A glimpse of mobile text entry errors and corrective behaviour in the wild.

Andreas Komninos
University of Patras
Rio, 26504, Greece
akomninos@ceid.upatras.gr

John Garofalakis
University of Patras
Rio, 26504, Greece
garofala@ceid.upatras.gr

Mark Dunlop
University of Strathclyde
26 Richmond St.
Glasgow, G1 1XH, UK
mark.dunlop@strath.ac.uk

Kyriakos Katsaris
Hellenic Open University
Parodos Aristotelous 18
Patras 26223, Greece
kykatsar@gmail.com

MobileHCI '18 Adjunct, September 3–6, 2018, Barcelona, Spain
© 2018 Copyright is held by the owner/author(s).
ACM ISBN 978-1-4503-5941-2/18/09.
https://doi.org/10.1145/3236112.3236143

Abstract

Research in mobile text entry has long focused on speed and input errors during lab studies. However, little is known about how input errors emerge in real-world situations or how users deal with these. We present findings from an in-the-wild study of everyday text entry and discuss their implications for future studies.

Author Keywords

Smartphones; Text entry; Field study; Error behaviour

ACM Classification Keywords

H.5.2. User Interfaces: Input devices and strategies (e.g., mouse, touchscreen)

Introduction

Most smartphone text entry research is carried out in lab settings and conditions. To our knowledge, very few studies have focused on text entry in ecologically valid settings. This is, in part, no great surprise. It wasn’t until 2011, when the Android OS allowed third-party developers to offer custom input methods (IMEs), i.e. virtual keyboards, which could fully replace the default operating system IME and be used in daily life. However, despite several years already having passed, practically almost no published study exists until today for contextually relevant mobile text entry behaviour in

Key Findings:

Study based on field data from real world text entry in smartphones;

We focus on errors and error management behaviour of users;

Error occurrence correlates only weakly with finger slippage;

Users frequently context-switch between keyboard and input area to spot and correct mistakes, thus slowing down;

Despite slowdown, users make ~2 word-level spelling mistakes per session which need correction

Users predominantly employ backspacing as an error correction strategy;

Support is needed to lessen the cognitive and temporal burden of frequent keyboard – text entry area context switching,

the wild. Furthermore, while significant effort has been made in previous research to address errors during input (mostly through better touch models, e.g. [12], intelligent deletion [1] or visual feedback [2][8]), very little is actually known about how errors in mobile text entry emerge in real life, and how users manage these. The result is a distinct lack of research into ways to support error management during smartphone text entry and, of course, a myriad of autocorrect memes on the Internet, each reflecting the funny, but simultaneously painful impact of text entry gone wrong.

A few studies [6][9][12] attempted to simulate real-life tasks (e.g. walking, driving) in the lab, to increase ecological validity. Closer to actually studying text entry in the field, in [7], user input was studied in the wild but only as part of a game, rather than use in real application contexts. In [10], users were asked to perform transcription tasks on their mobile, issued to them at random occasions via notification, effectively carrying the de-facto transcription task study method into the field. The closest study that exists is Buschek et al. [5], where a set of data captured from a full replacement IME is presented. In this study however, error emergence and error management behaviour is not reported, further from two interesting insights: the considerable ratio of backspaces compared to all keystrokes (8.9%) and that users of autocorrect resort to backspace use more than those who don’t.

This background motivates our paper, in which we attempt to shed light at the frequency of smartphone text entry errors and the strategies employed by users to manage their emergence, using data from an in-the-wild study of 12 young adult participants over 28 days.

Study Design

For our study, we used the MaxieKeyboard virtual QWERTY implementation available as open-source code on GitHub [8]. This is a full replacement IME for Android devices, which allows the logging to keystroke data and transmits the logged data to a remote server for future analysis (server-side code is also included in the GitHub repository). The keyboard has a range of configurable options to support entry by older adults, but for this study we configured it to function as a plain QWERTY implementation with a word suggestion bar but without the use of autocorrect.

MaxieKeyboard uses the concept of a text entry “session” during use of any app on the smartphone, which is measured from the time of invocation (appearance) of the keyboard on the screen, until its dismissal. For each session, the following data is recorded:

Session start and end time (milliseconds): Timestamps of the keyboard invocation and dismissal events.

- Application: The Android app package name in which the text entry is performed.
- Spelling errors during session: The number of “slight” or “serious” text entry spelling errors detected during typing. MaxieKeyboard includes a spell-checker which can offer candidates for completed words that the user has entered and which are not in its dictionary (i.e. classed as spelling mistakes). This check is performed after a sequence of characters terminated by a space or other punctuation mark. Slight mistakes are those for which a likely candidate can be found, serious are

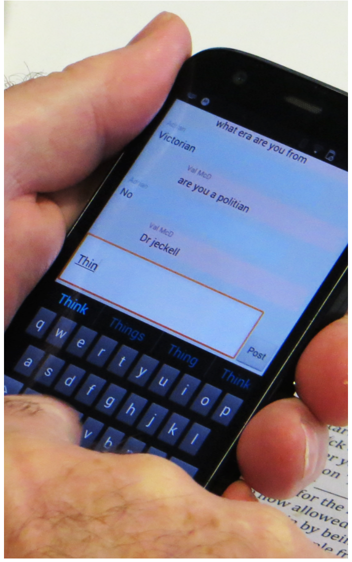


Figure 1: MaxieKeyboard running on an actual device. The suggestion bar at the top was the only assistance provided to users during entry, for this study.

those for which a candidate could not be recommended with confidence.

- Use of the suggestion bar: The number of times the user selected a suggestion from the bar during entry

Further to these, each keystroke entered by the user is also recorded and associated with the session in which it was typed. MaxieKeyboard offers a range of logging options, but to preserve participant anonymity and protect their sensitive information, we chose not to record the actual entered characters or touch coordinates, but the following non-identifying metrics for each keystroke:

- Horizontal and vertical slippage: This is measured as the touch-down and touch-up coordinate deltas on the horizontal and vertical axes.
- Inter-key time: The time elapsed between a keypress and the one immediately preceding it.
- Keypress duration: Time elapsed between touch-down and touch-up events
- Character & character code: Only instances of the backspace character were recorded – all other keycodes and corresponding characters were logged as '-400' and '$' accordingly.

Participants & study duration

We recruited 12 non-student participants (6 female) aged between 25-35. All participants reported as "expert" QWERTY users on their smartphones and none used gestural input (e.g. Swype). MaxieKeyboard was installed on their own devices and they were instructed not to change to another IME for the duration of the study, which was 28 days. After the study period ended, participants were notified to uninstall MaxieKeyboard from their device.

Data cleansing and preparation

Text entry in the wild takes place under a broad range of contexts, from simple search-bar queries to calendar entries, phone number dialing, URL typing, text messaging and more. For this analysis, we focus only on those text entry sessions that related to the composition of messages. Our main reason for this was because most text entry studies use the transcription task as a proxy to actual message composition. Since literature is therefore directed at measuring input speed and accuracy in this type task, to maintain some perspective over our findings, we kept only sessions from SMS, Instant Messaging, Email and social network apps, since micro-blogging in these (e.g. posting a comment on Facebook or a status update on Twitter) is quite similar in nature.

Study Results

How much do users type?

In our study, we recorded a total of 1629 text entry sessions totaling 54575 keystrokes. This makes the average text entry session quite short, having on average μ=33.87 keystrokes (σ=45.96) or μ=20.97sec (σ=36.73s), with the 3^{rd} quartile being 22.83 secs. About a quarter of these entry sessions (26.6%) are very short, amounting to just 10 keystrokes or less, which is reasonable as single-word replies to incoming messages are not uncommon (e.g. "OK", "done"). Overall, we note that the participants' entry speed was not as quick as many of the field studies; on average, just μ=17.51WPM (σ=5.40).

What types of error emerge?

In typical text entry studies, input accuracy is assessed after the user has committed a full phrase to the logging software, since it can be compared with the original text to be copied. In a field study we obviously can't have such a metric, but we can look at how many words are committed to the text input area which contain a "slight" or "serious" spelling mistake (i.e. one that the user did not detect while typing that word). We term these "word-level" errors, and overall, they not as infrequent as we might believe. In fact, we noted an average of μ=1.98 word-level errors of any type per session (σ=3.35). These are dominated by "serious" errors (μ=1.19, σ=2.72) rather than "slight" ones (μ=0.79, σ=1.83), which highlight the following conclusions: Users are generally careful typists since they make few "slight" mistakes – the spellchecker (GNU Aspell) is quite good at offering candidates for omission, substitution, and insertion types of mistakes. However, opportunities to help users with for managing errors arise in practically every text entry session, despite the fact that these sessions are quite short.

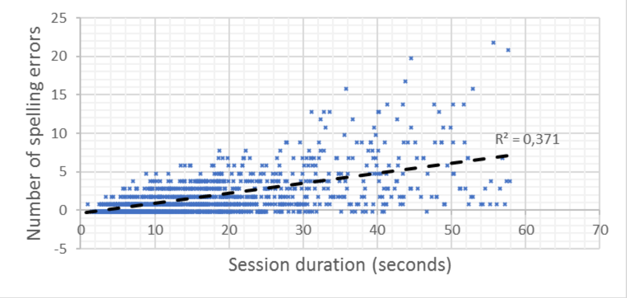


Figure 2: Total errors vs. session duration for sessions with a duration ≤ mean+1 SD (57.7s, 94.5% of all sessions)

In terms of associating the number of errors with the session duration, as it might be expected, a large positive association between the number of total "word-level" errors and session duration is found (Spearman's ρ =0.63, p<0.01) (**Figure 2**), though this association is weaker (medium) when looking at "slight" errors alone (ρ=0.41, p<0.01) or "serious" errors (ρ =0.40, p<0.01). The same applies for suggestion bar use (0.39, p<0.01), even though suggestion bar use remained quite low (μ=0.88 times / session, σ=1.42).

What causes errors?

Finger slippage is generally considered as one of the major factors in text entry error emergence [3]. We examined the average finger slippage for each keystroke and the occurrence of detected spelling mistakes per session. We found a weak statistically significant correlation (Spearman's ρ=0.326, p<0.01). Based on this, other factors must contribute more significantly to the emergence of errors.

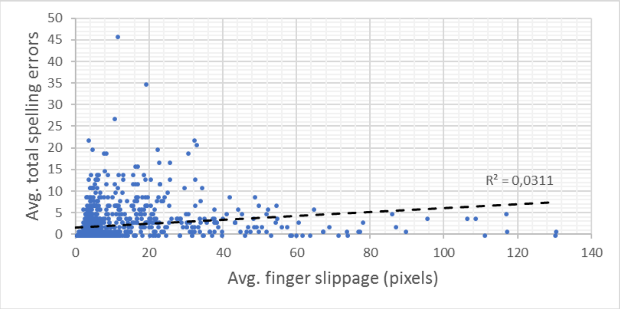
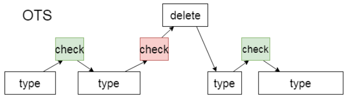


Figure 3: Finger slippage correlation with spelling mistakes.

How do users cope with errors?

So far we have observed that our participants type generally slowly and make few unnoticed mistakes at the word level (though every session contains two such mistakes on average). In [4], it is shown that expert typists will generally slow down to avoid the cost of correcting entry mistakes, increasing their movement

On-the-Spot (OTS): Short bursts of typing, checking and correcting are constantly interleaved. Users frequently context-switch between keyboard and entry area.



Type-and-Review (TAR): The user types longer on the keyboard, thus errors remain unnoticed until the user has finished a longer chunk of text and then checks it for mistakes. Mistakes can be corrected either by positioning the cursor and performing a short deletion sequence (**TAR-P**), or by long deletion sequences that require retyping all the deleted text (**TAR-D**).

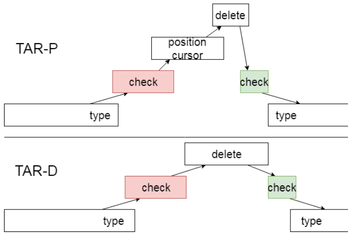


Figure 4: Error correction strategies. Activity box sizes are indicative of temporal length.

time. In this cost, motor performance is just one part of the equation: the error has to be first *detected*, hence possibly requiring a frequent context-switch between the entered text and the keyboard (this is also acknowledged in [4]). In [11], it is shown that typists have a "stopping" span, i.e. irrevocably commit to just one or two keystrokes. It is plausible thus to assume that if such a context-switch does occur, it would happen with a maximum frequency of once every couple of keystrokes, thus enabling typists to spot mistakes and correct them almost "on the spot".

In this context, we can classify error correction strategies as belonging to two broad categories, depending on how frequently the user diverts their attention from the keyboard to the text entry area: OTS and TAR, the latter including the variants TAR-D (deletions only) and TAR-P (positioning and deletion) (**Figure 4**). It would be common sense to assume that TAR-D option is the costliest (in terms of time and keystrokes) and therefore users would adopt either OTS or TAR-P strategies. To examine what type of strategy is most commonly found in our dataset, we looked at the backspace use behaviour of our users. A significant proportion of all keystrokes entered (μ=20.3%, σ=4.7%) are backspaces. These are mostly single (35.03%) or double consecutive ones (55.69% cumulative frequency) and sequences up to 5 backspaces are typically used to correct an error (82.92% c.f., **Figure 5**, blue line). This supports our assumption about TAR-D not being used.

In **Figure 5**, (orange line) we plotted the number of non-backspace keystroke sequences and their length. On average, the users spot errors and correct them (using a backspace) after typing μ=16.62 characters, although as we can see the distribution is quite flat (σ=86.62). The observed downward trend shows that as users type, the probability of making and spotting a mistake (i.e. using a backspace) increases linearly with the length of the preceding correct input. Seen together, it appears that the longer a user types, the more likely they are to *make and detect* a mistake, and this mistake is most likely fixed with a short backspace sequence (1-5 keystrokes).

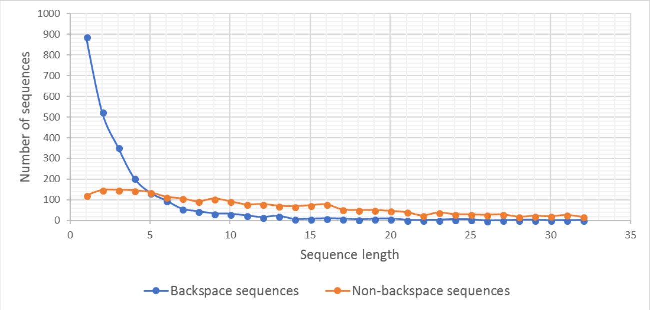


Figure 5: Distribution of backspace and non-backspace sequence length – sequences up to 32 keystrokes shown.

In [1] it is argued that user might not prefer a TAR-P strategy in real life, since cursor positioning actions are error-prone and require significant slowdowns (on average, 4.5s in a lab study), but this assumption has never been validated in real use. If TAR-P was being employed in real life, we might expect to see the average inter-key time between the last character entered and the first backspace in short backspace sequences, to be quite long (and ≈4.5s, as in [1]). In contrast, we find that the interkey time for the first backspace in short sequences (up to 5 backspaces) does not exceed the 1 second threshold (**Figure 6**).

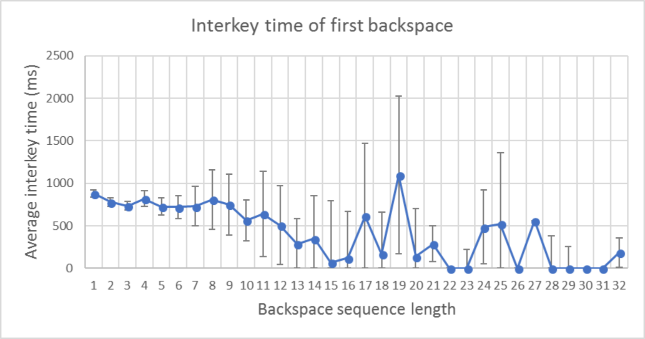


Figure 6: Average interkey time of the first backspace keystroke in sequences of up to 32 consecutive backspaces (Error bars at 95%c.i., negatives capped at 0).

Further, if it were true that participants used TAR-P strategies, we might expect that longer backspacing sequences have shorter average interkey times for the first backspace than short ones. This is because positioning the cursor and then performing a short deletion burst, should take longer than just immediately initiating a long deletion burst.

To determine whether the observed differences carry statistical significance, we grouped the backspace sequences is three bins according to their length (A=[1-5], B=[6-10], C=[11,15]) and performed an ANOVA (all bins normally distributed). The results show a statistically significant difference between the groups ($F_{(2,2443)}$=21.435, p<0.01) exists, but it indicates that shorter sequences have a lower average first backspace interkey time than longer ones (**Figure 7**). A Tukey post hoc test showed that the interkey time of the first backspace in shorter sequences (Bin A) (μ=0.846s, σ=0.613) is statistically significantly lower than Bin B (μ=1.053s, σ=0.886s, p<0.01) and Bin C (μ=1.215s, σ=0.991s, p<0.01). There was no statistically significant difference between bins B - C (p = 0.144).

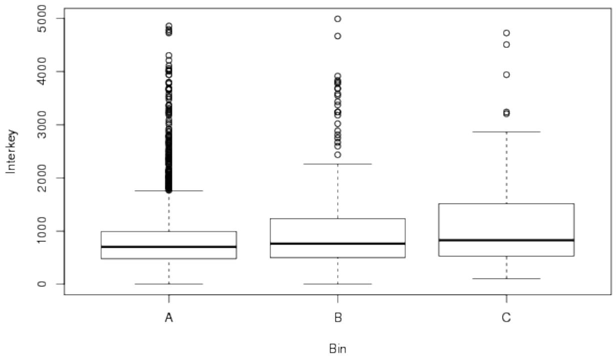


Figure 7: Boxplot of bin average first backspace interkey times (Error bars at 95%c.i.).

These results are the first confirmatory evidence that users indeed prefer OTS strategies in real life text entry. However, given the frequency of backspace use, it is not certain that TAR-P might be a worse alternative, as OTS incurs a context-switching cost to check entered text, not just for the input that needs correction, but also for the majority of input that doesn't (80% of all entered characters).

Discussion

Our work opens up a range of interesting research questions, which emerge from the analysis of our data, though it has some reasonable limitations (mostly from our sample size and age range).

Firstly, we note the fact that our participants typed quite slowly and carefully, ostensibly in an effort to avoid costly mistakes during text entry. Our slow observed entry speed contradicts the findings in [5] where speed was measured at an approximate 32WPM, albeit without filtering out any applications. This behaviour of slow input matches the observations in [4], where it was found that users deliberately slow their movement to avoid costly text entry mistakes. Further from the results in [4], we show that the slowing down of speed is not simply a product of motor behaviour alteration, but also of cognitive behaviour during the execution of the text entry composition task. We found evidence that suggests that users are frequently context-switching between text entry task (locating and pressing the appropriate key) and error-checking (looking at what was entered, and correcting mistakes). This constant shift of focus has a detrimental effect on their input speed, but even with this slowing-down and the constant context-switching, it appears that seldom does a text entry session go without some mistake slipping by undetected. With this in mind, we make the following recommendations for future research in text entry:

- Develop novel ways to keep the users' focus on the text entry task and decrease the frequency of context-switching for error checking, since an OTS strategy incurs this cost even for the majority of text that doesn't need correction: This could be achieved by providing more support (positive or negative feedback) in the screen area occupied by the keyboard, or using multimodal feedback to take advantage of the user's periphery. MaxieKeyboard already includes such mechanisms (visual feedback bar, haptic and audio feedback in case of detected spelling mistakes or autocorrects), but these remain unexplored.
- Develop better ways for managing deletion and correction of errors: Since use of the backspace key is the predominant error correction mechanism we need more research into how we can improve its function beyond its traditional behaviour.
- Develop better models and understanding of how text entry errors occur in the real world. Such models will allow us to design lab-based processes for evaluating text entry that have greater ecologic validity. In particular, we believe that we need to revisit the metrics we used for text entry method evaluation, e.g. reporting error rates as a comparison of the final submitted text with the requested text (as in transcription tasks).

Real life text entry is filled with errors. To increase the users' performance during text entry, we need not just methods that prevent errors or autocorrect them, but also methods that support the better detection and management of these.

References

1. Ahmed Sabbir Arif, Sunjun Kim, Wolfgang Stuerzlinger, Geehyuk Lee, and Ali Mazalek. 2016. Evaluation of a Smart-Restorable Backspace Technique to Facilitate Text Entry Error Correction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (CHI '16), 5151–5162. https://doi.org/10.1145/2858036.2858407
2. Ahmed Sabbir Arif, Cristina Sylla, and Ali Mazalek. 2016. Learning New Words and Spelling with Autocorrections. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces* (ISS '16), 409–414. https://doi.org/10.1145/2992154.2996790
3. Shiri Azenkot and Shumin Zhai. 2012. Touch Behavior with Different Postures on Soft

Smartphone Keyboards. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services* (MobileHCI ’12), 251–260. https://doi.org/10.1145/2371574.2371612

4. Nikola Banovic, Varun Rao, Abinaya Saravanan, Anind K. Dey, and Jennifer Mankoff. 2017. Quantifying Aversion to Costly Typing Errors in Expert Mobile Text Entry. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (CHI ’17), 4229–4241. https://doi.org/10.1145/3025453.3025695
5. Daniel Buschek, Benjamin Bisinger, and Florian Alt. 2018. ResearchIME: A Mobile Keyboard Application for Studying Free Typing Behaviour in the Wild. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (CHI ’18), 255:1–255:14. https://doi.org/10.1145/3173574.3173829
6. 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* (MobileHCI ’14), 339–348. https://doi.org/10.1145/2628363.2628408
7. Niels Henze, Enrico Rukzio, and Susanne Boll. 2012. Observational and Experimental Investigation of Typing Behaviour Using Virtual Keyboards for Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI ’12), 2659–2668. https://doi.org/10.1145/2207676.2208658
8. Andreas Komninos, Emma Nicol, and Mark D. Dunlop. 2015. Designed with Older Adults to SupportBetter Error Correction in SmartPhone Text Entry: The MaxieKeyboard. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct* (MobileHCI ’15), 797–802. https://doi.org/10.1145/2786567.2793703
9. Daniel Munger, Bruce Mehler, Bryan Reimer, Jonathan Dobres, Anthony Pettinato, Brahmi Pugh, and Joseph F. Coughlin. 2014. A Simulation Study Examining Smartphone Destination Entry While Driving. In *Proceedings of the 6th International Conference on Automotive User Interfaces and Interactive Vehicular Applications* (AutomotiveUI ’14), 28:1–28:5. https://doi.org/10.1145/2667317.2667349
10. 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* (CHI ’15), 679–688. https://doi.org/10.1145/2702123.2702597
11. Timothy A. Salthouse. 1986. Perceptual, cognitive, and motoric aspects of transcription typing. *Psychological Bulletin* 99, 3: 303–319. https://doi.org/10.1037/0033-2909.99.3.303
12. Daryl Weir, Henning Pohl, Simon Rogers, Keith Vertanen, and Per Ola Kristensson. 2014. Uncertain Text Entry on Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI ’14), 2307–2316. https://doi.org/10.1145/2556288.2557412