Speech Is 3x Faster than Typing for English and Mandarin Text Entry on Mobile Devices

Sherry Ruan
Computer Science Department
Stanford University
Stanford, CA, USA
ssruan@stanford.edu

Jacob O. Wobbrock
Information School
DUB Group
University of Washington
Seattle, WA, USA
wobbrock@uw.edu

Kenny Liou
Baidu
Sunnyvale, CA, USA
kennyliou@baidu.com

Andrew Ng
Baidu
Sunnyvale, CA, USA
andrewng@baidu.com

James Landay
Computer Science Department
Stanford University
Stanford, CA, USA
landay@stanford.edu

ABSTRACT
With laptops and desktops, the dominant method of text entry is the full-size keyboard; now with the ubiquity of mobile devices like smartphones, two new widely used methods have emerged: miniature touch screen keyboards and speech-based dictation. It is currently unknown how these two modern methods compare. We therefore evaluated the text entry performance of both methods in English and in Mandarin Chinese on a mobile smartphone. In the speech input case, our speech recognition system gave an initial transcription, and then recognition errors could be corrected using either speech again or the smartphone keyboard. We found that with speech recognition, the English input rate was 3.0x faster, and the Mandarin Chinese input rate 2.8x faster, than a state-of-the-art miniature smartphone keyboard. Further, with speech, the English error rate was 20.4% lower, and Mandarin error rate 63.4% lower, than the keyboard. Our experiment was carried out using Deep Speech 2, a deep learning-based speech recognition system, and the built-in QWERTY or Pinyin (Mandarin) Apple iOS keyboards. These results show that a significant shift from typing to speech might be imminent and impactful. Further research to develop effective speech interfaces is warranted.

ACM Classification Keywords
H.5.2. User Interfaces: Voice I/O.

Author Keywords
Mobile text entry; speech input; continuous speech recognition; text input; mobile devices; smartphones.

INTRODUCTION
Users today spend immense amounts of time texting using smartphones [28]. But a smartphone's miniature touch-based keyboard can be slow and frustrating to use, especially when a user wants to compose a long message. Given the amount of time users are spending on smartphones and other mobile devices, it remains important to design an effective off-desktop text entry method that can greatly reduce users' frustration and improve efficiency [39]. Different text entry methods have been designed and implemented in recent years and extensive research has been conducted to evaluate their effectiveness in different settings [7, 26, 36]. Most methods are focused on virtual or physical keyboards or keypads, often with alternative key arrangements or letter disambiguation algorithms [17, 38]. Speech has attracted some interest [23] and there have been several popular speech-based assistants, such as Apple's Siri, Microsoft's Cortana, Google Now, Amazon Echo's Alexa, and Baidu's Duer.

Despite decades of speech recognition research, speech recognition accuracy has not been sufficiently high for speech systems to enjoy widespread use. Indeed, back in 1999, Karat et al. [10] concluded that the accuracy of speech input was far inferior to keyboard input. Technical constraints included ambient noise and the lack of support for out-of-vocabulary words [40]. One year later, in 2000, Shneiderman [27] showed that the same cognitive and memory resources that are used in speech production are used in problem solving, hindering human performance in text composition.

However, in the last several years, there have been great advances in speech recognition due to the advent of deep learning models and computation [8, 24]. Indeed, speech recognition recently surpassed the threshold of having superior accuracy to human recognition, albeit in limited contexts [1]. In light of these advances, it is now pertinent to re-explore the potential of speech-based text entry, specifically for input into today's smartphones and other mobile devices. We hypothesize that the state-of-the-art speech recognition systems can now greatly improve text entry performance compared to those of 5 or 10 years ago. To test our hypothesis, we designed an interface integrating both speech and keyboard input. We conducted a rigorous comparative study between a state-of-the-art speech recognition system and a state-of-the-art miniature touch-based keyboard to quantitatively evaluate the performance of the two.

In our study, we found that text entry speeds, in words per minute (WPM), using speech were about 3.0 times faster than the keyboard for English (161.20 vs. 53.46 WPM) and about 2.8 times faster than the keyboard for Mandarin Chinese (108.43 vs. 31.31 WPM). Total error rates were also favorable to speech, with speech error rates being 20.4% lower than the keyboard error rates in English (2.93% vs. 3.68%), and 63.4% lower in Mandarin (7.51% vs. 20.54%). Thus, speech was demonstrably faster and more accurate than the keyboard.

The contribution of this work is the first rigorous empirical evaluation of a state-of-the-art deep learning-based speech recognition system and a state-of-the-art touch-based miniature keyboard for mobile text entry. Moreover, this contribution is made for two languages, English and Mandarin Chinese, which are the most influential language, and the most widely spoken language, respectively [13]. In addition, we offer a method of error correction that can utilize speech or the keyboard. We also report novel speech-based measures that can be reused in subsequent evaluations of speech-based text entry. Finally, we offer insights for how to improve interaction designs for speech-based text entry.

For smartphone users wishing to have a more efficient text input mechanism, this research suggests that modern deep learning-based speech recognition systems can be an effective mechanism.

RELATED WORK

In this section, we mainly discuss studies applicable to voice-based text entry. The reader is directed elsewhere for a thorough review of text entry methods [3, 17, 39].

Research has shown that humans' speaking rate can be as fast as 200 WPM for English [22] and 250 characters per minute (CPM) for Mandarin [37]. However, none of the usability studies to date have claimed to achieve such an English text entry rate on mobile devices. In a user study of Parakeet, a continuous speech recognition system for mobile phones, participants entered text at an average rate of 18 WPM when seated and 13 WPM when walking [31]. Another research study showed that users reached only 7.8 WPM for composition tasks and 13.6 WPM for transcription tasks using speech, compared to 32.5 WPM for a keyboard-mouse method [10]. In contrast, users were able to achieve a higher entry rate with elaborately designed keyboards and some practice. A longitudinal study of a mini-QWERTY keyboard showed that participants reached an average of 60 WPM after 20 twenty-minute typing sessions [6]. There are hardly any research results on the performance comparison of Mandarin speech and typing input methods.

Past research also reveals several limitations of speech recognition accuracy. Price et al. [21] observed a recognition error rate of about 33-44%, and they concluded that this may be partly due to background noise, a common and persistent problem for deployed speech-based systems. Furthermore, Bradford [2] claimed that recognizing user actions with speech recognition was inherently error prone and no reliable solution to this problem existed. However, part of his reasoning was built upon a research result from 1988 where the speech recognition systems were mostly based on signal processing and pattern matching [14].

The low accuracy of speech input methods may also be ascribed to the strict use of speech as the error correction mechanism. In fact, correcting with speech commands has shown to be susceptible to cascading failures, in which correction commands are misinterpreted by the speech recognition system and have to themselves be corrected [10]. As with mobile systems like smartphones, we incorporate a keyboard in our speech input method and provide the user with the flexibility to correct errors using either speech or the keyboard. Our results reveal that most users prefer keyboard to speech for error correction and this significantly improves performance.

Speech input methods have also been shown to be undesired by users. A longitudinal study showed that seven out of eight new users abandoned their speech recognition system after six months, mainly due to their unsatisfying user experience with speech recognition [11]. Another review expressed users' concern for speech input because of its lack of privacy, security, and confidentiality in social settings [25].

Admittedly, results from previous speech input studies were not competitive compared to those of mobile keyboard input methods. However, an important explanation for this is that the speech recognition techniques of the past were not as mature as they are today. Recently, speech recognition systems have made significant advances because of the availability of large amounts of data and sophisticated deep learning models [1]. We believe that today's speech recognition systems will no longer thwart the effectiveness and practicality of speech as a general-purpose text entry method.

EXPERIMENT

We conducted a study to evaluate the performance of two input methods, speech recognition and a touch screen keyboard, in two languages, English and Mandarin Chinese. Our goal was not only to capture high-level measures such as speed and accuracy, but also to further investigate how the speech input interface could be improved based on the low-level measures obtained.

Participants

A total of 38 people participated in the study. The data from six of them were discarded as they failed to properly follow instructions. Of the 32 remaining participants, 16 were native American English speakers and the remaining were native Mandarin Chinese speakers. All participants were university students majoring in different fields such as computer science, materials science, economics, chemistry, and business. Every participant was familiar with either an English QWERTY

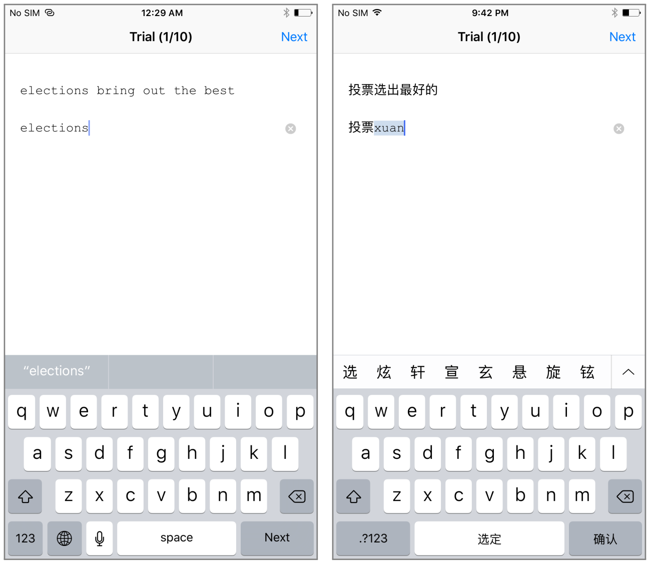

Figure 1. (a) Keyboard input interface with English QWERTY keyboard and (b) Pinyin QWERTY keyboard.

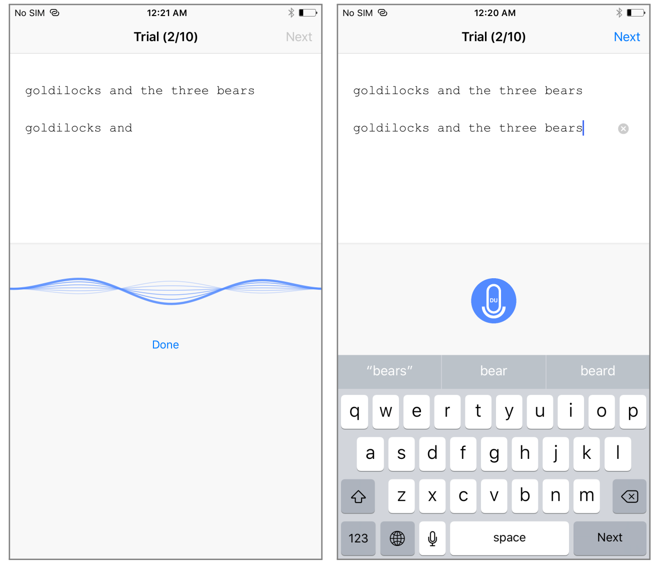

Figure 2. (a) Speech input interface: user is speaking and (b) user finishes speaking and a keyboard pops up for editing the initial spoken transcription.

keyboard or a Mandarin Pinyin QWERTY keyboard on an Apple iPhone. Participants ranged in age from 19 to 32 years old. Half of the participants performed a text transcription task in their native language using the keyboard input method followed by the speech input method, and the other half transcribed phrases with the input methods reversed. Eight of the 16 English participants were females and eight were males, and the same ratio held for Mandarin participants. The experiment was conducted under the supervision of the first author. The study task for each participant took about 30 minutes and each participant received $10 as compensation for their time. To minimize noise, we performed the study in a quiet meeting room.

Apparatus

We developed the experiment test-bed app with Swift 2 and Xcode 7 for iOS and connected it to a state-of-the-art speech recognition system, Baidu Deep Speech 2 [1]. The speech recognition system runs entirely on a server. As we were connected to Stanford University's high-speed network, there was no noticeable latency between the client iPhone and the speech server.

Our app allowed the user to perform transcription tasks using two input interfaces: keyboard or speech. Figure 1 shows the keyboard input interfaces with English QWERTY and Pinyin QWERTY keyboard respectively.

Figure 2 corresponds to the two modes of our custom speech input method: speech recognition mode and keyboard mode optionally used for error correction. The app is in speech recognition mode at the launch of each transcription task. The speech recognition system is on and the keyboard is hidden. The speech system recognizes the user's utterance and displays it in the input textbox. To indicate that he or she is finished speaking, the user touches the "Done" button or touches anywhere on the screen, which switches the app to error correction mode. In our error correction design, the user could either touch the mic button to turn on the speech system again or correct errors using the keyboard. These designs were inspired by the user interface built into Apple's iOS.

Although we chose to use the fastest typing-based keyboard we could find, we did not use advanced non-typing input methods such as stroke keyboards like Swype [9] or ShapeWriter [12] because such keyboards are not on every phone and require initial practice to reach proficiency. All participants used the same iPhone 6 Plus which had the experiment app installed.

Since QWERTY is the most common keyboard layout, we used it as the default for both English and Mandarin speaking participants to increase ecological validity. We used Pinyin QWERTY as the default Mandarin keyboard as shown in Figure 1(b). With a Mandarin Pinyin QWERTY keyboard, users input Mandarin characters by entering the Pinyin (phonetic transcriptions) of a Mandarin character, which triggers the presentation of a list of possible Mandarin Chinese characters matching the phonetic sound. Pinyins are composed of the same 26 English letters displayed in QWERTY layout. This is one of the most common ways Mandarin Chinese text is entered on smartphones and computers. Since the speech system already eliminates all invalid words before presenting a phrase, it was reasonable to allow autocorrect in the keyboard typing condition. The Mandarin Pinyin keyboard always outputs valid Mandarin characters which can be regarded as implicit built-in autocorrect and spell check features. Chinese keyboards also have a built-in prediction feature that can provide the user with a collection of possible characters based on their previous input. Therefore, to remain consistent across languages and text entry methods, we enabled the standard iOS text entry features which are spell check, autocorrect, and word completion for both languages and both methods.

Procedure

We had two equal phrase sets, A and B. Participants entered a series of text entry phrases drawn from one of the two phrase sets. Each participant entered text in only their first language, English or Mandarin Chinese, and the set of phrases used were the same in each language (translated, of course). Each phrase was regarded as one text entry trial. For each text entry method (speech or keyboard), participants completed 10 practice trials before beginning the test, which consisted of 50 trials. Participants were taught how to use each interface before the study and during practice. Actions and timestamps were logged in the background, and no timing information was ever visible to the participant. The app was set up so that the suitable input method, language, and phrase set were selected prior to handing the phone over to the user. After entering all the phrases with both input methods, the participant filled out a questionnaire regarding their demographic information and their opinions on the two text entry methods.

The study was fully counterbalanced so that for each language, one quarter of the participants (4 participants) started using the keyboard with phrase set A followed by speech with phrase set B, one quarter (4 participants) did the keyboard with phrase set B followed by speech with phrase set A, the other two groups did the speech before keyboard and the phrase set was likewise altered. Gender was balanced across all the conditions (text entry method order and phrase set) in our study.

Data Logging

Our app automatically logged all pertinent user behaviors during the experiment. In addition, we logged timestamps with these actions so as to have a record of the time at which the action occurred. During the study, a participant's actions fell into one of the following five categories. (The fifth item pertains only to the speech entry method.)

Insert. The user can insert a character using the keyboard or the speech recognition system.

Delete. The user can delete a single character using backspace, or multiple characters by selecting them first and backspacing, or simply delete everything by pressing the × button displayed at the end of the text box.

Auto-Correct. Auto-correct happens when a partial word or an existing word is replaced by a word suggested by the keyboard dictionary. The user presses the spacebar (i.e., continues typing) to confirm the auto-correct.

Complete. The user can insert multiple characters using word completion feature. Word completion happens when the user selects a word from the suggested word list. This can happen when the user in the beginning or in the middle of typing a word.

Speech. The speech system is turned on for the speech input method, but not for the keyboard method. A speech session means a sequence of the following actions in order: user presses the mic button, server starts to respond, user starts to speak, user stops speaking, user starts to speak, user stops speaking, . . . , user presses the done button, and server finishes responding. We are able to timestamp each of these actions on the client application.

During a trial in the speech condition, the user starts entering text by speaking the presented string, after which they can correct it using either their voice or the keyboard. Hence, multiple speech sessions may be recorded for a single trial.

Text Entry Phrase Set

We randomly selected 120 English phrases from a standard phrase set of 500 phrases [18] to create our text entry phrase set. The advantage of using these phrases was that they have moderate length and are representative of everyday English. These 120 phrases were randomly divided into two equal sets, A and B, which were used for the two input method conditions to prevent learning effects. Both A and B used the first 10 sentences as the practice phrase set and the remaining 50 as the test phrase set.

We also manually translated the 120 phrases to Mandarin Chinese and used them as our phrases for the Mandarin part of the study. The phrases in the Mandarin set had a one to one correspondence to the translations in the English set. They are exemplary of everyday Mandarin as well. The lengths of the English phrases varied from 17 to 39 characters ($M = 28.3$, $SD = 4.45$) and the lengths of Mandarin phrases ranged from 3 to 15 ($M = 8.1$, $SD = 2.43$) characters.

We excluded punctuation marks and capital letters, except for the single word "I". Automatically generating accurate punctuation marks with speech input is not regarded as a completely fair measure [4].

Measures

We present and discuss the following novel empirical measures of text entry performance [33]. We use T_i to denote the i-th transcribed string, P_i the i-th presented string, and S_i the i-th phrase returned by the speech system prior to any edits made by the user.

Words per Minute

Words per minute is the most commonly used measure for entry rates. The formal definition is given as follows:

$$WPM = \frac{|T_i| - 1}{t_i} \times 60 \times \frac{1}{L}$$

where t_i is time in seconds for the i-th trial. For the keyboard condition, it is measured from the entry of the first character to the entry of the last. For the speech condition, it is computed from the onset of the utterance of the first phoneme to the last edit made by the user. English words by convention are treated as having five characters [35], so we replace L with 5. For Mandarin Chinese, L was calculated as 1.5 since this is the average word length in Mandarin according to general statistics on the Chinese language [5].

Since we want to analyze not only the transcribed string but also what happens during the user's input, we need to define the input stream and classify characters in an input stream [34].

The input stream can be logged as a sequence of strings. Table 1 shows the input stream of a trial with the presented string

Current Input Stream	*Explanation*	*Action*	*Type*	*Value*
wear did I live my glasses	Insert an initial string using speech	Insert	Speech	“wear did I live my glasses”
	Delete the entire string using ×	Delete	Keyboard	“wear did I live my glasses”
wear did I leave my glasses	Insert a sentence using speech	Insert	Speech	“wear did I leave my glasses”
weardid I leave my glasses	Delete a space	Delete	Keyboard	“ ”
weadid I leave my glasses	Delete “r”	Delete	Keyboard	“r”
wedid I leave my glasses	Delete “a”	Delete	Keyboard	“a”
wdid I leave my glasses	Delete “e”	Delete	Keyboard	“e”
whdid I leave my glasses	Insert “h”	Insert	Keyboard	“h”
whedid I leave my glasses	Insert “e”	Insert	Keyboard	“e”
where did I leave my glasses	Select “where” from the predictive	Insert	Keyboard	“where”

Table 1. An example input stream when a user transcribed “where did I leave my glasses” using speech input with error corrections.

“where did I leave my glasses” with the speech condition. The first string is output by the speech system and the last one is also the transcribed string. By comparing two consecutive strings, we can see the user is making different types of changes (corrections): deleting the entire sentence using the × button, inserting a string using the speech system, deleting characters, or inserting characters. The actual actions that happened in the input stream are listed in the right column of the table. As we can see, these actions are either insertions or deletions. (Replacement can be treated as a deletion followed by an insertion). Note that users can insert or delete a single character or a whole string.

Error Rates

Text entry errors come in three forms: uncorrected errors, which remain in the transcribed string; corrected errors, which are fixed (e.g., backspaced) during entry; and total errors, which are the combination of the two [30]. Correspondingly, we can have uncorrected, corrected, and total error rates, which are normalized $[0,1]$. To compute error rates, we need to classify each character in the input stream, including backspaces, into one of four character classes: *Correct*, *Incorrect-not-fixed*, *Incorrect-fixed*, and *Fixes* [30].

Correct (C). All correct characters in the transcribed text. The size of the class is computed as $MAX(T,P) - MSD(T,P)$, where MSD is the *minimum string distance* (also called the *edit distance*) between two given strings [15, 29, 30, 32].

Incorrect-not-fixed (INF). All incorrect characters in the transcribed text. The size of the class is computed simply as $MSD(T,P)$ [30].

Incorrect-fixed (IF). All characters deleted during entry. The size of the class is computed as the sum of lengths of the value of all deletions.

Fixes (F). All delete actions. Examples are deleting a single character with backspace or deleting the entire sentence with the × button at the end of the text box.

With this classification, we are then able to compute the following measures for error rates [30, 33] for both the keyboard and the speech condition:

$$\text{uncorrected error rate} = \frac{INF}{C + INF + F}$$

$$\text{corrected error rate} = \frac{IF}{C + INF + F}$$

$$\text{total error rate} = \frac{INF + IF}{C + INF + F}$$

In addition, we can use the formula

$$\text{utilized bandwidth} = \frac{C}{C + INF + F + IF}$$

to characterize the efficiency of an input method.

Initial Speech Transcription Words per Minute

Previously we computed the entry rate using the transcribed string. Now we calculate it with the the initial speech transcription (IST) generated by the speech system to estimate the rate of the speech recognition system.

$$ISTWPM = \frac{|S_i| - 1}{t_i'} \times 60 \times \frac{1}{L}$$

where t_i' is defined as the time between the user presses the mic button and the server returns the last character and S_i is the initial string returned by the speech recognition system.

Initial Speech Transcription Error Rate

Instead of comparing the presented string to the transcribed one, we compare it against the initial output from speech recognition. In this way, we can have a general sense of how accurate speech recognition is from the outset before any error corrections occur. We calculated the four classes using the same formula presented earlier, however, IF and F are always 0 because no error correction mechanism is considered for speech-only measures.

Other Speech-Specific Measures

In addition, we present speech-specific measures which evaluate the general performance of the speech input method in terms of efficiency and effectiveness. These measures are intended to answer the following questions.

- How responsive is the server: $\frac{\text{user talking time}}{\text{server process time}}$

- How responsive is speech input: $\frac{\text{user talking time}}{\text{speech session time}}$
- How much delay is due to the user: $\frac{\text{user delay time}}{\text{speech session time}}$
- How much delay is due to the system: $\frac{\text{system delay time}}{\text{speech session time}}$
- What percentage of time is spent on the speech system: $\frac{\text{speech session time}}{\text{trial time}}$
- What percentage of time (characters) is related to the keyboard: $\frac{\text{keyboard time (characters)}}{\text{trial time (characters)}}$
- What percentage of time is spent in inputting an initial sentence using the speech system: $\frac{\text{first speech session time}}{\text{trial time}}$
- What percentage of time (characters) is used to make corrections: $\frac{\text{correction time (characters)}}{\text{trial time (characters)}}$:
- What percentage of correction time (characters) is related to the keyboard: $\frac{\text{correction using keyboard time (characters)}}{\text{correction time (characters)}}$
- What percentage of correction time (characters) is related to the speech system: $\frac{\text{correction using speech time (characters)}}{\text{correction time (characters)}}$

Subjective Measures

In addition to the performance measures, we asked the user to evaluate the demand of the tasks using the NASA TLX Likert 7-point scale across six categories. Users were also interviewed to gain an understanding of participants' qualitative experience using each method.

Design & Analysis

The experiment was a 2×2 mixed design. The input method factor was a within-subjects factor and the language factor was a between-subjects factor. Each experiment was divided into two sessions: speech and keyboard. Each session consisted of training and test trials. One of the two phrase sets (A or B) was assigned in each session period (either speech or keyboard) in alternating order from experiment to experiment. The order of the sessions (speech or keyboard) was also balanced between participants to reduce interaction effects.

Every keystroke was recorded together with a timestamp. For Pinyin QWERTY keyboard, we recorded both letters (Pinyin) and resulting Mandarin characters. The information was logged as a JSON file during the study. We wrote a parser to parse the auto-generated log file and a analyzer to calculate a series of text entry related measures such as entry rates and error rates. We ran appropriate statistical tests on the processed data to analyze results and obtain insights.

RESULTS

There are 32 log files and each contains 50 trials for the speech input method and 50 trials for the keyboard input method. Hence, we have 32 data points from the study. Our analysis of variance shows that neither gender nor method order exhibited a main effect for any measure, so we present our results as a function of input methods and languages.

Speed

We ran a log likelihood ratio test on linear mixed effect models with maximum likelihood estimator to examine a main effect of input methods (speech and keyboard). Results show that input method is a main effect for both languages: $\chi^2(1, N = 16) = 182.47, p < 0.001$ for English and $\chi^2(1, N = 16) = 160.24, p < 0.001$ for Mandarin. There is also a significant interaction between languages and input methods, $\chi^2(1, N = 32) = 38.20, p < 0.001$. This can be revealed from a box plot in Figure 3. In both languages, the initial speech transcription (IST) gives the highest entry rate, followed by speech input (including keyboard-based corrections), and keyboard input is the slowest one. Mandarin in general has higher entry rates than English across all the three conditions. However, the English entry rate and the Mandarin one are not directly comparable because average word lengths in two languages are inherently different.

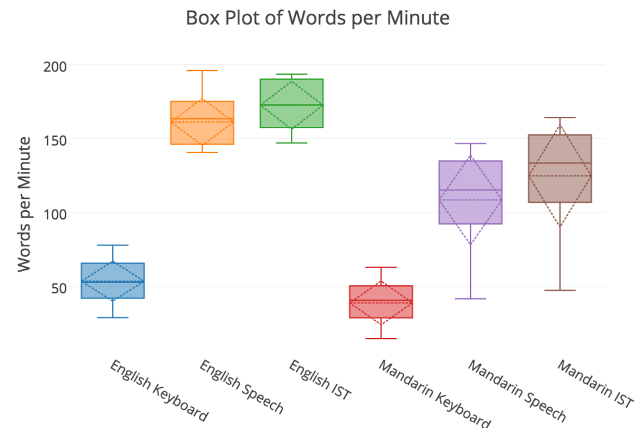

Figure 3. Words per minute as a function of language and input method.

We present means and corresponding standard deviations (in parentheses) of six measures in Table 2. The average entry rate of speech input is 3.01 times faster than keyboard input for English, and 2.79 times faster than Mandarin Pinyin input. This indicates that all users could experience a 2.8 times speedup using the speech input method and English users can accelerate more with the speech input method. Excluding time spent on error corrections, the speed of the initial speech transcription can be further improved: IST is 3.24 times faster than keyboard for English and 3.21 times faster for Mandarin. The ratio of average IST speed to average speech input speed is 1.07 for English and 1.15 for Chinese, which characterizes how much entry rate speed (in WPM) error correction of the IST is costing users. If the IST were always perfect, then IST would equal the speech entry rate and the ratio would be 1. We also observe a larger standard deviation in speech input compared to keyboard input for both languages (16.37 vs. 13.97 WPM for English and 31.31 vs. 15.26 WPM for Mandarin). This suggests that different users have different levels of familiarity with the speech input method. Hence, slow users may be able to further improve their speed after a period of practice.

	English			Mandarin Chinese		
Input Method	*Keyboard*	*Speech*	*IST*	*Keyboard*	*Speech*	*IST*
Words per Minute	53.46 (13.97)	161.20 (16.37)	172.54 (16.75)	38.78 (15.26)	108.43 (31.31)	124.74 (35.85)
Uncorrected Error Rate	0.19% (0.0017)	0.35% (0.0034)	3.14% (0.017)	1.40% (0.0094)	1.69% (0.011)	9.19%(0.048)
Corrected Error Rate	3.49% (0.018)	2.58% (0.014)	0.00% (0.00)	19.14% (0.098)	5.80% (0.032)	0.00% (0.00)
Total Error Rate	3.68% (0.018)	2.93% (0.015)	3.14% (0.017)	20.54 (0.094)	7.51% (0.029)	9.19% (0.048)
Utilized Bandwidth	93.84% (0.030)	96.26% (0.018)	96.86% (0.017)	73.34% (0.12)	90.30% (0.036)	90.81% (0.048)
Auto Correct Count	26.06 (20.87)	0.50 (1.51)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)

Table 2. Mean and corresponding standard deviation (in parentheses) of six measures for the two input methods and the initial speech transcription before error correction, for both languages.

Error Rates

A log likelihood ratio test on linear mixed effect models was ran on three types of error rates: uncorrected error rates, corrected error rates, and total error rates. Uncorrected error rates characterize the portion of errors that are not rectified by users, such as omission, extra insertions, and incorrect input. Results reveal that for uncorrected error rates, input method exerts a significant effect in English: $\chi^2(1, N=16) = 4.28, p < 0.05$ but not in Mandarin: $\chi^2(1, N=16) = 0.58, p = 0.45$ for Mandarin. This indicates that while English users tended to make more uncorrected errors using speech input, Mandarin users made comparable performance when rectifying errors using either input method.

In contrast, we found a reliable difference in input methods for both languages for corrected error rate, which refers to errors in the input stream but not in the final transcribed string. $\chi^2(1, N=16) = 4.64, p < 0.05$ for English and the effect is even more evident in magnitude for Mandarin, $\chi^2(1, N=16) = 78.01, p < 0.001$. This indicates that users spent significantly more time and effort editing the text when using the keyboard input method than the speech input method. The difference is more pronounced in Mandarin than in English (5.80% vs. 19.14% in Mandarin and 2.93% vs. 3.68% in English). One possible reason for this is that users have to press multiple keystrokes to enter a Pinyin (phonetic transcriptions) for a Mandarin character while only a single keystroke is needed for an English character.

We expect a main effect in the total error rate since it is simply the sum of the uncorrected error rate and the corrected error rate. Indeed, we see that $\chi^2(1, N=16) = 67.53, p < 0.001$ for Mandarin. The effect demonstrates that Mandarin users made significantly fewer errors using the speech input method than the keyboard input method. The difference is less evident in English: $\chi^2(1, N=16) = 2.77, p = 0.096$. In general, the total error rate of speech input is 63.4% lower than that of keyboard input for Mandarin and 20.4% lower for English, as shown in Table 2. Even if we exclude the error correction part, the initial speech transcription could achieve an error rate 55.3% lower than that of keyboard for Mandarin and 14.7% lower for English. We present a box plot for total error rates in Figure 4. We can see that keyboard input also results in greater standard deviations than speech input (0.018 vs. 0.015 for English and 0.094 vs. 0.029 for Mandarin). This may indicate that novice users tend to make more errors than experts using keyboard input. On the contrary, speech input does not require much expertise or training and both novice and experts perform similarly, as illustrated by a much smaller standard deviation.

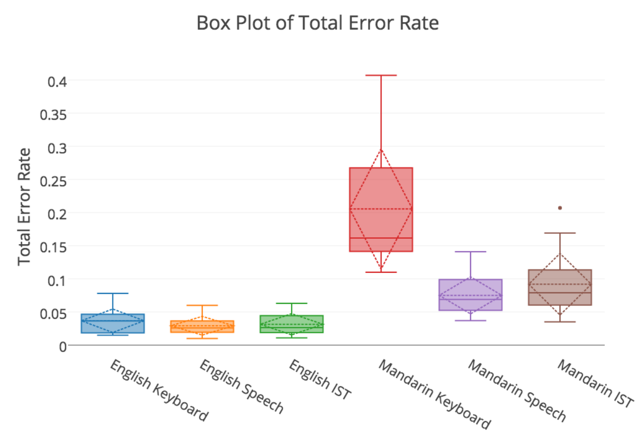

Figure 4. Total error rate as a function of language and input method.

Utilized bandwidth is a measure that characterizes the proportion of keystrokes that correspond to correct parts of the final string compared with the presented string. Our results reveal a reliable difference in this measure: $\chi^2(1, N=16) = 15.72, p < 0.001$ for English and $\chi^2(1, N=16) = 73.45, p < 0.001$ for Mandarin. A box plot is shown in Figure 5.

Finally, we present the total occurrences of auto corrects and word completion over 50 trials. As seen in Table 2, occurrence of these features was significantly higher for the keyboard than for speech in English, $\chi^2(1, N=16) = 111.43, p < 0.001$. Evidently, many more autocorrects occurred when participants used the keyboard input method (52.12%) than using the speech input method (1.00%). These features were not supported for the Pinyin QWERTY keyboard.

Speech-Specific Results

We present the following speech-specific measures to assess the performance of the speech input method in many respects. We first present Figure 6, which shows the time proportion the user spent on different tasks when entering text. As can be seen, only slightly more than one third of the total time

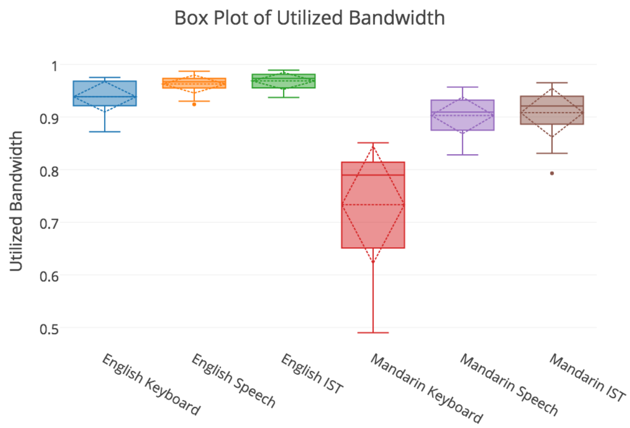

Figure 5. Utilized bandwidth as a function of language and input method.

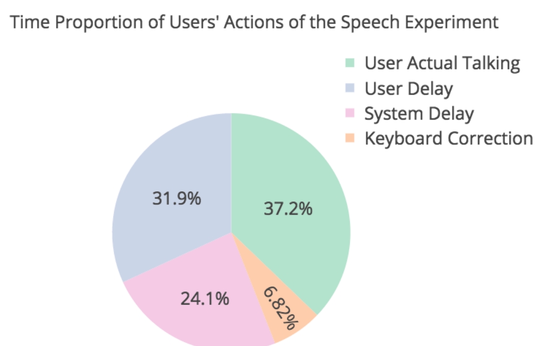

Figure 6. Time proportion of different users' actions with the speech input method.

(37.18%) is needed for users to actually speak a phrase. This can be explained by the short lengths of the presented phrases. 31.85% of time is wasted due to users' delay. For example, many users did not start to speak until one or two seconds after the speech system was turned on. Also, a number of users hesitated to press the done button even after finishing speaking, and they explained that this was because they were not sure if they wanted to talk more. This could lead to a substantial delay since the speech recognition system is context based [1], i.e., it does not finalize the output until the user clearly indicates he or she is finished speaking. All these waiting activities constituted a considerable user delay. This also suggests that as a user gains experience with speech input, they can significantly reduce the user delay, thus allowing speech to gain a further speed advantage. In contrast, less delay was due to the speech recognition system (24.14%). The delay mainly occurred when the system was still processing the user's utterance after the done button was pressed.

Figure 7 shows the time the user spent on speech input and keyboard input (error correction mode) in completing tasks using the speech input method. Users spent substantial time (91.93%) in producing the initial transcription using the speech

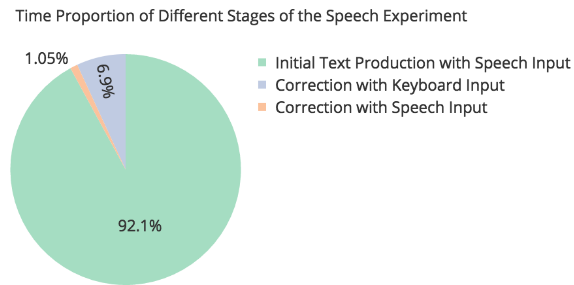

Figure 7. Time proportion of different stages with the speech input method.

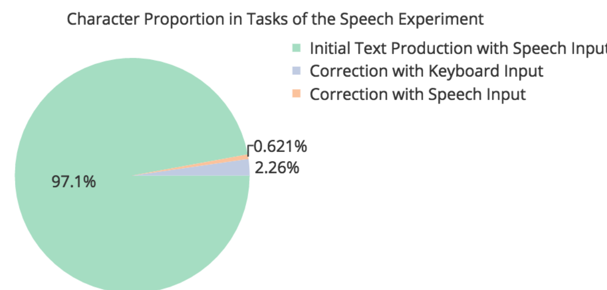

Figure 8. Character proportion in tasks with the speech input method.

recognition system. The user used the rest of the time (8.07%) to correct errors, and 86.5% of the time the user corrected errors using keyboard and only 13.5% of the time speech input was used. We can also view the same splits in terms of number of characters rather than time. As can be seen from Figure 8, 97.12% of the final characters were generated from the initial speech input directly. 2.26% and 0.62% of the final characters resulted from corrections using keyboard and speech, respectively. This suggests that users' confidence in the speech system's initial result was high enough that they barely resorted to the correction mode to modify the initial input. Furthermore, the user preferred to correct errors using the keyboard instead of speech most of the time. Our post-experiment interviews indicate that users found it more efficient and comfortable to correct errors with the keyboard.

NASA TLX Ratings

Participants' subjective ratings on the demand and performance of the two input methods are summarized in Figure 9 and 10. The lower the numbers are, the less demanding or the better performance the user found the method. As illustrated by Figure 9, participants found for both languages, the keyboard easier to correct errors with but speech easier to produce text.

On the NASA TLX 1-7 scale shown in Figure 10, English speech was ranked as the easiest input method among all four options across all six categories, followed by English keyboard, then Mandarin speech, and lastly Mandarin keyboard. Speech was rated better than keyboard in all six categories for both English and Mandarin. This indicates that speech input is more

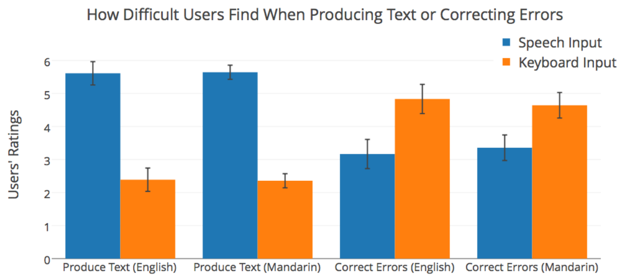

Figure 9. Participants' subjective evaluation of difficulty.

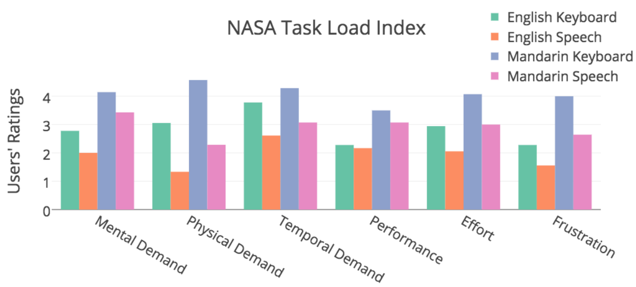

Figure 10. Participants' ratings on the NASA TLX Index.

desired than keyboard input and Mandarin input methods are generally harder to use than English ones.

Qualitative Responses

We also interviewed all participants for their opinions of the two input methods. They expressed that speech methods were generally easier to grasp and use, less demanding, and more accurate than their original expectation of speech recognition.

Most users expressed they liked the coherence and continuity of the speech system compared to keyboard input: "it is difficult to type an entire sentence using keyboard at a time, since a typo at the beginning of the sentence could be very hard to correct later." Users' attitudes towards keyboard input were that it was more comfortable to use but more error-prone: "I am comfortable typing on a keyboard. There was less uncertainty about what text was going to be produced. i.e. I could correct errors as they were made. That being said, it seemed like I introduced more errors typing than the speech system did." Also they pointed out speech could help address the problem of rough spelling: "compared to the speech method, I also had to worry about the rough spelling of the word which took more effort."

Despite all these advantages of speech input, users also indicated some deficiencies: "when I made mistakes it seemed like it took longer to correct, because I was switching from holding the phone to speak vs. holding to type."

Our Mandarin participants also provided some interesting insights with regards to Mandarin Chinese input. Some stated that the Mandarin Pinyin keyboard was inherently ambiguous because one Pinyin can correspond to many characters and selecting the right one took efforts. Some commented that they liked the flexibility and the ability to recognize accents with the Mandarin speech recognition system, especially when they were not sure about the exact Pinyin (phonetic sounds without any accent), which is usually required to produce the right characters. There were also challenges specific to Mandarin recognition: "some phrases in Mandarin sound the same, so if the speech recognition software does not have the context, it is very hard for it to guess the correct output."

DISCUSSION

Quantitative Analysis

Our results revealed that the speech input method was not only faster (3.0 times higher in English and 2.8 times higher in Mandarin) but also less error-prone (21.6% lower in English and 64.3% lower in Mandarin) than keyboard input. The speed of speech recognition for Mandarin was slightly slower than that for English, but the accuracy was substantially higher for Mandarin than English.

The low total error rate of the speech input method is explained by its low corrected error rate. We found that the Deep Speech 2 engine's speech transcription could already achieve an error rate 56.2% lower than the Mandarin Pinyin QWERTY keyboard and 16.2% lower than English QWERTY keyboard. Therefore, users only need to make few corrections on the input stream when using speech input. The 97.08% recognition rate for English and 92.49% recognition rate for Mandarin were both rather remarkable, especially considering that even humans do not have perfect speech recognition [16].

Qualitative Analysis

Users' subjective ratings and testimonies revealed their preference for the speech input method over the keyboard one. In general, speech is more natural, smooth, and capable of recognizing most of the words immediately. Moreover, it does not require much practice for novice users. Participants in our study got quite accustomed to it only after entering 10 phrases, and this made their confidence grow as they completed the main task. As a result, this reduced their mental and physical effort by a great amount.

Potential for Improvement

The high speed of speech input was achieved due to the responsiveness of the speech recognition system, as demonstrated by our results that system delay constituted only 24.4% of the entire experimental time. The user delay was 31.85% of the experimental time. The speech input speed might further improve if users reduce this delay through gaining experience with the system, or if we can create a design which helps even novice users reduce this delay. For example, we may be able to create a new interface which encourages users to signal the speech system as soon as they are finished speaking. Rather than having the user manually specify when they are done talking, we also think auto-detecting the end of speech holds significant promise. It also relieves the user of another manual task.

Another improvement we suggest would be an incorporation of customizable system features to the speech recognition system. During 50 trials, autocorrects and word completion occurred in about half of the trials (26.06 times) in the keyboard condition but barely appeared (0.50 times) in the English speech condition, as shown in Table 2. We could envision that a speech system with support of system features would be more powerful and desirable. For example, one design could be a system that not only gives the most likely words but also suggests a list of other possible words based on computed probabilities so that the user could select the right word from the list instead of typing it out. Moreover, the speech recognition system could analyze and learn from all the word selections made by the user. With the data collected, the system can keep updating the underneath machine learning model to make more accurate and customized predictions for each user.

Limitations and Future Work

Our research only focused on the situation where users were sitting in a relatively quiet environment talking on the phone. In reality, people need to use the speech functionality in a variety of ways and environments. A study on input speech under a variety of noise conditions would be interesting. It would also be interesting to explore the application of the speech recognition technique in other areas of mobile computing. With a more fully developed speech recognition technique, we may be able to design effective transcription applications and email composing tools, to name a few.

There have been other keyboards that have been developed as alternatives to the standard built-in iOS (and Android) keyboards, including swiping keyboards, and specialized Pinyin keyboards such as Baidu IME. These keyboards require some non-trivial training, unlike the speech interface (where we allowed only 10 sentences of training). Research showed that there was no significant difference in entry rate between Swype and QWERTY, but Swype was rated higher in subjective rankings [19]. Another study suggests that with about 13-19 hours of practice, users' typing performance using KALQ can be 34% more efficient than typing on split-screen QWERTY layouts [20]. A detailed comparison to these keyboards, especially in the regime of highly trained users, would also be interesting.

CONCLUSION

In this research, we designed and analyzed a speech input method for mobile devices in English and Mandarin and demonstrated its potency against keyboard input through a user study. We provide suggestions on how to further optimize speech input using the results obtained from the study. This work contributes the first empirical study demonstrating the practicality of speech input over keyboard input on mobile phones. Our work provides novel measures to evaluate the performance of speech-based text entry methods and potentially opens a new door for the study of the speech input method in mobile computing. We hope it will attract more researchers to develop effective speech-based mobile applications in the near future.

ACKNOWLEDGEMENT

We are grateful to He Dang of Baidu's Speech Technology group, who had performed a preliminary study that inspired this research project.

REFERENCES

1. Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Erich Elsen, Jesse Engel, Linxi Fan, Christopher Fougner, Tony Han, Awni Y. Hannun, Billy Jun, Patrick LeGresley, Libby Lin, Sharan Narang, Andrew Y. Ng, Sherjil Ozair, Ryan Prenger, Jonathan Raiman, Sanjeev Satheesh, David Seetapun, Shubho Sengupta, Yi Wang, Zhiqian Wang, Chong Wang, Bo Xiao, Dani Yogatama, Jun Zhan, and Zhenyao Zhu. 2015. Deep Speech 2: End-to-End Speech Recognition in English and Mandarin. *CoRR* abs/1512.02595 (2015). http://arxiv.org/abs/1512.02595

2. James H. Bradford. 1995. The Human Factors of Speech-based Interfaces: A Research Agenda. *SIGCHI Bull.* 27, 2 (April 1995), 61–67. DOI: http://dx.doi.org/10.1145/202511.202527

3. William Buxton, Mark Billinghurst, Yves Guiard, Abigail Sellen, and Shumin Zhai. 2011. *Human Input to Computer Systems: Theories, Techniques and Technology*. Technical Report. http://www.billbuxton.com/inputManuscript.html

4. C Julian Chen. 1999. Speech recognition with automatic punctuation. In *EUROSPEECH*.

5. Lee-Feng Chien and Hsiao-Tieh Pu. 1996. Important Issues on Chinese Information Retrieval. *Computational Linguistics and Chinese Language Processing* 1, 1 (Aug. 1996), 205–221.

6. Edward Clarkson, James Clawson, Kent Lyons, and Thad Starner. 2005. An Empirical Study of Typing Rates on mini-QWERTY Keyboards. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems (CHI EA '05)*. ACM, New York, NY, USA, 1288–1291. DOI: http://dx.doi.org/10.1145/1056808.1056898

7. James Clawson, Kent Lyons, Alex Rudnick, Robert A. Iannucci, Jr., and Thad Starner. 2008. Automatic Whiteout++: Correcting mini-QWERTY Typing Errors Using Keypress Timing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. ACM, New York, NY, USA, 573–582. DOI: http://dx.doi.org/10.1145/1357054.1357147

8. Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingsbury. 2012. Deep Neural Networks for Acoustic Modeling in Speech Recognition. *Signal Processing Magazine* (2012).

9. Swype Inc. 2002. Swype | Type Fast, Swype Faster. (2002). http://www.swype.com/

10. Clare-Marie Karat, Christine Halverson, Daniel Horn, and John Karat. 1999. Patterns of Entry and Correction in Large Vocabulary Continuous Speech Recognition Systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99)*. ACM, New York, NY, USA, 568–575. DOI: http://dx.doi.org/10.1145/302979.303160

11. Heidi Horstmann Koester. 2003. Abandonment of speech recognition by new users. In *Proc. RESNA*, Vol. 3.

12. Per-Ola Kristensson and Shumin Zhai. 2004. SHARK2: A Large Vocabulary Shorthand Writing System for Pen-based Computers. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology (UIST '04)*. ACM, New York, NY, USA, 43–52. DOI:http://dx.doi.org/10.1145/1029632.1029640

13. Accredited Language. 2016. The 10 Most Common Languages. (May 2016). https://www.alsintl.com/blog/most-common-languages/

14. Kai-Fu Lee and Raj Reddy. 1988. *Automatic Speech Recognition: The Development of the Sphinx Recognition System*. Kluwer Academic Publishers, Norwell, MA, USA.

15. V. I. Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady* 10 (Feb. 1966), 707.

16. John S. Logan, Beth G. Greene, and David B. Pisoni. 1989. Segmental intelligibility of synthetic speech produced by rule. *J Acoust Soc Am* 86, 2 (Aug 1989), 566–581. http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3507386/ 2527884[pmid].

17. I. Scott MacKenzie and R. William Soukoreff. 2002. Text Entry for Mobile Computing: Models and Methods,Theory and Practice. *Human-Computer Interaction* 17, 2-3 (2002), 147–198. DOI: http://dx.doi.org/10.1080/07370024.2002.9667313

18. 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. DOI:http://dx.doi.org/10.1145/765891.765971

19. Harry Nguyen and Michael C Bartha. 2012. Shape Writing on Tablets: Better Performance or Better Experience?. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 56. SAGE Publications, 1591–1593.

20. Antti Oulasvirta, Anna Reichel, Wenbin Li, Yan Zhang, Myroslav Bachynskyi, Keith Vertanen, and Per Ola Kristensson. 2013. Improving Two-thumb Text Entry on Touchscreen Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 2765–2774. DOI: http://dx.doi.org/10.1145/2470654.2481383

21. Kathleen J. Price, Min Lin, Jinjuan Feng, Rich Goldman, Andrew Sears, and Julie A. Jacko. 2004. *Data Entry on the Move: An Examination of Nomadic Speech-Based Text Entry*. Springer Berlin Heidelberg, Berlin, Heidelberg, 460–471. DOI: http://dx.doi.org/10.1007/978-3-540-30111-0_40

22. David A Rosenbaum. 2009. *Human motor control*. Academic press.

23. Ronald Rosenfeld, Dan Olsen, and Alex Rudnicky. 2001. Universal Speech Interfaces. *interactions* 8, 6 (Oct. 2001), 34–44. DOI:http://dx.doi.org/10.1145/384076.384085

24. Hasim Sak, Andrew W Senior, and Françoise Beaufays. 2014. Long short-term memory recurrent neural network architectures for large scale acoustic modeling.. In *INTERSPEECH*. 338–342.

25. Nitin Sawhney and Chris Schmandt. 2000. Nomadic Radio: Speech and Audio Interaction for Contextual Messaging in Nomadic Environments. *ACM Trans. Comput.-Hum. Interact.* 7, 3 (Sept. 2000), 353–383. DOI: http://dx.doi.org/10.1145/355324.355327

26. Andrew Sears and Ying Zha. 2003. Data entry for mobile devices using soft keyboards: Understanding the effects of keyboard size and user tasks. *International Journal of Human-Computer Interaction* 16, 2 (2003), 163–184.

27. Ben Shneiderman. 2000. The Limits of Speech Recognition. *Commun. ACM* 43, 9 (Sept. 2000), 63–65. DOI:http://dx.doi.org/10.1145/348941.348990

28. Aaron Smith. 2015. U.S. Smartphone Use in 2015. (2015). http://www.pewinternet.org/2015/04/01/us-smartphone-use-in-2015/

29. R. William Soukoreff and I. Scott MacKenzie. 2001. Measuring Errors in Text Entry Tasks: An Application of the Levenshtein String Distance Statistic. In *CHI '01 Extended Abstracts on Human Factors in Computing Systems (CHI EA '01)*. ACM, New York, NY, USA, 319–320. DOI:http://dx.doi.org/10.1145/634067.634256

30. R. William Soukoreff and I. Scott MacKenzie. 2003. Metrics for Text Entry Research: An Evaluation of MSD and KSPC, and a New Unified Error Metric. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*. ACM, New York, NY, USA, 113–120. DOI: http://dx.doi.org/10.1145/642611.642632

31. Keith Vertanen and Per Ola Kristensson. 2009. Parakeet: A Continuous Speech Recognition System for Mobile Touch-screen Devices. In *Proceedings of the 14th International Conference on Intelligent User Interfaces (IUI '09)*. ACM, New York, NY, USA, 237–246. DOI: http://dx.doi.org/10.1145/1502650.1502685

32. Robert A. Wagner and Michael J. Fischer. 1974. The String-to-String Correction Problem. *J. ACM* 21, 1 (Jan. 1974), 168–173. DOI: http://dx.doi.org/10.1145/321796.321811

33. Jacob O. Wobbrock. 2007. *Measures of Text Entry Performance*. Elsevier Inc., 47–74. DOI: http://dx.doi.org/10.1016/B978-012373591-1/50003-6

34. Jacob O. Wobbrock and Brad A. Myers. 2006. Analyzing the Input Stream for Character- Level Errors in Unconstrained Text Entry Evaluations. *ACM Trans. Comput.-Hum. Interact.* 13, 4 (Dec. 2006), 458–489. DOI:http://dx.doi.org/10.1145/1188816.1188819

35. H. YAMADA. 1980. A Historical Study of Typewriters and Typing Methods : from the Position of Planning Japanese Parallels. *Journal of Information Processing* 2, 4 (1980), 175–202. http://ci.nii.ac.jp/naid/110002673261/en/

36. Koji Yatani and Khai N. Truong. 2007. An Evaluation of Stylus-based Text Entry Methods on Handheld Devices in Stationary and Mobile Settings. In *Proceedings of the 9th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '07)*. ACM, New York, NY, USA, 487–494. DOI:http://dx.doi.org/10.1145/1377999.1378059

37. Jiahong Yuan, Mark Liberman, and Christopher Cieri. 2006. Towards an integrated understanding of speaking rate in conversation.. In *INTERSPEECH*.

38. Shumin Zhai and Per Ola Kristensson. 2012. The Word-gesture Keyboard: Reimagining Keyboard Interaction. *Commun. ACM* 55, 9 (Sept. 2012), 91–101. DOI:http://dx.doi.org/10.1145/2330667.2330689

39. Shumin Zhai, Per-Ola Kristensson, and Barton A. Smith. 2005. In search of effective text input interfaces for off the desktop computing. *Interacting with Computers* 17, 3 (2005), 229 – 250. DOI: http://dx.doi.org/10.1016/j.intcom.2003.12.007 Special Theme - Papers from Members of the Editorial Boards.

40. Lina Zhou, Yongmei Shi, Dongsong Zhang, and Andrew Sears. 2006. Discovering Cues to Error Detection in Speech Recognition Output: A User-Centered Approach. *J. Manage. Inf. Syst.* 22, 4 (April 2006), 237–270. DOI: http://dx.doi.org/10.2753/MIS0742-1222220409