

# Enabling Predictive Number Entry and Editing on Touchscreen-Based Mobile Devices

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

Azar Alizadeh  
Human-Computer Interaction Group  
University of California, Merced  
Merced, California  
aalizadeh@ucmerced.edu

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

## ABSTRACT

We propose a method for accommodating predictive number entry and editing on mobile devices via the suggestion bar of a virtual keyboard. We developed a simple predictive system to demonstrate the benefit of this method. It utilizes text-based querying and regular expression to suggest the most probable *next* numeric actions in the suggestion bar along with word suggestions. We evaluated this method in two user studies. The first explored number entry and the second explored editing. Results revealed that the proposed method significantly increases number entry and editing speed and accuracy. It reduces the number of actions needed per task. It also significantly reduces the time and effort needed to fix errors. Subjective analysis revealed that almost all participants found the method faster, more reliable, and easier to use than the conventional method, thus wanted to keep using it on their mobile devices.

## CCS CONCEPTS

- **Human-centered computing** → **Text input**; *Empirical studies in HCI*; • **Information systems** → *Web and social media search*; • **Computing methodologies** → *Information extraction*; • **Applied computing** → *Text editing*.

## KEYWORDS

Text entry, numeric and special characters, symbols, digits, predictive text, suggestion bar.

### ACM Reference Format:

Laxmi Pandey, Azar Alizadeh, and Ahmed Sabbir Arif. 2020. Enabling Predictive Number Entry and Editing on Touchscreen-Based Mobile Devices. In *2020 Conference on Human Information Interaction and Retrieval (CHIIR '20)*, March 14–18, 2020, Vancouver, BC, Canada. ACM, Vancouver, Canada, 10 pages. <https://doi.org/10.1145/3343413.3377957>

## 1 INTRODUCTION

Although the acquirement, input, and editing of numeric values (represented either by digits or text) are important parts of mobile text entry [3], these have not been well explored in the literature.

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 ACM 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](mailto:permissions@acm.org).

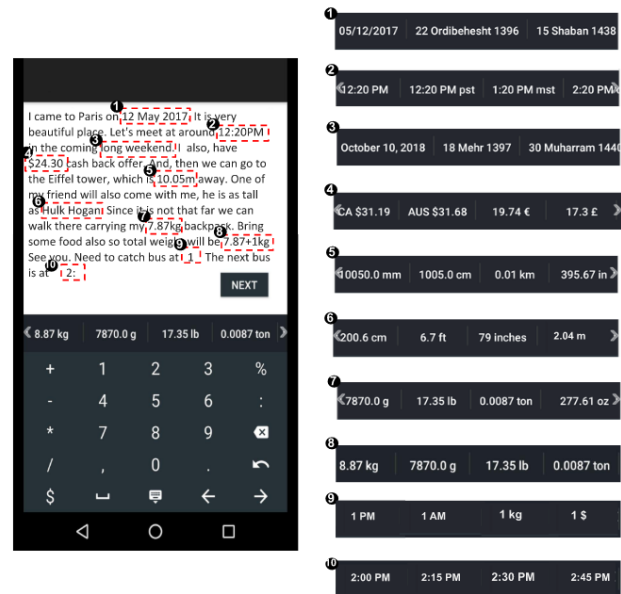
CHIIR '20, March 14–18, 2020, Vancouver, BC, Canada

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-6892-6/20/03...\$15.00

<https://doi.org/10.1145/3343413.3377957>

This negligence is evident in the suggestion bars of virtual keyboards. Nowadays, almost all virtual keyboards come with suggestion bars that present the most probable *next* words and seldom phrases using linguistic models [7, 21, 25]. However, none of these models provide the support for numeric values, thus the suggestion bars remain “blank” when these values are being entered or edited.



**Figure 1: The proposed method uses text-based querying and regular expression to detect numeric values and suggest the *next* probable number entry and editing tasks. The user can accept a suggestion by tapping on it. The left and right arrows indicate that there are more suggestions available, which the user can see by stroking left or right on the bar.**

The findings of an informal survey and user feedback from our previous studies revealed that mobile users desire an effective and user friendly method for interacting with numbers since they often work with numeric values on mobile devices. This includes entering an entirely new value (e.g., the date and time of a meeting), editing or converting an existing value (e.g., changing an amount from Euro to US Dollar), and performing arithmetic operations to a value (e.g., figuring out how to split a bill with friends before using a mobile payment app). Most users resolve to third-party apps for editing and conversion, such as calculators, unit converters, and

Web browsers. For instance, users tend to use a native or a Web app to find a time for a virtual meeting that is appropriate for all international attendees. Incriminating and decrementing numeric values also require the assistance of third-party apps since increasing and decreasing different units, such as time, currency, or length and weight, are fundamentally different from one another. This process is not only time-consuming and tedious but also distracts the user from the task at hand by forcing her to switch between different apps [1]. To mitigate this, we developed a simple predictive system that uses text-based querying and regular expression to identify numeric values to suggest the most likely *next* actions in the suggestion bar (Figure 1). However, the main contribution of this work is not the predictive system, but the demonstration that providing support for numeric values in the suggestion bar can radically simplify the task of entering and editing numbers on mobile devices. Figure 2 illustrates the architecture of the proposed system.

This paper starts with some examples of the predictive system. It then reviews the existing work in the area. It presents the proposed predictive system, evaluates it in two empirical studies, then discusses the findings of the studies and the limitations of the work. The paper concludes with speculations on future extensions.

## 2 EXAMPLES

The proposed predictive system suggests numeric values based on the context of a textual conversation. For this, the system uses a pre-trained DeepQA model [20] to process natural language with deep learning, then suggests relevant numeric values retrieved from a database or the Internet. Below are some examples.

Input	Suggestion
Let's go camping next Monday	The exact date: 10/08/2018
Is there a meeting in 45 minutes?	The absolute time: 12:45 PM
The distance between earth and moon is large	The exact distance: 238,900 miles

The predictive system suggests auto-completion for all numeric values. For this, it detects the nature of a value as the user types based on the context of the conversation, separators (decimal point, colon, etc.), symbols (dollar sign, percentage, etc.), and units (time, weight, etc.), then suggests the *next* probable numbers, words, suffixes, and prefixes for input. Below are some examples.

Input	Suggestion
12:	Quarter-hour increments: 12:00 PM, 12:15 PM, 12:30 PM
3.	Complete values with different units: \$3.99, 3.50 kg
61.52	Different units: 61.52 kg, \$61.52

The predictive system suggests unit conversions and arithmetic operations. For this, it detects the nature of a value as the user

types based on the context of the conversation, separators (decimal point, colon, etc.), symbols (dollar sign, percentage, etc.), units (time, weight, etc.), and arithmetic operation signs (plus, minus, division, multiplication, etc.), then suggests the most appropriate unit conventions and arithmetic operations. Below are some examples.

Input	Suggestion
I was in Paris on 05/12/2017	The date in different calendars: 15-Shaban-1438, ...
I went to a coffee shop at 12:20 PM	The time in different zones: 4:20 PM PST, ...
I had a Pizza for which I paid \$24.30	Converted currencies based on the current exchange rates from the Internet: €20.98, ...
I went to Eiffel tower that was about 10 m away	Sistance converted to different units: 0.01 km, ...
I tipped him 15% of \$25.55	The exact amount: \$3.83
55.42 + 37.97	The total: 93.39

The predictive system also facilitates editing existing numeric values. To edit an already entered value, the user taps on it to see all probable auto-completion, conversions, increment, decrement, and actual values retrieved from the Internet. Tapping on a value in the suggestion bar replaces the existing value with the selected one. Below are some examples.

Input	Suggestion
I am leaving on 10/21/19	Tapping on 10/21/19 suggests the date in different calendars: 29-Mehr-1398, 22-Safar-1441, ...
I have a meeting at 3 PM	Tapping on 3 PM suggests the time in different zones: 12 PM EST, 10 PM GMT, ...
I paid \$600 for the rent	Tapping on \$600 suggests the amount converted in different currencies: €534.47, £472.81, ...
He is as tall as Keanu Reeves	Tapping on "Keanu Reeves" suggests his actual height retrieved from the Internet in different units: 6.1 ft, 1.86 m, ...

## 3 RELATED WORK

Some have proposed novel keypad layouts to facilitate number entry on various devices. Isokoski and Käki [11] designed a gesture-based method for number entry on touchscreens. In a user study, this approach yielded a promising entry speed and accuracy. Cheng et al. [8] proposed a method for automatically adjusting the layout and position of a virtual keypad based on how the user is holding a mobile device. A study revealed that this approach increases entry speed by 42% compared to a manually adjustable keypad. Mary et al. [22] investigated the effects of different key sizes and spacing on touch characteristics in number entry tasks. They found out that touch force, impulse, and dwell time are significantly affected by key size. In a similar study, Tsang et al. [23] found out that both key size and layout affect input performance in terms of speed, accuracy,

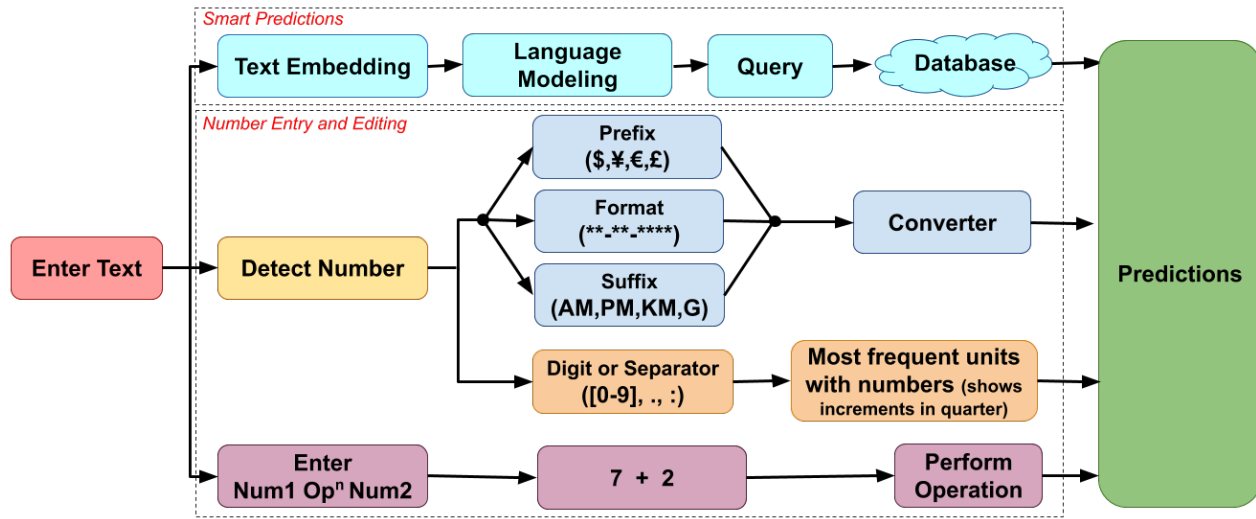


Figure 2: High level architecture of proposed predictive system.

and task completion rate. Chung et al. [9], in contrast, studied the effects of age on the usability of number entry with both virtual and physical keypads. They recruited adults aged 23–33 years and older adults aged 65–76 years. Results revealed that tactile feedback increases input accuracy of the virtual keypad, and both adults and older adults prefer the virtual keypad than the physical one.

Some have proposed novel keypad interactions to increase the security of conventional mobile user authentication approaches. Arif et al. [4] designed a keypad that enables the user to actively select digits and directional gestures as her passwords. Malek et al. [19] designed a force-based keypad that uses pressure as a binary input with two variances of the pattern-lock approach. Arif et al. [2] developed a different force-based keypad that uses three levels of pressure input as an extra security measure to the popular digit-lock approach.

Several recent works have focused on number entry on medical equipment since the existing approaches are not optimized for common data entry tasks in hospitals [28]. However, Oladimeji et al. [18] pointed out that these works explore only keypads and do not account for factors such as the range of values and the real-estate available on the equipment for keypads. Most of these works tend to show that input method significantly affects number entry speed and accuracy. In another work, Oladimeji et al. [17] reported that typically there are two types of number entry methods available for medical devices: serial (such as a keypad) and incremental (which uses knobs or keys to increment/decrement numeric values). They reported that incremental methods lead to more accurate input than serial methods. In a follow-up work, Wiseman et al. [27] proposed thirteen different codes for classifying errors, and using checksums to detect these errors [29]. In a separate work, Wiseman et al. [26] designed a novel layout to make the entry of the most frequently entered numbers easier on infusion pumps. For this, they collected data from infusion pumps programmed on the ward. They evaluated the method with three existing interfaces. Results revealed that it reduces the total number of keystrokes needed to complete a task.

But to our knowledge, no prior work has investigated how to accommodate number entry and editing in the context of mobile text entry.

## 4 THE PROPOSED METHOD

We propose a predictive system for number entry and editing that works in real-time and does not require using a different app to perform common actions with numbers. Figure 2 illustrates a high-level overview of the proposed framework. It comprises of three parts: 1) suggestions based on generic text-based queries on a database or search results from the Internet, 2) suggestions by detecting different types of numeric values using regular expression, and 3) suggestions based on real-time arithmetic operations and unit conversions.

### 4.1 Predictive System

The predictive system uses a semantically informed, context-aware language model that can understand natural language to retrieve necessary information from a knowledge base or the Internet for the suggestion bar. The system can suggest both words and numbers. It identifies the input that is semantically similar to the queries in the dataset [24], then displays numeric values associated with generic text-based queries in the suggestion bar along with conventional word predictions, when available. A numeric suggestion can be a date, a distance, or any other numeric value relevant to the context of the entered text. For this, the system uses an existing question-answering frameworks [24] and a pre-trained DeepQA model to perform high-level natural language processing tasks with deep learning [20]. DeepQA is built on top of Keras<sup>1</sup> and TensorFlow<sup>2</sup>. It uses a simple language model based on the seq2seq framework [15], which is trained for a conversational engine. This model can extract knowledge from both a domain specific dataset and a large, noisy,

<sup>1</sup>Keras: The Python Deep Learning library: <https://keras.io>

<sup>2</sup>TensorFlow: An End-to-End Open Source Machine Learning Platform: <https://www.tensorflow.org>

and general domain dataset of movie subtitles [24]. It is based on a recurrent neural network [14] that reads an input sequence (concatenation of what has been typed so far) and predicts the output sequence as reply. During training, the actual responses were given to the model to facilitate learning through backpropagation [14]. The model was trained to maximize the cross entropy of the correct responses for a given context [12]. We used this model because it can remember facts, understand contexts, perform common-sense reasoning without the complexity in traditional pipelines. Perhaps the most practical significant of the model is that it can generalize to new queries even when they do not appear in the training set. However, one drawback of this model is, sometimes it gives inconsistent answers to queries that are semantically comparable but not identical. We addressed this issue by using a predefined set of queries and by scraping off all unnecessary textual details from the responses. Below are some examples.

Query	Reply	Suggestion
What is two plus two?	It is 4	4
Two plus two equal to?	4	4
What time is it?	It's 8:30	8:30
What's the time?	It is 8:30	8:30

#### 4.2 Number Entry and Editing

The system identifies different types of numeric values using regular expression. It classifies each value based on its *prefix*, *suffix*, *format*, *separator*, and *operator* (Figure 2). Once the “type” of the value is determined, it acquires a list of suggestions (including digits, units, unit conversions, arithmetic operations, and words) and displays the most probable ones in the suggestion bar. The probability of all suggestions are determined using a custom frequency table<sup>3</sup> and a set of simple rules. For example, when the user enters “12:” the system classifies it as “time” since the frequency table suggests that the “:” separator is most commonly used with time values, then displays “12:00 PM”, “12:15 PM”, “12:30 PM”, and “12:45 PM” in the suggestion bar based on the rule “users prefer quarter-hour increments”. Note that the system suggests all time values in “PM” since “afternoon meetings are more preferred than late night meetings”. When a value does not have a prefix, suffix, separator, or operator, the system suggests the most common units based on the context of the input. For example, if the user types “I drove 15”, the system suggests the most common units, such as “miles” and “km”. The system also suggests common operations and unit conversions. For example, when the user enters “I drove 15 km”, the suggestion bar displays “9.3 miles”. Likewise, when the user types “Two third of 15 is”, the suggestion bar suggests the result “10”. The system reverts to the original value when the user uses the backspace key immediately after selecting a suggestion. For example, when the user converts “05/12/2017” to Iranian calendar “15-Shaban-1438” by mistake, she can press the backspace key to bring back the original input.

The system does not replace the conventional word and phrase suggestions. Instead, it displays numeric suggestions when the

<sup>3</sup>Since we were unable to find a frequency table for numeric values in mobile text entry, we developed a custom table based on the findings of an informal survey where we collected data from 35 frequent mobile users, average age 25.8 (SD = 3.3), 14 female, 21 male.

conventional method fails. In case both the conventional and the proposed method return suggestions, the suggestion bar displays results from both methods. Users can disable/enable the numeric suggestion feature by double-tapping on the suggestion bar.

### 5 EXPERIMENT 1: NUMBER ENTRY

We conducted a user study to compare the performance of the proposed predictive system with a conventional method of working with numbers (Google Quick Answer [13]) in various number entry tasks, including digit and symbol entries, metric conversions, arithmetic operations, and numeric answers of text-based queries.

#### 5.1 Apparatus

We used a Motorola Moto G<sup>5</sup> Plus smartphone (150.2×74×7.7 mm, 155 g) at 1080×1920 pixels in the study. We developed a custom keyboard and the proposed suggestion bar using the Android Studio 3.1, SDK 27, which looked and felt like the default Android keyboard (Figure 1 and Figure 3). It logged all actions with timestamps and calculated all performance metrics directly.

#### 5.2 Participants

Twelve volunteers aged 23–28 years (M = 25.3, SD = 1.61) took part in the study. They were all proficient in the English language. Four of them were female and eight were male. All of them were right-handed. They all chose to hold the device in portrait position. They all used Google Quick Answer and various third-party apps (e.g., calculators, converters) to work with numeric values.

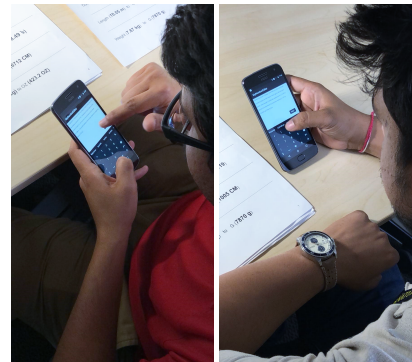


Figure 3: Two volunteers participating in the first user study exploring number entry.

#### 5.3 Design

The study used a within-subjects design. The independent variable (IV) was *method* and the dependent variables (DV) were the *performance metrics*. The final design was:

- 12 participants ×
- 2 conditions (conventional & predictive, counterbalanced) ×
- 10 tasks =
- 240 tasks in total, excluding practice tasks.



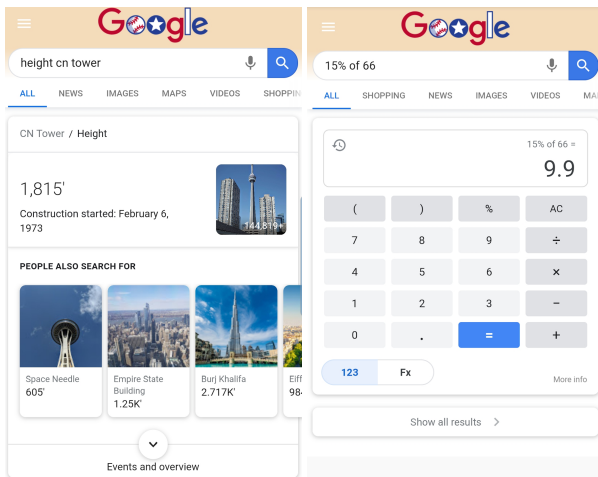
### 5.4 Metrics

We recorded the following performance metrics in the study.

- **Time per Task** represents the average time (seconds) users took to perform a task.
- **Actions per Task** signifies the average number of taps and gestures performed for each task. This metric is comparable to the Keystrokes per Character (KSPC) metric [5].
- **Error Rate** is the average percentage (%) of incorrect characters in the final text [5].
- **Correction per Task** is the average number of corrective actions (backspace, delete) performed per task.
- **Correction Time** signifies the average time (seconds) users took to correct input and edit errors.

### 5.5 Google Quick Answer

Google Quick Answers is a feature of Google Search Engine that places highlighted text snippet answers and links at the top of Search Engine Results Pages (SERP). This feature was introduced in September 2014 to enable users to acquire easy answers to “how to” or “what is” oriented queries. However, users are not required to inject “how to” or “what is” in their queries for this feature to work (Figure 4). To generate these answers, Google takes the relevant parts of content from Web pages that are already regarded as trustworthy. It then displays the answers in an outlined box on the top of the SERPs. About 40% of all queries on Google Search Engine display Quick Answers [13].



**Figure 4: Google Quick Answers displaying the height of the CN Tower (left) and 15% of 66 (right). Notice that the queries were not entered in full natural language.**

We used this method as the baseline since the informal survey<sup>3</sup> revealed that most users acquire, convert, and perform arithmetic operations to numeric values using this feature. During the user study, the mobile device had the Chrome Web browser<sup>4</sup> with Google as its default search engine opened. Participants were instructed to

<sup>4</sup>Google Chrome: <https://play.google.com/store/apps/details?id=com.android.chrome>

acquire and perform various operations to numeric values using this browser.

### 5.6 Procedure

First, we explained the study procedure to all participants and collected their consents. We then asked them to complete a short demographics and mobile usage questionnaire. Then, we demonstrated the Google Quick Answer and the proposed predictive method. We asked the participants to practice with both methods for about five minutes, although all of them were familiar with the Google Quick Answer. The tasks used in the practice were not repeated in the actual study. Participants could extend the practice period on request. Once they were familiar with both methods, they were instructed to perform ten tasks per condition (conventional and predictive). The conditions and the tasks were counterbalanced using a 4x4 balanced Latin square. We carefully selected the tasks to evaluate equal number of acquirement (search), auto-completions, conversions, arithmetic operations, and compound actions (e.g., conversion and arithmetic operation) in each condition. We also made sure that tasks require participants to select the same number of the *most* and the *least* probable suggestions to force them to swipe left and right on the suggestion bar. Some examples of the tasks used in the study are:

- Enter today’s date in Iranian calendar
- Enter US \$13.53 in Euro
- Enter 35 + 81 kg to oz
- Enter the time after 55 minutes in GMT
- Enter the height of Kiefer Sutherland in inches

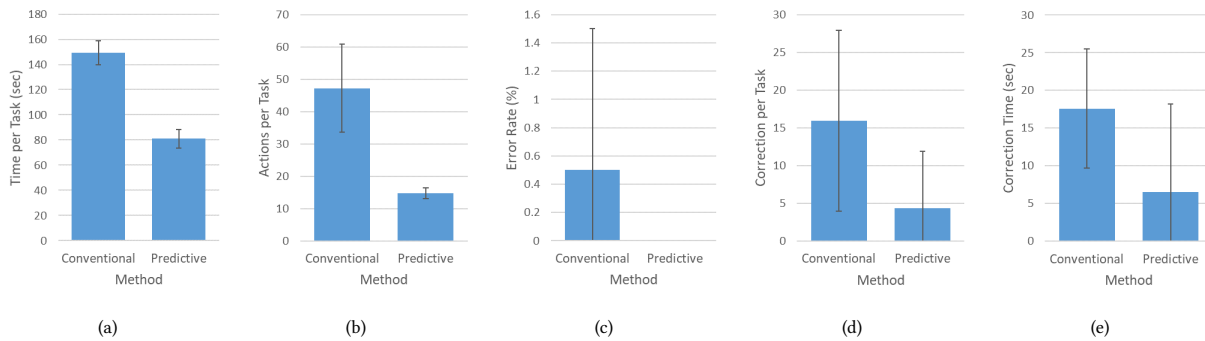
The tasks were presented in sheets of paper (Figure 3). We asked participants to complete each task as *fast and accurate* as possible. They were instructed to correct all errors as they notice them, but we did not enforce this. In the conventional condition, participants were instructed to complete the tasks using the Google Quick Answer and the default Android Qwerty keyboard<sup>5</sup>. The predictive system of the keyboard was enabled but did not provide any suggestions since it does not fully support number entry. In the predictive condition, participants were instructed to use the suggestion bar exclusively. Upon completion of the study, all participants completed a short post-study questionnaire.

### 5.7 Results

A complete session took 30–45 minutes, including demonstrations, practice periods, and breaks. A Shapiro–Wilk test and a Mauchly’s test confirmed that the data did not violate the normality or the sphericity assumptions. Thus, we used a repeated-measures ANOVA for all analysis.

**5.7.1 Time per Task.** An ANOVA identified a significant effect of method on time per task ( $F_{1,11} = 515.12, p < .0001$ ). On average participants spent 149.25 seconds (SD = 9.69) and 81.05 seconds (SD = 7.32) for each task with the conventional and the predictive method, respectively. Figure 5 (a) illustrates this.

<sup>5</sup>Gboard - the Google Keyboard: <https://play.google.com/store/apps/details?id=com.google.android.inputmethod.latin>



**Figure 5: Average (a) time per task, (b) actions per task, (c) error rate, (d) correction per task, and (e) correction time for the conventional (Google Quick Answers) and the proposed predictive method. Error bars represent ±1 standard deviation.**

**5.7.2 Actions per Task.** An ANOVA found a significant effect of method on actions per task ( $F_{1,11} = 5656.8, p < .0001$ ). On average participants performed 47.25 (SD =13.55) and 14.75 (SD = 1.6) actions per task with the conventional and the predictive method, respectively. Figure 5 (b) illustrates this.

**5.7.3 Error Rate.** An ANOVA also identified a significant effect of method on error rate ( $F_{1,11} = 25.51, p < .0005$ ). The average error rate for the conventional and the predictive method were 0.5% (SD = 1) and 0%, respectively. Figure 5 (c) illustrates this.

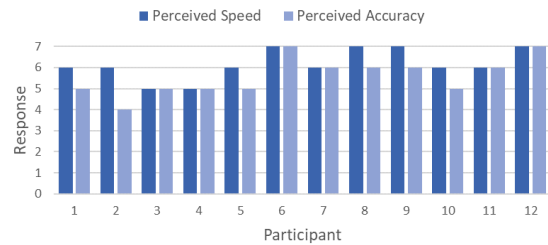
**5.7.4 Correction per Task.** An ANOVA identified a significant effect of method on correction per task ( $F_{1,11} = 264.43, p < .0001$ ). On average participants performed 15.98 (SD =11.98) and 4.4 (SD = 7.5) corrective actions per task with the conventional and the predictive method, respectively. Figure 5 (d) illustrates this.

**5.7.5 Correction Time.** An ANOVA identified a significant effect of method on correction time ( $F_{1,11} = 38.98, p < .0001$ ). On average participants spent 17.6 seconds (SD = 7.9) and 6.5 seconds (SD = 11.7) in performing corrective actions per task with the conventional and the predictive method, respectively. Figure 5 (e) illustrates this.

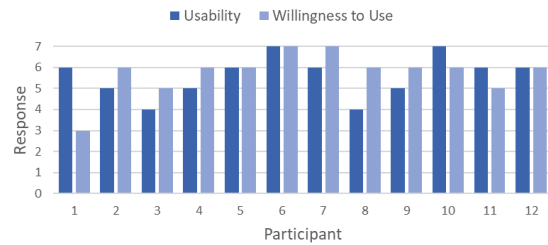
## 5.8 User Feedback

Upon completion of the study, all participants completed a short questionnaire where they rated the predictive method’s entry speed, accuracy, usability, and willingness to use on a 7-point Likert scale. We later converted these scales to 3-point scales using linear transformation to calculate ratios, which is common practice in statistics [10].

**5.8.1 Perceived Speed and Accuracy.** All participants (100%, N = 12) felt that the predictive method improved their entry speed. Most participants (92%, N = 11) also believed that it improved their accuracy. The remaining 8% (N = 1) was neutral. Figure 6 illustrates all responses, where one can see that all participants rated the predictive method “5” or higher for entry speed and “4” or higher for accuracy on 7-point Likert scales. The median user ratings of the method’s speed and accuracy were 6.0 (SD = 0.72) and 5.5 (SD = 0.9), respectively.



**Figure 6: User ratings of the statements that the predictive method increased their number entry speed and accuracy on 7-point Likert scales, where “1–7” represented “Strongly Disagree–Strongly Agree”.**



**Figure 7: User ratings of the statements that the predictive method was easy to use and they would use it frequently on 7-point Likert scales, where “1–7” represented “Strongly Disagree–Strongly Agree”.**

**5.8.2 Usability and Willingness to Use.** Most participants (83%, N = 10) found the predictive method easy to use, while two (17%) were neutral about it. Most of them (92%, N = 11) also wanted to use it frequently on their mobile devices, one (8%) was against it. Figure 11 illustrates all user responses, where one can see that the participants rated the predictive method “4” or higher for usability and “3” or higher for willingness to use on 7-point Likert scales. The median user ratings of the method’s usability and willingness to use were 6.0 (SD = 0.99) and 6.0 (SD = 1.05), respectively.

## 6 EXPERIMENT 2: NUMBER EDITING

We conducted a second study to investigate the predictive method’s performance in editing tasks (interactions with existing numeric values).

### 6.1 Apparatus

We used the same smartphone as the first study. However, we developed a new app that loads short paragraphs containing various numeric values for the participants to edit (Figure 1). We used the same keyboard and suggestion bar as the previous study. The custom app logged all actions with timestamps and calculated all performance metrics directly.

### 6.2 Participants

Twelve new volunteers aged 19–36 years ( $M = 25.5$ ,  $SD = 5.03$ ) participated in the study (Figure 8). Five of them were female and seven were male. They all were proficient in English. Ten of them were right-handed and two were left-handed. All of them chose to hold the device in portrait position. They all used Google Quick Answer and various third-party apps (e.g., calculators, converters) to work with numeric values. They received US \$10 for volunteering in the study.

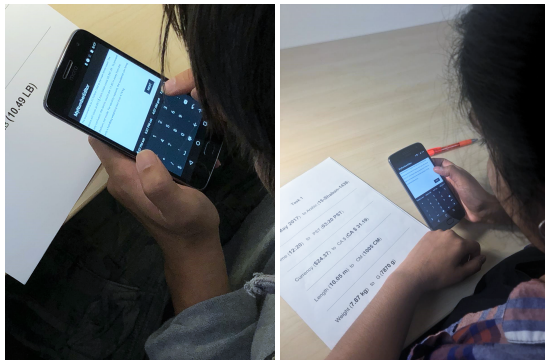


Figure 8: Two volunteers participating in the second study investigating predictive number editing.

### 6.3 Design

The study used a within-subjects design. The independent variable (IV) was *method* and the dependent variables (DV) were the *performance metrics*. The final design was:

- 12 participants ×
- 2 conditions (transcribe & predictive, *counterbalanced*) ×
- 3 paragraphs × 6 tasks =
- 360 tasks in total, excluding practice tasks.

### 6.4 Metrics

We recorded the *time per task*, *actions per task*, and *error rate* metrics as the first user study. We did not record the *correction per task* and *correction time* metrics since this study did not require participants to correct errors.

### 6.5 Procedure

This study used the same procedure as the first user study with the following exceptions. First, the printed sheets of paper included correct answers to all tasks since we wanted to find out if the proposed method outperforms the conventional method when participants already know the answers (do not have to search and cut/copy-paste the values using a third-party app). It also enabled us to complete the study within a reasonable time frame. In the first study, participants took on average 1.5 minutes to perform an input task with the conventional method. Performing an editing task would have taken more time since it involves the extra step of deleting and replacing an existing value. Second, we removed the Google Quick Answer from the demonstration since participants were not required to use it to acquire numeric values. Finally, we did not instruct participants to correct their errors since we wanted to emphasize *only* on the time and effort required for editing tasks, not on transcription/input errors.

In the study, participants edited three short paragraphs containing six different types of numeric values with each method (hence, six paragraphs in total) in a counterbalanced order. The custom app presented one paragraph at a time and asked participants to edit all numeric values as indicated in the printed task sheets. We used a similar approach as the first study to select the tasks. The following is an example paragraph (the underlined parts represent numeric values) and the respective tasks.

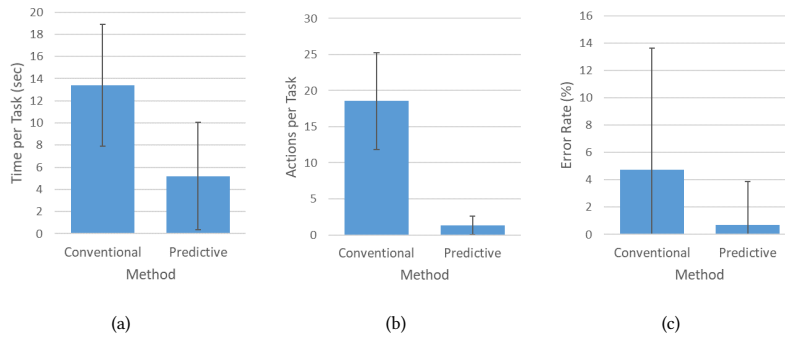
*“I went to the outlet mall on June 7, 2013. I arrived at around 19:45. I was looking for a table lamp and I found one for \$23.99. The distance from the mall to the parking was about 54.72 m. I had to carry that 8 kg lamp all the way up there! Just so you know, a bowling ball weights about ...”*

- Replace 7-Jun-2013 with Tamil calendar
- Replace 19:45 with PM
- Replace \$23.99 with Canadian Dollar
- Replace 54.72 meters with feet
- Replace 8 kg with pounds
- Replace “...” with actual weight of a bowling ball in kg

The app highlighted all numeric values in the paragraphs in red font for easier detection. In the conventional condition, participants simply had to transcribe the values in the paragraph using the default Android keyboard<sup>5</sup> since the task sheets provided all answers. In the predictive condition, they had to pick the values from the suggestion bar. Like the first study, the predictive system of the default Android keyboard was enabled but did not suggest numbers since it does not fully support number entry. Once done with editing a paragraph, participants had to press the “Next” button to see the next paragraph. Upon completion of the study, participants completed a short post-study questionnaire.

### 6.6 Results

A complete session took 30–45 minutes, including demonstrations, practice periods, and breaks. A Shapiro–Wilk test and a Mauchly’s test confirmed that the data did not violate the normality or the sphericity assumptions. Thus, we used a repeated-measures ANOVA for all analysis.

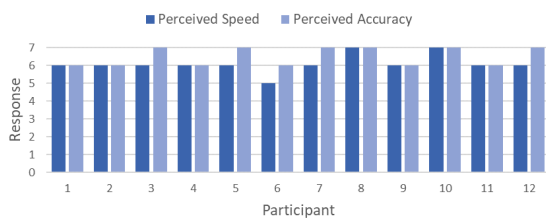


**Figure 9: Average (a) time per task, (b) actions per task, and (c) error rate for the conventional and the predictive method. Error bars represent ±1 standard deviation.**

**6.6.1 Time per Task.** An ANOVA identified a significant effect of method on time per task ( $F_{1,11} = 526.02, p < .0001$ ). On average participants spent 13.41 seconds (SD = 5.49) and 5.21 seconds (SD = 4.83) per task with the conventional and the predictive method, respectively. Figure 9 (a) illustrates this.

**6.6.2 Actions per Task.** An ANOVA found a significant effect of method on actions per task ( $F_{1,11} = 5236.4, p < .0001$ ). On average participants performed 18.56 (SD = 6.7) and 1.31 (SD = 1.3) actions per task with the conventional and the predictive method, respectively. Figure 9 (b) illustrates this.

**6.6.3 Error Rate.** An ANOVA also identified a significant effect of method on error rate ( $F_{1,11} = 31.94, p < .0005$ ). The average error rate for the conventional and the predictive method were 4.7% (SD = 8.9) and 0.7% (SD = 3.2), respectively. Figure 9 (c) illustrates this.

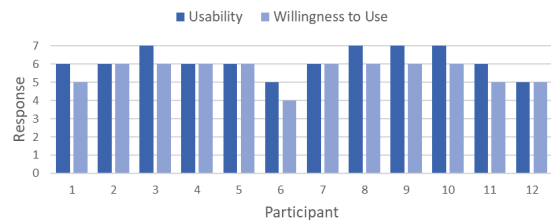


**Figure 10: User ratings of the statements that the predictive method increased their number editing speed and accuracy on 7-point Likert scales, where “1–7” represented “Strongly Disagree–Strongly Agree”.**

## 6.7 User Feedback

Once done with the study, all participants completed a short questionnaire where they rated the predictive method’s entry speed, accuracy, usability, and willingness to use on a 7-point Likert scale. Like the first study, we converted these scales to 3-point scales using linear transformation to calculate ratios [10].

**6.7.1 Perceived Speed and Accuracy.** All participants (100%, N = 12) felt that the predictive method improved their number editing



**Figure 11: User ratings of the statements that the predictive method was easy to use and they would use it frequently on 7-point Likert scales, where “1–7” represented “Strongly Disagree–Strongly Agree”.**

speed and accuracy. Figure 10 displays all responses, where one can see that all participants rated the predictive method “5” or higher for both editing speed and accuracy on 7-point scales. The median user ratings of the method’s speed and accuracy were 6.0 (SD = 0.51) and 6.5 (SD = 0.52), respectively.

**6.7.2 Usability and Willingness to Use.** All participants (100%, N = 12) found the predictive method easy to use. Most participants (92%, N = 11) also wanted to use it frequently on their mobile devices. One participant (8%) was neutral about it. Figure 11 illustrates all user responses, where one can see that all participants rated the predictive method “5” or higher for usability and “4” or higher for willingness to use on 7-point scales. The median user ratings of the method’s usability and willingness to use were 6.0 (SD = 0.72) and 6.0 (SD = 0.67), respectively.

## 7 DISCUSSION

The predictive method performed significantly better than the conventional method in both number entry and editing tasks. Results of the first user study revealed that number entry with the proposed method was 46% faster and required 69% fewer actions compared to the conventional method. It also required 72% fewer corrective actions, saving roughly 64% in correction time.

The conventional condition of the second user study did not require participants to acquire and calculate numeric values since



the task sheets included all answers. Yet, the proposed method was 61% faster and required 93% fewer actions than the conventional method. It was also 86% more accurate. This suggests that users are likely to take more time and make more mistakes with the conventional method in real-world scenarios where they have to acquire and calculate these values using third-party apps. Therefore, it can be said that the studies established that the proposed method can substantially improve mobile users' number entry and editing performance and experience.

Qualitative data also supports this. For input tasks, all participants responded that the predictive method improved their number entry speed. Most participants also felt that it improved their accuracy. Besides, almost all participants found the predictive method easy to use, thus wanted to keep using it on their mobile devices. Many of them expressed their enthusiasm about the method after the study. For example, one participant (female, 25 years) stated, *“Number prediction from text input is pretty cool and useful. [It] saves time.”* Some participants felt that the proposed method is useful not only for casual text entry but also for work related tasks. For example, one participant (female, 23 years) commented, *“The [...] method really allowed me to reduce time while allowing me to be accurate. I can see how this [method] can be useful for people that complete administrative tasks”*. Many participants liked the “Undo” feature of the system that enables them to revert to the original value by pressing the backspace key since it eliminates the need for re-typing the value. One participant (female, 22 years) commented, *“This option [...] needed a little getting used to, but once I got it, it worked quite fine. This is a very good way to ease up [correction effort]”*.

For edit tasks, all participants felt that the predictive method improved their editing speed and accuracy. Besides, they all found it easy to use, and almost all of them wanted to keep using it on their mobile devices. Although user feedback was encouraging for both input and edit tasks, participants were relatively more positive about the predictive method in the second study, which involved editing numeric values. We speculate that this is because participants struggle more with editing tasks than input tasks [6]. The fact that mobile users frequently acquire and share information with their family and friends may have contributed towards this as well (since they could relate to the tasks used in the study more). Relevantly, one participant (male, 25 years) responded, *“[editing and] prediction of metric conversion are really good [and] make sense [to me]”*.

Participants left no erroneous characters (0%) in the final text during the first user study since they were instructed to correct all errors. Figure 5 (a and e) illustrates the time and effort invested in error correction efforts during the study. In the second user study, the final text produced with the predictive method contained 0.7% erroneous characters, compared to the conventional method's 4.7% erroneous characters in the final text. A deeper analysis of the data revealed that almost all errors committed with the conventional method were due to transcription mistakes. The errors committed with the predictive method were rather diverse. About 75% of these errors were committed due to incorrect swipes on the suggestion bar, 17% were due to the selection of incorrect suggestions, and the remaining 8% were due to transcription errors when participants were manually editing the values.

## 8 LIMITATIONS

There are some limitations of this work. First, the frequency table<sup>3</sup> used in the predictive system is developed using a small sample of young adults ( $N = 35$ ) who were heavy smartphone users. Thus, the system evaluated in the studies is not generalizable to a larger audience. But as articulated earlier, the aim of this work is not to promote the developed system, instead to demonstrate how number prediction can radically simplify the task of entering and editing numbers on mobile devices. Second, we only used the tasks that will yield relevant suggestions in the user studies. It increased the internal validity but reduced the external validity. We decided against using a free-form text entry scenario in the studies because it lacks control for performance evaluations as it is difficult, if not impossible, to determine the participants' *true* intentions in free-form text entry episodes [16]. Finally, we used the Google Quick Answer as the baseline condition in the first user study since it is one of the most popular methods for acquiring numeric values. Alternative methods, such as using a third-party app or a different device, were not explored in the study.

## 9 CONCLUSION

We proposed enabling number entry and editing on mobile devices through the suggestion bar of a virtual keyboard. To showcase the benefit of this approach, we developed a simple predictive system that suggests the most probable *next* numeric values in the suggestion bar using text-based querying and regular expression. We evaluated this method in two user studies, the first focused on number entry and the second on number editing. Results revealed that the proposed method significantly increases number entry and editing speed and accuracy. It also reduces the number of actions needed per task, and the time and effort needed to correct errors. Subjective analysis revealed that most participants found the proposed method faster, more accurate, and easier to use. They all wanted to use it frequently on their mobile devices.

## 10 FUTURE WORK

We will improve the proposed predictive system to increase its reliability. We will apply machine learning approaches to provide mobile users with personalized suggestions based on their number entry and editing habits. We will also conduct a qualitative study to find out the most common types of numeric values entered and edited on mobile devices to enrich our frequency table. Besides, we will investigate the impact of our system on the performance of medical equipment. Previous research showed the existing number entry method on these devices are not reliable, thus can cause adverse events in data entry tasks for numerical drug dosing in hospitals [17, 28]. We will investigate whether the proposed method can mitigate this by suggesting numbers and auto-corrections based on previous inputs and prescriptions.

## ACKNOWLEDGMENTS

We thank ACM-W for awarding Laxmi Pandey a scholarship to present this work at CHIIR in Vancouver, BC, Canada. This work has been supported in part by the UC Merced Academic Senate Faculty Research Grants.

## REFERENCES

- [1] 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 (OzCHI '11)*. ACM, New York, NY, USA, 21–30. <https://doi.org/10.1145/2071536.2071539>
- [2] Ahmed Sabbir Arif, Ali Mazalek, and Wolfgang Stuerzlinger. 2014. The Use of Pseudo Pressure in Authenticating Smartphone Users. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 151–160. <https://doi.org/10.4108/icst.mobiquitous.2014.257919>
- [3] Ahmed Sabbir Arif, Michel Pahud, Ken Hinckley, and Bill Buxton. 2014. Experimental Study of Stroke Shortcuts for a Touchscreen Keyboard with Gesture-Redundant Keys Removed. In *Proceedings of Graphics Interface 2014 (GI '14)*. Canadian Information Processing Society, Toronto, ON, Canada, 43–50. <http://dl.acm.org/citation.cfm?id=2619648.2619657>
- [4] Ahmed Sabbir Arif, Michel Pahud, Ken Hinckley, and William Buxton. 2013. A Tap and Gesture Hybrid Method for Authenticating Smartphone Users. In *Proceedings of the 15th international conference on Human-computer interaction with mobile devices and services (MobileHCI '13)*. Association for Computing Machinery, Munich, Germany, 486–491. <https://doi.org/10.1145/2493190.2494435>
- [5] 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)*. 100–105. <https://doi.org/10.1109/TIC-STH.2009.5444533>
- [6] Ahmed Sabbir Arif and Wolfgang Stuerzlinger. 2010. Predicting the Cost of Error Correction in Character-Based Text Entry Technologies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 5–14. <https://doi.org/10.1145/1753326.1753329>
- [7] 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 (UIST '16)*. Association for Computing Machinery, Tokyo, Japan, 603–608. <https://doi.org/10.1145/2984511.2984584>
- [8] Lung-Pan Cheng, Hsiang-Sheng Liang, Che-Yang Wu, and Mike Y. Chen. 2013. Igrasp: Grasp-Based Adaptive Keyboard for Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. Association for Computing Machinery, Paris, France, 3037–3046. <https://doi.org/10.1145/2470654.2481422>
- [9] Min K. Chung, Dongjin Kim, Seokhee Na, and Donghun Lee. 2010. Usability Evaluation of Numeric Entry Tasks on Keypad Type and Age. *International Journal of Industrial Ergonomics* 40, 1 (Jan. 2010), 97–105. <https://doi.org/10.1016/j.ergon.2009.08.001>
- [10] John Dawes. 2008. Do Data Characteristics Change According to the Number of Scale Points Used? An Experiment Using 5-Point, 7-Point and 10-Point Scales. *International Journal of Market Research* 50, 1 (Jan. 2008), 61–104. <https://doi.org/10.1177/147078530805000106>
- [11] Poika Isokoski and Mika Käki. 2002. Comparison of Two Touchpad-Based Methods for Numeric Entry. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02)*. Association for Computing Machinery, Minneapolis, Minnesota, USA, 25–32. <https://doi.org/10.1145/503376.503382>
- [12] Katarzyna Janocha and Wojciech Marian Czarnecki. 2017. On Loss Functions for Deep Neural Networks in Classification. *arXiv:1702.05659 [cs]* (Feb. 2017). <http://arxiv.org/abs/1702.05659> arXiv: 1702.05659.
- [13] Kirill Kronrod. 2017. Google Answers, Quick Answers: Best Practices. <https://www.brightedge.com/blog/google-quick-answers/>
- [14] Zachary C. Lipton, John Berkowitz, and Charles Elkan. 2015. A Critical Review of Recurrent Neural Networks for Sequence Learning. *arXiv:1506.00019 [cs]* (Oct. 2015). <http://arxiv.org/abs/1506.00019> arXiv: 1506.00019.
- [15] Minh-Thang Luong, Ilya Sutskever, Quoc V. Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the Rare Word Problem in Neural Machine Translation. *arXiv:1410.8206 [cs]* (May 2015). <http://arxiv.org/abs/1410.8206> arXiv: 1410.8206.
- [16] I. Scott MacKenzie and R. William Soukoreff. 2003. Phrase Sets for Evaluating Text Entry Techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems (CHI EA '03)*. ACM, New York, NY, USA, 754–755. <https://doi.org/10.1145/765891.765971>
- [17] Patrick Oladimeji, Harold Thimbleby, and Anna Cox. 2011. Number Entry Interfaces and Their Effects on Error Detection. In *Human-Computer Interaction – INTERACT 2011 (Lecture Notes in Computer Science)*, Pedro Campos, Nicholas Graham, Joaquim Jorge, Nuno Nunes, Philippe Palanque, and Marco Winckler (Eds.). Springer, Berlin, Heidelberg, 178–185. [https://doi.org/10.1007/978-3-642-23768-3\\_15](https://doi.org/10.1007/978-3-642-23768-3_15)
- [18] Patrick Oladimeji, Harold Thimbleby, and Anna L. Cox. 2013. A Performance Review of Number Entry Interfaces. In *Human-Computer Interaction – INTERACT 2013 (Lecture Notes in Computer Science)*, Paula Kotzé, Gary Marsden, Gitte Lindgaard, Janet Wesson, and Marco Winckler (Eds.). Springer, Berlin, Heidelberg, 365–382. [https://doi.org/10.1007/978-3-642-40483-2\\_26](https://doi.org/10.1007/978-3-642-40483-2_26)
- [19] Mauricio Cantellano Orozco, Behzad Malek, Mohamad A. Eid, and Abdulmoteleb El Saddik. 2006. Haptic-Based Sensible Graphical Password. In *Proceedings of Virtual Concept*. Playa Del Carmen, Mexico, 1–10.
- [20] Etienne Pot. 2020. Conchylcultor/DeepQA. <https://github.com/Conchylcultor/DeepQA> original-date: 2016-07-07T17:17:20Z.
- [21] 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, San Jose, California, USA, 83–88. <https://doi.org/10.1145/2858036.2858305>
- [22] Mary E. Sesto, Curtis B. Irwin, Karen B. Chen, Amrish O. Chourasia, and Douglas A. Wiegmann. 2012. Effect of Touch Screen Button Size and Spacing on Touch Characteristics of Users with and Without Disabilities. *Human Factors* 54, 3 (June 2012), 425–436. <https://doi.org/10.1177/0018720811433831>
- [23] Steve N. H. Tsang, H. S. Chan, and K. Chen. 2013. A Study on Touch Screen Numeric Keypads: Effects of Key Size and Key Layout. In *Hong Kong (IMECS '13)*, Vol. II. IAING, Hong Kong, 4.
- [24] Oriol Vinyals and Quoc Le. 2015. A Neural Conversational Model. *arXiv:1506.05869 [cs]* (July 2015). <http://arxiv.org/abs/1506.05869> arXiv: 1506.05869.
- [25] 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 (CHI '14)*. ACM, New York, NY, USA, 2307–2316. <https://doi.org/10.1145/2556288.2557412>
- [26] Sarah Wiseman, Duncan P. Brumby, Anna Cox, and Orla Hennessy. 2013. Tailoring Number Entry Interfaces To The Task of Programming Medical Infusion Pumps. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 57, 1 (Sept. 2013), 683–687. <https://doi.org/10.1177/1541931213571148>
- [27] Sarah Wiseman, Paul Cairns, and Anna Cox. 2011. A Taxonomy of Number Entry Error. In *Proceedings of the 25th BCS Conference on Human-Computer Interaction (BCS-HCI '11)*. BCS Learning & Development Ltd., Newcastle-upon-Tyne, United Kingdom, 187–196.
- [28] Sarah Wiseman, Anna L. Cox, and Duncan P. Brumby. 2013. Designing Devices With the Task in Mind: Which Numbers Are Really Used in Hospitals? *Human Factors* 55, 1 (Feb. 2013), 61–74. <https://doi.org/10.1177/0018720812471988>
- [29] Sarah Wiseman, Anna L. Cox, Duncan P. Brumby, Sandy J.J. Gould, and Sarah O'Carroll. 2013. Using Checksums to Detect Number Entry Error. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. Association for Computing Machinery, Paris, France, 2403–2406. <https://doi.org/10.1145/2470654.2481332>