Validating State-of-the-Art Virtual Keyboard Simulators with Empirical Data

IOULIA SIMOU, University of Patras, Greece
ANDREAS KOMNINOS, University of Patras, Greece
IOANNIS GAROFALAKIS, University of Patras, Greece

Recent research in the field of mobile text entry aims to address the need for faster and more generalisable evaluation by introducing computational models that simulate human behavior. In this paper, we examine a state-of-the-art open-source simulator of text entry with virtual keyboards (RoboType), and present a comparison between empirical data of human text entry performance and the data generated under the same conditions by RoboType. We find that while the simulator does indeed generate realistic behaviours, metrics derived from it deviate from real data on a like-for-like comparison basis. We conclude by reflecting on potential corrective actions to better approach true human typing behavior.

CCS Concepts: • **Computing methodologies** → **Simulation evaluation**; • **Human-centered computing** → *Ubiquitous and mobile computing*; **Text input**.

Additional Key Words and Phrases: Simulation, Text Input, Evaluation

ACM Reference Format:
Ioulia Simou, Andreas Komninos, and Ioannis Garofalakis. 2018. Validating State-of-the-Art Virtual Keyboard Simulators with Empirical Data. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 9 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Mobile text entry is a key aspect of human – computer interaction (HCI), enabling communication across various devices and in several contexts. The rapid development of technology has created the need for efficient and user-friendly text entry interfaces, the evaluation of which is a topic of ongoing research. Novel text entry systems are typically evaluated in laboratory settings, using transcription tasks, which require participants to quickly and accurately copy a set of phrases presented to them during the experiment. The effort required for organizing such studies can be quite substantial and costly for the researchers, mainly due to the recruitment, need for consent and compensation of participants [5]. Because of constraints like these, lab experiments in text entry are often limited in sample numbers, and further, studies with diverse populations (e.g. non-fluent speakers, elderly or disabled persons) are even harder to organise and conduct. This creates an increasing need for models able to predict and mimic diverse human mobile touchscreen typing behaviors, while achieving comparable performance.

Simulation is a powerful tool used in the field of HCI, to model and analyze the performance of a system. In the context of mobile text entry, simulation can be utilized to reduce the costs and time needed to perform necessary experiments, for example to evaluate novel text entry interfaces, by mimicking the behavior of human participants.

Authors' Contact Information: Ioulia Simou, University of Patras, Rio, Greece, simo@ceid.upatras.gr; Andreas Komninos, University of Patras, Rio, Greece, akomninos@ceid.upatras.gr; Ioannis Garofalakis, University of Patras, Rio, Greece, garofala@ceid.upatras.gr.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.
Manuscript submitted to ACM

We present an evaluation of the typing behavior of RoboType, an open-source simulator developed by Komninos [8], in comparison to the typing behavior exhibited by participants of a mobile text entry experiment [6]. We present our findings, identifying limitations and potential future improvements of the simulator.

2 Background

Over recent years, there has been an increasing interest in simulating human behavior in text entry tasks. There are several attempts where text entry has been studied mainly in terms of motor performance. Bi et al. [3] modelled finger touch-based input on modern touchscreen devices by proposing an expanded form of Fitts' law, based on a dual distribution hypothesis, and provided a more accurate prediction of movement time and user performance on touchscreens. In Cao et al. [4], researchers developed a model for simulating transcription typing tasks on a touchscreen keyboard, incorporating the QN-ACTR cognitive architecture.

However, studies have shown that typing is not solely a matter of motor performance, as visual attention has been shown to play a crucial role during the typing process. A study of mobile text entry in lab settings, using gaze-tracking alongside other data capture methods, showed that there indeed exists a frequent allocation of attention between the keyboard and text entry area, during transcription tasks to perform error-checking actions [6]. Another gaze-tracking mobile text entry study discovered that focusing solely on the text entry area accounted for up to 25% of the user's fixations during lab transcription tasks [9].

Jokinen et al. [7] introduced a model implemented via reinforcement learning. The model simulated co-ordination of eye and finger movements, under the theory of optimal supervisory control. It assumed the agent's decisions are optimal, given its bounds and those of the design, in pursuit of maximal typing performance. A more recent work, CRTypist [10], introduced a model that enabled accurate simulation of typing in terms of typing speed and error correction strategies. Among other results, CRTypist generated eye and finger movements that were comparable to those of human typists, being able to represent a broad range of individual differences.

2.1 The RoboType text entry simulation platform

Although previous literature has used simulation as a research tool to explore theory or design spaces, we are aware of only one simulation tool currently available freely to the community (RoboType [8]). RoboType is an open-source simulator for text entry written entirely in Python, with the ability to support and evaluate various text entry systems. RoboType can be used to set up complete experimental protocols with the use of synthetic (virtual) participants, thus, allowing researchers to overcome the obstacles of costly participant recruitment and experiment execution. Two main components are used to simulate text entry, the *Keyboard* and the *Finger*. Each synthetic user is associated with appropriate motor and cognitive parameters, giving researchers the ability to simulate various types of real participants (e.g. participants with visual or motor impairments).

The Keyboard component is constructed by reading an XML file that contains the desired keyboard layout properties, such as key width, height and labels. The Keyboard object is also responsible for maintaining the key coordinates, which are utilized during text input, to identify the pressed key. The Finger component is responsible for mimicking the movement of an actual finger around the keyboard during text entry. To simulate a transcription task, the Finger component takes as a parameter a target sentence and attempts to type the string by moving the finger from target to target over the Keyboard. For further details on the RoboType simulator, readers are encouraged to consult the original paper [8], where the authors provide evidence that the simulator is able to produce realistic findings that match results from empirical research.

3 Motivation, Materials and Method

3.1 Motivation and Materials

Despite the promising evidence shown in [8], the authors did not evaluate performance of the simulator against empirical data on a like-for-like basis (i.e. utilising the exact same experimental protocol). The aim of this short paper is to evaluate the typing behavior of RoboType in comparison to real human behavior, under the conditions of one finger typing and immediate correction of typing errors using the backspace key. To achieve this, we utilized open source typing data provided by Jiang et.al [6], using exclusively data from typing with the index finger. This dataset has also been used in [8] in order to derive the synthetic participant profile parameters and to construct finger kinematics models, therefore, we should be able to directly compare the actual human performance, as presented in the dataset, against RoboType, by replicating in the simulation the exact protocol ran in the original study by [6]. The dataset contains keystroke data from 30 participants, of whom 27 were right-handed, each conducting 20 unique trials (i.e. attempts to enter a complete phrase) for this typing condition. Each actual participant in [6] has a synthetic counterpart in [8], therefore a direct comparison between real and synthetic participant performance can be made. For a detailed description of the data and collection process, readers can consult the original paper by Jiang et al. [6].

3.2 Procedure

Prior to simulating the typing process, it was necessary to create a corresponding keyboard layout for RoboType that matched the one used during data collection. Based on information and careful analysis of screenshot material in [6], an XML file describing the corresponding Finnish keyboard layout was created and incorporated into RoboType (Fig. 1).

As RoboType assumes that errors are immediately detected and corrected by the synthetic participants, we removed from the Jiang et al. dataset all typing trials where *consecutive* touches of the backspace key were present. The rationale here is that consecutive backspaces indicate that a user only detected an input error after several characters had been committed to the input stream, therefore deleting all of them to get to the mistake and fix it. This reduced the number of trials to 346 (out of 599 in the original dataset) from 29 participants. After cleaning the Jiang et al. dataset from redundant data that were present, we used RoboType to simulate the typing process. Each synthetic participant was assigned the same target sentences as those used in the corresponding real participants' trials. The motor parameters for each synthetic participant were maintained as in the original paper [8], while the parameter of each synthetic user for Visual Scan Time was randomly selected from a Gaussian distribution with a mean set at 50 ms and standard deviation at 0.2.

To compare the performance of real and synthetic participants, we computed the following text entry metrics [1, 6, 11]: Words per Minute (WPM): the number of words divided by the time spent typing during a trial, Keystrokes per Character (KSPC): ratio of the number of entered characters to the number of characters in the transcribed text, Keystrokes per Second (KSPS): average number of keypresses per second during a trial, Inter-Key Interval (IKI): time (in seconds) between consecutive keypresses, Corrected Error Rate (CER): rectified incorrect keypresses as a percentage of the sum of incorrect and correct keypresses, Uncorrected Error Rate (UER): incorrect key presses as a percentage of the sum of incorrect and correct keypresses, Total Error Rate (TER): sum of the previous two error rate metrics, and Backspaces: average number of Backspace presses.

The metrics for real participants were computed directly from their typing data. The following procedure was used to calculate the metrics for the synthetic participants: 1) each synthetic user was assigned a set of target sentences (as previously described) with which to perform trials; 2) RoboType, using a synthetic user, was instructed to perform each

trial (i.e. type each target sentence) 20 times; 3) the text entry metrics were computed and recorded for each iteration; 4) after completing all 20 iterations, the average metrics for each synthetic user, per trial, were computed and recorded; 5) the overall average metrics of each synthetic user were calculated upon completion of their target sentences. In step 2, we performed repeated runs of the same trial in order to avoid result biases due to the stochastic nature of the simulator. Total execution time for the synthetic experiment was under 1 minute. RoboType also affords the ability to modify the speed with which a synthetic user types, by applying a multiplier to their original finger kinematics parameters. Therefore at speed $V_p = 1.0$ a synthetic participant uses kinematic parameters that are directly equivalent to their real-life counterpart, while other values of V_p may make the user slower or faster by a factor equal to that value.

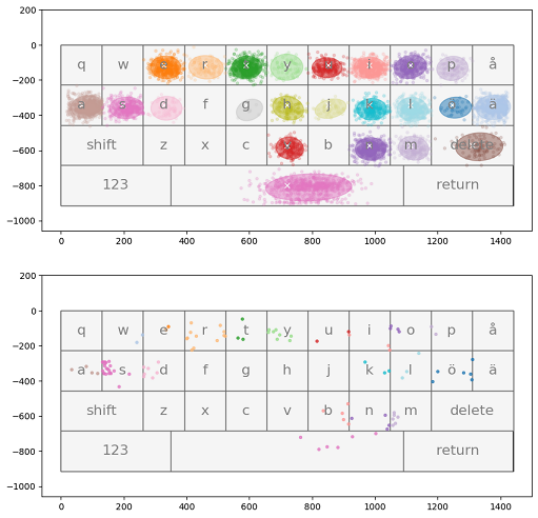
4 Results

A visualisation of the key touches is presented in Figure 1. We distinguish touches in two categories: Hits are the touches which resulted in the desired letter being input, while Misses are the opposite. When simulating input with RoboType, it is easy to know both a synthetic participant's intended key target, as well as the outcome of an attempt, and in fact RoboType logs both for input stream analysis. Knowing a real participant's intended key target however is impossible. We can, however, infer it with *reasonable* accuracy, since we can assume the participant intended to copy the presented phrase accurately, and thus we can identify misses as the outcomes which the participant backspaced to and replaced with another character, or outcomes which do not match the target phrase, and were left uncorrected. This, of course, is not always true, as, for example, the participant may have memorised the phrase to be copied slightly differently from the original, or may intentionally type incorrectly due to dyslexia or inability to recall the correct spelling a word (e.g. "I saw a *pair*" vs. the original "I saw a *pear*"). Nevertheless, we proceed with the assumption that there would not be that many cases in the dataset. As such, the correct key touches (hits) of real participants are displayed in the upper part of Figure 1a, while the lower part shows the incorrect key touches (misses), i.e. the ones followed by a backspace and consequently deleted. Similarly, Figure 1b illustrates the hits and misses of key presses produced by RoboType for one iteration of all target sentences. It is important to note that Figure 1b includes presses on modifier keys that do not affect the final input, such as Shift and Return, whereas this information might not have been logged for real participants' data and is therefore not presented in Figure 1a.

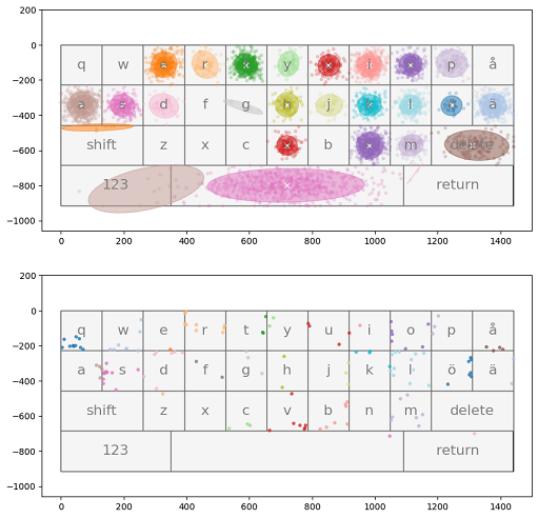
4.1 Initial results for text entry metrics

The mean and standard deviation of computed metrics are presented in the columns "Real participants" and "$V_p = 1.0$" (for synthetic participants) of Table 1. Given the sample sizes (29 real and 29 synthetic participants), the Shapiro-Wilk test was utilized to assess the normality of each metric. The results were further validated by examining corresponding Q-Q plots. The independent-samples t-test was used for comparing the data for each metric between real and synthetic participants when both groups met the assumption of normality. When the normality assumption was not satisfied, the Mann-Whitney U test was employed as a non-parametric alternative. Statistical results are reported with a significance level of $\alpha = 0.05$.

The mean typing speed for the group of real participants was 30.98 WPM ($\sigma = 5.83$), compared to 25.70 WPM ($\sigma = 2.91$) for the synthetic participants group. An independent-samples t-test indicated that real participants typed significantly faster than synthetic participants, $t(56) = 6.306$, $p < 0.001$. This is also supported by an independent-samples t-test performed for the KSPS metric (real: $\bar{x} = 2.66$, $\sigma = 0.46$; synthetic: $\bar{x} = 2.22$, $\sigma = 0.27$), $t(56) = 5.788$, $p < 0.001$, and by a Mann-Whitney U test performed for the IKI metric (real: $\bar{x} = 0.39$, $\sigma = 0.07$; synthetic: $\bar{x} = 0.45$, $\sigma = 0.07$), $U = 167.0$, $p < 0.001$). Mann-Whitney U tests were used for the rest of the metrics. The results indicated no significant difference for



(a) Correct (upper part) and incorrect (lower part) key touches of real participants.



(b) Correct (upper part) and incorrect (lower part) key touches of synthetic participants.

Fig. 1. Key touches of real and synthetic participants (ellipses at the 95% confidence interval).

the two groups for metrics KSPC (U = 423.0, p = 0.975), Number of Backspaces (U = 395.0, p = 0.697), CER (U = 405.0, p = 0.816), and TER (U = 462.0, p = 0.524). Significant difference was found for metric UER (U = 580.0, p < 0.001), an expected result, since the current settings of RoboType force immediate correction of typing errors, thus zeroing the UER statistic for each synthetic participant. Hence, UER and TER (which includes UER) will be left out in the next sections.

4.2 Increasing the typing speed of RoboType and recalculating the metrics

As indicated by the results in the previous section, RoboType tends to underestimate the typing speed of real participants under the conditions of immediate correction of a typing error and index finger typing. To address this, the typing speed setting of RoboType (initially at $V_p = 1.0$) was adjusted to $V_p = 1.1$ and $V_p = 1.2$ in an attempt to better approximate the typing speed of real participants. The complete results can be found in Table 1.

The mean typing speed for the synthetic participants group for $V_p = 1.1$ was 27.36 WPM ($\sigma = 2.94$) and for $V_p = 1.2$ was 28.79 WPM ($\sigma = 2.98$). In both cases, independent-samples t-tests indicated that real participants still typed significantly faster than synthetic participants, $V_p = 1.1$: t(56) = 4.420, p < 0.001; $V_p = 1.2$: t(56) = 2.800, p = 0.007. In the case of IKI, for $V_p = 1.1$, a Mann-Whitney U test indicated significant difference between the two groups (U = 270.0, p = 0.020), while for $V_p = 1.2$, the test indicated no significant difference between the groups (U = 413.0, p = 0.913). Similar results are observed in the case of KSPS, where independent-samples t-tests indicated significant difference when $V_p = 1.1$: t(56) = 3.525, p < 0.001, but no significant difference when $V_p = 1.2$: t(56) = 1.273, p = 0.208.

Given the observed increase in the mean number of Backspaces ($V_p = 1.1$: $\bar{x} = 0.56$, $\sigma = 0.39$, U = 269.5, p = 0.019; $V_p = 1.2$: $\bar{x} = 0.80$, $\sigma = 0.50$, U = 167.0, p < 0.001), one can posit that, in conjunction with the findings presented earlier, as the speed setting increases, RoboType begins to approximate the entry rates of real participants, albeit at the expense of accuracy. This inference is further supported by the corresponding increase in the mean error rate metrics. For these and the remaining metrics, statistical tests revealed significant differences between the two groups.

		Real participants	$V_p = 1.0$	$V_p = 1.1$	$V_p = 1.2$
WPM	$\bar{x}$	30.98	**25.70**	**27.36**	**28.79**
	σ	5.83	2.91	2.94	2.98
KSPS	$\bar{x}$	2.66	**2.22**	**2.40**	2.58
	σ	0.46	0.27	0.28	0.31
KSPC	$\bar{x}$	1.03	1.04	1.05	1.08
	σ	0.06	0.03	0.04	0.05
IKI	$\bar{x}$	0.39	**0.45**	**0.45**	0.39
	σ	0.07	0.07	0.17	0.06
CER (%)	$\bar{x}$	1.42	1.76	2.51	3.58
	σ	2.63	1.34	1.69	2.13
UER (%)	$\bar{x}$	0.53	**0.00**	**0.00**	**0.00**
	σ	4.68	0.00	0.00	0.00
TER (%)	$\bar{x}$	1.95	1.76	2.51	3.58
	σ	5.31	1.34	1.69	2.13
Num. of Backspaces	$\bar{x}$	0.30	0.39	**0.56**	**0.80**
	σ	0.57	0.30	0.39	0.50

Table 1. Metrics across different speeds.

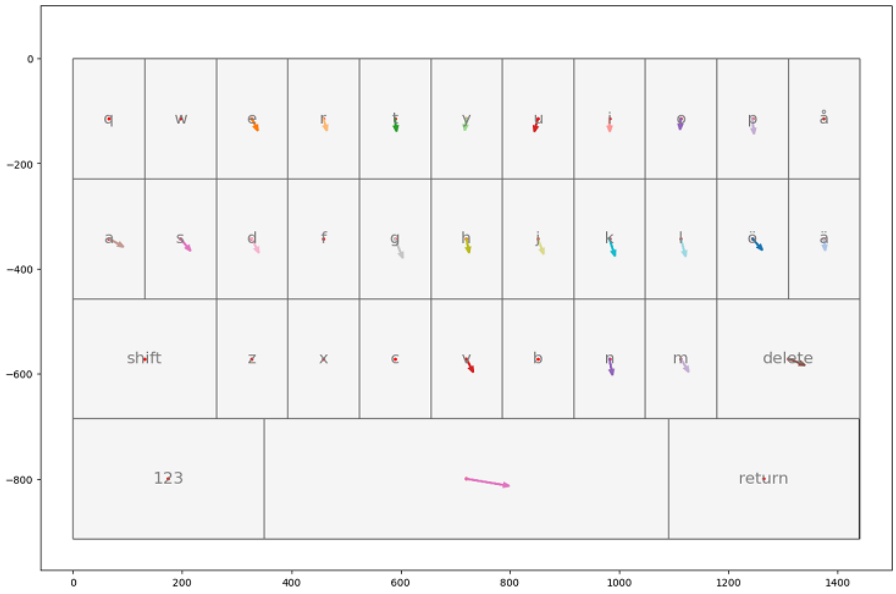


Fig. 2. Offset vectors based on real participants' key press distributions.

4.3 Incorporating offset in the touch target of RoboType

Further following up on the original RoboType paper, we decided to investigate whether real participants' touch distributions are indeed offset from key centers in this case. To accomplish this, we retained only the correct real participants' key touches (hits). As can be seen in Fig. 2, the centers of the participants' touch distributions are located mostly downwards and to the right, relatively to the center coordinates of the target button.

	Real participants	V_p = 1.0	V_p = 1.1	V_p = 1.2
WPM	$\bar{x}$ = 30.98	$\bar{x}$ = **25.43**	$\bar{x}$ = **27.05**	$\bar{x}$ = **28.38**
	σ = 5.83	σ = 2.80	σ = 2.86	σ = 2.90
KSPS	$\bar{x}$ = 2.66	$\bar{x}$ = **2.22**	$\bar{x}$ = **2.41**	$\bar{x}$ = 2.58
	σ = 0.46	σ = 0.27	σ = 0.29	σ = 0.30
KSPC	$\bar{x}$ = 1.03	$\bar{x}$ = 1.05	$\bar{x}$ = **1.07**	$\bar{x}$ = **1.10**
	σ = 0.06	σ = 0.04	σ = 0.04	σ = 0.06
IKI	$\bar{x}$ = 0.39	$\bar{x}$ = **0.45**	$\bar{x}$ = **0.41**	$\bar{x}$ = 0.40
	σ = 0.07	σ = 0.07	σ = 0.05	σ = 0.07
CER (%)	$\bar{x}$ = 1.42	$\bar{x}$ = 2.26	$\bar{x}$ = **3.25**	$\bar{x}$ = **4.38**
	σ = 2.63	σ = 1.57	σ = 1.95	σ = 2.45
Num. of Backspaces	$\bar{x}$ = 0.30	$\bar{x}$ = 0.50	$\bar{x}$ = **0.73**	$\bar{x}$ = **0.99**
	σ = 0.57	σ = 0.35	σ = 0.45	σ = 0.58

Table 2. Metrics across different V_p values with offset target. Statistically significant results in bold.

Based on the length and angle of the offset vectors, we computed (in pixels) the vertical and horizontal offsets as:

$$\text{Offset in x axis} = \text{length} \times \cos(\text{angle}) \tag{1}$$

$$\text{Offset in y axis} = \text{length} \times \sin(\text{angle}) \tag{2}$$

where *length* is the Euclidean distance between the actual key center and the center of the touch distribution for the corresponding key and *angle* is in radians.

After computing the offsets, we integrated them into the RoboType code, instructing this way the synthetic finger of RoboType to target the position calculated using these offsets rather than the actual key centers. The mean horizontal offset for keys was at 9.22 pixels (σ = 14.31), while the mean vertical offset was at -16.72 pixels (σ = 4.64). The negative vertical offset indicates that, on average, the key touches occurred below the actual key centers. The SPACE key has the largest horizontal offset, which is expected given its greater width compared to the other keys. These findings are largely consistent with results of earlier studies as well [2, 13].

The text entry metrics were recalculated, varying the speed setting as previously. The results can be found in Table 2. With different V_p values, the statistical tests generally showed consistent results regarding the significance of the differences between the two groups (Table 3), indicating that incorporating the offset targets into the simulator did not significantly impact or improve its performance.

4.4 Altering the Visual Scan Time Component

As already described, the motor system parameters were maintained as the original paper. However, the Visual Scan Time parameter was randomly selected from a Gaussian distribution with a mean of 50 ms and standard deviation at 0.2, which was assumed to be representative of an average to expert user [8]. Additionally to our primary analysis, we conducted a preliminary investigation into different settings for VST, aiming to identify the actual impact of the setting on the text entry rates. Although brief, the results of Table 4 show an expected, yet noteworthy, impact of the setting on WPM and Backspaces.

While the speed setting remained at 1.0, real participants had higher mean WPM (remaining significantly different) and lower mean Backspaces (with no significant difference) in all VST settings than synthetic participants. When the speed setting was adjusted to 1.1, once the mean of the Gaussian distribution for VST was reduced to 20ms and 10 ms,

	V_p = 1.0	V_p = 1.1	V_p = 1.2
WPM	T = **6.698**	T = **4.811**	T = **3.292**
	p <0.001	p <0.001	p = 0.002
KSPS	T = **5.84**	T = **3.387**	T = 1.282
	p <0.001	p = 0.0013	p = 0.205
KSPC	U = 337.0	U = **213.0**	U = **138.0**
	p = 0.197	p = 0.0013	p <0.001
IKI	U = **164.0**	T = **-2.416**	U = 379.0
	p <0.001	p = 0.019	p =0.524
CER (%)	U = 322.0	**U = 205.0**	U = **124.0**
	p = 0.128	p = 0.0008	p <0.001
Num. of Backspaces	U = 316.5	U = **195.0**	U = **118.0**
	p = 0.107	p <0.001	p <0.001

Table 3. Statistical test results across different V_p values with offset target.

	V_p = 1.0		V_p = 1.1	
VST	WPM	Backspaces	WPM	Backspaces
50 ms	$\bar{x}$ = 25.70	$\bar{x}$ = 0.39	$\bar{x}$ = 27.36	$\bar{x}$ = 0.56
	σ = 2.91	σ = 0.30	σ = 2.94	σ = 0.39
40 ms	$\bar{x}$ = 26.42	$\bar{x}$ = 0.39	$\bar{x}$ = 28.08	$\bar{x}$ = 0.58
	σ = 3.17	σ = 0.28	σ = 3.07	σ = 0.39
30 ms	$\bar{x}$ = 27.01	$\bar{x}$ = 0.38	$\bar{x}$ = 28.85	$\bar{x}$ = 0.59
	σ = 3.32	σ = 0.29	σ = 3.33	σ = 0.40
20 ms	$\bar{x}$ = 27.69	$\bar{x}$ = 0.39	$\bar{x}$ = 29.65	$\bar{x}$ = 0.58
	σ = 3.38	σ = 0.28	σ = 3.47	σ = 0.42
10 ms	$\bar{x}$ = 28.40	$\bar{x}$ = 0.37	$\bar{x}$ = 30.56	$\bar{x}$ = 0.57
	σ = 3.56	σ = 0.29	σ = 3.78	σ = 0.37

Table 4. WPM and Backspace metrics across different VST values.

the differences between the two groups in terms of WPM were found non-significant: 20ms: WPM: t(56) = 1.730, p = 0.089, Backspaces: U = 279.5, p = 0.029; 10ms: WPM: t(56) = 0.749, p = 0.457, Backspaces: U = 253.3, p = 0.0096.

We observe that by reducing the VST value, the WPM approaches and then falls withing the theoretical bounds set in [8, 12]. However, consistent with the results of the previous sections, this happens at the expense of accuracy. The differences between the average number of Backspaces become significant, as the average number increases. This also indicates that, when RoboType is instructed to type faster, it makes more errors – given that it is tested under the condition of immediate error correction. We intend to further explore these variations, along with variations in speed and target offset, in more detail, to determine their impact on the performance of RoboType.

5 Conclusion

In this paper, we evaluated the text entry performance of RoboType, an open-source simulator, in direct comparison to real human text entry behavior. Based on the results of the previous sections, we found that RoboType, with the current motor and cognitive parameters and speed setting, underestimated real participants' typing speed. We saw that, by adjusting the speeding setting, RoboType began to approach the text entry metrics of real participants, albeit still exhibiting significant differences and at the expense of accuracy, as the average error metrics increased and presented

significant differences from the corresponding values of real participants. This does not detract from the value or ability of the simulator to reproduce expected differences across various alternative input method designs, but highlights that the theoretical boundaries of performance discovered by the simulator, may not necessarily match actual human performance. These results could be attributed to inaccuracies in the participant profiling parameters, requirements for better modelling of the stochastic processes within the simulator, or modelling of performance variations that are dependent on user attitudes towards risk taking (i.e. increasing or decreasing typing speed dynamically during a trial, as part of the feedback loop with the external environment and the monitoring of success of operations). While our findings could facilitate further research in the development and refinement of RoboType to better approximate the behavior of actual participants, we recognize that our text entry data was limited to a single dataset, which may not represent the broader population. Future studies should also include a more diverse sample to enhance the generalizability of the findings.

References

[1] A. S. Arif and W. Stuerzlinger. 2009. Analysis of text entry performance metrics. In *2009 IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH)*. 100–105. https://doi.org/10.1109/TIC-STH.2009.5444533

[2] 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)*. Association for Computing Machinery, New York, NY, USA, 251–260. https://doi.org/10.1145/2371574.2371612

[3] Xiaojun Bi, Yang Li, and Shumin Zhai. 2013. FFitts law: modeling finger touch with fitts' law. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. Association for Computing Machinery, New York, NY, USA, 1363–1372. https://doi.org/10.1145/2470654.2466180

[4] Shi Cao, Anson Ho, and Jibo He. 2018. Modeling and Predicting Mobile Phone Touchscreen Transcription Typing Using an Integrated Cognitive Architecture. *International Journal of Human–Computer Interaction* 34, 6 (June 2018), 544–556. https://doi.org/10.1080/10447318.2017.1373463 Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/10447318.2017.1373463.

[5] Karen Church, Denzil Ferreira, Nikola Banovic, and Kent Lyons. 2015. Understanding the Challenges of Mobile Phone Usage Data. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '15)*. Association for Computing Machinery, New York, NY, USA, 504–514. https://doi.org/10.1145/2785830.2785891

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

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

[8] Andreas Komninos. 2023. RoboType: Realistic Mobile Text Entry Evaluations with Synthetic Users. In *Proceedings of the 25th International Conference on Mobile Human-Computer Interaction (MobileHCI '23 Companion)*. Association for Computing Machinery, New York, NY, USA, 1–7. https://doi.org/10.1145/3565066.3608693

[9] Andreas Komninos, Angeliki Tsiouma, Georgia Gogoulou, and John Garofalakis. 2023. Don't Look Up: The Cost of Attention to Stimulus Phrases in Mobile Text Entry Evaluations. In *Proceedings of the 26th Pan-Hellenic Conference on Informatics (PCI '22)*. Association for Computing Machinery, New York, NY, USA, 341–346. https://doi.org/10.1145/3575879.3576015

[10] Danqing Shi, Yujun Zhu, Jussi P. P. Jokinen, Aditya Acharya, Aini Putkonen, Shumin Zhai, and Antti Oulasvirta. 2024. CRTypist: Simulating Touchscreen Typing Behavior via Computational Rationality. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24)*. Association for Computing Machinery, New York, NY, USA, 1–17. https://doi.org/10.1145/3613904.3642918

[11] Jacob O. Wobbrock. 2007. Measures of text entry performance. *Text entry systems: Mobility, accessibility, universality* (2007), 47–74. Publisher: San Francisco: Morgan Kaufmann.

[12] Shumin Zhai, Michael Hunter, and Barton Smith. 2002. Performance Optimization of Virtual Keyboards. *Human-computer Interaction* 17 (Sept. 2002), 229–269. https://doi.org/10.1207/S15327051HCI172&3_4

[13] Suwen Zhu, Tianyao Luo, Xiaojun Bi, and Shumin Zhai. 2018. Typing on an Invisible Keyboard. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, Montreal QC Canada, 1–13. https://doi.org/10.1145/3173574.3174013