Effect of Context on Smartphone Users' Typing Performance in the Wild

ELGİN AKPINAR, Middle East Technical University
YELİZ YEŞİLADA, Middle East Technical University Northern Cyprus Campus
PINAR KARAGÖZ, Middle East Technical University

Smartphones play a crucial role in daily activities, however, situationally-induced impairments and disabilities (SIIDs) can easily be experienced depending on the context. Previous studies explored the effect of context but mainly done in controlled environments with limited research done in the wild. In this article, we present an in-situ remote user study with 48 participants' keyboard interaction on smartphones including the performance and context details. We first propose an automated approach for error detection by combining approaches introduced in the literature and with a follow-up study, show that the accuracy of error detection is improved. We then investigate the effect of context on the typing performance based on five dimensions: environment, mobility, social, multitasking, and distraction, and reveal that the context affects participants' error rate significantly but with individual differences. Our main contribution is providing empirical evidence with an in-situ study showing the effect of context on error rate.

CCS Concepts: • **Human-centered computing → HCI design and evaluation methods**; **Empirical studies in HCI**; **Ubiquitous and mobile computing design and evaluation methods**; **Empirical studies in ubiquitous and mobile computing**; *Interaction techniques*; *Ubiquitous and mobile devices*;

Additional Key Words and Phrases: Context, smartphones, text entry, user study

ACM Reference Format:
Elgin Akpinar, Yeliz Yeşilada, and Pinar Karagöz. 2022. Effect of Context on Smartphone Users' Typing Performance in the Wild. *ACM Trans. Comput.-Hum. Interact.* 30, 3, Article 36 (June 2023), 44 pages.
https://doi.org/10.1145/3577013

ONLINE REPOSITORY

All the materials and data of this study (instructions and consent form of the user study and individual performance comparisons) are available in our external online repository at https://iam.ncc.metu.edu.tr/cabas/.

1 INTRODUCTION

Smartphones play a significant role in our daily lives. Through the years, their use has drastically increased, reaching almost 3.8 billion users in 2021 [1]. An average smartphone user checks their

Authors' addresses: E. Akpinar and Pinar Karagöz, Middle East Technical University, Üniversiteler Mahallesi, Dumlupınar Bulvarı No:1, Çankaya/Ankara 06800, Turkey; emails: {elgin.akpinar, karagoz}@metu.edu.tr; Y. Yeşilada, Middle East Technical University Northern Cyprus Campus, Kalkanlı, Güzelyurt, Mersin 10, 99738, Turkey; email: yyeliz@metu.edu.tr.

1073-0516/2023/06-ART36 $15.00
https://doi.org/10.1145/3577013

device 58 times a day and spends about three hours daily [71]. Smartphones are no longer just used for communication, but they are also used to perform most of the daily tasks [10]. They are widely used for text entry[1] tasks, such as writing text messages or e-mails [53].

Smartphones can be used in different environments, and while using smartphones, the users might be engaged with different and parallel tasks [52]. In the literature, temporary reductions in user performance due to context are referred to as **situationally-induced impairments and disabilities** (**SIIDs**) [79]. This phenomenon was defined as "difficulty accessing computers due to the context or situation one is in, as opposed to a physical impairment" [65, 66]. There can be many factors causing SIIDs, and the main observation is that some of these factors might be related to the current context. This article refers to context as "any information that characterizes a situation related to the interaction between humans, applications, and the surrounding environment" [18] (p. 106). Although this is a broad definition, the research on the effect of context on users' performance has been limited to a few contextual factors, such as mobility [2]. Our previous systematic review identified five contextual dimensions including environment, mobility, social, multitasking, and distraction, and showed that there is very limited research in understanding the effect of context on smartphone users' performance in typing text [2]. Most existing studies have been based on experimental tasks conducted in controlled environments (see Section 2). Participants typically asked to transcribe given phrases under different contextual factors in controlled laboratory settings. This approach of course provides a consistent way of measuring typing speed and errors made. However, this approach can also miss some difficulties in real-world usage [51]. Furthermore, in controlled studies, users can only use specific interaction methods, and this restriction may also jeopardize the validity of these studies [21]. Collecting data from actual users' context where the users' are not prescribed to type a specific text, can address these kinds of issues. However, collecting such free text data also has some challenges. Reproducibility is an issue [21] and since the users' intentions are not fully known, the reliability of performance measurement can also be questioned [51]. Even though these are important issues to consider, the existing literature also shows that it is possible to conduct an in-situ user study without a specific task model and still detect errors with a good accuracy [21].

This article presents an in-situ remote user study that aims to investigate the effect of context on users' text entry performance in real-world settings. Real-world text entry data is collected from 48 participants during their everyday interactions. To collect data, an existing framework called AWARE [23] is extended such that it also allows participants to label their current context via an **Experience Sampling Method** (**ESM**) [39]. This framework allowed us to conduct a remote user study in the wild to collect not only text entry data but also sensor data, and context labels from the participants (see Section 3). To compare the user's performance under different conditions, we needed to interpret user performance in terms of several metrics. One crucial metric was typing errors to measure users' performance. Several studies have identified typing errors in the wild; however, these studies had some limitations. For instance, daily texting language was not considered in these approaches. Therefore, we combined several existing approaches to detect typing errors and distinguish between edits and corrections using the text entry data (see Section 4). Finally, we investigated the effect of context on user performance by combining text entry data and context labels in five dimensions: environment, mobility, social, multitasking, and distraction (see Section 5).

The **contributions** of this study and the article are as follows:

- Most of the text entry studies have been conducted in controlled laboratory environments. In this study, we collected text entry and sensor data in the wild. We extended an existing

[1] This article uses the terms "typing" [51] and "text entry" [21] synonymously to refer to writing text using a keyboard.

framework to capture the participants' keyboard interactions, a set of sensor data, and context labels submitted by the participants.

- Recent approaches to detect typing errors by using free text have been based on lookup approaches. We combined the approaches of Evans and Wobbrock [21], Nicolau et al. [51], and Torunoğlu and Eryiğit [70] to cover daily texting language and detect typing errors in both English and Turkish.
- The effect of context on users' text entry performance has been primarily investigated for different mobility conditions in the literature. This study considered the context in five dimensions: environment, mobility, social, multitasking, and distraction. According to our findings, being in an outdoor environment, being mobile, presence of other people, multitasking, and having distractions increase error rate but have no effect on typing speed. This study provides the first empirical evidence on the effect of context on users' typing performance in an in-situ study (see Section 6).

2 LITERATURE REVIEW

The main aim of this study is to investigate the effect of context on users' typing performance. We start our literature review by identifying the metrics to measure typing performance. We continue our review to identify the typing errors and typing behaviour in daily life. Then, we review the literature surrounding how context affects these performance metrics in the text entry task domain. The studies reviewed have been conducted in controlled settings, and a systematic understanding of how context affects users' typing performance in their daily life is still lacking. Finally, we review the studies that automatically measure typing performance in the wild.

2.1 Text Entry Metrics

Several metrics are used to measure typing performance. In terms of typing speed, **words per minute** (**WPM**) and **keystrokes per second** (**KSPS**) are the most popular metrics. WPM considers only the length of transcribed text and how long it takes to produce it. It considers a word every five characters entered and measures the number of words entered in a minute [77]. KSPS is used to measure the number of keystrokes made in a second. It is useful when taking error corrections into account [77]. **Keystroke per character** (**KSPC**) and **error rate** (**ER**) are widely used for accuracy. KSPC is the ratio of the total entered character count to the length of the transcribed string [67]. ER is the ratio of incorrect characters to all characters entered [67]. Minimum string distance between intended and transcribed text can also be used for ER [77]. Error rates can be assessed in several ways especially for the studies conducted in the wild without a predefined task model.

2.1.1 *Unintentional Errors.*

Text entry errors can be classified into unintentional and intentional typing errors. For unintentional errors, Durham et al. [19] identified four types of word-level text errors as follows: transposition, the wrong letter, extra letter, and missing letter. According to Chen et al. [13], mobile device users experience character ambiguity, missing or additional character, bounce (repeating characters), long-press, and transposition errors. Greene et al. [29] also reported extra or missing character, incorrect shifting, wrong character, adjacent character, transposition, and misplaced character errors. A word in a text can contain many errors, and the number of errors even can exceed the number of correct characters [12].

2.1.2 *Intentional Errors.*

The intentional typing errors are referred to as "text-speak" [32] and consists of intentional corruptions on the words [68] for several reasons such as mirroring positive and negative emotions [25], increasing perceived playfulness [34], typing faster to reduce latency

Table 1. Literature Summary (↓: Decreased, ↑: Increased, ⊘: No Significant Effect, -: NA) WPM: Words Per Minute, KSPS: Keystrokes Per Second, KSPC: Keystroke Per Character, ER: Error Rate

Ref.	Context	Factor	WPM	KSPS	KSPC	ER
[54]	Environment (lab/indoor real-world)	Being in a public place	-	⊘†	-	⊘
[42]	Mobility (stable/mobile)	Walking	-	⊘	-	↑
[33]	Mobility (stable/mobile)	Being in a subway train	-	-	⊘	⊘
[17]	Mobility (stable/mobile)	Walking	⊘	-	-	↑
[26]	Mobility (stable/mobile)	Walking	⊘	-	-	↑
[50]	Mobility (stable/mobile)	Walking	⊘	-	-	↑
[24]	Mobility (stable/mobile)	Walking	-	-	-	↑
[15]	Mobility (stable/mobile)	Walking	↓	-	-	⊘
[49]	Mobility (stable/mobile)	Walking	↓	-	-	⊘
[54]	Mobility (stable/mobile)	Walking	-	↓†	-	↑
[22]	Mobility (stable/mobile)	Walking	↓	-	-	↑
[61]	Urban noise (indoor/outdoor)	Outdoor noise	-	↓*	-	-
[61]	Speech (meaningful/meaningless)	Meaningful noise	-	↓*	-	-
[60]	Ambient light	Dimmed light or sunglasses	-	⊘*	-	⊘
[59]	Multitasking	Presence of stress task	-	⊘*	-	⊘
[16]	Multitasking	Avoiding hazards	-	-	-	↑
[35]	Distractions	Presence of distraction	↑	-	↓	↑

The metrics used in corresponding studies were (∗) time per character entry, and (†) character per minute which can be interpreted as KSPS.

in a synchronized way of communication [14, 68], or common words in communication slang [57, 74]. Using text-speak, users compress the text by employing abbreviations, phonetic substitutions, and character strategies [14]. Table 13 in Appendix A illustrates common text-speak techniques in daily texting use and examples for these techniques. Since the users are intentionally typing in this way, they should not be associated with a performance problem.

2.1.3 Corrected/Uncorrected Errors. Wobbrock and Myers [80] classified errors into insertions, omissions, and substitutions and considered whether these errors were corrected or uncorrected. The corrected errors do not appear in the transcribed text; however, they can be traced using the input stream and can help measure the text entry performance better. There might also be cases when a user did not make an error but somehow thought that he/she did and deleted the corresponding text to rewrite it (corrected no-errors). Uncorrected errors are the errors that remain in the final transcribed text. The total ER is then can be calculated by the sum of the corrected ER and uncorrected ER [80].

2.2 The Effect of Context on Users' Text Entry Performance

Table 1 presents a summary of the research on the effect of contextual factors on text entry performance. Instead of a character level entry rate, Hoggan et al. [33] used time to enter phrases. They showed that sitting in a subway train decreased the entry time than sitting in the laboratory. Similarly, Crease et al. [16] used task completion time and showed that walking and avoiding hazards together decreased the task completion time.

Most of the previous research has focused on mobility conditions, while a relatively small body of literature has covered other contextual factors. The popularity of mobility conditions may be explained by the fact that different mobility conditions can be easily simulated by ensuring identical experimental settings across all sessions. On the other hand, social context and physical contextual factors such as lighting level, temperature, or ambient noise are hard to control, and they can easily differ between sessions.

Although mobility has been a popular contextual factor, there have been inconsistent findings on its effect on typing speed and error rate. Several studies have shown that environment [54], ambient light [60], and multitasking [59] did not affect typing speed and error rate. Jain and Balakrishnan [35] demonstrated that the presence of distraction increased typing speed. They commented that the increase in typing speed might be related to higher attention caused by higher distraction.

Table 1 only encloses the most relevant studies to text entry. However, there is considerable research on the effect of different contextual factors on the other task domains. Sarsenbayeva et al. [58] and Goncalves et al. [28] showed the effect of ambient temperature on target selection time. Barnard et al. [6, 7] compared low and high lighting levels for reading and searching tasks and indicated that there is a main effect of lighting level on task completion time and workload but not on error rate. Encumbrance also has a main effect on target selection accuracy and time [45–48]. Further detailed review on the effect of context on users' performance can be found in our systematic review [2].

2.3 Text Entry Studies in the Wild

Several methods are used to detect texting errors in the wild which include the following:

Using transcribed text. Palin et al. [53] conducted a study with considerably large number of participants. However, instead of allowing participants to enter free text during their daily activities, they presented texts for participants to transcribe. Similarly, Reyal et al. [55] compared two different keyboard methods in the wild. Although participants used their own devices during their daily activities, they performed transcription tasks. Schlögl et al. [64] and Wimmer et al. [76] used game-based approaches to measure a large number of text entry metrics for different soft keyboards.

Using an offline lexicon. Evans and Wobbrock [21] aimed to measure desktop text entry performance in the wild. They used WPM, uncorrected, and corrected ERs. To detect errors and distinguish between corrections and edits, they used an offline lexicon (English Lexicon Project). If a word was in the lexicon, it was considered correct. Nicolau et al. [51] conducted a study with blind users to observe their everyday typing behaviour on mobile devices. They used the Hunspell lexicon for error detection.

Using a spell-checker. Komninos et al. [38] observed typing error and correction behaviour in the wild. They used a spell-checker to classify errors as slight and severe concerning the suggestions for entered text. Wong et al. [81] used Aspell for spelling error detection in chat records.

Using an online query service. Evans and Wobbrock [21] used Bing API in addition to the offline lexicon. The API returned suggestions if the word is incorrect. They considered these suggestions the intended words. Wong et al. [81] used an online resource to expand abbreviations. Varnhagen et al. [74] used NetLingo and UrbanDictionary as helper services.

Manual analysis. Battestini et al. [8] conducted an in-situ study to analyze text message topics. They analyzed whom the participants texted with, why they sent text messages and their thoughts on text messaging. They manually categorized topics of conversation. Nicolau et al. [51] also manually analyzed words that do not appear in the offline lexicon to detect text entry errors.

Rodrigues et al. [56] compared transcription, composition, and passive sensing approaches in terms of the effort of the participants, the effect on the typing behaviour, and the participants' perception of privacy by conducting a study in the wild. They observed that the amount of effort put on the participants was the least for passive sensing and the most for composition tasks. Moreover, they ensured a policy that no raw data was collected during the study and provided a mechanism to pause capturing data. These helped to create a perception of privacy and trust among the participants. On the other hand, the composition task, in which participants were asked to compose a text describing their daily activities, caused more cognitive effort and privacy issues.

Using transcribed text in a controlled environment may increase the consistency of a study; however, these studies fail to cover real-world cases. On the other hand, detecting users' intention when there is no task model, and users enter free text in daily settings is challenging [51]. Evans and Wobbrock [21], and Nicolau et al. [51] used offline lexicons to detect typing errors along with other resources such as an online search API or manual analysis. However, this method may not be practical due to many out-of-vocabulary words for morphologically rich languages, such as Turkish [70]. Using offline lexicons fail when the text contains words changed with text-speak for daily language. Torunoğlu and Eryiğit [70] carried out a study to normalize Turkish text on social media. The transformations they applied on out-of-vocabulary tokens include letter case transformation, removal of character repetitions, transformations on emo style writing, proper noun detection, deasciification, vowel restoration, and accent normalization.

According to Evans and Wobbrock [21], if a participant deletes some characters and enters text again, there are two possibilities: participant either corrects an error or changes his/her mind to enter a new word. They used a straightforward approach. If an online query returned suggestions for removed words and reentered words matched with one of these suggestions, it was identified as an error correction. Otherwise, it was considered an edit. Nicolau et al. [51] noted that blind users tend to correct errors as soon as possible. As a result, they needed to check incomplete words with final words to distinguish between errors and edits. First, they checked whether removed and reentered characters were adjacent. If all characters were adjacent, it was considered an error correction. Then, they used Hunspell to retrieve spelling suggestions for the removed text. It was considered an error correction if the final text was in the spelling suggestions. Finally, if the minimum string distance between deleted and final word was more than half of the words' length, they considered it an edit. Otherwise, it was considered an error correction.

In summary, our literature review showed that the effect of the context on users' typing performance had been investigated mainly in controlled settings. Conducting studies in the wild is essential to collect more realistic data on the tasks users do in their daily lives. Processing the text entries in daily lives requires a mechanism to measure typing performance automatically. There have been several attempts to achieve this; however, such studies remain narrow in focus dealing only with formal writing. Morphologically rich languages and daily texting language should also be considered.

3 USER STUDY—IN THE WILD

We conducted a user study in which we aimed to collect user performance data and corresponding context factors in the wild. In general, we adopted the ESM [39] for context labels and automated collecting performance data. This section explains the methodology of our study in full detail.

3.1 Data Collection Framework

We used the AWARE Framework for data collection [23]. AWARE is a framework that provides logging mechanisms for a variety of available sensors in Android devices. It also enables data collection using ESM. One of the significant advantages of AWARE is that it is open-source, and anyone can extend it for specific purposes. Moreover, it provides mechanisms to register and unregister to studies, pause and resume the data collection, disable data synchronization when the battery level is low, or the smartphone is not connected to Wi-Fi, and monitor the studies.

AWARE is a general-purpose framework and did not have certain features required within our study. Therefore, we implemented several features on the AWARE framework for our study. First, we embedded the informed consent form in the app and made it the opening page after installation (see "Online Repository" Section on page 1). The participants could participate in the study only if they read and accepted the informed consent form. We also created a demographics form (see Section 3.6 for the questions and available options). After participants registered in the study, we asked them to complete this form once. The app retrieved sensor configurations in JSON format from a web service and configured the study automatically. Since we were interested in sensor data only when participants entered text, the app disabled all sensors and stopped recording when the screen was off or locked. When the screen was on or unlocked, it again enabled sensors. This optimization helped us reduce bandwidth use and storage required for overall study data. We also ignored the sensor data for the sessions that participants did not enter text. If a participant entered text longer than five characters, the app asked the participant to answer five questions about the context. To not interrupt the participants during a task, the app showed these questions when the participants returned to the home screen. We removed all unnecessary permission requests by disabling irrelevant modules, such as cameras or contacts.

During the study, the app collected data from the following sensors: accelerometer, applications, barometer, battery, communication, gravity, gyroscope, light, linear accelerometer, locations, magnetometer, proximity, rotation, screen, significant motion, telephony, and Wi-Fi. Moreover, after each keystroke, the text before and after the keystroke was recorded. The app did not take pictures, capture videos or audio, collect passwords, or collect screen content. It also did not send messages on behalf of the participants.

We deployed the AWARE server application on METU NCC servers. The interaction between the app and the server was handled with the HTTPS protocol. We used this application for monitoring and data collection purposes. Finally, we created a web page for the study.[2]

3.2 Methodological Decisions

In general, we followed the guidelines provided by van Berkel et al. [73] and focused on having an unobtrusive study as much as possible. We aimed to minimize participants' burden; therefore, we presented a set of options for each context dimension (details are given in the following section) and asked participants to select only one option for each question. With this approach, we avoided free text entry inputs. Each notification was triggered after a text entry event. If participants did not respond to questionnaires, they expired in 30 seconds and were removed from the notification panel. This notification timeout aimed to ensure that the participants answered the questions within the context of the text entry. We put at least 15 minutes between two questionnaires to not overload participants, and participants received these questionnaires at most eight times a day. We asked participants to keep the app installed for at least one week to capture context data during different daily activities. Finally, participants were informed that they could pause data collection any time they felt uncomfortable sharing their private data.

3.3 Context Labelling Questions

In our systematic review, we investigated the effect of the context under five dimensions: physical, temporal, social, task, and technical contexts [2]. We also reviewed the relevant ESM-based research and collected the contextual factors used. Then, we combined our findings with our

[2] https://iam.ncc.metu.edu.tr/cabas-user-study/, last access: 21.01.2022.

systematic review. In this study, we investigated the effect of context based on the following dimensions: environment (physical context), mobility (physical context), social, multitasking (task context), and distraction (task context). We used the following questions to collect context labels in participants' perspectives:

- Which one of these best describes your current location? *(environment)*
- Which one of these best describes your mobility condition? *(mobility)*
- Which one of these best describes people around you? *(social)*
- Did you handle any other task along with text entry? *(multitasking)*
- Is there anything that interrupted/distracted your interaction with your mobile device? *(distractions)*

We provided a set of options for each question and asked participants to select only one option at a time. For instance, the options for the environment consisted of *indoors*, *outdoors*, *stairs*, *in a vehicle*, *crosswalk*, and *others*. These options are created based on our findings from the systematic review [2] and previously conducted ESM-based research studies. Overall options available for these questions are listed in Appendix B.

3.4 Procedure

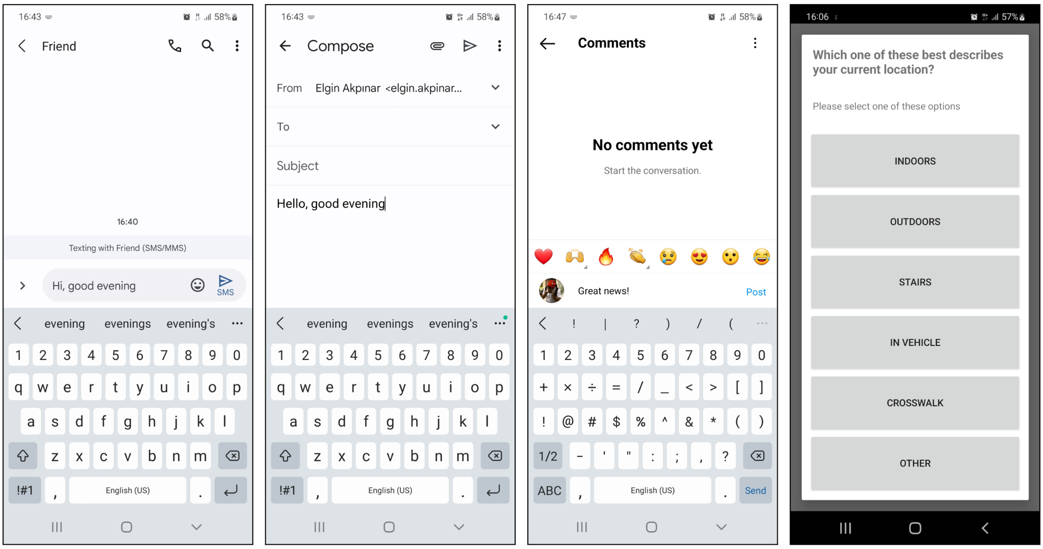
The participants were provided with a set of instructions for installing the app and registering for the study. These instructions were published online on the user study page.[3] The participants had to confirm that they read the consent form and voluntarily signed up for the study. Then, they were asked to fill a demographics form. After they completed this step, the app was activated to collect data. There was no specific task model; the participants interacted with their smartphones like they usually do. The app captured any text entered by the participants, such as while sending a text message (i.e., Samsung Messages, Figure 1(a)), composing an e-mail (i.e., Gmail, Figure 1(b)), or posting comments on social media (i.e., Instagram, Figure 1(c)). During their interactions, the app asked them to answer a set of questions regarding the current context (Figure 1(d)). The data synchronization process was fully automated; background services posted the data to the server after the interaction was completed. To quit the study, participants removed the app from their smartphones. The participants were rewarded with $10/70TL worth of a gift card from Amazon or a preferred local shopping site if they completed the study for at least a week.

3.5 Administration

This study was approved by the METU Applied Ethics Research Center with 516 ODTU 2019 protocol number.[4] In the consent form, it was clearly stated that the participation was voluntary. Moreover, we also stated that we would not collect the content of the password fields and share or publish the textual content collected during the study. We indicated that the data would be evaluated with an automated process for academic purposes only. We ensured that the questions used during the study would not include questions that would cause personal discomfort. We stated that any participant could leave the study for any reason by just removing the app. Finally, we explained how to pause and resume data collection if the participants had any privacy concerns.

We adopted the Snowball Sampling technique and started our user experiment with personal contacts on July 27th, 2020. Then we announced the study on social media including Facebook, Instagram, and LinkedIn, and via various email groups. The study was designed to be conducted fully remotely. We instructed participants if they had problems with the setup and warned them if

[3] https://iam.ncc.metu.edu.tr/cabas-user-study-instructions/, last access: 21.01.2022.
[4] http://ueam.metu.edu.tr/, last access: 20.12.2021.



Hi, good evening

From Elgin Akpınar <elgin.akpinar...

Hello, good evening

No comments yet

Start the conversation.

Great news!

Which one of these best describes your current location?

Please select one of these options

(a) Sending a text message (b) Composing an email (c) Posting on social media (d) Sample ESM question

Fig. 1. Sample text entry activities captured and ESM question.

there was a problem with the data flow. We also notified them when one week period of the study was over. The study was conducted and administered for 58 days and completed on September 22nd, 2020.

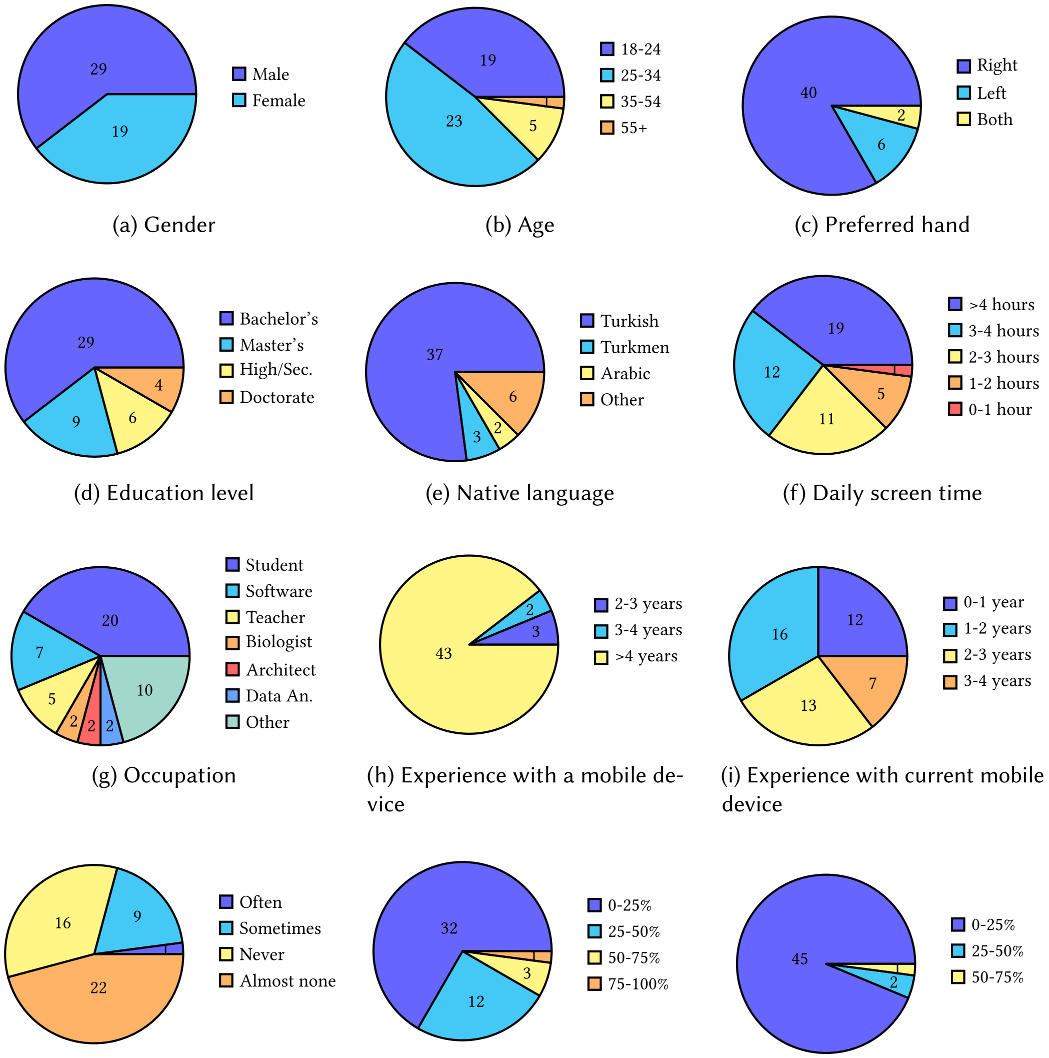
3.6 Participation and Demographics

Overall, 55 participants downloaded and installed our app on their devices. Seven participants either had a technical problem or decided not to participate in the study; thus, they uninstalled the app within the same day of installation. Other 48 participants kept the app installed from 3 days to 10 days (mean is 7.3 days and median is 7 days). In our data analysis, we did not exclude any of these 48 participants' data.

Figure 2 shows the demographics of the participants. Among 48 participants, 29 were male, and 19 were female. A total of 23 participants were aged between 25 and 34, 19 participants were 18–24, 5 participants were 35–54, and 1 participant was over 55. The majority of the participants (40) used their right hands as their dominant hands. A total of 29 participants had Bachelor's Degree, 9 had a Master's Degree, 6 completed high/secondary school, and 4 had a Doctorate. A total of 43 participants have been using mobile devices for more than four years.

Reported occupations included student (20), software engineer (7), teacher (5), biologist (2), architect (2), data analytics manager (2), pilot (1), QA (1), business analyst (1), network admin (1), communications manager (1), machine engineer (1), game designer (1), doctor (1), researcher (1), and DB admin (1). The majority of the participants (37) reported Turkish as their native language. Other native languages were Turkmen (3), Arabic (2), Persian (1), Urdu (1), Korean (1), English (1), Hindi (1), and Dutch (1). Finally, participants sent data from different countries, including the United States, Senegal, Mauritania, Germany, Netherlands, Belgium, Greece, Russia, Turkmenistan, India, Pakistan, Kyrgyzstan, and Kazakhstan.

All participants were smartphone users. The device sizes ranged between 5.1 and 6.67 inches (mean: 5.9, median: 6.0). None of the participants were excluded due to the device size. The diversity of the device types was unexpectedly high. Participants used 37 different models of eight



(a) Gender

(b) Age

(c) Preferred hand

(d) Education level

(e) Native language

(f) Daily screen time

(g) Occupation

(h) Experience with a mobile device

(i) Experience with current mobile device

(j) The frequency of experiencing collision with objects or other people while preoccupied with a mobile device

(k) Percent of mobile device use (talking, texting, or other) while walking

(l) Percent of mobile device use (talking, texting, or other) while crossing a street

Fig. 2. Demographic data.

brands and five different SDK versions. The keyboard apps used by the participants were Samsung Keyboard (18), Gboard—the Google Keyboard (17), and Microsoft SwiftKey Keyboard (12). Table 14 in Appendix C provides a summary of participants' devices.

We asked our participants to ignore all of the context labelling questions whenever they felt that paying attention to the questions would cause safety problems, such as while driving. The overall compliance rate to the context labelling questions is 55.32%. Maximum and minimum compliance rates among 48 participants are 100.00% and 2.34%, respectively, and the mean compliance rate among the participants is 58.68% (standard deviation is 27.56%, and median is 65.22%). Figure 3 illustrates the histogram for participants' context labels.

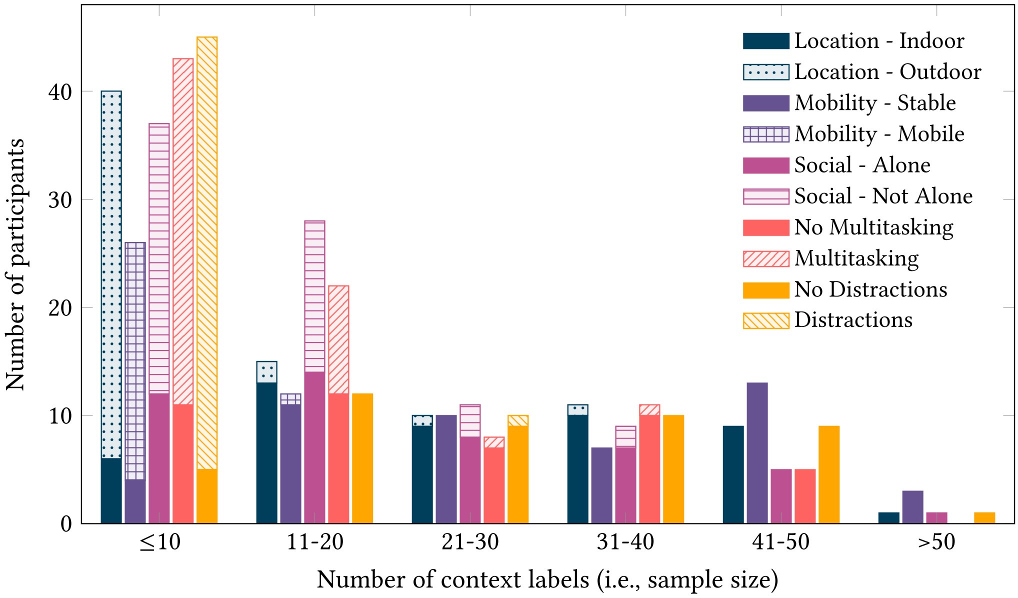


Fig. 3. Histogram for participants' context labels.

Table 2. Participants' Responses to Whether They Made a Typing Error in the Current Session

Participants' response	Count	Percent (%)
No	787	76.93
Yes	158	15.44
Maybe	78	7.62

Table 3. Responses to Typing Error Causes

Cause of Typing Error	Count	Percent (%)
No particular reason	142	60.17
Something that interrupts me	16	6.78
Other task I am busy with	15	6.36
My current mobility situation	15	6.36
People around me	11	4.66
My current location	10	4.24
Multiple of these	5	2.12
Other	18	7.63
No response	4	1.69

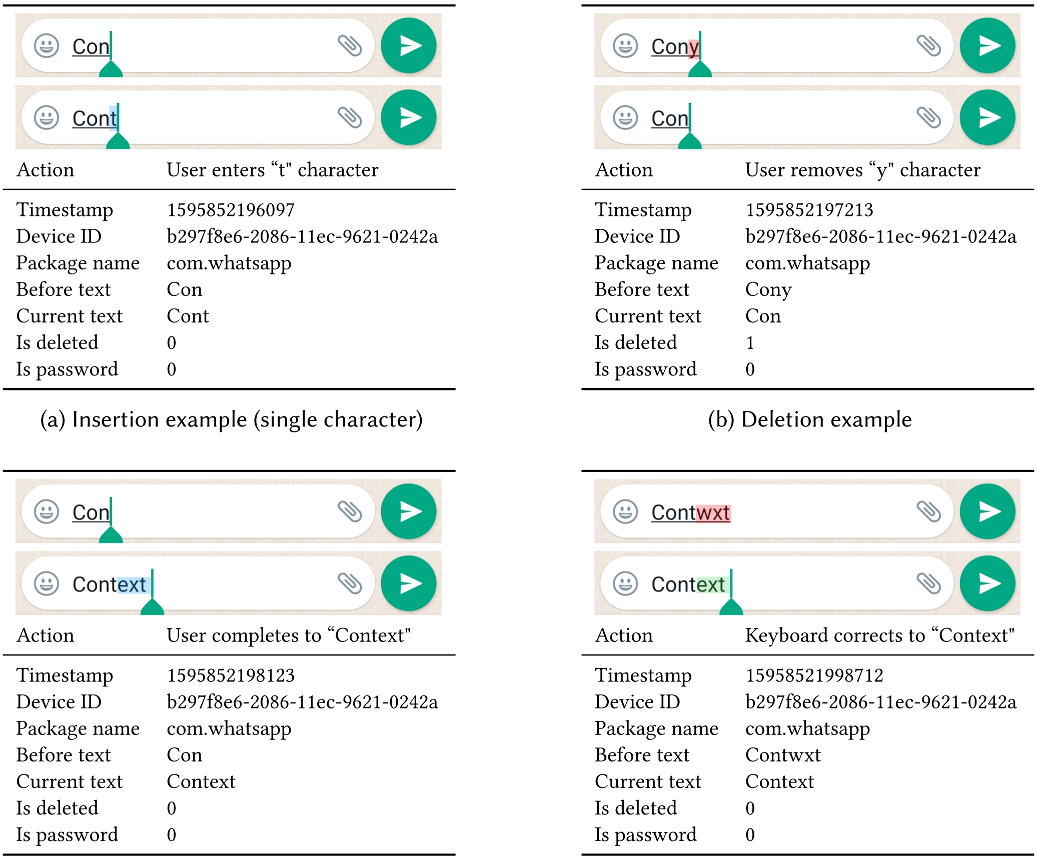
3.7 Participants' Self Evaluation on Typing Errors

After context labelling questions, we asked participants who had deleted any text during the current session whether they had made a typing error. Table 2 presents the participants' responses to this question. According to this table, the majority of text removals were not caused by a typing error.

If the participants selected yes or maybe options, we asked a further question regarding the cause of this typing error. Table 3 illustrates the participants' responses to this question. The participants indicated that there was no particular reason for their typing error in the majority of the cases.

4 USER PERFORMANCE MODELLING: DETECTION AND CORRECTION

As can be seen from the previous section, instead of transcribing the given text, in our study, participants entered text to complete their daily tasks without having a predefined task model. This section explains the techniques employed to process user data, and evaluate the users' performance and in particular, the techniques used to assess the users' typing errors and corrections.



Action	User enters “t" character
Timestamp	1595852196097
Device ID	b297f8e6-2086-11ec-9621-0242a
Package name	com.whatsapp
Before text	Con
Current text	Cont
Is deleted	0
Is password	0

(a) Insertion example (single character)

Action	User removes “y" character
Timestamp	1595852197213
Device ID	b297f8e6-2086-11ec-9621-0242a
Package name	com.whatsapp
Before text	Cony
Current text	Con
Is deleted	1
Is password	0

(b) Deletion example

Action	User completes to “Context"
Timestamp	1595852198123
Device ID	b297f8e6-2086-11ec-9621-0242a
Package name	com.whatsapp
Before text	Con
Current text	Context
Is deleted	0
Is password	0

(c) Insertion example (multiple characters)

Action	Keyboard corrects to “Context"
Timestamp	15958521998712
Device ID	b297f8e6-2086-11ec-9621-0242a
Package name	com.whatsapp
Before text	Contwxt
Current text	Context
Is deleted	0
Is password	0

(d) Substitution example

Fig. 4. Sample actions and corresponding data collected via the AWARE framework.

4.1 Data Model

The AWARE Framework logs the keyboard events at the character level and adopts the transcription sequence paradigm by capturing the entire transcription after every keyboard action [85]. A keyboard log is recorded for every keyboard interaction that either inserts or removes a character. Figure 4 illustrates the table columns for the keyboard logs and sample data for single character insertion (Figure 4(a)), single character deletion (Figure 4(b)), multiple character insertion (auto-completion, Figure 4(c)), and substitution (auto-correction, Figure 4(d)). The columns include the timestamp of the keyboard event, an ID for interacting users, and the package name of the app used during the keyboard event. Moreover, the text just before the keyboard event and the text just after the keyboard event are also logged in two separate columns. A boolean field indicates if the field that the text entered is a password field. The AWARE Framework masks the text entered into password fields; therefore, it does not collect the password phrases. We used this field to ignore such phrases in the overall process. Finally, we added a boolean field to indicate if the user is entering or deleting text. Using this data model, it is possible to generate the input stream and distinguish between the type of actions by comparing two consecutive transactions. If the simultaneous actions edit discontiguous parts of the text, they are considered as substitution [85].

4.2 Trial Identification and Tokenization

Evans and Wobbrock [21] and Nicolau et al. [51] refer to the set of keyboard interactions when participants complete a single task as trials, as in the laboratory experiments. To analyze data systematically, we have also applied several steps to segment overall keyboard data into trials. First, the overall text input stream was grouped by the participants, and then the timestamp values sorted the keyboard data list of each participant in ascending order. Then, iterating over the keyboard data lists, we compared two consecutive keyboard data to check if a new trial had started. If the participant switched to another app, we considered a new trial. Finally, we compared *before text* and *current text* values of two consecutive keyboard data. For instance, if a non-empty *current text* was followed by an *empty before* text, it indicated that the user either submitted or cleared the text just entered, and a new trial started. Unlike Evans and Wobbrock [21], and Nicolau et al. [51], we did not use screen events for starting a new trial since they may indicate an interception due to context. However, we adopted Evans and Wobbrock [21] and Nicolau et al. [51]'s approach to detect pauses. In summary, we calculated the mean time interval between keyboard events and added three standard deviations to obtain a threshold. Using the dataset obtained from 48 participants, we calculated that the mean difference between two successive non-backspaces or backspaces was 285 ms, a non-backspace following a backspace was 742 ms, and a backspace following a non-backspace was 899 ms. After adding three standard deviations to each mean value, we had 2,346 ms, 9,867 ms, and 23,189 ms, respectively, as the pause segmentation times. Overall, we segmented 938,431 keyboard interaction data into 42,018 trials.

4.2.1 Trial Validation. We excluded the trials if they only included a URL, a numeric value, a password, or a text with less than five characters. Moreover, since our participants entered text in any language they like, we also applied language criteria. We used Apache Tika tika-langdetect package[5] and language-detector library[6] for language detection. We ignored the trials other than Turkish and English. The text language was used later to determine the proper resource to check if a token was correct or a typing error. Details are given below. Overall, we excluded 9,497 (22.6%) trials.

4.2.2 Tokenization and Token Selection. We tokenized the trials by using the whitespaces. We did not use punctuation characters in tokenization to detect typing errors caused by unintentional punctuation characters between the tokens. However, we removed punctuation characters and emojis at the end of the tokens. Some types of tokens serve a particular purpose in the text; however, they are out-of-vocabulary due to their structural appearance. They are very prevalent, especially in social media, and can be listed as follows [20]:

- URLs
- E-mail addresses
- Mentions (i.e., @mention)
- Hashtags (i.e., #hashtag)
- Emojis (i.e., :D)
- Vocatives (i.e., hahaha[7])

In addition to these, we recognized serial numbers (one or two upper case letters followed by a set of numeric characters), websites or domain names (i.e., metu.edu.tr), and file names with extensions (i.e., sample.pdf). We used regular expressions to check if a token matched with one

[5] https://tika.apache.org/, last access: 29.09.2021.
[6] https://github.com/optimaize/language-detector, last access: 29.09.2021.
[7] Turkish word equivalent to lol in English.

Table 4. Overview of the Dataset in Terms of Trials and Actions

	Size	Min	Max	Mean	Median	Std. Dev	Overall
Keystrokes in trials	32,301	5	325	23.56	18.00	20.40	760,980
Characters entered in trials	32,301	5	302	24.31	19.00	19.34	785,383
Participants' daily trials	384	1	1,220	84.12	38.00	121.05	32,301
Participants' overall trials	48	46	3,307	672.94	369.50	776.84	32,301
Insertions	48	800	68,018	14,508.79	9,305.00	16,548.20	696,422
Deletions	48	55	4,830	1,151.21	652.50	1,328.65	55,258
Substitutions	48	0	761	108.23	28.00	173.46	5,195
Auto completions	48	0	952	73.60	27.00	162.18	3,533

of these cases. When our implementation found a match, it excluded the token from the dataset, similar to Han and Baldwin [31]. We excluded 4,574 tokens and had 135,254 valid tokens overall with 38,768 distinct tokens. Table 4 illustrates an overview of the dataset in terms of trials and actions after trial and token validations.

4.3 Error Detection

The previous section explained how we segmented the input stream into trials and tokenized each trial. The next step was to process these tokens to calculate the performance metrics. First, we needed to identify if a token had a typing error. Then, we had to distinguish between typing error corrections and edits when participants removed some characters and reentered new text. The final transcribed text does not contain the removed part when a user corrects a typing error. As a result, typing speed also decreases while the error rate increases. On the other hand, when users change their minds and decide to write something else, the overall transcribed text also includes the removed part as it was written intentionally. In this case, there is no adverse effect on typing speed and error rate. Therefore, we had to distinguish between these two cases to measure better the metrics related to typing speed.

Algorithm 1 represents the pseudocode to validate a token. To check if a token has a typing error, we used several resources. First, we used Hunspell spellchecker [44] as it has been widely used in similar studies and supports multiple languages, including Turkish. Moreover, we checked if a token appeared in METU Turkish Corpus [63], a collection of 2 million words of Turkish text. Finally, we used the spellchecker implementation of the Zemberek project [3]. We only used METU Turkish Corpus and spellchecker of Zemberek if the participant's native language or text language was Turkish. If a token was identified as correct in one of these tools, it was accepted as a correct word without any typing error. In addition to these, we used several resources for lookup purposes. These resources include location names and country codes [86], Turkish abbreviations,[8] and a set of Turkish slang and text speak words [20]. Finally, we used Hunspell and Zemberek suggestions for vowel restoration.

Daily conversations or social media posts may also include some out-of-vocabulary but valid words, such as brand names, social media accounts, or technical terms. Even if a user intends to type such words, offline resources fail to identify these words as correct. Therefore, we used Bing Spell Check and Search APIs, similar to Evans and Wobbrock [21]. To reduce the number of calls to these APIs, we only sent requests for tokens that offline tools could not recognize. Moreover, we did not send the overall text content of the trial for the spell checking; we only sent a single token at a time. Finally, we used additional query options such as filtering results for Urban

[8] https://tdk.gov.tr/wp-content/uploads/2019/01/K%c4%b1saltmalar_Dizini.pdf, last access: 02.10.2021.

ALGORITHM 1: Algorithm to Validate a Token

```
Require: $token \neq$ “”
procedure isValid ($token, lang$)
    $hunspellInstance \leftarrow Hunspell.instance(lang)$
    if $hunspellInstance.isValid(token)$ then return TRUE
    end if
    if $lang =$ “$tr$” then
        if $zemberek.isValid(token)$ then return TRUE
        else if $metuCorpus.isValid(token)$ then return TRUE
        else if $addressLookup.contains(token)$ then return TRUE
        else if $abbreviationLookup.contains(token)$ then return TRUE
        else if $textSpeakLookup.contains(token)$ then return TRUE
        end if
    else if $lang =$ “$en$” then
        if $textSpeakLookup.contains(token)$ then return TRUE
        end if
    end if
    if $Bing.query(token, options =$ “$spellcheck : true$”$) \neq$ EMPTY then return TRUE
    else if $Bing.query(token, options =$ “$site : tureng.com$”$) \neq$ EMPTY then return TRUE
    else if $Bing.query(token, options =$ “$site : urbandictionary.com$”$) \neq$ EMPTY then return TRUE
    end if
     return FALSE
end procedure
```

Dictionary[9] and Tureng Multilingual Dictionary[10] sites to retrieve specific search results. Urban Dictionary is a crowdsourced resource and can be used to check the words in English slang and daily language [75]. Tureng Dictionary is a Turkish and English dictionary [72], and it makes use of resources in many different fields, such as engineering, law, and medicine.

Overall, we combined the approaches of Evans and Wobbrock [21], Nicolau et al. [51], and Torunoğlu and Eryiğit [70]. To check if a token is valid in the corresponding language, we mainly followed the approaches of Evans and Wobbrock [21] and Nicolau et al. [51], except for the manual analysis. To identify text-speak words, we followed Torunoğlu and Eryiğit [70]. Moreover, we converted the words to lower, upper and proper noun cases and checked if they were valid. We applied a set of transition rules on the tokens. For instance, we removed repeating characters and checked if the resulting word was valid. Table 5 presents these rules with corresponding algorithms and examples. If the transformed word was valid, then it was accepted as correct.

We considered the following cases as typing errors:

— transposition errors (i.e., cont[xe]t),
— punctuation marks separating two words without any whitespace (i.e., context[.]factor),
— invalid tokens becoming valid after changing some characters with adjacent characters on the keyboard (i.e., cont[r]xt),
— tokens with missing or extra characters with respect to Hunspell and Zemberek suggestions (i.e., cont[]xt, conte[r]xt),
— one of the adjacent characters to spacebar separating two words (i.e., context[n]factor),
— two consecutive words as a token without any whitespace (i.e., context[]factor).

[9] https://www.urbandictionary.com/, last access: 14.01.2022.
[10] https://tureng.com/, last access: 14.01.2022.

Table 5. Token Validation Rules

Rule	Description	Algorithm		Examples
Case alternatives	Tokens that become valid after converting to lower, upper, and proper noun cases	**return** $isValid(toLower(token))$ **or** $isValid(toUpper(token))$ **or** $isValid(toProper(token))$	*en.* *tr.* *en.*	usa → USA ankara → Ankara COME → come
Dialectical or accent use	Tokens that are written in informal forms in text speak and become valid after applying dialectical and accent transitions	$dSet \leftarrow \{dialectSet\}$ $tSet \leftarrow \{transitionSet\}$ **for** $i = 0; i < dSet.length; i{+}{+}$ **do** … **if** $token.contains(dSet[i])$ **then** … … $token.replace(dSet[i], tSet[i])$ … … **if** $isValid(token)$ **then** … … … … **return** $true$ **return** $false$	*tr.* *tr.* *tr.* *en.*	yapcaz → yapacağız (we will do) yapıyom → yapıyorum (I am doing) yapmicam → yapmayacağım (I won't do) goin → going
Repeating characters	Tokens that become valid after removing repetitive characters, that are generally used for expressing emotions	**for** $i = 1; i < token.length; i{+}{+}$ **do** … **if** $token[i] = token[i-1]$ **then** … … $n \leftarrow token.remove(i)$ … … **if** $isValid(n)$ **then return** $true$ **return** $false$	*tr.* *en.*	evettttt → evet (yes) hiiiii → hi
Deasciification	Tokens that become valid after applying deasciification, to detect use of “i”, “o”, “u”, “c”, “g”, and “s” instead of “ı”, “ö”, “ü”, “ç”, “ğ”, and “ş” characters	$ascii \leftarrow \{i, o, u, c, g, s\}$ $tr \leftarrow \{ı, ö, ü, ç, ğ, ş\}$ **for** $i = 0; i < ascii.length; i{+}{+}$ **do** … $d \leftarrow token.replace(ascii[i], tr[i])$ … **if** $isValid(d)$ **then return** $true$ **return** $false$	*tr.* *tr.*	Turkce → Türkçe (Turkish) isik → ışık (light)
English and French words	English and French words in a non-English or non-French text	**return** $isValid(token, “en”)$ **or** $isValid(token, “fr”)$	*en.* *en.* *fr.*	playlist data voilà

Proper nouns	Proper nouns with missing apostrophes, generally ignored in text speak	**for** $i = 1; i < token.length; i{+}{+}$ **do**
		… $n \leftarrow token.put(i, “'”)$
		… **if** $isValid(n)$ **then return** $true$
		return $false$
			tr.
			tr.
			en.
			en.
				Elginin → Elgin'in
				Ankaraya → Ankara'ya
				Elgins → Elgin's
				Ill → I'll

Phonetic substitution	Tokens that are intentionally corrupted by replacing some characters with phonetically similar forms or nonalphabe-tic characters	$pSet \leftarrow \{phoneticRuleSet\}$ $tSet \leftarrow \{transitionSet\}$ **for** $i = 0; i < pSet.length; i{+}{+}$ **do** … **if** $token.contains(pSet[i])$ **then** … $token.replace(dSet[i], tSet[i])$ … … **if** $isValid(token)$ **then** … … … …**return** $true$ **return** $false$	*tr.* *tr.* *tr.* *tr.* *en.*	kardeshim → kardeşim (my sister/brother) qanqa → kanka (dude) \$eker → Şeker (Sugar) yawrum → yavrum (my little one) c@ → cat

Misspelled conjunction	Tokens that ends with a frequently misspelled conjunction	**for** c **in** $conjunctionSet$ **do**
		… **if** $token.endsWith(c)$ **and** $isValid(token.remove(s), “tr”)$
		… … **then return** $true$
		return $false$
			tr.
			tr.
				tamammı → tamam mı (is it OK)
				alırmısın? → alır mısın? (would you take?)

Frequents	Frequent spelling mistakes	$fmSet \leftarrow \{frequentMistakesSet\}$ **return** $fmSet.contains(token)$	*tr.* *en.*	yalnış → yanlış (wrong) succesful → successful
Removing vowels	Tokens constructed by removing vowels from a valid token	$suggests \leftarrow suggestions(token)$ **for** s **in** $suggests$ **do** … **if** $token = s.removeVowels()$ … … **then return** $true$ **return** $false$	*tr.* *tr.* *en.*	tmm → tamam (OK) slm → selam (hi) msg → message

Neologism	Non-Turkish words followed by a Turkish suffix	**for** s **in** $suffixSet$ **do**
		… **if** $token.endsWith(s)$ **and** $isValid(token.remove(s), “en”)$
		… … **then return** $true$
		return $false$
			tr.
			tr.
			tr.
			tr.
				hack-lemek (hacking)
				item-ler (items)
				edit-lemek (editing)
				pick-leyip (picking)

To distinguish between edits and error corrections, we first checked for adjacent character errors similar to Nicolau et al. [51]. For this purpose, we modified the minimum string distance calculation to accept two characters as equal if they are adjacent on the keyboard. If this new distance value is zero but removed and reentered texts are different, it is accepted as a correction of adjacent character error. In addition to the method of Nicolau et al. [51], we also checked for transposition, missing and extra character, bounce (repetition), and wrong character errors. We detected the difference between removed and reentered text. If the removed text segment is the reverse of the reentered text segment, it is considered a correction of a transposition error. If the removed text segment and reentered text segment have only one character, it is considered missing, extra, or wrong character error. If the removed text segment is a repetition of a single character, it is considered a bounce error correction. We observed that unintentional space characters and punctuation were commonly corrected. Finally, we applied suggestion checks similar to Nicolau et al. [51], and Evans and Wobbrock [21].

According to Zhang et al. [84], using auto-correction could help to prevent typos; while, it may also result in typos. Unfortunately, Nicolau et al. [51] and Evans and Wobbrock [21] did not consider the effect of auto-correction on typing errors. If the user removed an auto-corrected text, we considered it as an error correction. We also checked if the removed text is valid. If so, we considered it as an edit. Finally, we calculated the edit distance between removed and reentered text. If the edit distance is more than half of the lengths of both texts, we considered it an edit. Otherwise, it was classified as an error correction. According to Arif and Stuerzlinger [4]'s experiments, half of the users correct typing errors immediately (character-level), and the other half correct after a few keystrokes (word-level). The majority of the users that apply word-level correction correct after two to five characters. For this reason, we applied this method to both in-text and end-of-text replacements.

Algorithm 2 represents the pseudocode to distinguish between error corrections and edits. The corresponding procedure accepts non-empty removed and reentered text and a boolean value to indicate whether an auto-correction event occurred within the removed text's typing process. We compared the current text with the before text value to check this. If the minimum string distance between the current text and the before text is more than one, it indicates either an auto-complete or an auto-correction. If the current text starts with the before text, it is an auto-complete event (see Figure 4(c)); otherwise, it is an auto-correction event (see Figure 4(d)).

Out of 21,683 text changes, we classified 18,192 (83.9%) error corrections and 3,491 (16.1%) edits. Participants corrected errors 379 times on average (standard deviation: 480.62, median: 204.5) and edited 72 times (standard deviation is 82.81, median is 46).

4.4 Evaluation

Before using our findings to investigate the effect of context on users' typing performance, we had to evaluate our error detection implementation. Due to our commitments in the consent form (see Section 3.5), we conducted a follow-up study with the same participants in our first user study and asked them to evaluate our implementation on the data they sent during the first experiment. As we have indicated in the consent form, we only automatically processed their data.

4.4.1 Procedure. In this follow-up study, we automatically prepared Excel files that included user data and our system's classification typing error and correction. These files are used to collect users' feedback such that we could compare users' feedback with the system's classification. This study mainly included the following three parts:

(1) Invitation: We sent invitation e-mails to the participants who provided their e-mail addresses (46 participants). We briefly explained the purpose of the study and asked the

ALGORITHM 2: Algorithm to distinguish between error corrections and edits

```
Require: $removed \neq$ “”, $reentered \neq$ “”
procedure errorCorrectionOrEdit ($removed, reentered, autoCorrection$)
    $msd_{adj} \leftarrow MSD_{adj}(removed, reentered)$          ▷ Adjacent characters are accepted as equal in MSD
    if $msd_{adj} = 0$ thenreturn CORRECTION          ▷ Adjacent character error
    else if $startsWith(removed, “z”)$ and $startsWithUppercase(reentered)$
      and $startsWith(toLowerCase(reentered), toLowerCase(removeFirst(removed)))$ then
      return CORRECTION          ▷ Failing to switch to uppercase error
    end if
    $removed_{diff} \leftarrow getDifference_{removed}(removed, reentered)$          ▷ Consider only the difference
    $reentered_{diff} \leftarrow getDifference_{reentered}(removed, reentered)$
    if $removed_{diff} = reverse(reentered_{diff})$ then
      return CORRECTION          ▷ Transposition error
    else if removeSpaces($removed_{diff}$) $= removeSpaces(reentered_{diff})$ then
      return CORRECTION          ▷ Missing space error
    else if removeRepeatedChars($removed_{diff}$) $= removeRepeatedChars(reentered_{diff})$ then
      return CORRECTION          ▷ Bounce error
    else if $length(removed_{diff}) = 0$ and $length(reentered_{diff}) = 1$ then
      return CORRECTION          ▷ Missing character error
    else if $length(removed_{diff}) = 1$ and $length(reentered_{diff}) = 0$ then
      return CORRECTION          ▷ Extra character error
    else if $length(removed_{diff}) = 1$ and $length(reentered_{diff}) = 1$ then
      return CORRECTION          ▷ Wrong character error
    else if $getHunspellSuggestions(removed).contains(reentered)$ then
      return CORRECTION
    else if $getZemberekNormalizations(removed).contains(reentered)$ then
      return CORRECTION
    else if $autoCorrection$ and $startsWith(reentered, beforeCorrection(removed))$ then
      return CORRECTION          ▷ Auto-correction error
    else if $MSD(removed, reentered) > (length(removed) + length(reentered))/4$ then
      return EDIT          ▷ Edit by distance
    else if $!autoCorrection$ and $isValid(removed)$ then
      return EDIT          ▷ Edit by removing a correct word
    end if
      return CORRECTION
end procedure
```

participants to reply if they agree to participate voluntarily in the follow-up study. We did not offer compensation for this follow-up study. For the analysis of this evaluation, the participants were asked to permit to process the responses manually. They were free to leave the study anytime they wanted. Moreover, we asked them to remove any text from the file without changing the row order if they feel uncomfortable sharing it.

(2) Uncorrected Error Detection Task: For the first section, we randomly selected 10 words that our system classified as correct and 10 words that our system classified as typing errors. These were automatically chosen. Next, we listed the overall text participant entered with the selected words and asked participants to enter “F” if they think they made a typo and “T” otherwise.

(3) Edits & Error Correction Detection Task: In the second section, we selected 10 cases classified as edit and 10 cases classified as error correction. These were again automatically chosen. Next, we listed the overall text participant entered with the removed and

Table 6. Evaluation Results of the Follow-Up Study

	Proposed Approach		Nicolau et al. [51]		Evans and Wobbrock [21]	
	Error Detection	Error Corr. Detection	Error Detection	Error Corr. Detection	Error Detection	Error Corr. Detection
Accuracy	0.797	0.761	0.661	0.744	0.651	0.460
Sensitivity	0.818	0.726	0.979	0.671	0.839	0.234
Specificity	0.789	0.849	0.536	0.918	0.577	0.993
Precision	0.603	0.922	0.453	0.951	0.438	0.987
F1 Score	0.694	0.782	0.620	0.760	0.576	0.525

reentered texts and asked participants to enter "F" if they think they corrected an error and "T" otherwise.

4.4.2 Material. For both sections, we only selected Turkish and English texts. To help participants to remember the context, we provided the overall text participant entered. Moreover, we selected the texts with at least three words. For the words and cases to be evaluated, we selected the words with at least three characters. The Excel files were created automatically and sent to participants without manual revision.

We did not provide the verdict of our system in the Excel file that we sent to participants. Moreover, the words and cases were randomly listed in the Excel file so that participants could not predict the system verdict. In a separate file, we saved the system verdict in the same order in the Excel file. We asked participants not to change the order of the words and cases to match the system verdict with the participant response. We provided the instructions with relevant examples to better guide the participants.

4.4.3 Study Duration and Participation. We sent the initial invitation on April 13th, 2021. As of April 19th, 2021, we sent the Excel files to all participants who responded positively. The overall evaluation process was completed on May 15th, 2021. 30 of 46 participants agreed to participate in the follow-up study. We had to eliminate one participant since there was not enough text in Turkish and English. One participant changed their mind and decided not to participate due to their busy schedule. Two participants did not respond after we sent the Excel file. Overall, we received evaluation results from 26 participants.

4.4.4 Results. We compared participants' responses to the system verdict and calculated the system accuracy. We also implemented the approaches proposed by Evans and Wobbrock [21] and Nicolau et al. [51] to compare our results with the literature. Table 6 presents the evaluation results. According to these results, our system has higher accuracy than Evans and Wobbrock [21], and Nicolau et al. [51]'s approaches. They are more sensitive since they classify the words that do not appear in offline lexicon and online query services. On the other hand, this results in lower specificity.

We created confusion matrixes to analyze the results of the evaluation. When deciding on the error rule set, one of our assumptions was that there must be a space character after punctuation characters. Our system classified 26 cases as typing errors in the evaluation dataset. However, participants labelled 21 of these cases as correctly spelt text. Moreover, we used Hunspell suggestions and Tureng query results, which resulted in 10 and 13 incorrect classifications, respectively.

We compared removed text with the reentered text of the same length to detect edits and error corrections similar to Evans and Wobbrock [21], and Nicolau et al. [51]. However, this resulted in higher string distances in case of unintentional or missing characters. Moreover, we assumed that if the removed text consists of valid words, it was an edit. Unfortunately, participants labelled 29

Table 7. Evaluation Results of the Revised System

	Error Detection	Error Correction Detection
Accuracy	0.913	0.871
Sensitivity	0.923	0.881
Specificity	0.909	0.849
Precision	0.800	0.935
F1 Score	0.857	0.813

of such cases as error correction. Finally, we observed that auto-completed text replacement might not indicate an error correction in all cases.

Based on these observations, we updated the above rules to increase the system's accuracy. For error detection, we accepted commonly made mistakes as correct since the participants may have written them intentionally (i.e., *tommorow-tomorrow* (English) or *lavobo-lavabo* (Turkish)). Moreover, we assumed that punctuations should follow the last word without a space character, and a space character must be inserted after the punctuation. However, some participants stated that they either put no space character after the punctuation or intentionally put a space character before the punctuation. Finally, we assumed that if a participant used any Turkish characters in a trial, any deasciified character corresponds to a typing error. However, we observed that some participants deasciified specific Turkish characters while using the others without deasciification. Therefore, we relaxed this assumption. To distinguish between error detection and edits, we calculated the edit distance between the removed text and the reentered text with the same length. However, this method failed with the missing character problems. We moved forward in the reentered text as long as the edit distance decreased. In some cases, the participants unintentionally tapped on adjacent characters while switching to uppercase mode. We implemented a rule for these cases. Finally, we assumed that it was an edit if both removed and reentered text were correct words. However, the participant responses showed that it was not a valid assumption. After these changes, we compared the new verdicts with the participant responses. Table 7 presents the evaluation results of the modified system.

This section began by describing error and edit/correction detection mechanisms. It went on to describe the process of the follow-up study to evaluate these mechanisms and their results. The following section presents the statistical procedures and the results obtained from them to investigate the effect of context on user performance by using the data we collected in our main user study explained in Section 3.

5 THE EFFECT OF THE CONTEXT ON USER PERFORMANCE

After we completed error and edit/error correction detection mechanisms and evaluated them, we investigated the effect of the context on user performance in text entry tasks. This section explains the procedure and results of this investigation.

5.1 Design and Procedure

In Section 2.1, the metrics for typing performance were identified. We used those four metrics in our investigation as dependent variables. We calculated the total error rate for ER metric by summing up the corrected and uncorrected error rates. Moreover, we accepted intentional errors due to text-speak as correct and did not include them in the ER calculation. For independent variables, we used participants' responses to context labels in five dimensions: environment, mobility, social, multitasking, and distraction. Table 8 shows the groups for the independent variables and corresponding context labels. We excluded the participants' data from the dataset of the contexts

Table 8. Independent Variables and Corresponding Context Labels

Context	Groups	Options
Environment	Indoor	Indoors, In vehicle
	Outdoor	Outdoors, Crosswalk
Mobility	Stable	Lying down, Sitting, Standing
	Mobile	Walking, Running
Social	Alone	Alone
	Not Alone	With 2–4 friends/family members/colleagues, With a friend/family member/colleague, With more than 4 friends/family members/colleagues, With strangers (crowded), With strangers (not crowded)
Multitasking	Nothing	Nothing
	Multitasking	I am carrying a box/bag/other, I am doing home-activities (cleaning, cooking, etc), I am having a conversation with someone around me, I am having breakfast/lunch/dinner, I am shopping, I am trying to avoid collision while walking, I am working, Multiple of these
Distractions	Nothing	Nothing
	Multitasking	I am in a hurry, I am interrupted by someone, I am interrupted by something unexpected, I need to check something from time to time, There are obstacles/people/cars on walking path, Multiple of these

if the participant's context labels did not include samples for two groups. For instance, if participants did not provide samples for both indoor and outdoor groups, we excluded their data from all statistical calculations regarding environment context. We calculated the performance metrics for each sample and associated them with the contextual labels users have assigned.

Our study did not provide a predefined task. The participants have interacted with their smartphones as in their daily lives. Some participants have spent more time with their smartphones and entered text more than the others. Figure 5 illustrates the histogram for the number of trials each participant made during the study. Moreover, a Kruskal-Wallis H test showed that there was a statistically significant difference between the participants' performances, in terms of WPM ($\chi^2(47) = 6923.066$, $p < 0.0001$), KSPS ($\chi^2(47) = 8563.796$, $p < 0.0001$), KSPC ($\chi^2(47) = 1630.444$, $p < 0.0001$), and ER ($\chi^2(47) = 1156.542$, $p < 0.0001$). Comparing the samples of each context group in such an unbalanced data could cause biases in the results. Therefore, we calculated mean and median values of each metric for all participants under different context groups.

Further statistical analysis showed that typing speed metrics (WPM and KSPS) were normally distributed for most participant and context group pairs. In contrast, error rate metrics (KSPC and ER) significantly deviated from a normal distribution for all participant and context group pairs. The histograms for both KSPC and ER were in the long tail form. A KSPC value of 1 means no correction and corresponds to 56.1% of the cases in the data. Similarly, an ER value of 0 means no uncorrected typing error and corresponds to 59.5% of the cases. WPM and KSPS, on the other hand, had no such values that dominated the sample. Moreover, WPM and KSPS were measured based on the text length and duration of the corresponding trial. On the contrary, KSPC and ER were calculated based on our error detection implementation results. The mean for these metrics would result in a poor estimate of central tendency, while the median would yield more valid results [11]. As a result, we used the mean values of WPM and KSPS and median values of KSPC and ER to investigate context effects on user performance. In our statistical analysis, the value of

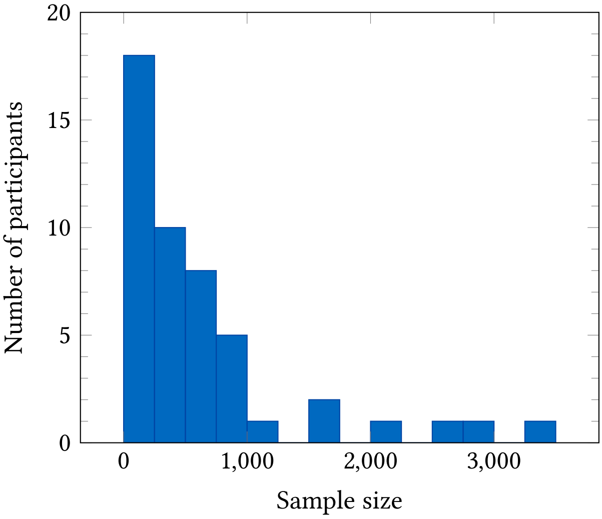


Fig. 5. Histogram for participants' sample sizes.

each performance metric under one context group was compared to the other group in a pairwise manner on each context dimension. Therefore, the p-value was adjusted using the Bonferroni correction method to reduce Type-I errors [43] and divided by the number of pairwise comparisons (0.05/5 = 0.01).

Our statistical analysis first checked if the data were normally distributed for all groups for each context factor. We conducted Kolmogorov-Smirnov and Shapiro-Wilk tests. We used the Wilcoxon Signed-Rank Test when the test results showed that the data significantly deviated from a normal distribution. Otherwise, we used Paired T-Test to compare the user performance under two context factors. All tests were conducted in 95% confidence intervals.

5.2 Research Questions

We addressed the following research questions in our investigation:

- **R1 – Environment:** *Does being in an outdoor environment affect text entry performance in terms of typing speed (WPM and KSPS), and error rate (KSPC and ER) compared to being in an indoor environment?*
- **R2 – Mobility:** *Does walking affect text entry performance in terms of typing speed (WPM and KSPS), and error rate (KSPC and ER) compared to being stable?*
- **R3 – Social context:** *Does the presence of other people around affect text entry performance in terms of typing speed (WPM and KSPS), and error rate (KSPC and ER) compared to being alone?*
- **R4 – Multitasking:** *Does multitasking affect text entry performance in terms of typing speed (WPM and KSPS), and error rate (KSPC and ER) compared to having no multitasking?*
- **R5 – Distractions:** *Does the presence of distractions affect text entry performance in terms of typing speed (WPM and KSPS), and error rate (KSPC and ER) compared to having no distractions?*

5.3 Results

Tables 9 and 10 present the results of our investigation. Table 11 summarizes these results for each context and performance metric.

Table 9. Paired T-Test Results for the Effect of Context on Users' Mean WPM and KSPS Values

Context	Metric	Group	N	Mean	Median	Std. dev.	Results
Environment	WPM	Indoor	31	41.832	41.975	6.834	$t(30) = 0.499, p = 0.622$
		Outdoor	31	41.307	41.155	7.467	
	KSPS	Indoor	31	3.670	3.688	0.604	$t(30) = -0.079, p = 0.938$
		Outdoor	31	3.677	3.663	0.646	
Mobility	WPM	Stable	15	45.466	46.047	5.501	$t(14) = -0.912, p = 0.377$
		Mobile	15	46.785	44.064	9.441	
	KSPS	Stable	15	4.011	4.032	0.460	$t(14) = -0.517, p = 0.613$
		Mobile	15	4.069	3.894	0.763	
Social	WPM	Alone	38	42.281	42.203	6.868	$t(37) = 1.001, p = 0.323$
		Not Alone	38	41.500	41.225	6.636	
	KSPS	Alone	38	3.711	3.731	0.606	$t(37) = 0.614, p = 0.543$
		Not Alone	38	3.670	3.636	0.619	
Multitasking	WPM	Nothing	35	41.461	42.049	7.287	$t(34) = -1.377, p = 0.178$
		Multitask	35	42.635	43.364	6.415	
	KSPS	Nothing	35	3.644	3.697	0.609	$t(34) = -2.217, p = 0.033$
		Multitask	35	3.793	3.850	0.569	
Distractions	WPM	Nothing	35	41.987	41.959	6.836	$t(34) = -0.169, p = 0.867$
		Multitask	35	42.145	41.941	8.290	
	KSPS	Nothing	35	3.704	3.726	0.597	$t(34) = -0.364, p = 0.718$
		Multitask	35	3.732	3.712	0.727	

N Sample size, $p < 0.01$ to show the significance level.

Table 10. Wilcoxon Signed-Rank Test Results for the Effect of Context on Users' Median KSPC and ER Values

Context	Metric	Group	N	Mean	Median	Std. dev.	Results
Environment	KSPC	Indoor	31	1.013	1.000	0.028	$Z = 82.0, p = 0.001^{*}$
		Outdoor	31	1.052	1.024	0.083	
	ER	Indoor	31	0.311	0.000	1.028	$Z = 42.0, p < 0.0001^{**}$
		Outdoor	31	1.217	0.000	1.822	
Mobility	KSPC	Stable	15	1.016	1.000	0.033	$Z = 30.0, p = 0.095$
		Mobile	15	1.040	1.031	0.045	
	ER	Stable	15	0.442	0.000	1.256	$Z = 10.0, p = 0.003^{*}$
		Mobile	15	1.057	0.000	1.443	
Social	KSPC	Alone	38	1.013	1.000	0.031	$Z = 115.0, p < 0.0001^{**}$
		Not Alone	38	1.028	1.008	0.038	
	ER	Alone	38	0.303	0.000	0.932	$Z = 96.0, p < 0.0001^{**}$
		Not Alone	38	0.593	0.000	1.153	
Multitasking	KSPC	Nothing	35	1.027	1.000	0.063	$Z = 157.0, p = 0.009^{*}$
		Multitask	35	1.042	1.014	0.059	
	ER	Nothing	35	0.494	0.000	1.410	$Z = 129.0, p = 0.002^{*}$
		Multitask	35	1.223	0.000	2.160	
Distractions	KSPC	Nothing	35	1.017	1.000	0.034	$Z = 81.0, p < 0.001^{*}$
		Multitask	35	1.052	1.008	0.081	
	ER	Nothing	35	0.467	0.000	1.136	$Z = 99.0, p < 0.001^{*}$
		Multitask	35	1.460	0.000	2.122	

N Sample size, $*p < 0.01$, $**p < 0.0001$.

Table 11. The Effect of Context on User Performance (↓: Decreased, ↑: Increased, ⊘: no Significant Effect)

		Typing Speed		Error Rate	
Context	Factor	WPM	KSPS	KSPC	ER
Environment (indoor/outdoor)	Being outdoors	⊘	⊘	↑	↑
Mobility (stable/mobile)	Being mobile	⊘	⊘	⊘	↑
Social (alone/not alone)	Presence of other people	⊘	⊘	↑	↑
Multitasking (with/without multitask)	Presence of multitasking	⊘	⊘	↑	↑
Distraction (with/without distraction)	Presence of distraction	⊘	⊘	↑	↑

R1 – Environment. Environment of the participant significantly affects user performance in terms of KSPC (Z = 82.0, p = 0.001) and ER (Z = 42.0, $p < 0.0001$). Participants in outdoor condition had higher KSPC (1.052 ± 0.083) than participants in indoor condition (1.013 ± 0.028). Similarly, ER was higher for outdoor condition (1.217 ± 1.822) than indoor condition (0.311 ± 1.028). Environment does not significantly affect user performance in terms of WPM (t(30) = 0.499, p = 0.622) and KSPS (t(30) = −0.079, p = 0.938).

R2 – Mobility. Mobility of the participant significantly affects user performance in terms of only ER (Z = 10.0, p = 0.003). ER was lower for stable condition (0.442 ± 1.256) than mobile condition (1.057 ± 1.443). Mobility does not significantly affect user performance in terms of WPM (t(14) = −0.912, p = 0.377), KSPS (t(14) = −0.517, p = 0.613), and KSPC (Z = 30.0, p = 0.095).

R3 – Social context. Social context significantly affects user performance in terms of KSPC (Z = 115.0, $p < 0.0001$) and ER (Z = 96.0, $p < 0.0001$). The presence of other people resulted in higher KSPC (1.028 ± 0.038) than participants in alone condition (1.013 ± 0.031). Similarly, ER increased with the presence of other people (0.593 ± 1.153) compared to alone condition (0.303 ± 0.932). The presence of other people does not significantly affect user performance in terms of WPM (t(37) = 1.001, p = 0.323) and KSPS (t(37) = 0.614, p = 0.543).

R4 – Multitasking. Multitasking significantly affects user performance in terms of KSPC (Z = 157.0, p = 0.009) and ER (Z = 129.0, p = 0.002). Multitasking resulted in higher KSPC (1.042 ± 0.059) than no multitasking (1.027 ± 0.063). Similarly, ER was higher for multitasking conditions (1.223 ± 2.160) than no multitasking condition (0.494 ± 1.410). Multitasking does not significantly affect user performance in terms of WPM (t(34) = −1.377, p = 0.178) and KSPS (t(34) = −2.217, p = 0.033).

R5 – Distractions. Distractions significantly affect user performance in terms of KSPC (Z = 81.0, $p < 0.0001$) and ER (Z = 99.0, $p < 0.0001$). Presence of distraction resulted in higher KSPC (1.052 ± 0.081) than no distraction (1.017 ± 0.034). Similarly, ER was higher for distraction condition (1.460 ± 2.122) than no distraction condition (0.467 ± 1.136). Distractions do not significantly affect user performance in terms of WPM (t(34) = −0.169, p = 0.867) and KSPS (t(34) = −0.364, p = 0.718).

Task context. Participants entered text in 231 different apps during our study, and 139 apps left after trial and token validations. The most frequently used apps include WhatsApp, Instagram, Messenger, Google Chrome, Tinder, and Telegram. We retrieved the category of each app by using the categories in Google Play Store.[11] Then, we grouped the apps by their categories and selected the most frequently used app categories: communication (i.e., Whatsapp), social (i.e., Instagram), tools (i.e., Google), and productivity (i.e., Notes). Finally, we investigated the effect of the category of the app used on user performance. A repeated-measures ANOVA with a Greenhouse-Geisser

[11] https://play.google.com/store/apps/details?id=com.whatsapp&hl=en_US&gl=US, last access: 21.05.2022.

correction determined that mean WPM differed statistically significantly between different app types (F(1.878, 20.663) = 3.955, p = 0.037). Participants were fastest while using a communication app (44.857 ± 1.962), slowest while using a productivity app (36.083 ± 2.604), had 42.037 ± 1.955 WPM in social apps, and had 43.549 ± 3.861 WPM in tool apps. Post hoc analysis with a Bonferroni adjustment revealed that WPM was statistically significantly increased from productivity apps to communication apps (8.774 (95% CI, 1.034 to 16.514), p = 0.036), and from social apps to communication apps (2.820 (95% CI, 0.201 to 5.439), p = 0.048), but not from productivity apps to social apps (5.954 (95% CI, −2.128 to 14.036), p = 0.295) and not from tools to others. Similarly, a repeated-measures ANOVA with a Greenhouse-Geisser correction determined that mean KSPS differed statistically significantly between app types (F(1.970, 21.672) = 4.859, p = 0.018). Participants were fastest while using a communication app (3.964 ± 0.171), slowest while using a productivity app (3.179 ± 0.203), had 3.708 ± 0.172 KSPS in social apps, and had 3.765 ± 0.319 KSPS in tool apps. Post hoc analysis with a Bonferroni adjustment revealed that KSPS was statistically significantly increased from productivity apps to communication apps (0.784 (95% CI, 0.202 to 1.366), p = 0.012), and from social apps to communication apps (0.256 (95% CI, 0.043 to 0.469), p = 0.025), but not from productivity apps to social apps (0.198 (95% CI, −0.444 to 0.840), p = 1.000) and not from tools to others. On the other hand, the effect of using different app types on error rate was not statistically significant in terms of KSPC ($\chi^2(3) = 4.480$, p = 0.214) and ER ($\chi^2(3) = 2.418$, p = 0.490).

Language context. Before participating in our study, we asked about our participants' native language. The distribution of the participants' native languages is illustrated in Figure 2(e) in Section 3.6. We also detected the language of the texts entered during the study (see Section 4.2.1 for details). Using participants' native language and the language of text they entered, we investigated the effect of using the native or a non-native language on the users' performance. Pairwise comparisons adjusted with Bonferroni showed statistically significant differences in terms of WPM (t(31) = 8.139, p < 0.0001), KSPS (t(31) = 7.641, p < 0.0001), KSPC (Z = 67.0, p < 0.01), and ER (Z = 19.0, p < 0.0001) between native and non-native language usage. The participants were faster while typing in their native languages (WPM: 43.005 ± 6.211, KSPS: 3.793 ± 0.551) than in a non-native language (WPM: 36.253 ± 7.704, KSPS: 3.241 ± 0.673). Moreover, the participants were more accurate in their native language (KSPC: 1.014 ± 0.029, ER: 0.277 ± 0.965) than in a non-native language (KSPC: 1.057 ± 0.071, ER: 1.565 ± 2.574).

Technical context. We further investigated the effect of technical context on the participants' performance in smartphone brands, screen size, and keyboards used. One-way ANOVA tests showed that there were no statistically significant differences between the participants who used smartphones in different brands in terms of WPM (F(7,40) = 0.740, p = 0.639) and KSPS (F(7,40) = 0.963, p = 0.471). Similarly, Kruskal-Wallis H tests showed that the differences between smartphone brands in terms of KSPC ($\chi^2(7) = 10.853$, p = 0.145) and ER ($\chi^2(7) = 6.753$, p = 0.455) were not statistically significant.

We divided the participants into three based on the screen sizes of their smartphones: small, medium, and large screens. First, we calculated Q1 (5.5 inches) and Q3 (6.32 inches) based on the overall samples of screen sizes. Then, we classified the screen sizes smaller than Q1 as small screens, those larger than Q3 as large screens, and the others as medium screens. One-way ANOVA tests showed that there were no statistically significant differences between the participants who used smartphones in different screen sizes in terms of WPM (F(2,45) = 0.055, p = 0.947) and KSPS (F(2,45) = 0.061, p = 0.941). Similarly, Kruskal-Wallis H tests showed that the differences between screen sizes in terms of KSPC ($\chi^2(2) = 0.759$, p = 0.684) and ER ($\chi^2(2) = 1.595$, p = 0.450) were not statistically significant.

Table 12. Percent of the Participants That Corresponding Metric is Higher for the Context Factor (Participants who had the Same Value under Both Conditions are Excluded)

Context	Factor	WPM (%)	KSPS (%)	KSPC (%)	ER (%)
Environment	Indoor	54.8	48.4	16.1	6.5
	Outdoor	45.2	51.6	51.6	35.5
Mobility	Stable	53.3	46.7	26.7	6.7
	Mobile	46.7	53.3	46.7	33.3
Social	Alone	57.9	55.3	13.2	7.9
	Not Alone	42.1	44.7	42.1	23.7
Multitasking	Nothing	37.1	25.7	20.0	14.3
	Multitasking	62.9	74.3	40.0	28.6
Distractions	Nothing	45.7	42.9	11.4	11.4
	Distraction	54.3	57.1	42.9	37.1

Our participants used four different types of soft keyboards during the study: Samsung, Microsoft SwiftKey, Gboard, and Fleksy keyboards (see Table 14). We excluded the Fleksy keyboard from our statistical analysis since only one participant used this keyboard. One-way ANOVA tests showed that there were no statistically significant differences between different keyboard groups in terms of WPM (F(2,44) = 1.382, p = 0.262) and KSPS (F(2,44) = 1.686, p = 0.197). Similarly, Kruskal-Wallis H tests showed that the differences between three keyboard groups in terms of KSPC ($\chi^2(2) = 4.558$, p = 0.102) and ER ($\chi^2(2) = 1.633$, p = 0.442) were not statistically significant.

Demographics. Our statistical analysis could not find a main effect of demographic groups, including age, gender, education level, experience with a mobile device, experience with the current mobile device, daily screen time, and occupation on typing speed and error rate performance metrics.

Individual user performances. To investigate the individual user performances, we repeated the same procedure in Section 5.1 on the dataset of each participant by using the same performance metrics. Figures 6–10 in Appendix D illustrate the individual performance metrics for each participant under different contextual factors. Table 12 also illustrates the percent of the participants for each metric that have a higher value for each context factor. For example, 54.8% of participants had higher WPM in indoor conditions, while 45.2% had higher WPM in outdoor conditions. It is possible to observe individual differences in the effect of context. The effects of all context dimensions on each participant are available in our online repository (see "Online Repository" Section on page 1).[12] These results show that some participants' typing speed or error rate increase under certain context factors, while the same factor decreases the other participants' typing speed or error rate.

6 DISCUSSION

In this study, we investigated the effect of context on smartphone users' text entry performance in real-world settings. We conducted a user study in the wild and collected participants' text entry data, sensor data, and context labels. We identified a set of performance metrics to measure users' typing performance systematically. We combined several existing approaches to detect typing

[12] https://iam.ncc.metu.edu.tr/cabas-individual-context-comparisons/, last access: 21.01.2022.

errors and distinguish between edits and corrections to better measure typing speed. Finally, we investigated the effect of context on user performance by combining the performance metrics and context labels in five dimensions: environment, mobility, social, multitasking, and distraction.

In reviewing the literature, the text entry studies investigating the effect of context on users' performance have been conducted in controlled settings. Our study, on the other hand, was conducted in the wild. For this purpose, we extended an existing framework and captured the participants' keyboard interactions, a set of sensor data, and context labels submitted by the participants. In our user experiment, the participants interacted with their smartphones as they do in their daily lives without a predefined task model. This approach helped us to collect user data in more realistic settings.

Measuring user performance without a task model is challenging. There are several approaches to detect typing errors and measure typing speed; however, these lookup-based approaches handle daily texting language manually or treat them as typing errors. Daily texting language is too common that considering them as typing errors since they are out-of-vocabulary would yield incorrect interpretations about the effect of context on users' performance. On the other hand, manual analysis introduces privacy issues and is not applicable for possible applications of error detection mechanisms. We combined several existing approaches to cover daily texting language and detect typing errors in English and Turkish. Our evaluation showed that our implementation improved the error detection accuracy compared to the literature. However, even though we applied the text speak rules in the literature, some participants' verdicts for error detection introduced new text speak uses that we did not cover initially. Therefore, an error detection mechanism should learn common usage patterns and adapt itself to users.

The majority of the text entry studies investigating the effect of context on users' performance have primarily focused on different mobility conditions. It may be the case that different mobility conditions can be easily replicated during a study. Moreover, there was contradicting evidence in the literature regarding the effect of mobility. This study considered the context in a broader perspective in five dimensions: environment, mobility, social, multitasking, and distraction. The results of our experiment yielded that being in an outdoor environment, being mobile, presence of other people and having distractions increased error rate, while they did not affect typing speed. Multitasking increased the number of keystrokes in a second and error rate. These are the first empirical evidence on the effect of context on users' typing performance in a study conducted in the wild.

6.1 Findings

In this study, we focused on five research questions, and each research question addressed the effect of different contextual factors on users' text entry performance. For the environment, the error rate was significantly lower for the indoor group than for the outdoor group in terms of KSPC and ER. However, no significant difference between the two groups was evident for WPM and KSPS. Generally, we are exposed to more external factors in the outdoor environments. Therefore, people likely pay more attention to these external factors than typing, or some factors make it difficult to type, resulting in higher error rates. Prior studies have focused on a single aspect of the environment. Sarsenbayeva et al. [61] considered ambient noise and Sarsenbayeva et al. [60] investigated the effect of ambient light. The present study was designed to consider all aspects of the environment.

Mobility had a significant effect on the uncorrected ER. Participants' error rate was higher when mobile than when they were stable in terms of ER. No significant difference between the two groups was evident for WPM, KSPS, and KSPC. We mainly focus on our surroundings to avoid hazards when we are walking. In general, therefore, it seems that this causes more typing errors. It also seems possible that users do not correct their typing errors in mobile conditions. Comparison of the findings with those of other studies confirms the increase in error rate in the case of mobility. In contrast to earlier findings, however, no evidence of the effect of mobility on typing speed was detected.

For the social context, the presence of other people increased the error rate in terms of KSPC and ER. The participants made fewer typing errors alone than when there were people around. Similar to the environment and mobility, it did not significantly affect the typing speed. The presence of other people and social interaction with them may have shifted the focus from the text entry task to the interaction. Therefore, this resulted in more typing errors. In reviewing the literature, no data was found on the effect of social context on users' typing performance.

Multitasking affected the participants' error rate, increasing both KSPC and ER. Multitasking did not have a significant effect on typing speed. Like the social context, focusing on other tasks may have increased the error rate. This finding was also reported by Crease et al. [16]. However, the findings of the current study do not support Sarsenbayeva et al. [59] who reported no significant effect of multitasking on error rate.

The presence of distractions increased the error rate in terms of KSPC and ER; however, it did not affect typing speed. Distraction factors took the participants' primary focus, similar to the environment and mobility, and the participants made more typing errors when interrupted. This outcome is partially contrary to that of Jain and Balakrishnan [35], who found an increase in error rate, typing speed, and a decrease in error corrections when participants were distracted. This contradiction might be related to the experimental task used by Jain and Balakrishnan [35]. In our study, participants tended to correct their typing errors as they were dealing with their real-world tasks rather than experimental tasks.

It is interesting to see individual differences in the effect of context on different participants. A context factor may affect a participant negatively by reducing the typing speed or increasing the error rate, while the same factor may improve another participant's performance by increasing the typing speed or reducing the error rate. Ability-based design is an approach in which users do not adapt themselves to a system; instead, it measures the user performance and adapts itself. For instance, if a user has problems tapping on a key on the keyboard, the system may increase the size of the keys to prevent the error. The ability-based design identifies and exploits users' abilities rather than their disabilities to enhance interaction using available resources [79]. Overall, these results show that ability-based design could be an approach to better consider users' context. Further research is needed to show the actual effect of ability-based designed applications on the users' performance.

6.2 Implications

Using a smartphone itself can cause performance problems similar to those experienced by users with motor impairments [82]. Contextual factors such as the environment, the current position, or the accessories worn can cause additional problems. Users generally adopt different strategies to overcome these problems. For instance, smartphone users change their current locations or use their hands for shadows when exposed to direct sunlight [69]. Other possible adaptations to prevent situational visual impairments include removing accessories that may introduce temporary visual impairments, adjusting the smartphone's brightness, or postponing their task [69]. On the other hand, these performance problems can be addressed by applying adaptations similar to those for users with physical impairments [9]. For example, to maintain the same performance in a stable condition while walking, target sizes might be increased [40].

The adaptation process can be automated by using different available data sources. Goel et al. [26] showed that the accelerometer sensor can be used to overcome the performance problems experienced while walking. According to Goel et al. [27], detecting hand posture can prevent situations like carrying something with the dominant hand or grabbing a handle in public transportation from causing performance problems. Furthermore, Sarsenbayeva et al. [62] suggested that using the smartphone's battery temperature can help adapt to interaction in cold environments. This study collected text entry data in the wild and processed this data offline to measure the users' performance. The same approach can be applied to measure the performance online and support the adaptation of the user interfaces to the users' abilities and context.

Section 3.7 presents the participants' self-evaluations on whether they made a typing error or the reason for the typing error as they perceived. In most cases, participants did not correlate their editing/correcting behaviour with a typing error, even though our error correction/edit classification implementation classified most cases as error corrections. Moreover, the participants did not associate the typing problem with a particular reason in most typing error cases. Like environmental factors, people's emotional or cognitive states can also cause them to make various typing errors. Moreover, in some cases, the users might not perceive the contextual factors as a source of typing errors. Therefore, further studies should be conducted to investigate the effect of context on user performance at the sensor level.

Our results also showed individual differences between the effects of different contextual factors on participants. The same context factor has different effects on each participant. It may reduce the typing speed or increase the error rate for one participant, while it increases the typing speed or reduces the error rate for another participant. A possible explanation for these results may be the different strategies employed to overcome SIIDs. For instance, a user who needs to send a text message while walking in a public area may wish to complete the typing tasks as soon as possible, increasing the typing speed and possibly increasing the error rate. Another user, on the other hand, may prioritize paying attention to the surroundings and decrease his/her typing speed. Therefore, each strategy users intentionally or unintentionally employ under different scenarios may affect user performance differently. For this reason, user-specific ability-based interfaces that adapt themselves based on the users' abilities should be considered [36]. As software libraries to sense the context become available, mobile app developers can use them to create adaptive interfaces [78]. For instance, a background process can send broadcasts whenever a user is in a situation that may affect his/her performance, and the apps receiving these broadcasts can apply different adaptations based on the requirements of the user interface.

6.3 Challenges of Conducting Studies in the Wild

There are several issues related to conducting a user study remotely in the wild. Since our study was remote, participants were asked to install an application on their smartphones and share their data during the study. The app running as a background service consumed battery and bandwidth with data collection and participants' attentional resources with questionnaires. Finding voluntary participants that would install such an app and keep it for at least three days was challenging even if we offered a small amount of compensation. Overall, we collected data from 48 participants. Another significant issue is privacy. When asked to share daily data with strangers, people could have privacy concerns. We clearly explained how and why we processed the data to address the participants' concerns. Moreover, we provided a mechanism to pause and resume the experiment so that participants could stop sharing data when they felt uncomfortable. Still, we could find more participants if we did not transfer keyboard data to our server and process them on the participants' devices. However, we needed keyboard data to work on a typing error detection mechanism. Data security and anonymity are essential in such studies, and researchers should pay attention to these issues.

Conducting a study in the wild enabled us to collect real-world data from the users while doing their daily tasks in their everyday context. However, controlling the samples to maintain a balance between independent groups is challenging in such studies. This balance is typically ensured in controlled studies. The researchers can specify the number of observations required for each context factor and continue the experiment until the expected number of samples is collected. In this study, on the other hand, we collected data labels during participants' daily life. We did not ask participants to change their normal behaviour and use their smartphones under conditions they would not normally do. Some participants may prefer not to use their smartphones under specific conditions, such as while walking. Moreover, some participants may not have encountered certain context factors during the experiment. The imbalance of the contextual factors is a tradeoff between controlled and in-situ studies.

The study was conducted during the Covid19 pandemic. During this pandemic, people were encouraged to isolate themselves from each other and stay at home. To not risk researchers and participants, we have conducted this study as a completely remote study. Since the participants would download, install, and configure the app independently, we had to set clear instructions for this process. When a participant failed to complete this process, we tried to assist him/her remotely. Some participants abandoned early due to some technical problems, and we could not investigate the problem effectively since we did not have access to the smartphones. Moreover, since people were at home most of the time, this might have limited the coverage of contextual factors in submitted questionnaire answers. On the other hand, we could reach participants with a broad range of demographic profiles by conducting a remote experiment.

6.4 Limitations

Our study is not without limitations. Even though we combine many different techniques, our accuracy is still not 100% for calculating performance metrics, therefore there is always a risk of not assessing the users' typing errors fully. Furthermore, the following cases were a true challenge for our automated assessment. First, some of the text-speak uses are identical to typing errors. For instance, character repetitions may both indicate emotions and a typing error. Therefore, some out-of-vocabulary words that we identified as intentional errors or text speak may correspond to unintentional typing errors.

A typing error may result in another valid word. Moreover, the spelling of a word may be correct; however, it may not be grammatically correct in the sentence. Our implementation does not detect these errors. This problem could be addressed by checking the occurrence frequencies of the tokens with surrounding words. However, further studies are needed to explore such **Natural Language Processing** (**NLP**) techniques.

6.5 Future Work

During our user study, we collected data from a set of available sensors on the smartphones along with the text entered and context labelled by the participants. These sensors include motion sensors (accelerometer, gravity, gyroscope, and rotation), environment sensors (barometer and light), position sensors (magnetometer and proximity), and other sensors such as location, WiFi, and telephony. This article explained how we calculated several performance metrics (WPM, KSPS, KSPC, and ER) and how context affects them. In our future work, we plan to implement a mechanism to infer the relationship between these metrics and the sensor data and predict performance problems caused due to the context. This mechanism is intended to be used to propose adaptations to overcome SIIDs.

Our analysis showed that each contextual factor affected the individuals differently. Further research can also be conducted to investigate which context has the highest performance impact on a particular user.

7 CONCLUSION

This study aimed to observe the effect of context on users' text entry performance in their daily settings and without any predefined task model. We conducted an in-situ user study and collected text entry interactions and corresponding context factors in the wild. We implemented a mechanism to determine whether a text contains typing errors. Using this mechanism, we calculated a set of performance metrics and associated them with the corresponding context labels. Finally, we investigated the effect of context on the participants' performance by using these metrics. Our findings show that contextual factors mainly affect participants' typing performance in terms of error rate; however, they do not significantly affect the typing speed. Moreover, individual comparisons reveal that their effects on each participant differ. The findings reported here shed new light on the potential of ability-based design to monitor individual user performance problems due to context and act accordingly. Our future work will also explore and utilize collected sensor data to predict user performance issues.

APPENDICES

A TEXT-SPEAK EXAMPLES

Table 13 illustrates common text-speak techniques in daily texting use and examples for these techniques.

Table 13. Common Text-Speak Techniques and Their Examples

Text-speak technique	Example	References
Deletion of vowels	"msg" for "message"	[14, 41]
Deletion of repeated characters	"tomorow" for "tomorrow"	[14]
Shortening of words	"lab" for "laboratory"	[14, 41, 74]
Deletion of punctuation	"dont" for "don't"	[41, 74]
Deletion of the "g" at the end in words ending "ing"	"goin" for "going"	[41]
Deletion of the final characters	"hav" for "have"	[41]
Phonetic substitution	"2" for "too" or "c" for "see"	[14, 41, 74]
Abbreviation	"lol" for "laughs out loud"	[14, 41, 74]
Dialectal and informal usage	"gonna" for "going to"	[14, 41, 74]
Deletion of function words and pronouns	"readin bk" for "I am reading the book"	[14]
Missed capitalization	"i'd" for "I'd"	[41, 74]
Spelling as pronunciation	"fone" for "phone" "gidicem" for "gideceğim"	[41, 70, 74]
Onomatopoeic/ exclamatory	"ha", "yay"	[41, 74]
Repeating characters for expression	"whaaaat" to express surprise	[74]
Using upper case/extra punctuation for emotion	"WHAT?????"	[74]
Using insider words	"hottie", "fugly"	[74]
Prevention of using Turkish characters	"kacmis" for "kaçmış"	[5, 37, 70]
Separation errors	"birşey" for "bir şey" "hiç biri" for "hiçbiri"	[37]
Use of English words in Turkish text		[37]
Neologisms[13]	"hack-lemek"	[37]
Incorrect use of some suffixes	"kitapda" for "kitap da"	[83]

[13] non-Turkish word with Turkish suffixes.

B OVERALL ESM QUESTIONS

This section presents the ESM questions used in the study.

B.1 ESM Questions for Labelling Context

If the participants entered text longer than five characters, the app sent notifications to ask them to answer a set of questions related to their current context. The overall questions for context labelling and provided options are as follows:

1. Which one of these best describes your current location?
    - Indoors
    - Outdoors
    - Stairs
    - In vehicle
    - Crosswalk
    - Other
2. Which one of these best describes your mobility condition?
    - Lying down
    - Sitting
    - Standing
    - Walking
    - Running
    - Other
3. Which one of these best describes people around you?
    - Alone
    - With a friend/family member/colleague
    - With 2–4 friends/family members/colleagues
    - With more than 4 friends/family members/colleagues
    - With strangers (not crowded)
    - With strangers (crowded)
    - Other
4. Did you handle any other task along with text entry?
    - Nothing
    - I am carrying a box/bag/other
    - I am trying to avoid collision while walking
    - I am having a conversation with someone around me
    - I am working
    - I am shopping
    - I am doing home-activities (cleaning, cooking, etc)
    - I am having breakfast/lunch/dinner
    - Multiple of these
    - Other
5. Is there anything that interrupted/distracted your interaction with mobile device?
    - Nothing
    - There are obstacles/people/cars on walking path
    - I am in a hurry
    - I need to check something from time to time (i.e., a child or cook)
    - I am interrupted by someone
    - I am interrupted by something unexpected

- Multiple of these
- Other

B.2 ESM Questions For Participants' Self Evaluation On Typing Errors

If the participants deleted any characters during a session, we asked participants if they made a typing error after context questions. If the participants selected yes or maybe options, we asked them to specify the cause of the typing problem. The overall questions for self-evaluation and provided options are as follows:

(1) Did you just make a typing error?
- Yes
- No
- Maybe

(2) What do you think caused this typing error?
- My current location
- My current mobility situation
- People around me
- Other task I am busy with
- Something that interrupts me
- Multiple of these
- Other

C PARTICIPANTS' DEVICE SUMMARY

Table 14 provides a summary of participants' devices. The brand and model names and Android SDK versions were retrieved from participants' devices. The screen sizes were collected from product specifications [30].

Table 14. Participants' Smartphone Brands, Models, Android SDK Versions, and Screen Sizes

Brand	Model	SDK	Size (inches)	Keyboard	#
Asus	ASUS_X00QD (Zenfone 5)	28	6.2	Gboard	1
Google	Pixel 3	29	5.5	Gboard	1
Huawei	ANE-LX1 (P20 lite)	28	5.84	Microsoft SwiftKey	2
Huawei	BLA-L09 (Mate 10 Pro)	29	6.0	Gboard	1
Huawei	ELE-L29 (P30)	29	6.1	Microsoft SwiftKey	1
Huawei	FIG-LX1 (P smart)	28	5.65	Microsoft SwiftKey	1
Huawei	RNE-L21 (Mate 10 Lite)	26	5.9	Microsoft SwiftKey	1
Huawei	SNE-LX1 (Mate 20 lite)	29	6.3	Microsoft SwiftKey	1
Huawei	VTR-L09 (P10)	28	5.1	Microsoft SwiftKey	1
Lenovo	Lenovo P2a42 (P2)	24	5.5	Gboard	1
Nokia	Nokia 6.1	29	5.5	Gboard	1
Nokia	Nokia 7.2	29	6.3	Gboard	1
OnePlus	ONEPLUS A6000	29	6.28	Microsoft SwiftKey	1
Samsung	SM-A305F (Galaxy A30)	29	6.4	Samsung	1
Samsung	SM-A307FN (Galaxy A30s)	29	6.4	Samsung	1
Samsung	SM-A505F (Galaxy A50)	29	6.4	Samsung	2
Samsung	SM-A520F (Galaxy A5)	26	5.2	Samsung	2

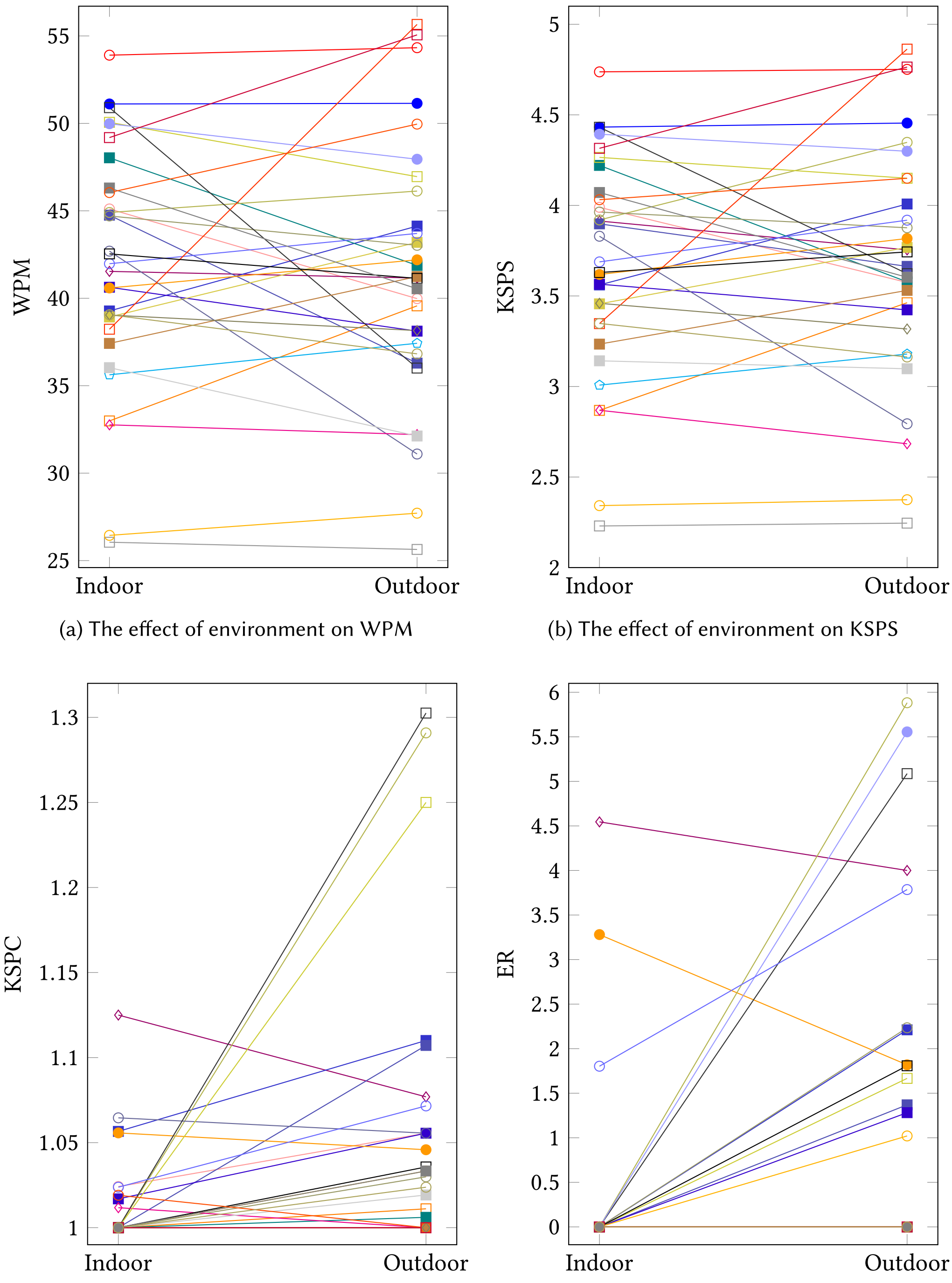
(Continued)

Table 14. Continued

Brand	Model	SDK	Size (inches)	Keyboard	#
Samsung	SM-A710F (Galaxy A7)	24	5.5	Microsoft SwiftKey	1
Samsung	SM-G610F (Galaxy J7 Prime)	24	5.5	Samsung	1
Samsung	SM-G610F (Galaxy J7 Prime)	27	5.5	Samsung	2
Samsung	SM-G930F (Galaxy S7)	26	5.1	Microsoft SwiftKey	1
Samsung	SM-G935F (Galaxy S7 Edge)	26	5.5	Fleksy	1
Samsung	SM-G935F (Galaxy S7 Edge)	26	5.5	Samsung	1
Samsung	SM-G950F (Galaxy S8)	28	5.8	Samsung	1
Samsung	SM-G950U (Galaxy S8)	28	5.8	Samsung	1
Samsung	SM-G965F (Galaxy S9+)	26	6.2	Samsung	1
Samsung	SM-G965U1 (Galaxy S9+)	29	6.2	Samsung	1
Samsung	SM-J710FQ (Galaxy J7)	27	5.5	Samsung	1
Samsung	SM-N950F (Galaxy Note8)	28	6.3	Samsung	1
Samsung	SM-N960F (Galaxy Note9)	29	6.4	Samsung	2
Xiaomi	MI 6	28	5.15	Gboard	1
Xiaomi	MI 6	28	5.15	Microsoft SwiftKey	1
Xiaomi	MI 8 Lite	29	6.26	Gboard	1
Xiaomi	MI CC 9e	28	6.01	Gboard	1
Xiaomi	Mi 9T	28	6.39	Gboard	1
Xiaomi	Redmi 6	28	5.45	Microsoft SwiftKey	1
Xiaomi	Redmi Note 5 Pro	28	5.99	Gboard	1
Xiaomi	Redmi Note 8	28	6.3	Gboard	1
Xiaomi	Redmi Note 8 Pro	28	6.53	Gboard	1
Xiaomi	Redmi Note 8 Pro	29	6.53	Gboard	3
Xiaomi	Redmi Note 9 Pro	29	6.67	Gboard	1

D THE EFFECT OF CONTEXT ON PARTICIPANTS' INDIVIDUAL PERFORMANCE

Figures 6, 7, 8, 9, and 10 illustrate how individual performances change under different context groups of environment, mobility, social context, multitasking, and distractions, respectively. The shape of the marker indicates the age of the participants (circle (○): 18–24, square (□): 25–34, diamond (◇): 35–54, triangle (△): 55+). Female participants are represented as filled (●), and male participants are represented as unfilled (○). The participants are illustrated with the same colors in all figures.



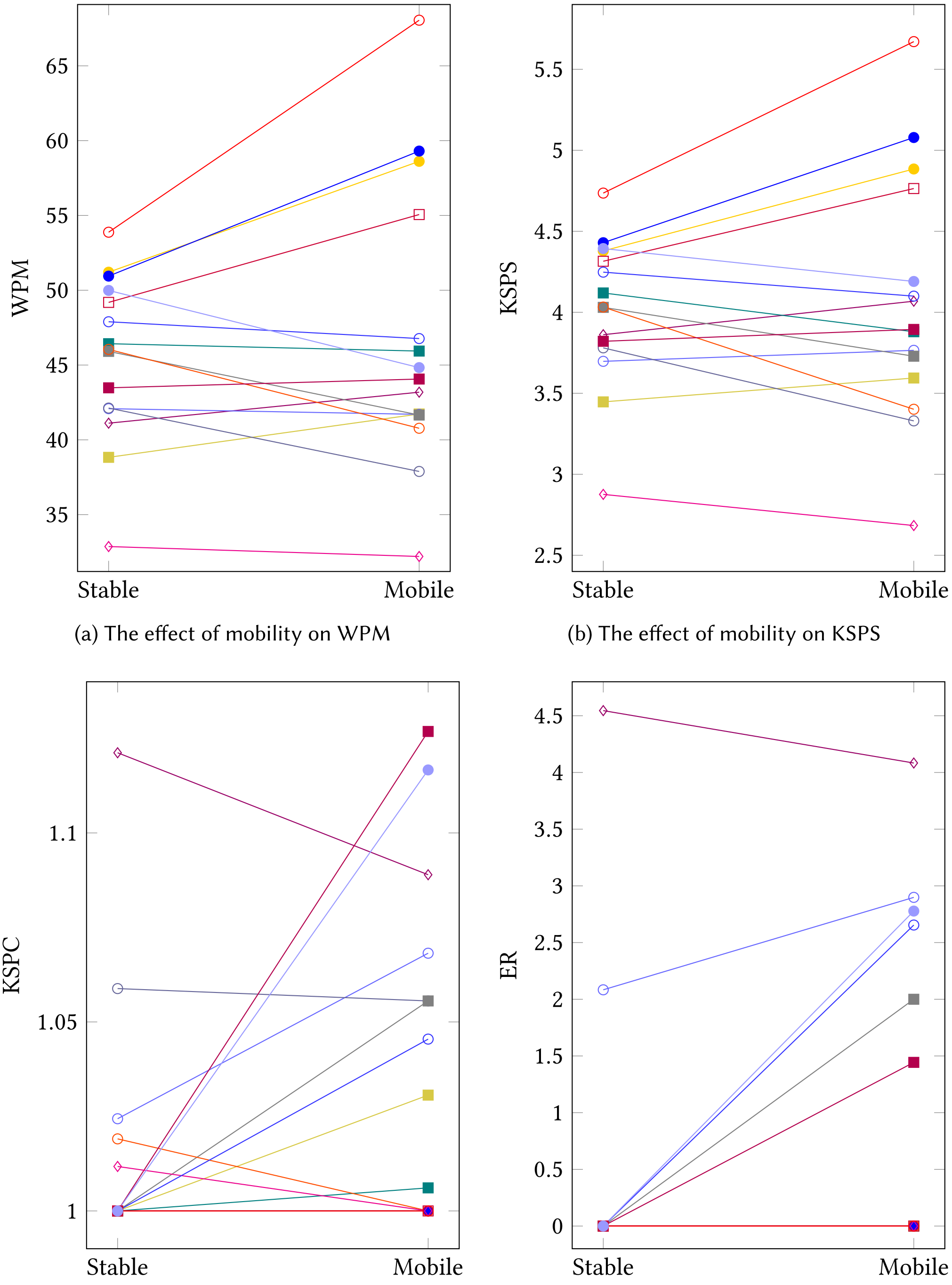
(a) The effect of environment on WPM

(b) The effect of environment on KSPS

(c) The effect of environment on KSPC

(d) The effect of environment on ER

Fig. 6. The effect of environment on individual performances.



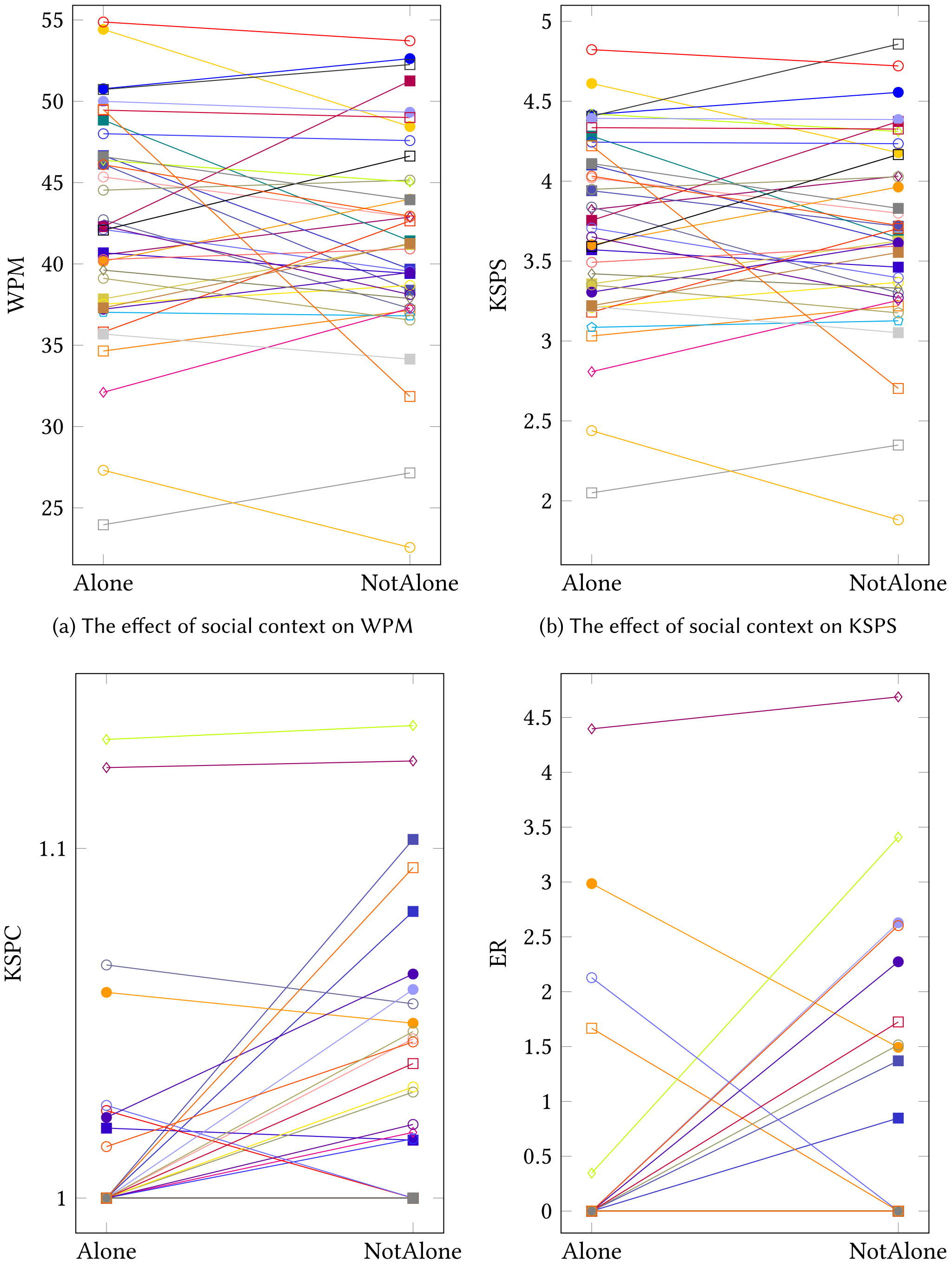
(a) The effect of mobility on WPM

(b) The effect of mobility on KSPS

(c) The effect of mobility on KSPC

(d) The effect of mobility on ER

Fig. 7. The effect of mobility on individual performances.



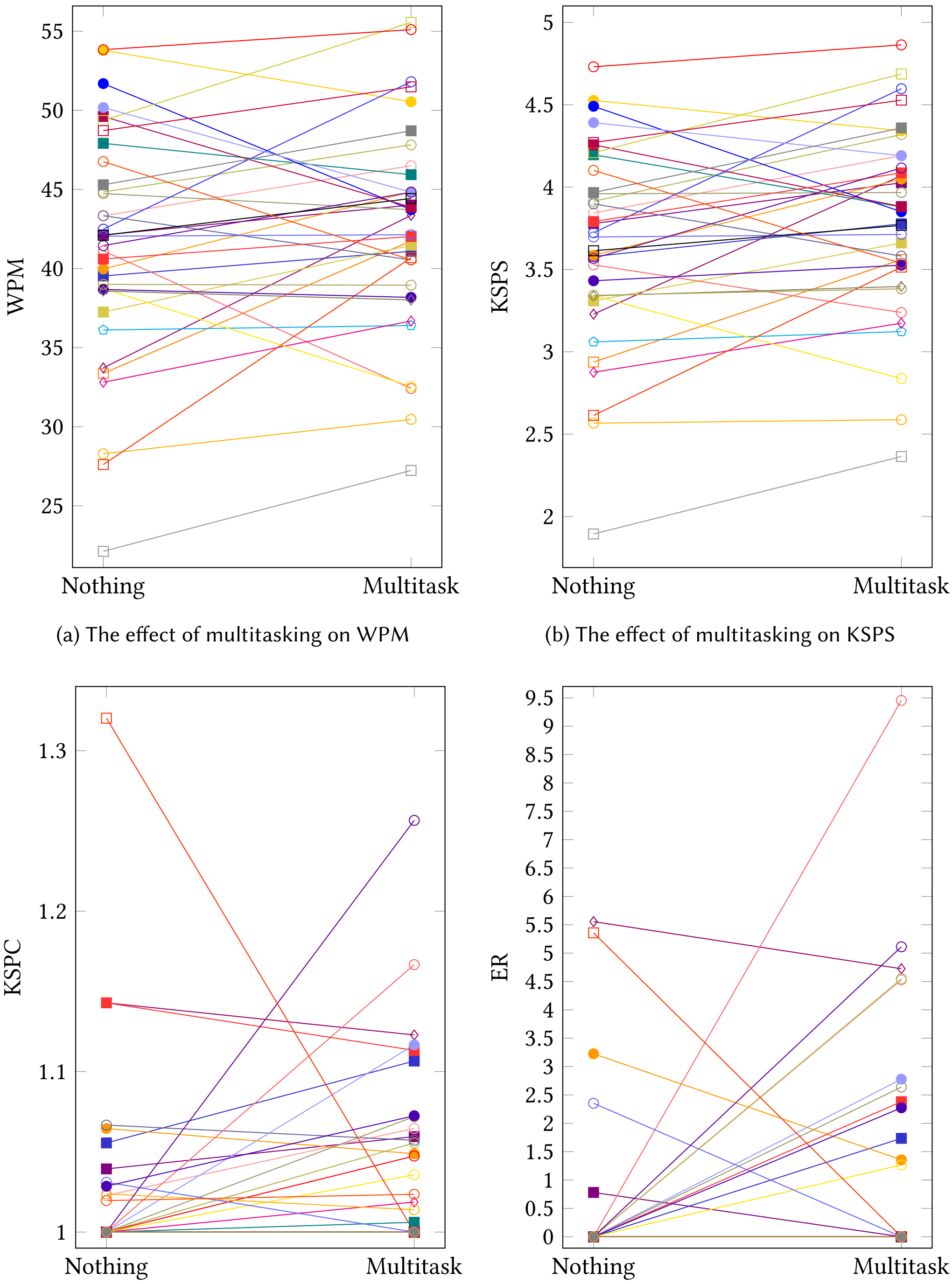
(a) The effect of social context on WPM

(b) The effect of social context on KSPS

(c) The effect of social context on KSPC

(d) The effect of social context on ER

Fig. 8. The effect of social context on individual performances.



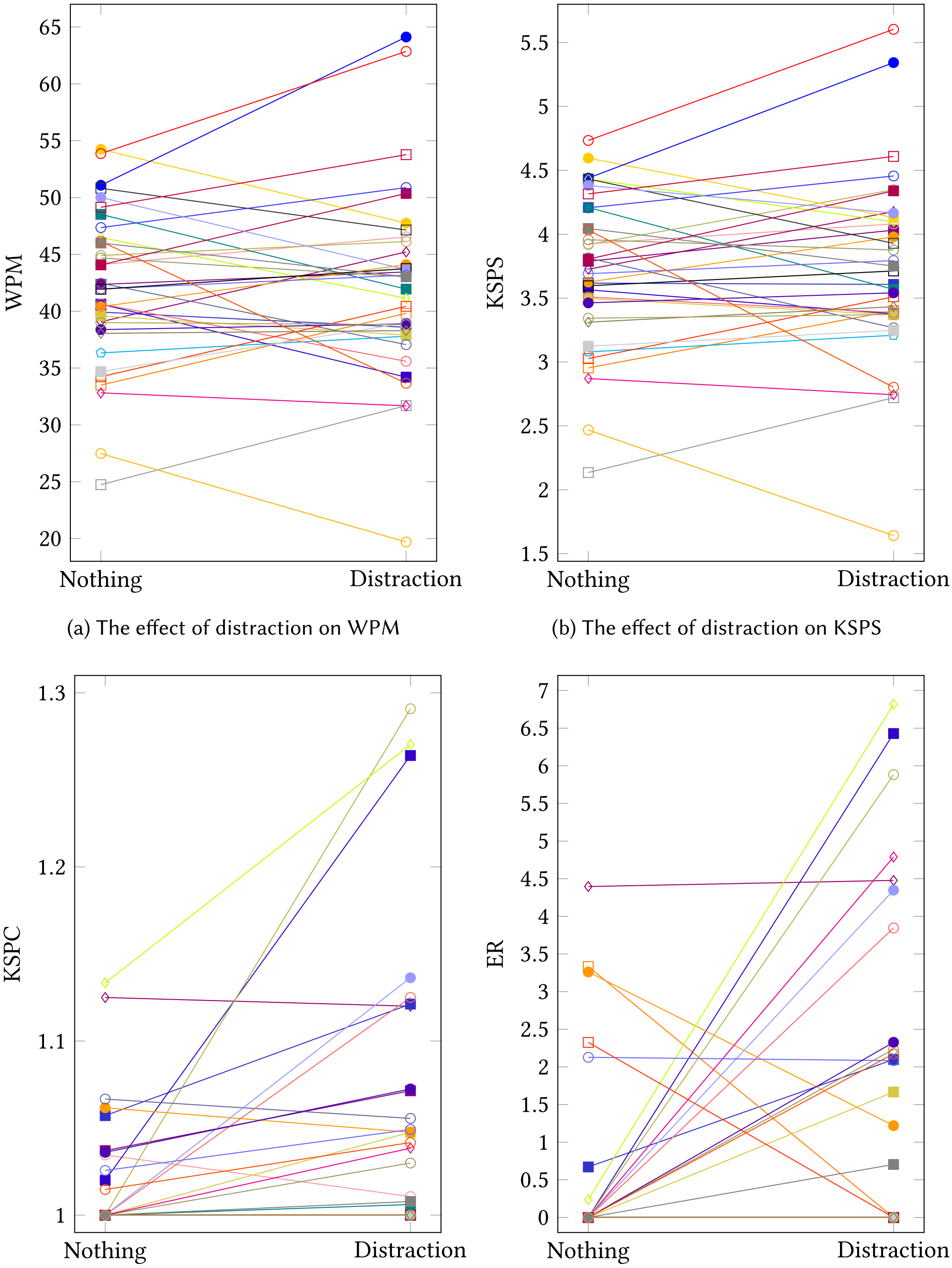
(a) The effect of multitasking on WPM

(b) The effect of multitasking on KSPS

(c) The effect of multitasking on KSPC

(d) The effect of multitasking on ER

Fig. 9. The effect of multitasking on individual performances.



(a) The effect of distraction on WPM

(b) The effect of distraction on KSPS

(c) The effect of distraction on KSPC

(d) The effect of distraction on ER

Fig. 10. The effect of distraction on individual performances.

ACKNOWLEDGMENTS

We thank our participants who voluntarily participated in our study and provided their data.

REFERENCES

[1] Md Sabbir Ahmed, Rahat Jahangir Rony, and Nova Ahmed. 2021. Identifying high and low academic result holders through smartphone usage data. In *Proceedings of the Asian CHI Symposium 2021*. ACM, New York, NY, 114–121. DOI: https://doi.org/10.1145/3429360.3468192

[2] Elgin Akpinar, Yeliz Yeşilada, and Selim Temizer. 2020. The effect of context on small screen and wearable device users' performance - a systematic review. *ACM Computing Surveys* 53, 3, Article 52 (May 2020), 44 pages.

[3] Ahmet Afşın Akın and Mehmet Dündar Akın. 2007. Zemberek, an open source NLP framework for Turkic Languages. *Structure* 10, 2007 (2007), 1–5.

[4] Ahmed Sabbir Arif and Wolfgang Stuerzlinger. 2010. Predicting the cost of error correction in character-based text entry technologies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, 5–14.

[5] Ahmet Arslan. 2016. DeASCIIfication approach to handle diacritics in Turkish information retrieval. *Information Processing and Management: An International Journal* 52, 2 (March 2016), 326–339.

[6] Leon Barnard, Ji Soo Yi, Julie A. Jacko, and Andrew Sears. 2005. An empirical comparison of use-in-motion evaluation scenarios for mobile computing devices. *International Journal of Human-Computer Studies* 62, 4 (April 2005), 487–520.

[7] Leon Barnard, Ji Soo Yi, Julie A. Jacko, and Andrew Sears. 2007. Capturing the effects of context on human performance in mobile computing systems. *Personal and Ubiquitous Computing* 11, 2 (Jan. 2007), 81–96.

[8] Agathe Battestini, Vidya Setlur, and Timothy Sohn. 2010. A large scale study of text-messaging use. In *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services*. ACM, New York, NY, 229–238.

[9] P. Biswas, P. M. Langdon, J. Umadikar, S. Kittusami, and S. Prashant. 2014. How interface adaptation for physical impairment can help able bodied users in situational impairment. In *Inclusive Designing*. P. M. Langdon, J. Lazar, A. Heylighen, and H. Dong (Eds.), Springer International Publishing, Cham, 49–58.

[10] Matthias Böhmer, Brent Hecht, Johannes Schöning, Antonio Krüger, and Gernot Bauer. 2011. Falling asleep with Angry Birds, Facebook and Kindle: A large scale study on mobile application usage. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*. ACM, New York, NY, 47–56.

[11] Peter Bruce, Andrew Bruce, and Peter Gedeck. 2020. *Practical Statistics for Data Scientists: 50+ Essential Concepts Using R and Python*. O'Reilly Media, Incorporated, Sebastopol, CA, 11–11.

[12] Sau Kwan Chan, Ben He, and Iadh Ounis. 2005. An in-depth study of the automatic detection and correction of spelling mistakes. In *Proceedings of the 5th Dutch-Belgian Information Retrieval Workshop*. Utrecht University, Utrecht, 71–81.

[13] Tianyi Chen, Yeliz Yesilada, and Simon Harper. 2010. What input errors do you experience? Typing and pointing errors of mobile Web users. *International Journal of Human-Computer Studies* 68, 3 (2010), 138–157.

[14] M. Choudhury, Rahul Saraf, Vijit Jain, Animesh Mukherjee, Sudeshna Sarkar, and Anupam Basu. 2007. Investigation and modeling of the structure of texting language. *International Journal on Document Analysis and Recognition* 10, 3–4 (2007), 157–174.

[15] James Clawson, Thad Starner, Daniel Kohlsdorf, David P. Quigley, and Scott Gilliland. 2014. Texting while walking: An evaluation of mini-qwerty text input while on-the-go. In *Proceedings of the 16th International Conference on Human-Computer Interaction with Mobile Devices & Services*. ACM, New York, NY, 339–348.

[16] Murray Crease, Jo Lumsden, and Bob Longworth. 2007. A technique for incorporating dynamic paths in lab-based mobile evaluations. In *Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI…But Not As We Know It*. Vol. 1, British Computer Society, Swinton, 99–108.

[17] Liwei Dai, Andrew Sears, and Rich Goldman. 2009. Shifting the focus from accuracy to recallability: A study of informal note-taking on mobile information technologies. *ACM Transactions on Computer-Human Interaction* 16, 1, Article 4 (April 2009), 46 pages.

[18] Anind K. Dey, Gregory D. Abowd, and Daniel Salber. 2001. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human–Computer Interaction* 16, 2 (Dec. 2001), 97–166.

[19] Ivor Durham, David A. Lamb, and James B. Saxe. 1983. Spelling correction in user interfaces. *Communications of the ACM* 26, 10 (Oct. 1983), 764–773.

[20] Gülşen Eryiğit and Dilara Torunoğlu-Selamet. 2017. Social media text normalization for Turkish. *Natural Language Engineering* 23, 6 (2017), 835–875.

[21] Abigail Evans and Jacob Wobbrock. 2012. Taming wild behavior: The input observer for obtaining text entry and mouse pointing measures from everyday computer use. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, 1947–1956.

[22] Jody A. Feld and Prudence Plummer. 2019. Visual scanning behavior during distracted walking in healthy young adults. *Gait & Posture* 67 (2019), 219–223. https://www.sciencedirect.com/journal/gait-and-posture/vol/67/suppl/C.

[23] Denzil Ferreira, Vassilis Kostakos, and Anind K. Dey. 2015. AWARE: Mobile context instrumentation framework. *Frontiers in ICT* 2 (2015), 6. https://www.frontiersin.org/journals/ict/volumes.

[24] Daniel Fitton, I. Scott MacKenzie, Janet C. Read, and Matthew Horton. 2013. Exploring tilt-based text input for mobile devices with teenagers. In *Proceedings of the 27th International BCS Human Computer Interaction Conference*. British Computer Society, Swinton, Article 25, 6 pages.

[25] Flora-Jean Forbes and E. Buchanan. 2019. "Textisms": The comfort of the recipient. *Psychology of Popular Media Culture* 8, 4 (2019), 358–364.

[26] Mayank Goel, Leah Findlater, and Jacob Wobbrock. 2012. WalkType: Using accelerometer data to accomodate situational impairments in mobile touch screen text entry. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, 2687–2696.

[27] Mayank Goel, Alex Jansen, Travis Mandel, Shwetak N. Patel, and Jacob O. Wobbrock. 2013. ContextType: Using hand posture information to improve mobile touch screen text entry. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, 2795–2798.

[28] Jorge Goncalves, Zhanna Sarsenbayeva, Niels van Berkel, Chu Luo, Simo Hosio, Sirkka Risanen, Hannu Rintamäki, and Vassilis Kostakos. 2017. Tapping task performance on smartphones in cold temperature. *Interacting with Computers* 29, 3 (2017), 355–367.

[29] Kristen K. Greene, Melissa A. Gallagher, Brian C. Stanton, and Paul Y. Lee. 2014. I Can't type that! P@$$w0rd entry on mobile devices. In *Proceedings of the 2nd International Conference on Human Aspects of Information Security, Privacy, and Trust*. Vol. 8533, Springer-Verlag, Berlin, 160–171.

[30] GSMArena. 2000. GSMArena.com - Mobile Phone Reviews, News, Specifications and More . . . Retrieved from https://www.gsmarena.com/. Accessed 29 June 2022.

[31] Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Vol. 1, Association for Computational Linguistics, 368–378.

[32] James Head, Paul N. Russell, Martin J. Dorahy, Ewald Neumann, and William S. Helton. 2012. Text-speak processing and the sustained attention to response task. *Experimental Brain Research* 216, 1 (01 Jan 2012), 103–111. DOI: https://doi.org/10.1007/s00221-011-2914-6

[33] Eve Hoggan, Stephen A. Brewster, and Jody Johnston. 2008. Investigating the effectiveness of tactile feedback for mobile touchscreens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, 1573–1582.

[34] Sara H. Hsieh and Timmy H. Tseng. 2017. Playfulness in mobile instant messaging: Examining the influence of emoticons and text messaging on social interaction. *Computers in Human Behavior* 69, C (2017), 405–414.

[35] Mohit Jain and Ravin Balakrishnan. 2012. User learning and performance with bezel menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, 2221–2230.

[36] Shaun K. Kane, Jacob O. Wobbrock, and Ian E. Smith. 2008. Getting off the treadmill: Evaluating walking user interfaces for mobile devices in public spaces. In *Proceedings of the 10th International Conference on Human Computer Interaction with Mobile Devices and Services*. ACM, New York, NY, 109–118.

[37] Asiye Tuba Koksal, Ozge Bozal, Emre Yürekli, and Gizem Gezici. 2020. #Turki$hTweets: A benchmark dataset for Turkish text correction. In *Findings of the ACL: (EMNLP'20)*, Trevor Cohn, Yulan He, and Yang Liu (Eds.). ACL, Online, 4190–4198. https://aclanthology.org/volumes/2020.findings-emnlp/.

[38] 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*. ACM, New York, NY, 221–228.

[39] Reed Larson and Mihaly Csikszentmihalyi. 2014. *The Experience Sampling Method*. Springer, Dordrecht, 21–34. DOI: https://doi.org/10.1007/978-94-017-9088-8_2

[40] Min Lin, Rich Goldman, Kathleen J. Price, Andrew Sears, and Julie Jacko. 2007. How do people tap when walking? An empirical investigation of nomadic data entry. *International Journal of Human-Computer Studies* 65, 9 (Sept. 2007), 759–769.

[41] Fiona Lyddy, Francesca Farina, James Hanney, Lynn Farrell, and Niamh Kelly O'Neill. 2014. An analysis of language in university students' text messages. *Journal of Computer-Mediated Communication* 19, 3 (2014), 546–561.

[42] Sachi Mizobuchi, Mark Chignell, and David Newton. 2005. Mobile text entry: Relationship between walking speed and text input task difficulty. In *Proceedings of the 7th International Conference on Human Computer Interaction with Mobile Devices & Services*. ACM, New York, NY, 122–128.

[43] Matthew A. Napierala. 2012. What is the Bonferroni Correction? *AAOS Now - American Academy of Orthopaedic Surgeons*. Retrieved from https://www.aaos.org/aaosnow/2012/apr/research/research7/. Accessed 29 June 2022.

[44] László Németh. 2016. Hunspell. Retrieved October 2, 2021 from http://hunspell.github.io/.

[45] Alexander Ng, Stephen Brewster, and Andrew Crossan. 2011. The Effects of Encumbrance on Mobile Gesture Interactions. *MobileHCI'11*.

[46] Alexander Ng, Stephen A. Brewster, and John Williamson. 2013. The impact of encumbrance on mobile interactions. In *Human-Computer Interaction – INTERACT 2013*. Paula Kotzé, Gary Marsden, Gitte Lindgaard, Janet Wesson, and Marco Winckler (Eds.), Springer Berlin Heidelberg, Berlin, 92–109.

[47] Alexander Ng, Stephen A. Brewster, and John H. Williamson. 2014. Investigating the effects of encumbrance on one- and two- handed interactions with mobile devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, 1981–1990.

[48] Alexander Ng, John H. Williamson, and Stephen A. Brewster. 2014. Comparing evaluation methods for encumbrance and walking on interaction with touchscreen mobile devices. In *Proceedings of the 16th International Conference on Human-Computer Interaction with Mobile Devices & Services*. ACM, New York, NY, 23–32.

[49] Hugo Nicolau, Tiago Guerreiro, David Lucas, and Joaquim Jorge. 2014. Mobile text-entry and visual demands: Reusing and optimizing current solutions. *Universal Access in the Information Society* 13, 3 (01 Aug 2014), 291–301.

[50] Hugo Nicolau and Joaquim Jorge. 2012. Touch typing using thumbs: Understanding the effect of mobility and hand posture. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, 2683–2686.

[51] Hugo Nicolau, Kyle Montague, Tiago Guerreiro, André Rodrigues, and Vicki L. Hanson. 2017. Investigating laboratory and everyday typing performance of blind users. *ACM Transactions on Accessible Computing* 10, 1, Article 4 (March 2017), 26 pages.

[52] Antti Oulasvirta, Sakari Tamminen, Virpi Roto, and Jaana Kuorelahti. 2005. Interaction in 4-second bursts: The fragmented nature of attentional resources in mobile HCI. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, 919–928.

[53] Kseniia Palin, Anna Maria Feit, Sunjun Kim, Per Ola Kristensson, and Antti Oulasvirta. 2019. How do people type on mobile devices? Observations from a study with 37,000 volunteers. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM, New York, NY, Article 9, 12 pages.

[54] Prudence Plummer, Sarah Apple, Colleen Dowd, and Eliza Keith. 2015. Texting and walking: Effect of environmental setting and task prioritization on dual-task interference in healthy young adults. *Gait & Posture* 41, 1 (2015), 46–51.

[55] Shyam Reyal, Shumin Zhai, and Per Ola Kristensson. 2015. Performance and user experience of touchscreen and gesture keyboards in a lab setting and in the wild. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, New York, NY, 679–688.

[56] André Rodrigues, Hugo Nicolau, André Santos, Diogo Branco, Jay Rainey, David Verweij, Jan David Smeddinck, Kyle Montague, and Tiago Guerreiro. 2022. Investigating the tradeoffs of everyday text-entry collection methods. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, Article 378, 15 pages. DOI: https://doi.org/10.1145/3491102.3501908

[57] Larry D. Rosen, Jennifer Chang, Lynne Erwin, L. Mark Carrier, and Nancy A. Cheever. 2010. The relationship between "Textisms" and formal and informal writing among young adults. *Communication Research* 37, 3 (2010), 420–440.

[58] Zhanna Sarsenbayeva, Jorge Goncalves, Juan García, Simon Klakegg, Sirkka Rissanen, Hannu Rintamäki, Jari Hannu, and Vassilis Kostakos. 2016. Situational impairments to mobile interaction in cold environments. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, New York, NY, 85–96.

[59] Zhanna Sarsenbayeva, Niels van Berkel, Danula Hettiachchi, Weiwei Jiang, Tilman Dingler, Eduardo Velloso, Vassilis Kostakos, and Jorge Goncalves. 2019. Measuring the effects of stress on mobile interaction. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 1, Article 24 (March 2019), 18 pages.

[60] Zhanna Sarsenbayeva, Niels van Berkel, Weiwei Jiang, Danula Hettiachchi, Vassilis Kostakos, and Jorge Goncalves. 2019. Effect of ambient light on mobile interaction. In *Human-Computer Interaction – INTERACT 2019*. David Lamas, Fernando Loizides, Lennart Nacke, Helen Petrie, Marco Winckler, and Panayiotis Zaphiris (Eds.), Springer International Publishing, Cham, 465–475.

[61] Zhanna Sarsenbayeva, Niels van Berkel, Eduardo Velloso, Vassilis Kostakos, and Jorge Goncalves. 2018. Effect of distinct ambient noise types on mobile interaction. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 2, Article 82 (July 2018), 23 pages.

[62] Zhanna Sarsenbayeva, Niels Van Berkel, Aku Visuri, Sirkka Rissanen, Hannu Rintamaki, Vassilis Kostakos, and Jorge Goncalves. 2017. Sensing cold-induced situational impairments in mobile interaction using battery temperature. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3, Article 98 (Sept. 2017), 9 pages.

[63] Bilge Say, Deniz Zeyrek, Kemal Oflazer, and Umut özge. 2004. Development of a corpus and a treebank for present-day written Turkish, (proceedings of the eleventh international conference of turkish linguistics, August, 2002). In *Current Research in Turkish Lingustics*. Kamile İmer and Gürkan Doğan (Eds.), Eastern Mediterranean University Press, Famagusta, North Cyprus, 183–192.

[64] Richard Schlögl, Christoph Wimmer, and Thomas Grechenig. 2019. Hyper typer: A serious game for measuring mobile text entry performance in the wild. In *Proceedings of the Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, 1–6.

[65] A. Sears, M. Lin, J. Jacko, and Y Xiao. 2003. When computers fade … pervasive computing and situationally-induced impairments and disabilities. In *Proceedings of the HCII 2003*. Lawrence Erlbaum, Mahwah, NJ, 1298–1302.

[66] Andrew Sears and Mark Young. 2003. Physical disabilities and computing technologies: An analysis of impairments. In *The Human-Computer Interaction Handbook*. Julie A. Jacko and Andrew Sears (Eds.), L. Erlbaum Associates Inc., Hillsdale, NJ, 482–503.

[67] R. William Soukoreff and I. Scott MacKenzie. 2003. Metrics for text entry research: An evaluation of MSD and KSPC, and a new unified error metric. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, 113–120.

[68] L. Venkata Subramaniam, Shourya Roy, Tanveer A. Faruquie, and Sumit Negi. 2009. A survey of types of text noise and techniques to handle noisy text. In *Proceedings of the 3rd Workshop on Analytics for Noisy Unstructured Text Data*. ACM, New York, NY, 115–122.

[69] Garreth W. Tigwell, David R. Flatla, and Rachel Menzies. 2018. It's not just the light: Understanding the factors causing situational visual impairments during mobile interaction. In *Proceedings of the 10th Nordic Conference on Human-Computer Interaction*. ACM, New York, NY, 338–351. DOI: https://doi.org/10.1145/3240167.3240207

[70] Dilara Torunoğlu and Gülşen Eryiğit. 2014. A cascaded approach for social media text normalization of Turkish. In *Proceedings of the 5th Workshop on Language Analysis for Social Media*. ACL, Gothenburg, 62–70.

[71] Eve Sarah Troll, Malte Friese, and David D. Loschelder. 2021. How students' self-control and smartphone-use explain their academic performance. *Computers in Human Behavior* 117 (2021), 106624. https://www.sciencedirect.com/journal/computers-in-human-behavior/vol/117/suppl/C.

[72] Alaettin Ucan, Behzad Naderalvojoud, Ebru Akcapinar Sezer, and Hayri Sever. 2016. SentiWordNet for new language: Automatic translation approach. In *Proceedings of the 2016 12th International Conference on Signal-Image Technology Internet-Based Systems*. IEEE, Washington, DC, 308–315. DOI: DOI: https://doi.org/10.1109/SITIS.2016.57

[73] Niels van Berkel, Denzil Ferreira, and Vassilis Kostakos. 2017. The experience sampling method on mobile devices. *ACM Computing Surveys* 50, 6, Article 93 (Dec. 2017), 40 pages. DOI: https://doi.org/10.1145/3123988

[74] Connie Varnhagen, G. Peggy McFall, Nicole Pugh, Lisa Zederayko(Routledge), Heather Sumida-MacDonald, and Trudy Kwong. 2010. Lol: New language and spelling in instant messaging. *Reading and Writing* 23, 6 (07 2010), 719–733.

[75] Steven R. Wilson, Walid Magdy, Barbara McGillivray, and Gareth Tyson. 2020. Analyzing temporal relationships between trending terms on Twitter and urban dictionary activity. In *Proceedings of the 12th ACM Conference on Web Science*. ACM, New York, NY, 155–163. DOI: https://doi.org/10.1145/3394231.3397905

[76] Christoph Wimmer, Richard Schlögl, Karin Kappel, and Thomas Grechenig. 2019. Measuring mobile text entry performance and behaviour in the wild with a serious game. In *Proceedings of the 18th International Conference on Mobile and Ubiquitous Multimedia*. ACM, New York, NY, Article 8, 11 pages.

[77] Jacob Wobbrock. 2007. Measures of text entry performance. In *Text Entry Systems*. Kumiko Tanaka-Ishii and I. Scott MacKenzie (Eds.), Morgan Kaufmann, San Francisco, Chapter 3, 47–74.

[78] Jacob O. Wobbrock. 2019. Situationally aware mobile devices for overcoming situational impairments. In *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems*. ACM, New York, NY, Article 1, 18 pages.

[79] Jacob O. Wobbrock, Shaun K. Kane, Krzysztof Z. Gajos, Susumu Harada, and Jon Froehlich. 2011. Ability-based design: Concept, principles and examples. *ACM Transactions on Accessible Computing* 3, 3, Article 9 (April 2011), 27 pages.

[80] Jacob O. Wobbrock and Brad A. Myers. 2006. Analyzing the input stream for character- level errors in unconstrained text entry evaluations. *ACM Transactions on Computer-Human Interaction* 13, 4 (dec 2006), 458–489. DOI: https://doi.org/10.1145/1188816.1188819

[81] Wilson Wong, Wei Liu, and Mohammed Bennamoun. 2006. Integrated scoring for spelling error correction, abbreviation expansion and case restoration in dirty text. In *Proceedings of the 5th Australasian Conference on Data Mining and Analystics*. Vol. 61. Australian Computer Society, Inc., AUS, 83–89.

[82] Yeliz Yesilada, Simon Harper, Tianyi Chen, and Shari Trewin. 2010. Small-device users situationally impaired by input. *Computers in Human Behavior* 26, 3 (2010), 427–435.

[83] Savas Yildirim and T. Yildiz. 2015. An unsupervised text normalization architecture for Turkish language. *Research in Computing Science* 90, 1 (2015), 183–194.

[84] Mingrui Ray Zhang, He Wen, and Jacob O. Wobbrock. 2019. Type, then correct: Intelligent text correction techniques for mobile text entry using neural networks. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, 843–855.

[85] Mingrui Ray Zhang and Jacob O. Wobbrock. 2019. Beyond the input stream: Making text entry evaluations more flexible with transcription sequences. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, 831–842. DOI: https://doi.org/10.1145/3332165.3347922

[86] Gökhan Şeker and Gülşen Eryiğit. 2017. Extending a CRF-based named entity recognition model for Turkish well formed text and user generated content1. *Semantic Web* 8, 5 (01 2017), 1–18.

Received 25 January 2022; revised 1 November 2022; accepted 21 November 2022