Towards Psycho-Cognitive Modelling of Mobile Text Entry Behaviour

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

We present a novel model for simulating human text entry behaviour, based on psycho-cognitive theory. The model marks a significant departure from statistical abstractive models in past research, providing higher explanatory power of human behaviour in planning, target finding and target selection during text entry. We describe the formulation and implementation of the model, and confirm its validity based on hypotheses derived from extant literature. We release the model as open-source for use by the community.

CCS Concepts: • **Human-centered computing → User models**; **Ubiquitous and mobile computing design and evaluation methods**.

Additional Key Words and Phrases: Mobile text entry, simulation, user models

ACM Reference Format:
Andreas Komninos, Ioulia Simou, and Ioannis Garofalakis. 2018. Towards Psycho-Cognitive Modelling of Mobile Text Entry Behaviour. In *Proceedings of Pan-hellenic Conference on Informatics (PCI'24) (Conference acronym 'XX)*. ACM, New York, NY, USA, 10 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

The ubiquity of mobile devices has made efficient text entry a critical area of research. As smartphones and tablets become central to daily life, the need for fast, accurate, and inclusive text input methods drives continuous advances in input techniques that can overcome the limitations of small screens and virtual keyboards. Computer models and simulations are essential for evaluating novel text entry methods, allowing researchers to test and refine new input techniques without the time and expense of full-scale user studies. By simulating user behaviour, researchers can quickly identify potential issues of their designs and optimize performance before moving to real-world testing.

Simulations are particularly valuable in understanding user behaviours during the learning phase of new text entry methods. Learning curves can be modeled to predict how quickly users adapt to a novel input technique and what challenges they may encounter, helping researchers design more intuitive interfaces and develop effective training strategies to accelerate user adoption. Furthermore, computer models can simulate long-term use scenarios, providing insights into how users might interact with a text entry method over extended periods. This is crucial for predicting fatigue, ergonomic issues, and potential repetitive strain injuries that short-term studies might miss. These insights help researchers focus their human studies on the most promising designs and problem areas, making the research process more efficient and effective. Most existing literature on modelling human behaviour in mobile text entry relies on the

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

formulation of statistical models based on the analysis of empirical data gathered from real users. While immensely effective in predicting outcomes, statistical models primarily capture observable patterns in user behaviour without necessarily explaining the underlying cognitive processes, which places significant limits on our understanding of why users make certain choices or errors. Consequently, they may indicate the need for design changes but do not offer insights on how to implement those changes effectively. Further, statistical models often lack generalisability, as models based on specific datasets may not accurately predict behaviour in novel situations or with different user populations.

In this paper, we present a novel model for simulating human text entry behaviours, based on psycho-cognitive theory, and therefore marking a significant departure from statistical abstractive models used in previous research. We describe the formulation and implementation of the model, and present preliminary insights into its use and capabilities. We release the model as open-source for use by the community.

2 Related Work

To overcome the challenges of design space exploration and the high costs of human-based evaluations, researchers have turned towards simulation using computational models of human behaviour during a task. Theoretical models of human behaviour during mobile text entry already surfaced at the time of physical mobile keyboards (e.g. [7]) and later expanded to touchscreens (e.g. [3, 17]), based on extensions of Fitts' Law and empirical derivation of model parameters from data obtained from human subjects. Recent attempts, aimed to improve models by incorporating aspects of human motor and cognitive processes (e.g. [5, 10]). Theoretical modelling has recently expanded into the effects of behaviour during error discovery (e.g. [2]) and use of autocompletions or word suggestions (e.g. [1, 20]). Spatial and language models (e.g. [22]) to improve input accuracy are a key element of all commercial virtual keyboards. However, unlike the use of touch data to construct models, other sources of data of human sensory, postural, cognitive or motor functions have not been fully explored. Eye movement, myoskeletal and brain activity data are only recently being considered as sources for understanding human text entry behaviour (e.g. [9, 15, 19]). Past literature has attempted to model typing tasks as a form of hierarchical control, using a queuing network (QN-ACTR, QN-MHP models) consistent of two main steps (loops); one that transforms text or thought into a series of words, and one that transforms each word into a series of keystrokes, specifying hand, finger, and direction [5]. A simpler agent-based model of hierarchical control that conflates cognitive and motor system details using Markov Decision Processes with reward functions is proposed in [11]. Machine learning using linear and non-linear Gaussian Processes has been demonstrated to more accurately simulate touch dynamics, including the known phenomenon of finger slippage, though not in a text entry context [4]. More recently, researchers have demonstrated that the use of GANs can produce reasonably good approximations of keystroke dynamics during text entry [6]. In [23], a bounded rationality model of text entry was proposed, incorporating the use of word suggestions as a valid course of action during input, as well as accounting for interruptions during text entry, and erroneous behaviours. This is probably the most complete user model to date, accounting for sub-optimal behaviours (i.e. continuing to type instead of selecting a word suggestion) as well as corrective actions for erroneous input, though the latter are limited to wrong key selection. Further, the model does not account for factors such as unfamiliarity with a keyboard layout and the related cognitive task of target finding prior to selection.

In summary, extant literature contains limited attempts to model human behaviour during mobile text entry, that have never been unified under a single theoretical framework which incorporates cognitive, visual and motor functions that are implicated during text entry. This leaves IME design reliant on empirical approaches, intuition, design skill and, perhaps sheer imagination, rather than grounded theory. As AI increasingly integrates into the human creative processes, including communication and text composition, it is crucial to develop a comprehensive model to understand

both the human process and its integration with AI-assistance. This will help create better IME interfaces and guarantee that human creativity during written expression will continue to be supported, not undermined, by AI systems.

3 Modelling user behaviour in text entry based on psycho-cognitive theory

Our work on modelling user behaviour in text entry is inspired by Miller's TOTE (Test-Operate-Test-Exit) model theory [16], and Norman's Action Cycle (AC) model [18]. The TOTE model is a cognitive framework that explains how individuals execute tasks. It suggests that a goal is established, then tested against the current state. If there's a discrepancy, an operation is performed to reduce it and repeated until the goal is met (test-operate-test), leading to exit. The TOTE model allows for the nesting of TOTE units, as is required in order to achieve a higher-level goal by breaking it down into sub-tasks. Although the model does not discuss the formulation of goals and their conversion into plans, it provides a basic understanding of goal-directed behaviour. The Action Cycle expands this by introducing the concepts of intention, execution, and evaluation. Intention involves forming a goal, which sets the direction for subsequent actions. Execution entails planning and carrying out the actions necessary to achieve the goal. This involves selecting appropriate actions, coordinating movements, and monitoring the progress of the task. Finally, evaluation assesses the outcome of the actions against the original intention. This evaluation stage is crucial for learning and adaptation, as it allows individuals to identify discrepancies between their goals and outcomes, leading to adjustments in future behaviour. The Action Cycle highlights the iterative nature of human behaviour, emphasizing the constant interplay between goal setting, action execution, and outcome evaluation. This cyclical process is not always linear, as individuals may encounter unexpected challenges that require adjustments to their plans. Furthermore, the evaluation stage may reveal that the initial goal was not entirely appropriate or achievable, leading to a revision of the intention.

We focus on modelling of user behaviour during transcription tasks in a laboratory environment [21], where a user is asked to copy a stimulus phrase as quickly and as accurately as possible. Despite concerns about their ecological validity, transcription tasks are the standard for evaluating text entry methods and serve as a solid foundation for our model. Additionally, we limit our model to single-digit input (index finger only).

3.1 Psycho-cognitive Modelling of execution using the TOTE model

In typical text entry, the high-level goal is to enter a chunk of text, usually a phrase. Longer texts may be composed of multiple phrases but we concern ourselves here to the single phrase alone, as this is a typical task asked of participants during text entry evaluations. This goal can be broken down into the subgoal of entering the tokens (words) that make up the phrase, and each token can be broken down into the subgoal of entering the single letters that comprise it. At the phrase level, the user Tests the system state (is the phrase completed?), then Operates a sequence of token entry attempts, then Tests the system (has the phrase been completed?), and then Exits. At the token level, the user Tests the system state (is the token completed?), Operates a sequence of keystrokes, then Tests the system again before Exiting (moving on to input the next token). At the keystroke level, the user Tests whether the desired letter has already been input. While the answer here is typically no, the stage is still essential as the desired letter might have been accidentally pressed during a previous operation and therefore might be present in the system's output. Then, a user will Operate, i.e. attempt to hit the desired key. The outcome of this operation is Tested (was the desired key actually pressed and present in the system output?). If yes, then the user Exits the unit. Alternatives might be having pressed the wrong key or having pressed on an inactive area of the keyboard. Interleaved in this sequence of keystroke subgoals, we interject another possibility - checking whether the desired token can be completed faster by selecting a word suggestion, instead of completing the keystroke sequence. Therefore, as the user types, they should check the available

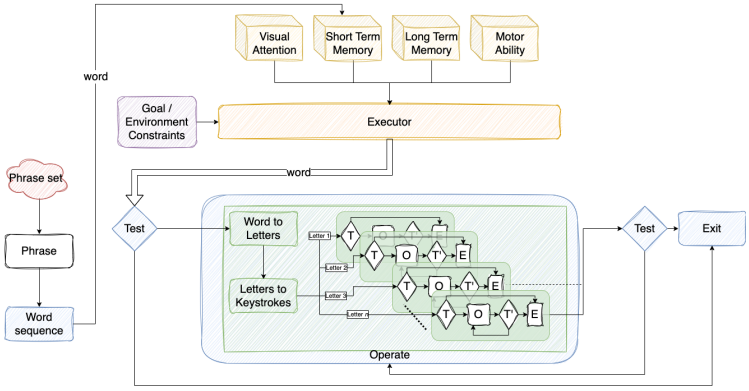


Fig. 1. Conceptual diagram of our model for text entry behaviour.

suggestions before proceeding to the next keystroke. The conceptual diagram of our model is presented in Fig. 1, where the various interacting components and nested TOTE modules are shown. In this concept, the short-term memory holds the formulated plan for the current token. The long-term memory is used to stochastically model users' familiarity with the keyboard. An Executor module coordinates information exchange between modules and oculomotor commands (visual attention, and motor ability, which hold state information on the eye and finger positions, and stochastically model their movement). Keyboard layout and state is modelled in the Goal/Environment Constraints module.

3.2 Psycho-cognitive Modelling of goal formulation using the AC model

Achieving the high-level goal requires formulating a plan, which may need adjustments as the system state changes with each keystroke TOTE unit completion. At the highest level, the plan represents the sequence of tokens to be input. This plan may change during natural text entry as the user reconsiders their choice of words while typing. In transcription tasks, however, the plan should remain immutable, as the goal is set from the start and users must adhere precisely to copying the presented phrase. At the token level, a sub-plan must be formulated at the start, consisting of the sequence of characters that comprise the token (including any token terminators or token separators). This sub-plan is much more susceptible to revision, since errors during the input process, or the availability of desired word suggestions, may drive the user to take alternative actions. This process can be modeled by the Action Theory as an iterative and continuous adaptations of user goals in the Gulf of Evaluation (assessing system state).

Consider the plan for inputting the token *hello*, formulated as the sequence of characters ['h', 'e', 'l', 'l', 'o'], where each character is a step in the plan. More generally, the plan steps are not necessarily characters, but targets that represent interactive areas of the keyboard, including letter keys, modifier keys, output modification keys such as backspace, and suggestion keys. As the user types, they might make a mistake, so for example instead of hitting the 'e' target at step 1 of the original plan, they hit the proximal letter 'r'. In this case, the plan needs to be modified, therefore we interject a sequence of corrective actions (i.e. pressing a backspace key). Thus the plan becomes ['h', 'DEL', 'e', 'l', 'l', 'o']. If the user makes another mistake and instead of pressing the 'DEL' key, they press a proximal character (e.g. 'm'), the plan becomes ['h', 'DEL', 'DEL', 'e', 'l', 'l', 'o']. If the user presses an ineffective area of the keyboard (e.g. the spacing between keys, or some key whose functionality is not implemented in the simulation), the plan should remain the same, but execution must not proceed until the current step is successfully completed. Another possibility is that the complete

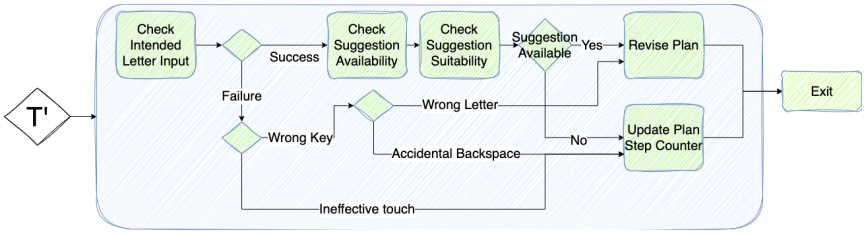


Fig. 2. Plan reformulation in the second T of TOTE units.

token becomes available as a suggestion, after a few characters have been input. Suppose the user is at step 4 of the last plan, and the keyboard's suggestion engine provides the tokens ["help", "held", "hello"] as options to the user. Then the plan might become ['h', 'DEL', 'DEL', 'e', 'l', 'SUG2', 'l', 'o'], meaning that the user's intention is now to select the last suggestion key in order to complete the token. Note that our implementation does not discard the plan history, nor does it truncate the subsequent plan steps once a suggestion becomes available, enabling inspection of plan revisions and potential reversion to the original plan (e.g. if the cost of completing a word is only one character, i.e. the same as selecting a suggestion, or to model user attitudes towards using suggestions). Finally we should clarify that to execute the plan, each target on the keyboard is treated the same, i.e. a TOTE unit is executed regardless of the target's nature (letter, suggestion or other key). This process is shown conceptually in as the second T in TOTE modules (T') in Fig. 2.

3.3 Modelling visual perception and finger kinematics

Goal formulation and operation execution depend on the user's ability to visually inspect the system and take appropriate actions. During the Operation phase in the TOTE unit, the user must first visually locate the target on the keyboard, and then attempt to hit it. This phase consists of two distinct TOTE units - one for locating the target visually, and one for attempting to hit it with the finger. For individual character or the backspace key, we model the search process stochastically, where the user moves their gaze to a random location sampled from a double Gaussian distribution around the target's center coordinates, where one standard deviation in the distribution equates to the target's key horizontal and vertical size. We limit the sampling to n standard deviations, in order to model familiarity of a user with the current keyboard layout. When $n = 0$, the user's mental model of the keyboard's key locations is perfect - the user will always direct their gaze to the correct location (the center of the target). As n increases, so does the probability of moving the gaze to an inappropriate location (i.e. searching randomly around the keyboard until the desired target is located, while remaining within the bounds of the keyboard area). When the gaze is moved to some coordinates according to this approach, the user checks the keys at that location. To model the user's foveal vision (i.e. visual clarity) we assume that the user can simultaneously read the labels of a number of targets within a parametrisable range from these coordinates, therefore all the keys within that parametrisable *proximity* to the gaze coordinates are returned as found by the user. If the desired target is within this set, then we consider the search operation successful (and exit).

For suggestion key targets, we follow a slightly modified approach. Since users typically begin checking word suggestions after a few characters are entered, as the chance of finding the desired token increases with input length, we place a minimum character limit entered manually by the user, before they begin checking on the suggestion list after entering a subsequent character. Then, we proportionally increase the likelihood that the suggestion bar will be checked by increasing the probability of the user's attention to execute a suggestion TOTE unit according to the number of characters input, up to a maximum of 1, after a number of letters have been input. Since the user can be

assumed to always know where the suggestion keys are (but not their labels), to examine if the desired token is present in the suggestion list, the user first moves their gaze to the left-most suggestion key center coordinates and continues to the next one, until they either find the desired token or they don't, in which case they may continue to the next character that's part of the plan. Our suggestion engine is built using a frequency dictionary of English words. It does not take previous input tokens as part of the context, considering only characters already input as part of the current token, and provides suggestions ordered by their frequency score. The engine is written as a separate module so it could be easily replaced by a more advanced one without other changes to the simulator code.

For finger kinematics, we leverage the open-source model from [12]. The model is based on a stochastic approach to modelling kinematics, based on velocity profiles and accuracy models derived from empirical data. It has been shown to accurately reflect human performance in a variety of text entry contexts, including unfamiliar and optimised keyboard layouts. Its release includes several pre-parametrised user profiles, therefore we employ these for our evaluation.

4 Simulation results

To demonstrate the functionality of our model, we conduct a comparative simulation with and without word suggestions, and across varying expertise levels (Expert, Familiar and Novice). We measure the participants typing speed (Words per Minute) as well as input efficiency in terms of KSPC (Keystrokes Per Character). The simulator uses a 1400x1000px keyboard, with a standard (letter) $keywidth$ of 140px. To verify the user behavioural model, we make the following hypotheses based on the reasonable expectations from our implementation. H1 - Typing efficiency (KSPC) will be higher and typing speed (WPM) will be lower when word suggestions are used (as per [1]). H2 - Gaze distribution across the keyboard will be more uniform when the user's expertise (familiarity with the keyboard) is lower [8]. H3 - As a direct consequence of H2, expert users will achieve higher typing speeds [8]. To explore these hypotheses, we ran a synthetic transcription task experiment, using all 30 virtual participants and their respective finger kinematics models in [12], and randomly selecting 40 phrases from the Enron memorable phrase set. Each virtual participant typed the same phrases but we modified their expertise levels by parametrising the visual perception model as described previously (Expert:$n = 0.1$, Familiar:$n = 1.0$ and Novice:$n = 2.0$). Statistical tests are reported after choosing parametric or non-parametric tests, dependent on the examination of the necessary assumptions.

As shown in Fig. 3 (top), typing speeds are impacted by both user expertise, and the availability of suggestions. For each level of expertise, we compared the typing speed depending on the availability of suggestions. Expert and Familiar users were faster when suggestions were not available (E: $\bar{x} = 18.96WPM, \sigma = 2.39$; F: $\bar{x} = 17.49WPM, \sigma = 2.176$) than when suggestions were being used (E: $\bar{x} = 17.407WPM, \sigma = 2.6$; F: $\bar{x} = 16.488WPM, \sigma = 2.436$), and the differences are statistically significant (E: Wilcoxon $Z = -4.762, p < 0.001$; F: Wilcoxon $Z = -4.720, p < 0.001$). Novice users, however, typed faster with the presence of suggestions (N: $\bar{x} = 13.116WPM, \sigma = 1.63$) than without (N: $\bar{x} = 12.854WPM, \sigma = 1.369$). The difference is again statistically significant (N: Wilcoxon $Z = -2.417, p < 0.05$). Within each condition of suggestion availability the differences across user expertise levels are statistically significant. Where suggestions were not available, a Friedman test indicates statistically significant differences ($\chi^2 = 60.0, p < 0.001$). Post-hoc pairwise tests with Bonferroni correction show the differences are also significant between Expert and Familiar (Wilcoxon $Z = -4.782, p < 0.001$), Expert and Novice (Wilcoxon $Z = -4.782, p < 0.001$) and Familiar and Novice (Wilcoxon $Z = -4.782, p < 0.001$). Where suggestions were available, a Friedman test indicates statistically significant differences ($\chi^2 = 56.267, p < 0.001$). Post-hoc pairwise tests with Bonferroni correction show the differences are also significant between Expert and Familiar (Wilcoxon $Z = -4.572, p < 0.001$), Expert and Novice (Wilcoxon $Z = -4.782, p < 0.001$) and Familiar and Novice (Wilcoxon $Z = -4.782, p < 0.001$).

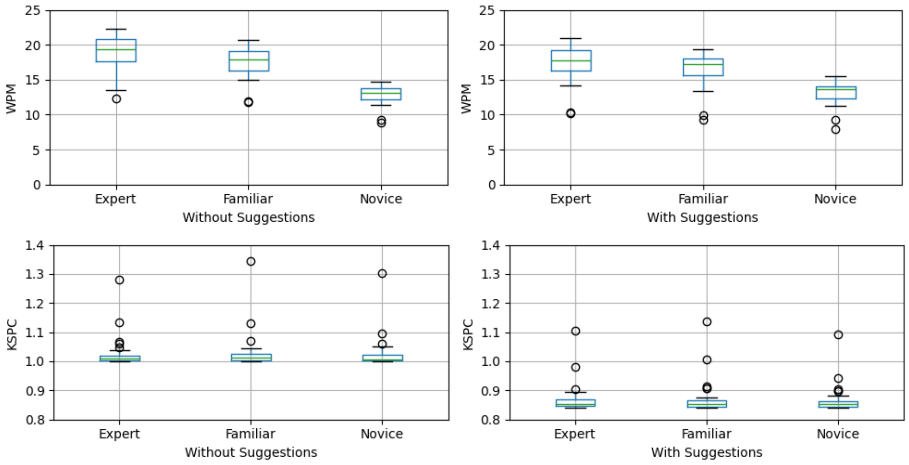


Fig. 3. Typing speed in WPM (top) and efficiency in KSPC (bottom).

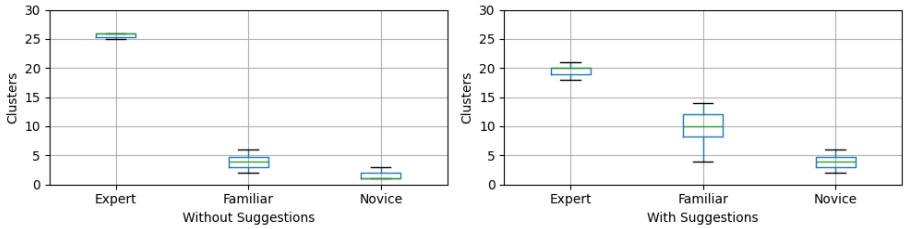
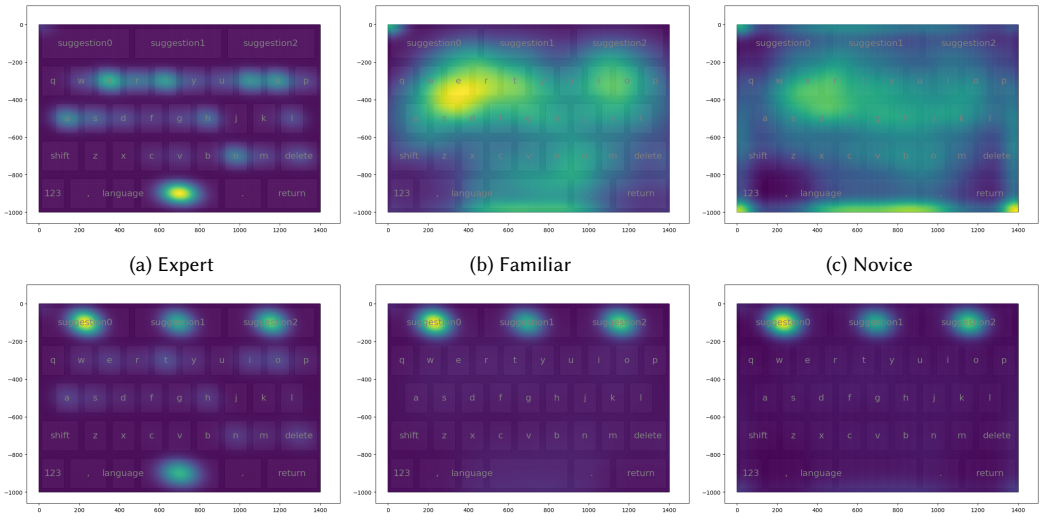


Fig. 4. Number of gaze clusters

In Fig. 3 (bottom), we note that input efficiency is mainly affected by the availability of suggestions. For each level of expertise, we compared the typing efficiency depending on the availability of suggestions. All levels of expertise were less efficient when suggestions were not available (E: $\bar{x} = 1.028 KSPC, \sigma = 0.055$; F: $\bar{x} = 1.03 KSPC, \sigma = 0.065$, N: $\bar{x} = 1.025 KSPC, \sigma = 0.057$) compared to when suggestions were being used (E: $\bar{x} = 0.871 KSPC, \sigma = 0.053$; F: $\bar{x} = 0.873 KSPC, \sigma = 0.06$, N: $\bar{x} = 0.869 KSPC, \sigma = 0.048$). The differences are statistically significant (E: Wilcoxon $Z = -4.782, p < 0.001$; F: Wilcoxon $Z = -4.782, p < 0.001$; N: Wilcoxon $Z = -4.782, p < 0.001$). Within each condition of suggestion availability, there are no statistically significant differences across user expertise levels (with suggestions: Friedman $\chi^2 = 2.737, p = 0.255$; without suggestions: Friedman $\chi^2 = 5.357, p = 0.069$). These results seem to partially confirm H1, since we were not able to reproduce the effects found in [1] for Novice users. However, in that research, users were already familiar with the keyboard layout, therefore we are not able to directly compare Novice users. We also confirm H3, since user expertise clearly results in higher text entry speeds, due to much shorter visual search times.

For H2, as shown in Fig. 5, we note that eye gaze distribution is more uniformly distributed with lesser expertise, in both cases. To verify, we ran the DBSCAN clustering algorithm to determine the number of distinct gaze clusters that can be identified in each condition, setting $e = keywidth/2, min_{pts} = 20$ for all cases (see Fig. 4). For the condition without suggestions, we note that indeed the number of clusters increases with participant expertise (E: $\bar{x} = 25.733, \sigma = 0.45$; F: $\bar{x} = 3.733, \sigma = 1.311$, N: $\bar{x} = 1.467, \sigma = 0.681$). A Friedman test confirms the differences are statistically significant



(a) Expert (b) Familiar (c) Novice

(d) Expert (e) Familiar (f) Novice

Fig. 5. Eye gaze distributions without suggestions (top) and with suggestions (bottom)

($\chi^2 = 58.207, p < 0.001$). The differences are also statistically significant in all post-hoc pairwise test with Bonferroni correction: Expert and Familiar (Wilcoxon $Z = -4.811, p < 0.001$), Expert and Novice (Wilcoxon $Z = -4.883, p < 0.001$) and Familiar and Novice (Wilcoxon $Z = -4.485, p < 0.001$). In the case where suggestions were available, we observe again that the number of clusters increases with participant expertise. An ANOVA test confirms the differences are statistically significant ($F = 692.263, p < 0.001$). The differences are also statistically significant in all post-hoc pairwise test with Bonferroni correction: Expert and Familiar (t-test $t = 18.557, p < 0.001$), Expert and Novice (t-test $t = 50.958, p < 0.001$) and Familiar and Novice (t-test $t = 11.697, p < 0.001$). Thus, H2 is supported.

5 Discussion

We presented a novel model to simulate human behaviour during text entry. The model behaves according to expectations from empirical data, as found in extant literature. However, there are limitations to address in future work. Gaze and visual attention should be modelled more accurately with a full kinematic model. Visual attention outside the keyboard area to check for input is not currently considered, though it is a real and impactful effect [14]. We assume users immediately notice mistakes made, which is a simplification of actual strategies [13]. We do not model autocorrects and resultant user revising actions. Therefore, improvements to the planning process to account for missed errors and text revision should be made. Further improvements may consider user typing speed adjustment after mistakes have been made [2]. Additionally, we do not model two-thumb entry or glide input, which is common amongst users. Despite these limitations, we have demonstrated the TOTE and AC models' suitability to capture human behaviour in text entry, and contribute significantly towards a complete model able to offer high explanatory power when evaluating text entry methods with computational methods. Our model is openly available at [ANON URL] for use by the community.

References

[1] Ohoud Alharbi, Wolfgang Stuerzlinger, and Felix Putze. 2020. The Effects of Predictive Features of Mobile Keyboards on Text Entry Speed and Errors. *Proceedings of the ACM on Human-Computer Interaction* 4, ISS (Nov. 2020), 183:1–183:16. https://doi.org/10.1145/3427311

[2] 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)*. ACM, New York, NY, USA, 4229–4241. https://doi.org/10.1145/3025453.3025695

[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] Daniel Buschek and Florian Alt. 2015. TouchML: A Machine Learning Toolkit for Modelling Spatial Touch Targeting Behaviour. In *Proceedings of the 20th International Conference on Intelligent User Interfaces (IUI '15)*. Association for Computing Machinery, New York, NY, USA, 110–114. https://doi.org/10.1145/2678025.2701381

[5] 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

[6] Jeremy Chu, Yan Ma, Shumin Zhai, Xianfeng David Gu, and Xiaojun Bi. 2023. TouchType-GAN: Modeling Touch Typing with Generative Adversarial Network. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3586183.3606760

[7] Edward Clarkson, Kent Lyons, James Clawson, and Thad Starner. 2007. Revisiting and Validating a Model of Two-Thumb Text Entry. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. Association for Computing Machinery, New York, NY, USA, 163–166. https://doi.org/10.1145/1240624.1240650

[8] Xinhui Jiang, Jussi P. P. Jokinen, Antti Oulasvirta, and Xiangshi Ren. 2022. Learning to Type with Mobile Keyboards: Findings with a Randomized Keyboard. *Computers in Human Behavior* 126 (Jan. 2022), 106992. https://doi.org/10.1016/j.chb.2021.106992

[9] Xinhui Jiang, Yang Li, Jussi P P Jokinen, Viet 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*. ACM, Honolulu, USA, 14. https://doi.org/10.1145/3313831.3376711

[10] Jussi Jokinen. 2017. Touch Screen Text Entry as Cognitively Bounded Rationality. In *CogSci 2017 Proceedings*. COGNITIVE SCIENCE SOCIETY. https://research.aalto.fi/en/publications/touch-screen-text-entry-as-cognitively-bounded-rationality

[11] 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*. ACM, Yokohama Japan, 1–14. https://doi.org/10.1145/3411764.3445483

[12] 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

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

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

[15] Young D. Kwon, Kirill A. Shatilov, Lik-Hang Lee, Serkan Kumyol, Kit-Yung Lam, Yui-Pan Yau, and Pan Hui. 2020. MyoKey: Surface Electromyography and Inertial Motion Sensing-based Text Entry in AR. In *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. 1–4. https://doi.org/10.1109/PerComWorkshops48775.2020.9156084

[16] George A. Miller, Galanter Eugene, and Karl H. Pribram. 1968. Plans and the Structure of Behaviour. In *Systems Research for Behavioral Science*. Routledge.

[17] Josip Musić and Roderick Murray-Smith. 2016. Nomadic Input on Mobile Devices: The Influence of Touch Input Technique and Walking Speed on Performance and Offset Modeling. *Human–Computer Interaction* 31, 5 (Sept. 2016), 420–471. https://doi.org/10.1080/07370024.2015.1071195

[18] Donald A. Norman. 1990. *The Design of Everyday Things*. Doubleday.

[19] Felix Putze, Tilman Ihrig, Tanja Schultz, and Wolfgang Stuerzlinger. 2020. Platform for Studying Self-Repairing Auto-Corrections in Mobile Text Entry Based on Brain Activity, Gaze, and Context. (2020), 13.

[20] Philip Quinn and Shumin Zhai. 2016. A Cost-Benefit Study of Text Entry Suggestion Interaction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. Association for Computing Machinery, New York, NY, USA, 83–88. https://doi.org/10.1145/2858036.2858305

[21] 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 and Services (MobileHCI '11)*. Association for Computing Machinery, Stockholm, Sweden, 295–298. https://doi.org/10.1145/2037373.2037418

[22] 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 33rd Annual ACM Conference on Human Factors in Computing*

Systems (CHI '15). Association for Computing Machinery, New York, NY, USA, 659–668. https://doi.org/10.1145/2702123.2702135

[23] Boyin Yang and Per Ola Kristensson. 2023. Imperfect Surrogate Users: Understanding Performance Implications of Augmentative and Alternative Communication Systems through Bounded Rationality, Human Error, and Interruption Modeling. *Proc. ACM Hum.-Comput. Interact.* 7, MHCI (Sept. 2023), 213:1–213:33. https://doi.org/10.1145/3604260

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009