0.019	0.007
roc_auc	0.829	0.069	0.025	0.875	0.07	0.025

power consumption. One main limitation of our study is the need for testing different machine learning techniques and comparing their results, regarding their ability to predicting the imminent gaze shifts during text entry. Deep learning architectures such as recurrent neural networks might provide better results, however they would likely require much more data than we have available, and real-time inference performance on the mobile environment would need to be examined.

A further limitation of our study is that we depend on data from an external source and a limited population sample, which was collected in a carefully controlled laboratory environment and not in-the-wild. Therefore, can only generalise about our findings with reservation. Unfortunately, data like this, combining both typing and gaze events, is difficult to find and even collect. We plan to conduct further investigation, to validate the model in real-life settings. Additionally, the proposed model, and models that will result from training different machine learning algorithms, should be evaluated in settings where transcription tasks are not the selected evaluation method or in settings where the text entry area is positioned close to the keyboard area, in contrast with the experiment in [13]. Moreover, to assess its performance, we are considering comparing our ML-derived model to the model proposed by Jokinen et al. [14],



(a) One-finger typing

(b) Two-thumbs typing

Figure 2: ROC curves for all participants using Leave-One-Person-Out validation.

which treats the text entry process as a supervisory control problem to explain gaze shifts.

Finally, the work presented here rests on the assumption that the mobile device is somehow able to track users' gazes therefore collecting the necessary information for some of our features. Although gaze tracking on smartphones without external hardware has been recently demonstrated to be feasible with reasonable accuracy [3, 23], this has only been done in a laboratory setting and it remains unknown whether such techniques might work as well as wearable eye-trackers in real life settings. We would like to examine different models which do not rely on eye-tracking ability, therefore exploiting only data which can be reliably extracted from the device, such as touch and device positional data.

6 CONCLUSION

In this paper, we introduced a predictive model yielding encouraging results in predicting when a user will shift their attention away from the mobile keyboard area during text entry. Our findings can facilitate further research in integrating lightweight machine learning models into the source code of a mobile keyboard, aiming to improving users' efficiency. Incorporating such models in the code of mobile keyboards to provide visual feedback to users, about the correctness of the input text, and encouraging the user to continue, rather than disrupt the flow of the text composition process, could possibly aid the improvement of users' typing efficiency.

REFERENCES

[1] Ohoud Alharbi, Ahmed Sabbir Arif, Wolfgang Stuerzlinger, Mark D. Dunlop, and Andreas Komninos. 2019. WiseType: A Tablet Keyboard with Color-Coded Visualization and Various Editing Options for Error Correction. In *Proceedings of Graphics Interface 2019 (GI 2019)*. Canadian Information Processing Society, Kingston, Ontario. https://doi.org/10.20380/GI2019.04

[2] Ohoud Alharbi and Wolfgang Stuerzlinger. 2022. Auto-Cucumber: The Impact of Autocorrection Failures on Users' Frustration. In *Graphics Interface 2022*. https://openreview.net/forum?id=dcbsb4qTmnt

[3] Riku Arakawa, Mayank Goel, Chris Harrison, and Karan Ahuja. 2022. RGBDGaze: Gaze Tracking on Smartphones with RGB and Depth Data. In *Proceedings of the 2022 International Conference on Multimodal Interaction (ICMI '22)*. Association for Computing Machinery, New York, NY, USA, 329–336. https://doi.org/10.1145/3536221.3556568

[4] Ahmed Sabbir Arif, Sunjun Kim, Wolfgang Stuerzlinger, Geehyuk Lee, and Ali Mazalek. 2016. Evaluation of a Smart-Restorable Backspace Technique to Facilitate Text Entry Error Correction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. Association for Computing Machinery, New York, NY, USA, 5151–5162. https://doi.org/10.1145/2858036.2858407

[5] Ahmed Sabbir Arif and Wolfgang Stuerzlinger. 2009. Analysis of Text Entry Performance Metrics. In *2009 IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH)*. 100–105. https://doi.org/10.1109/TIC-STH.2009.5444533

[6] Nikola Banovic, Varun Rao, Abinaya Saravanan, Anind K. Dey, and Jennifer Mankoff. 2017. Quantifying Aversion to Costly Typing Errors in Expert Mobile Text Entry. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. Association for Computing Machinery, New York, NY, USA, 4229–4241. https://doi.org/10.1145/3025453.3025695

[7] Nikola Banovic, Ticha Sethapakdi, Yasasvi Hari, Anind K. Dey, and Jennifer Mankoff. 2019. The Limits of Expert Text Entry Speed on Mobile Keyboards with Autocorrect. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3338286.3340126

[8] Sirine Bouriga and Thierry Olive. 2021. Is Typewriting More Resources-Demanding than Handwriting in Undergraduate Students? *Reading and Writing* 34, 9 (Nov. 2021), 2227–2255. https://doi.org/10.1007/s11145-021-10137-6

[9] Neil D. B. Bruce and John K. Tsotsos. 2009. Saliency, attention, and visual search: An information theoretic approach. *Journal of Vision* 9, 3 (March 2009), 5. https://doi.org/10.1167/9.3.5

[10] Yuichi Higashiyama, Katsuhiko Takeda, Yoshiaki Someya, Yoshiyuki Kuroiwa, and Fumiaki Tanaka. 2015. The Neural Basis of Typewriting: A Functional MRI Study. *PLoS ONE* 10, 7 (July 2015), e0134131. https://doi.org/10.1371/journal.pone.0134131

[11] Andrey Ignatov, Radu Timofte, William Chou, Ke Wang, Max Wu, Tim Hartley, and Luc Van Gool. 2018. AI Benchmark: Running Deep Neural Networks on Android Smartphones. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. 0–0. https://openaccess.thecvf.com/content_eccv_2018_workshops/w25/html/Ignatov_AI_Benchmark_Running_Deep_Neural_Networks_on_Android_Smartphones_ECCVW_2018_paper.html

[12] Shamsi T Iqbal and Eric Horvitz. 2007. Disruption and Recovery of Computing Tasks: Field Study, Analysis, and Directions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 677–686.

[13] Xinhui Jiang, Yang Li, Jussi P.P. Jokinen, Viet Ba Hirvola, Antti Oulasvirta, and Xiangshi Ren. 2020. How We Type: Eye and Finger Movement Strategies in Mobile Typing. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376711

[14] Jussi Jokinen, Aditya Acharya, Mohammad Uzair, Xinhui Jiang, and Antti Oulasvirta. 2021. Touchscreen Typing As Optimal Supervisory Control. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3411764.3445483

[15] Andreas Komninos. 2022. The Value of Open Data in HCI: A Case Report from Mobile Text Entry Research. *Multimodal Technologies and Interaction* 6, 9 (Sept. 2022), 71. https://doi.org/10.3390/mti6090071

[16] Andreas Komninos, Mark Dunlop, Kyriakos Katsaris, and John Garofalakis. 2018. A Glimpse of Mobile Text Entry Errors and Corrective Behaviour in the Wild. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct (MobileHCI '18)*. Association for Computing Machinery, New York, NY, USA, 221–228. https://doi.org/10.1145/3236112.3236143

[17] Andreas Komninos, Emma Nicol, and Mark D. Dunlop. 2015. Designed with Older Adults to SupportBetter Error Correction in SmartPhone Text Entry: The MaxieKeyboard. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct (MobileHCI '15)*. Association for Computing Machinery, New York, NY, USA, 797–802. https://doi.org/10.1145/2786567.2793703

[18] Andreas Komninos, Angeliki Tsiouma, Georgia Gogoulou, and John Garofalakis. 2022. Don't Look up: The Cost of Attention to Stimulus Phrases in Mobile Text Entry Evaluations. In *26th Pan-Hellenic Conference on Informatics with International Participation*. ACM, ACM, Athens, Greece. https://doi.org/10.1145/3575879.3576015

[19] Yang Li, Sayan Sarcar, Sunjun Kim, and Xiangshi Ren. 2020. Swap: A Replacement-based Text Revision Technique for Mobile Devices. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3313831.3376217

[20] Gordon D. Logan and Matthew J. C. Crump. 2011. Chapter One - Hierarchical Control of Cognitive Processes: The Case for Skilled Typewriting. In *Psychology of Learning and Motivation*, Brian H. Ross (Ed.). Advances in Research and Theory, Vol. 54. Academic Press, 1–27. https://doi.org/10.1016/B978-0-12-385527-5.00001-2

[21] Andrea McIntosh, Safwat Hassan, and Abram Hindle. 2019. What Can Android Mobile App Developers Do about the Energy Consumption of Machine Learning? *Empirical Software Engineering* 24, 2 (April 2019), 562–601. https://doi.org/10.1007/s10664-018-9629-2

[22] Tadashi Okoshi, Julian Ramos, Hiroki Nozaki, Jin Nakazawa, Anind K. Dey, and Hideyuki Tokuda. 2015. Attelia: Reducing user's cognitive load due to interruptive notifications on smart phones. In *2015 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. 96–104. https://doi.org/10.1109/PERCOM.2015.7146515

[23] Nachiappan Valliappan, Na Dai, Ethan Steinberg, Junfeng He, Kantwon Rogers, Venky Ramachandran, Pingmei Xu, Mina Shojaeizadeh, Li Guo, Kai Kohlhoff, and Vidhya Navalpakkam. 2020. Accelerating Eye Movement Research via Accurate and Affordable Smartphone Eye Tracking. *Nature Communications* 11, 1 (Sept. 2020), 4553. https://doi.org/10.1038/s41467-020-18360-5

[24] Christopher D. Wickens. 2002. Multiple Resources and Performance Prediction. *Theoretical Issues in Ergonomics Science* 3, 2 (Jan. 2002), 159–177. https://doi.org/10.1080/14639220210123806