A Dataset of Noisy Typing on QWERTY Keyboards

Keith Vertanen
Michigan Technological University
Houghton, Michigan, United States
vertanen@mtu.edu

Per Ola Kristensson
University of Cambridge
Cambridge, United Kingdom
pok21@cam.ac.uk

ABSTRACT

Text entry is a common and important part of many intelligent user interfaces. However, inferring a user's intended text from their input can be challenging: motor actions can be imprecise, input sensors can be noisy, and situations or disabilities can hamper a user's perception of interface feedback. Numerous prior studies have explored input on touchscreen phones, smartwatches, in mid-air, and on desktop keyboards. Based on these prior studies, we are releasing a large and diverse data set of noisy typing input consisting of thousands of sentences written by hundreds of users on QWERTY-layout keyboards. This paper describes the various subsets contained in this new research dataset as well as the data format.

CCS CONCEPTS

• **Human-centered computing → Text input**.

KEYWORDS

text entry, mobile text input, touchscreen keyboard, mid-air keyboard

ACM Reference Format:
Keith Vertanen and Per Ola Kristensson. 2023. A Dataset of Noisy Typing on QWERTY Keyboards. In *28th International Conference on Intelligent User Interfaces (IUI '23 Companion), March 27–31, 2023, Sydney, NSW, Australia.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3581754.3584174

1 INTRODUCTION

The QWERTY keyboard is a common method for entering English text on desktop computers and mobile phones. It is also widely used on other devices, such as smartwatches and via mid-air keyboards in Virtual Reality (VR) and Augmented Reality (AR). However, a user's input can be noisy due to a variety of factors, such as imprecise motor control or device sensor inaccuracies. This has led to much research on ways to provide fast and accurate text input despite this noisy input.

Some prior text entry datasets are available. For example, Dhakal et al. [2] released a large crowdsourced dataset on desktop typing and Palin et al. [8] released a large crowdsourced dataset on mobile keyboard typing. Dudley et al. [3] released a high-level dataset on ten-finger typing in VR, though it unfortunately does not include low-level input data. Foy et al. [6] released a dataset on co-activations in mid-air VR typing. The dataset presented in this paper extends such prior efforts by providing a dataset covering a wide range of typing data for a variety of applications. We specifically focus on *noisy* typing on QWERTY keyboards where data was captured in such a way that it allows for detailed annotation of the dataset.

Two popular ways to address noisy input is to provide auto-correct and word predictions. In auto-correct, a user's noisy input for a sequence of keypresses is replaced by the most likely text given the noisy keypress locations. In word predictions, a noisy prefix of keypresses is used to generate the most likely words the user is writing. Both methods can benefit from improved algorithms.

While it is possible to compare different algorithms in user studies, this can be time consuming and detecting subtle differences can be difficult. Therefore carrying out comparisons via computational simulations is an attractive alternative as it can be performed repeatedly with high reproducibility without the variance and costs associated with human-in-the-loop experiments. While it is possible to simulate noisy keypresses by, for example, introducing Gaussian noise to a key's center coordinate [5], this fails to simulate other aspects of user noise, such as accidental extra keypresses, missing keypresses, and undershoot/overshoot errors resulting from a keypress' preceding or following letters.

Instead of introducing synthetic noise, a more realistic simulation might use data from previous user input of words or sentences. This paper describes the release of a large set of typing data collected over numerous previous studies involving hundreds of study participants typing a total of 16,468 sentences. We have also included development data (typically recorded by a paper's authors) of 3,881 sentences. The dataset includes instances of people typing on touchscreen phones, smartwatches, mid-air AR/VR keyboards, and desktop keyboards.

2 DATASET DESCRIPTION

In our dataset[1], we separated our past studies into four groups based on the type of input device used:

(1) **Phone** — Touchscreen input on a mobile phone. This group contains 2,597 development sentences and 8,695 test sentences.
(2) **Watch** — Touchscreen input on a smartwatch. This group contains 225 development sentences, 724 practice sentences (sentences typed by participants but not analyzed in the original paper), and 2,684 test sentences.
(3) **Mid-air** — Input on a keyboard that appeared in mid-air in either VR or AR. This group contains 693 development sentences and 2,739 test sentences.

IUI '23 Companion, March 27–31, 2023, Sydney, NSW, Australia

ACM ISBN 979-8-4007-0107-8/23/03.
https://doi.org/10.1145/3581754.3584174

[1] https://osf.io/5xwng/

(4) **Desktop** — Typing on a conventional desktop keyboard. This group contains 367 development sentences and 2,350 test sentences.

We now describe the papers in each group along with any unique aspects of the collected data. For full details, see the `readme.txt` contained in the associated directory of the dataset.

2.1 Phone

Data in this group was collected on a touchscreen phone. In all studies, participants typed sentences from the Enron mobile dataset [11].

In the first paper [13], participants typed on a standard sized touchscreen phone keyboard and several smaller keyboards (i.e. similar in size to a smartwatch). The paper tested typing with and without visual feedback prior to recognition. The paper tested different options for signifying the spaces between words, including using a spacebar, swiping to the right, or skipping spaces entirely.

In the second paper [14], sighted participants typed both normally and while blindfolded. The study compared signifying the space between words by a spacebar and by swiping to the right.

Both of the studies involved participants typing an entire sentence prior to recognition. Word-at-a-time input is a more common input method. To support researchers interested in word-at-a-time input, we provide not only the original user input, but also data files where we have separated a user's input into each word in the reference sentence. We did this by using the VelociTap decoder [13] to force align the input with the reference text. We dropped sentences that failed to force align (for example, when a user typed something different from the reference text).

2.2 Watch

Data in this group was collected on a Sony Smartwatch 3. Participants copied sentences from the Enron mobile dataset [11] or from Twitter [10]. Twitter messages were either entirely in-vocabulary with respect to a 100Ḱ word list, or had a single out-of-vocabulary word. In some cases participants composed their own novel messages. In the case of composition tasks, we provide a reference text via the crowdsourcing procedure in [12], by having participants type their intended text on a laptop, or by having participants dictate their text to the experimenter [7].

In a paper examining the impact of word, multiple word, and sentence decoding [9], participants typed one word, two words, or an entire sentence at-a-time. Participants also copied phrases or composed novel messages in which they decided how much text to input prior to recognition. We provide force aligned versions of conditions in which participants typed multiple words prior to recognition.

In a paper on smartwatch typing [10], participants typed one word-at-a-time. This study compared different ways users might interact with the predictions offered by a smartwatch keyboard. It also tested two different keyboard designs, one offering multiple predictions, and a simpler keyboard offering only two choices. Some experiments tested a lock letter feature which allowed users to specify letters in a word that were not subject to auto-correction. Locked letters are explicitly denoted in our data.

We dropped sentences in which participants used a word prediction prior to typing all of a word. We did this to make the data consistent with the other studies in the dataset where we have input events for all letters. Having a complete trace of all input for a word can also be necessary when conducting computational simulations. For example, when testing a new word prediction algorithm, the algorithm might fail to predict a user's intended word as early as the system in the original study. In this case, subsequent noisy keypresses are required to faithfully simulate performance.

In a paper examining composition tasks [7], participants composed sentences they thought would be easy or hard for the auto-correct algorithm. Participants typed one word-at-a-time. Similar to the paper examining smartwatch typing [10], participants could lock letters and these events are denoted in the data. We dropped sentences in which participants used word predictions.

2.3 Mid-air

Data in this group was collected using mid-air keyboards displayed either in VR or AR. A participant typed by poking their index finger through the keyboard plane. All studies had participants copy sentences from the Enron mobile dataset [11]. Participants typed one word-at-a-time.

In a paper examining AR typing [4], participants typed on an AR keyboard using a Microsoft HoloLens. The interface tracked a participant's hands using the built-in HoloLens sensor. The study compared a keyboard with and without letter labels. It also investigated a feature allowing participants to precisely select letters. These precisely selected letters are explicitly denoted in our data.

In a paper investigating VR typing [1], participants typed on a VR keyboard using an HTC Vive. The interface tracked a participant's hands via a Leap Motion controller attached to the VR headset. The study compared typing with one or two hands, compared a normal versus split QWERTY keyboard layout, and had users type on an invisible mid-air keyboard. In the study, we tracked which hand a participant tapped a key with and have denoted this in our data. For the invisible keyboard experiment, participants could choose the keyboard's size and location. We provide in our dataset each participant's chosen keyboard layout.

2.4 Desktop

While typing on a desktop keyboard is normally relatively easy, it can be more challenging when a user cannot see the keyboard (as may be the case while wearing a VR headset). In the studies in this group, each keypress on a physical keyboard was treated as noisy by assuming it was a touch input event located at the center of a key on a virtual keyboard layout that mirrored the physical keyboard. In this way, typing mistakes (e.g. accidentally hitting an adjacent key) can be auto-corrected. Participants copied sentences from the Enron mobile dataset [11]. Participants typed an entire sentence before recognition.

In the first desktop typing paper [15], participants typed without visual occlusion, occlusion via a physical box, and occlusion via a VR headset. In the second desktop typing paper [16], participants typed without visual occlusion, while wearing a VR headset with no visual feedback, and while wearing a VR headset that displayed visual feedback denoting which keys were being hit.

```
ID: mobile1700
LEFT: Tracy and Jim are trying to track down six more copies.
ORIG: Could you see where this stands?
RIGHT: We are in 49C1.
REF: could you see where this stands
IN: -266,-457,0|494,23,3120|262,11,7201|907,-135,11520|-521,-117,16020
IN: -13,103,21760|681,71,25160|244,120,27581
IN: -772,-146,39160|-610,34,42300|-543,15,44880
IN: -792,10,50461|23,-160,54220|-626,26,57120|-461,29,59880|-652,-1,61800
IN: -277,48,66420|-3,-75,68360|295,168,69780|-689,-180,72940
IN: -754,-201,77600|-251,36,79600|-998,-97,82300|63,-349,84260|-552,-159,86081|-746,-174,87600
```

Table 1: Log for a sentence in "Experiment 1: Selection Method Evaluation" from [4]. In this experiment, input was word-at-a-time and the six IN lines show the input for each word typed. The first word "could" has five letters and the first IN line contains five keypress events separated by vertical bars. The first event was located at an (x, y) of $(-266, -457)$, 3120 ms later the second key was pressed at $(494, 23)$, and so on. REF is the text the participant was asked to copy. LEFT and RIGHT give the text in the original Enron email that appeared to the left and right of the sentence being copied. ORIG shows the sentence being copied before any removal of punctuation or changes to case. ID is a unique identifier for the reference phrase.

3 DATASET ORGANIZATION

3.1 Directory and file structure

Our dataset contains a separate subdirectory for every typing experiment in the cited papers. We have separated each participant's input into a separate file. In experiments that involved a participant completing several conditions, each condition is in a separate file. For example, in a visual feedback versus no visual feedback experiment [13], the `phone/vt_exp1/test` subdirectory contains the files `p1_feed.log` and `p1_nofeed.log` for the data from participant 1 in the feedback and no feedback conditions respectively.

3.2 Log file format

A participant log file consists of a sequence of sentence input tasks in the order they received them in the experiment. The data for each input task appears on a sequence of lines with separate tasks separated by a blank line. The meaning of each line is determined by an uppercase keyword followed by a colon at the start of a line. Table 1 shows an example of an input task with the following required fields (as well as a few optional ones):

- **ID** — Unique ID for the phrase the participant was asked to copy. In case of composition tasks, this is simply a string followed by a 0-based index.
- **REF** — The text the participant was asked to copy. In the case of composition tasks, this is the text that was determined via a crowdsourced procedure [12] or provided by the participant [7].
- **IN** — The sequence of inputs that were provided by a participant for a particular recognition event. In the case of word-at-a-time experiments, there is a separate IN line for every word written. In the case of sentence-at-a-time experiments, there is a single IN line.
  - The separate letter inputs in a given IN line are separated by a vertical bar.
  - Letter input data consists of an x-location, y-location, and timestamp. The x- and y-location are relative to the keyboard layout defined in a separate file associated with each experiment. The timestamp is the milliseconds that have elapsed since the first input event of the current sentence.
  - For some input devices, multiple sensor readings were recorded for a given letter input. For example, touchscreen phones can record all x- and y-locations from the touch down to touch up event. Where this was available, we have included the sequence of coordinates separated by semicolons. For example, the letter input "`404,713,176;397,709,200;399,710,213;399,710,214`" indicates the user touched down on the screen 176 ms after the first letter in a sentence and touched up after 214 ms. The touchscreen in this case recorded four slightly different (x, y) locations during the tap.
  - For some interfaces, users could signal that a particular letter was not subject to auto-correct. When users provided such a signal, the data for that letter's input is suffixed with "`:1.0:0`". The `1.0` indicates complete certainty in the user's input (i.e. no chance of auto-correct). The `0` indicates that this letter should not be subject to deletion in the recognizer's search. In the released data, there are no interfaces that made a softer decision about whether a letter should be subject to auto-correct. We adopted this format to allow such data in the future.
  - For some interfaces, users could use a backspace key to delete a previous tap prior to recognition. Our input sequences do not contain an event for the backspace key or the previous tap that was deleted. This was necessary as many of the interfaces did not log the details of the backspace event or the deleted tap. This also simplifies use of our data as each input sequence represents the sequential typing of each letter in a sentence or word. However, this may mean some sequences are overall less noisy since the final input events are the coordinates after any user correction of individual taps.

For some experiments, log files may contain additional fields:

- **LEFT** — The text (if any) appearing to the left of the text the participant was copying. If the left context contains line

```
# Keyboard for HoloLens
# Format: character, x-center, y-center, width, height
# row 1
q;-900;0;200;200
w;-700;0;200;200
e;-500;0;200;200
r;-300;0;200;200
t;-100;0;200;200
y;100;0;200;200
```

Table 2: Part of the keyboard definition for "Experiment 1: Selection Method Evaluation" in [4]. The 5-tuples define a key's character, (x, y) center, width and height.

breaks, these have been replaced with vertical bars. This may be useful for experiments considering language modeling leveraging surrounding context.

- **RIGHT** — The text (if any) appearing to the right of the text the participant was copying.
- **ORIG** — The original text of the phrase being copied before any changes to case or symbols. For example, the ORIG text of the Enron mobile phrase ID mobile1192 is "If not can I call you?" but the text the participant was asked to copy (the REF field) was "if not can i call you".
- **DEL** — In some experiments, participants could delete the entire previous word and type it again. In cases where this was recorded, we have provided the input for these deleted words. This might be useful as instances of more difficult input that was initially recognized incorrectly.

3.3 Keyboard definition files

The x- and y-coordinates provided in the log files are relative to the keyboard used in a study. The geometry of the keyboard is defined in a file in the corresponding experiment directory in the dataset. This file consists of lines containing 5-tuples. The 5-tuple defines a key's: character, x-center coordinate, y-center coordinate, width, and height. The elements of the 5-tuple are separated by semicolons. Lines starting with the # symbol are comments. We use "<sp>" to denote the spacebar key, and "<b>" to denote the backspace key (if any). Table 2 shows an example of part of a keyboard definition.

4 CONCLUSIONS

We have described a new dataset containing noisy typing data on QWERTY keyboards from nine previous papers. We have wrangled the previously unreleased data into a single common directory structure and file format. The diversity of device types and experimental conditions allow researchers to test both under "easy" noisy input conditions (e.g. typing on a large phone keyboard) and under more challenging conditions (e.g. typing on an invisible mid-air keyboard). We believe this dataset will be helpful to other researchers interested in improving text entry algorithms and interfaces.

ACKNOWLEDGMENTS

This dataset was created from studies supported by Google Faculty awards (K.V. and P.O.K.), EPSRC grants EP/N010558/1, EP/N014278/1, EP/R004471/1 (P.O.K.), NSF IIS-1909248 and NSF IIS-1750193 (K.V.).

REFERENCES

[1] Jiban Adhikary and Keith Vertanen. 2021. Typing on Midair Virtual Keyboards: Exploring Visual Designs and Interaction Styles. In *Human-Computer Interaction – INTERACT 2021*. 132–151. https://doi.org/10.1007/978-3-030-85610-6_9

[2] Vivek Dhakal, Anna Maria Feit, Per Ola Kristensson, and Antti Oulasvirta. 2018. Observations on typing from 136 million keystrokes. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–12. https://doi.org/10.1145/3173574.3174220

[3] John Dudley, Hrvoje Benko, Daniel Wigdor, and Per Ola Kristensson. 2019. Performance envelopes of virtual keyboard text input strategies in virtual reality. In *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 289–300. https://doi.org/10.1109/ISMAR.2019.00027

[4] John J. Dudley, Keith Vertanen, and Per Ola Kristensson. 2018. Fast and Precise Touch-Based Text Entry for Head-Mounted Augmented Reality with Variable Occlusion. *ACM Transactions on Computer-Human Interaction (TOCHI)* 25, 6, Article 30 (12 2018), 40 pages. https://doi.org/10.1145/3232163

[5] Andrew Fowler, Kurt Partridge, Ciprian Chelba, Xiaojun Bi, Tom Ouyang, and Shumin Zhai. 2015. Effects of Language Modeling and Its Personalization on Touchscreen Typing Performance. In *Proceedings of the SIGCHI Conference on Human factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*. ACM, New York, NY, USA, 649–658. https://doi.org/10.1145/2702123.2702503

[6] Conor R Foy, John J Dudley, Aakar Gupta, Hrvoje Benko, and Per Ola Kristensson. 2021. Understanding, Detecting and Mitigating the Effects of Coactivations in Ten-Finger Mid-Air Typing in Virtual Reality. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–11. https://doi.org/10.1145/3411764.3445671

[7] Dylan Gaines, Per Ola Kristensson, and Keith Vertanen. 2021. Enhancing the Composition Task in Text Entry Studies: Eliciting Difficult Text and Improving Error Rate Calculation. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 725, 8 pages. https://doi.org/10.1145/3411764.3445199

[8] 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*. 1–12. https://doi.org/10.1145/3338286.3340120

[9] Keith Vertanen, Crystal Fletcher, Dylan Gaines, Jacob Gould, and Per Ola Kristensson. 2018. The Impact of Word, Multiple Word, and Sentence Input on Virtual Keyboard Decoding Performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. ACM, New York, NY, USA, Article 626, 12 pages. https://doi.org/10.1145/3173574.3174200

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

[11] Keith Vertanen and Per Ola Kristensson. 2011. A Versatile Dataset for Text Entry Evaluations Based on Genuine Mobile Emails. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices & Services* (Stockholm, Sweden) *(MobileHCI '11)*. ACM, New York, NY, USA, 295–298. https://doi.org/10.1145/2037373.2037418

[12] Keith Vertanen and Per Ola Kristensson. 2014. Complementing Text Entry Evaluations with a Composition Task. *ACM Transactions of Computer Human Interaction* 21, 2, Article 8 (February 2014), 33 pages. https://doi.org/10.1145/2555691

[13] Keith Vertanen, Haythem Memmi, Justin Emge, Shyam Reyal, and Per Ola Kristensson. 2015. VelociTap: Investigating Fast Mobile Text Entry Using Sentence-Based Decoding of Touchscreen Keyboard Input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*. ACM, New York, NY, USA, 659–668. https://doi.org/10.1145/2702123.2702135

[14] Keith Vertanen, Haythem Memmi, and Per Ola Kristensson. 2013. The Feasibility of Eyes-free Touchscreen Keyboard Typing. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility* (Bellevue, Washington) *(ASSETS '13)*. ACM, New York, NY, USA, Article 69, 2 pages. https://doi.org/10.1145/2513383.2513399

[15] James Walker, Scott Kuhl, and Keith Vertanen. 2016. Decoder-Assisted Typing using an HMD and a Physical Keyboard. In *Extended Abstracts of the the ACM Conference on Human Factors in Computing Systems* (San Jose, CA) *(CHI EA '16)*.

[16] James Walker, Bochao Li, Keith Vertanen, and Scott Kuhl. 2017. Efficient Typing on a Visually Occluded Physical Keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 5457–5461. https://doi.org/10.1145/3025453.3025783