Context-Sensitive App Prediction on the Suggestion Bar of a Mobile Keyboard

Laxmi Pandey
Human-Computer Interaction Group
University of California, Merced
Merced, California
lpandey@ucmerced.edu

Ahmed Sabbir Arif
Human-Computer Interaction Group
University of California, Merced
Merced, California
asarif@ucmerced.edu

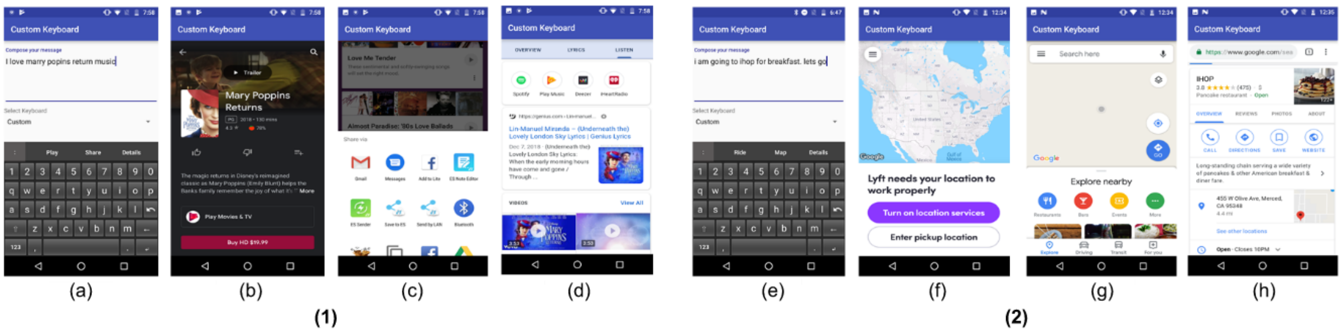

Figure 1: The proposed real-time app prediction method. With this method, the user types as she usually would. (1) As soon as the "music" keyword is detected (a) the suggestion bar displays options for the detected music: (b) tapping on "Play" displays a music player, (c) "Share" directly shares the music details, and (d) "Details" displays the details about the identified music. (2) When the "go" keyword is detected: (a) the suggestion bar displays options for the detected location/destination, (b) tapping on "Ride" launches a ridesharing app, (b) "Map" displays the map of the destination/location, and "Details" displays details about the particular destination/location.

ABSTRACT

This work augments context-sensitive app prediction feature to the suggestion bar of a mobile virtual keyboard to accommodate fast and easy information acquisition and sharing in textual conversations. The purpose is to eliminate the need for switching between apps while typing. A user study revealed that the proposed method improves performance both in terms of speed and effort for common tasks, such as playing a song or finding and sharing the address of a restaurant. Post-study questionnaire revealed that all participants found the method fast, easy, and likely to facilitate more engaging and meaningful textual conversations. All wanted to keep using it on their mobile devices.

CCS CONCEPTS

• **Human-centered computing → Text input**; *User studies*; *User interface design.*

KEYWORDS

Apps, smartphones, text entry, prediction bar, suggestion bar.

ACM Reference Format:
Laxmi Pandey and Ahmed Sabbir Arif. 2019. Context-Sensitive App Prediction on the Suggestion Bar of a Mobile Keyboard. In *MUM 2019: 18th International Conference on Mobile and Ubiquitous Multimedia (MUM 2019), November 26–29, 2019, Pisa, Italy*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3365610.3368414

1 INTRODUCTION

Mobile devices have become a part and parcel of today's modern world. Humans are using mobile devices to all extents, some for daily activities and some for specific purposes [11, 19, 26]. A large part of these activities involves text entry [18]. Predictive text is one of the recent advances that has significantly improved text entry on mobile devices [10, 16]. Predictive text facilitates typing on mobile devices by suggesting words and phrases in a suggestion bar that users may intend to enter [4, 23, 27]. Hence, text entry using the suggestion bar has penetrated the daily communication practices of many modern mobile users.

Nowadays, almost all virtual keyboards come with suggestion bars that suggest words and phrases based on complex language models. These suggestion bars do not provide the support for apps although users tend to use various apps, including maps, media player, rideshare, camera, and social media, to acquire information for sharing in textual conversations. Currently, this process involves

MUM 2019, November 26–29, 2019, Pisa, Italy

ACM ISBN 978-1-4503-7624-2/19/11.
https://doi.org/10.1145/3365610.3368414

leaving the input environment to locate, launch, and use these apps, cut/copy the information, switch back to the input environment, and paste the acquired information. This process is not only time-consuming and difficult but also distracts user attention from the current task [2].

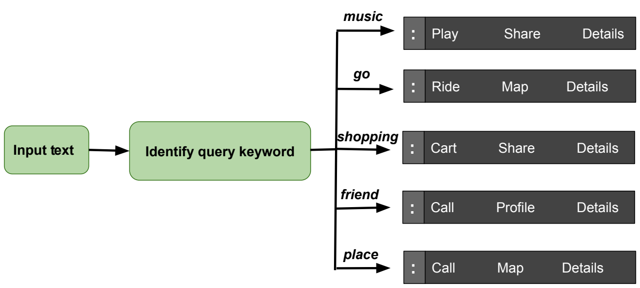

Figure 2: Architecture of the proposed framework.

To address this, here we propose a predictive system that can predict different apps and display Web search results based on the contextual information of an ongoing conversation. The proposed system uses keyword-based querying to suggest relevant apps as users enter text on mobile devices (Figure 2). For queries, it uses an ontological approach to identify keywords in text as users type to suggest related apps and Web search results. It is important to note that the proposed system does not replace the existing word prediction but extends its support to apps.

2 RELATED WORK

Recent innovations in mobile text entry include gesture keyboards [1, 20, 29], key-target re-sizing [14, 24], alternative layouts [7, 9, 22], and sensor-based adaptation [5, 15]. Most mobile keyboards include a suggestion bar that facilitates word completion and correction. It also enables users to actively select an intended word from a list of the most probable next words to improve text entry speed and accuracy [13]. Few studies have investigated suggestions of new content. For example, a case-based reasoning system makes suggestions in the form of short phrases that are mined from product reviews [8]. Although suggestion bars play a central role in virtual keyboard design, it has received a limited attention in academic research [6]. We introduce a simple extension to the familiar mobile keyboard suggestion interface that presents app suggestions based on the context of an ongoing conversation.

To the best of our knowledge, no research has investigated app prediction based on the context of an ongoing conversation. Some have used deep reinforcement learning [25] and other methods to predict the apps that will be opened next to prefetch dynamic data for those apps to reduce latency [21], recommended the most interesting and relevant apps using context-aware collaborative filtering algorithm [17], and predicted the most relevant apps for a given location using a transfer learning technique [28]. This line of research, however, is outside the scope of this work.

The most relevant work to our system is a commercial solution. Google has recently added a feature to the Gboard keyboard [12] that enables users to access options for emojis, GIFs, translation, and keyboard themes on the suggestion bar by tapping on the

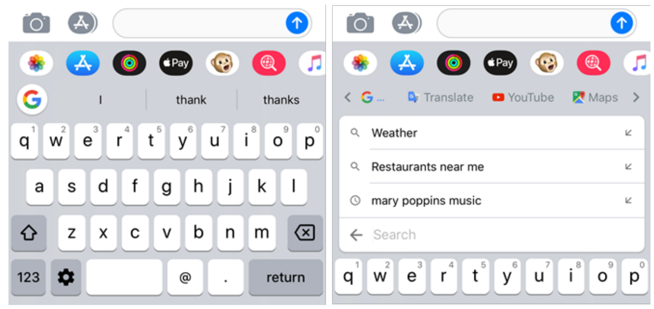

Figure 3: The Google Keyboard's (Gboard) information acquisition and sharing method: (left) tapping on the "G" icon displays a Google search area, which (right) enables the user to search and share information while typing.

"G" icon (Figure 3, left). Tapping on the arrow icons ("<" and ">") displays additional options (Figure 3, right). This enables users to perform Google search directly on the suggestion bar and share results in text messages. Users could also search and share contact information after enabling the contacts search option from the settings. However, this approach is time-consuming as it requires users to actively select from a list of options. It also compromises user attention by forcing them to alternate their focus between the Google search area and the text input area [2].

3 PROTOTYPE

We implemented a standalone app prediction feature for the suggestion bar that works in real-time and does not require users to switch to different apps while chatting for tasks such as searching and sharing information. It comprises of automatic detection of specific keywords for app prediction and Internet search. It uses an ontological approach to identify keywords as the user types. It includes an editing environment, a keyboard, and a suggestion bar that includes both word and app prediction.

Currently, the system only supports options for multimedia, shopping, destination or location, and phone calls (Figure 1). However, support for other scenarios can be easily added, for example news and social media. Users could disable and (re)enable the app prediction feature by tapping on the ":" icon on the suggestion bar (Figure 1).

3.1 Example Scenarios

Below are some example scenarios.

- User types *"Did you book an appointment with Supercuts?"*, suggestion bar displays "Call", "Map", and "Details".
- User types *"I love Marry Poppins Returns soundtrack"*, suggestion bar displays "Play", "Share", and "Details".
- User types *"I do want to buy an iPad"*, suggestion bar displays "Cart", "Share", and "Details".
- User types *"I'm going to IHOP for breakfast"*, suggestion bar displays "Ride", "Map", and "Details".
- User types *"I'm meeting Jane tomorrow"*, suggestion bar displays "Call", "Profile", and "Details".

4 PILOT STUDY

We conducted a pilot study to investigate how users acquire and share information when entering text on mobile devices.

4.1 Apparatus

Participants used their own smartphones during the pilot. These devices were connected to a reliable Wi-Fi network to reduce the chance of data loss during Internet search.

4.2 Participants

Six volunteers from the local university, aged 22-26 years, participated in the pilot. Two of them were female and four were male. They all had at least 4 years of experience with smartphones. All were native or fluent speakers of the English language. All of them were right-handed (Figure 4).

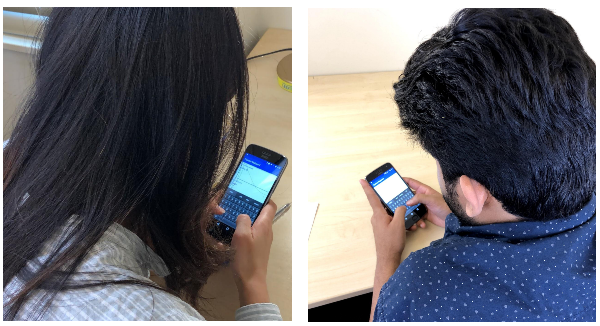

Figure 4: Two participants volunteering in the study.

4.3 Metrics

We recorded the following performance metrics in the study.

- Actions per Task (APT) signifies the average number of taps and gestures performed per task, which is comparable to the Keystrokes per Character (KSPC) metric [3].
- Time per Task (TPT) represents the average time (in seconds) users took to perform a task.

4.4 Design and Procedure

In the pilot, each participant performed 15 tasks involving three different categories: action (play, ride, call, etc.), share, and lookup. Each category included 5 tasks (5×3=15 tasks). Some examples of these tasks are: *"Play Mary Poppins Returns soundtrack"*, *"Share iPad Pro specifications with your friend"*, and *"Search location of a nearby IHOP"*. To eliminate a potential confounding variable, we asked all participants to perform the same tasks. Participants were instructed to use their own smartphone, hold it as they usually would, and use the method they typically use to perform the tasks (to acquire and share information).

4.5 Results

All participants chose to hold their devices in portrait position. They all used mobile apps to perform the action tasks, but mobile browsers to perform the lookup tasks. For the share tasks, one user took screenshots, two used the cut/copy-paste approach, while the remaining three used the secondary apps' "Share" option.

4.5.1 Time per Task (TPT). The average TPT was 165.35 seconds (SD = 26.95). Participants took on average 135.81, 188.61, and 171.63 seconds to perform the action, share, and lookup tasks, respectively. An ANOVA identified a significant effect of task type on TPT ($F_{2,5}$ = 637.13, p < .0001). A Tukey-Kramer test revealed that the action tasks were significantly faster, while the share tasks were significantly slower compared to the other tasks. See Figure 5 (a, b).

4.5.2 Actions per Task (APT). The average APT was 20.06 (SD = 3.25). Participants performed on average 16.8, 23.3, and 20.1 taps and gestures to perform the action, share, and lookup tasks, respectively. An ANOVA identified a significant effect of task type on APT ($F_{2,5}$ = 8.32, p < .01). A Tukey-Kramer test revealed that the action tasks took significantly fewer actions (taps and gestures), while the share tasks took significantly more actions than the other tasks. See Figure 5 (c, d).

5 FINAL STUDY

We conducted a study to evaluate the performance, preference, and learnability of the proposed method.

5.1 Apparatus

We used a Motorola Moto G^5 Plus smartphone (155 g, 150.2×74×7.7 mm) at 1080×1920 pixels. The custom app recorded all interactions with timestamps. The device was connected to a reliable Wi-Fi network to reduce the chance of data loss during Internet search.

5.2 Participants

Twelve new volunteers from the local university, aged from 24 to 28 years (M = 25.3, SD = 1.5), participated in the study. Two of them were female and ten were male. They all had at least 4 years of experience with smartphones. All participants were native or fluent speakers of the English language. All of them were right-handed. Like the pilot, they all chose to hold the device in portrait position (Figure 4).

5.3 Design and Procedure

We used the same design, tasks, and metrics as the pilot study. However, participants were instructed to use the suggestion bar to complete these tasks. The study was held in a quiet office room. Upon arrival, we explained the research to all participants and collected their consents. They then completed a demographics and mobile usage questionnaire. We demonstrated the proposed system and enabled all participants to practice with it for about one minute. However, they could extend this practice period on request. After they all were familiar with the design, we asked them to enter a predetermined set of phrases that we knew would trigger app prediction. Once predictions were displayed, they were asked to perform explicit tasks associated with the entered phrase (same tasks as the pilot study). These tasks were provided in a sheet of paper. Timing started from the first touch-down and ended with the last touch-up.

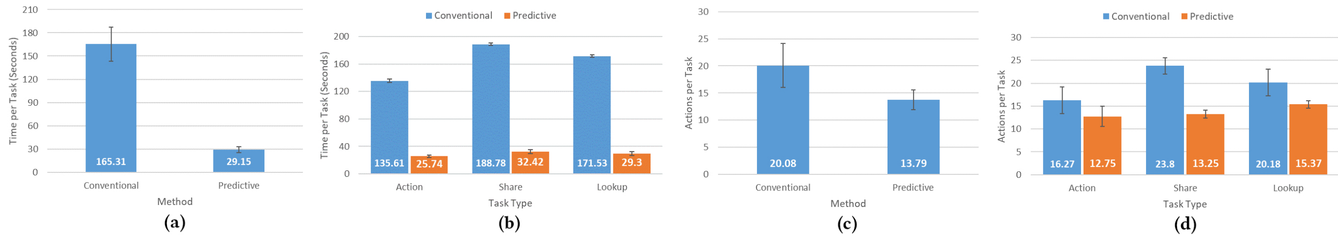

Figure 5: Results of the two studies: average Time per Task (TPT) for the (a) two methods and (b) three different types of tasks; and average Action per Task (APT) for the (c) two methods and the (d) three different types of tasks. Error bars represent ±1 standard deviation (SD).

5.4 Results

A complete session took about 30 minutes, including demonstration, practice, and breaks. We used the results of the pilot study as the baseline condition (the conventional method), therefore used a between-subjects ANOVA to compare the two methods. However, we used a repeated-matures ANOVA to compare the types of task, like in the pilot study.

5.4.1 Time per Task (TPT). An ANOVA identified a significant effect of method on TPT ($F_{1,16} = 29791.46$, $p < .0001$). An ANOVA also found a significant effect of task type on TPT ($F_{2,11} = 22.7$, $p < .0001$). A Tukey-Kramer test revealed that the action tasks were performed significantly faster and the share tasks were performed significantly slower than the other tasks. See Figure 5 (a, b).

5.4.2 Actions per Task (APT). An ANOVA identified a significant effect of method on APT ($F_{1,16} = 156.13$, $p < .0001$). An ANOVA also identified a significant effect of task type on APT ($F_{2,5} = 8.32$, $p < .001$). A Tukey-Kramer test revealed that the lookup tasks took significantly more taps and gestures than the other tasks. See Figure 5 (c, d).

6 DISCUSSION

Results revealed that the proposed method improved performance both in terms of time and effort. Participants performed the tasks significantly faster and with significantly fewer actions than the conventional method. Unsurprisingly, different types of tasks were significantly different in terms of time and effort with both methods. Share was the most time-consuming task with the conventional method since it took significantly more actions than the other tasks. Interestingly, lookup was the most time-consuming with the proposed method, most probably because participants took extra time in scanning the presented information.

After the study, participants completed a short questionnaire that asked them to rate various aspects of the proposed method on a 7-point Likert scale. Results (Figure 6) revealed that all participants (100%) found the method easy to use. They all felt that it will enable them to accomplish the task of acquiring and sharing information while texting faster than the methods they typically use. One participant (female, 25 years) commented, *"[The method is] very useful while chatting and sharing information together. I specifically liked the searching location feature in the suggestion bar"*. They all felt that it will facilitate more engaging and meaningful

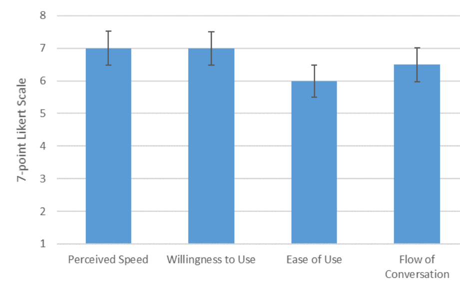

Figure 6: Median user ratings of the proposed method's perceived speed, willingness to use, ease of use, and the quality of conversation on a 7-point Likert scale. The choices were: (1) Strongly Disagree, 2) Disagree 3) Somewhat Disagree, 4) Neutral, 5) Somewhat Agree, 6) Agree, and 7) Strongly Agree. Error bars represent ±1 standard deviation (SD).

conversation by eliminating the need for switching between various apps. All of them wanted to keep using it on their devices. One participant (male, 27 years) stated, *"I would like to use it in my day to day conversations"*. Another (male, 26 years) wrote, *"The suggestion bar is pretty handy. I think this suggestion bar is more useful than predicting words because nowadays, humans can type very fast and most of the people don't even use predictive words but performing these complex tasks from suggestion bar is really cool"*.

7 CONCLUSION

We augmented a context-sensitive app prediction feature to the suggestion bar of a virtual keyboard to facilitate fast and easy information acquisition and sharing while texting. A user study revealed that this method improves performance both in terms of speed and effort. Besides, all participants found it fast, easy, and to facilitate more engaging and meaningful conversations. They all wanted to use it on their mobile devices.

7.1 Future Work

A major limitation of this work is the use of specific keywords to predict relevant apps. We will address this in a future work. We will also include additional features to the system, such as predicting relevant news, flight and hotel booking, and money sharing features, based on the context of an ongoing textual conversation.

REFERENCES

[1] Jessalyn Alvina, Joseph Malloch, and Wendy E. Mackay. 2016. Expressive Keyboards: Enriching Gesture-Typing on Mobile Devices. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, 583–593.

[2] Ahmed Sabbir Arif, Benedikt Iltisberger, and Wolfgang Stuerzlinger. 2011. Extending Mobile User Ambient Awareness for Nomadic Text Entry. In *Proceedings of the 23rd Australian Computer-Human Interaction Conference*. ACM, 21–30.

[3] Ahmed Sabbir Arif and Wolfgang Stuerzlinger. 2009. Analysis of Text Entry Performance Metrics. In *2009 IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH)*. IEEE, 100–105.

[4] Kenneth C. Arnold, Krzysztof Z. Gajos, and Adam T. Kalai. 2016. On Suggesting Phrases vs. Predicting Words for Mobile Text Composition. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, 603–608.

[5] 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*. ACM, 251–260.

[6] Xiaojun Bi, Tom Ouyang, and Shumin Zhai. 2014. Both Complete and Correct?: Multi-Objective Optimization of Touchscreen Keyboard. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*. ACM, 2297–2306.

[7] Xiaojun Bi, Barton A. Smith, and Shumin Zhai. 2010. Quasi-Qwerty Soft Keyboard Optimization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 283–286.

[8] Derek Bridge and Paul Healy. 2010. Ghostwriter-2.0: Product Reviews with Case-Based Support. In *International Conference on Innovative Techniques and Applications of Artificial Intelligence*. Springer, 467–480.

[9] Daniel Buschek, Alexander De Luca, and Florian Alt. 2015. There is More to Typing than Speed: Expressive Mobile Touch Keyboards via Dynamic Font Personalisation. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM, 125–130.

[10] Mark D. Dunlop and Andrew Crossan. 2000. Predictive Text Entry Methods for Mobile Phones. *Personal Technologies* 4, 2-3, 134–143.

[11] Ben S. Gerber, Melinda R. Stolley, Allison L. Thompson, Lisa K. Sharp, and Marian L. Fitzgibbon. 2009. Mobile Phone Text Messaging to Promote Healthy Behaviors and Weight Loss Maintenance: A Feasibility Study. *Health informatics journal* 15, 1, 17–25.

[12] Google. 2019. Gboard - The Google Keyboard. Retrieved 2018-06-01 from https://play.google.com/store/apps/details?id=com.google.android.inputmethod.latin

[13] Jason Griffin. 2005. Predictive Text Input System for a Mobile Communication Device. Google Patents. US Patent App. 10/783,901.

[14] Asela Gunawardana, Tim Paek, and Christopher Meek. 2010. Usability Guided Key-Target Resizing for Soft Keyboards. In *Proceedings of the 15th international conference on Intelligent user interfaces*. ACM, 111–118.

[15] Mariam Hassib, Daniel Buschek, Paweł W. Wozniak, and Florian Alt. 2017. HeartChat: Heart Rate Augmented Mobile Chat to Support Empathy and Awareness. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 2239–2251.

[16] Yijue How and Min-Yen Kan. 2005. Optimizing Predictive Text Entry for Short Message Service on Mobile Phones. In *Proceedings of HCII*, Vol. 5.

[17] Alexandros Karatzoglou, Linas Baltrunas, Karen Church, and Matthias Böhmer. 2012. Climbing the App Wall: Enabling Mobile App Discovery Through Context-Aware Recommendations. In *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 2527–2530.

[18] Amanda Lenhart et al. 2012. Teens, Smartphones & Texting. *Pew Internet & American Life Project* 21, 1–34. https://www.pewinternet.org/2012/03/19/teens-smartphones-texting

[19] Andrew Lepp, Jacob E. Barkley, and Aryn C Karpinski. 2014. The Relationship Between Cell Phone Use, Academic Performance, Anxiety, and Satisfaction with Life in College Students. *Computers in Human Behavior* 31, 343–350.

[20] Anders Markussen, Mikkel Rønne Jakobsen, and Kasper Hornbæk. 2014. Vulture: A Mid-Air Word-Gesture Keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1073–1082.

[21] Abhinav Parate, Matthias Böhmer, David Chu, Deepak Ganesan, and Benjamin M Marlin. 2013. Practical Prediction and Prefetch for Faster Access to Applications on Mobile Phones. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*. ACM, 275–284.

[22] Henning Pohl, Dennis Stanke, and Michael Rohs. 2016. EmojiZoom: Emoji Entry via Large Overview Maps. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM, 510–517.

[23] 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)*. ACM, New York, NY, USA, 83–88. https://doi.org/10.1145/2858036.2858305

[24] Dmitry Rudchenko, Tim Paek, and Eric Badger. 2011. Text Text Revolution: A Game That Improves Text Entry on Mobile Touchscreen Keyboards. In *International Conference on Pervasive Computing*. Springer, 206–213.

[25] Zhihao Shen, Kang Yang, Wan Du, Xi Zhao, and Jianhua Zou. 2019. DeepAPP: A Deep Reinforcement Learning Framework for Mobile Application Usage Prediction. In *ACM SenSys*.

[26] Ran Wei and Ven-Hwei Lo. 2006. Staying Connected While on the Move: Cell Phone Use and Social Connectedness. *New Media & Society* 8, 1, 53–72.

[27] Daryl Weir, Henning Pohl, Simon Rogers, Keith Vertanen, and Per Ola Kristensson. 2014. Uncertain Text Entry on Mobile Devices. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*. ACM, 2307–2316.

[28] Donghan Yu, Yong Li, Fengli Xu, Pengyu Zhang, and Vassilis Kostakos. 2018. Smartphone App Usage Prediction Using Points of Interest. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 4 (2018), 174.

[29] Shumin Zhai and Per Ola Kristensson. 2012. The Word-Gesture Keyboard: Reimagining Keyboard Interaction. *Commun. ACM* 55, 9 (2012), 91–101.