New Data on Text Entry with Multitap and T9

Sajida Sajida

University of Tampere
School of Information Sciences
Interactive Technology
M.Sc. thesis
Supervisor: Poika Isokoski
June 2012

University of Tampere
School of Information Sciences
Interactive Technology
Sajida Sajida: New Data on Text Entry with Multitap and T9
M.Sc. thesis, 64 pages, 6 index + appendix pages
June 2012

Abstract

Mobile phone technology is advancing rapidly. Text entry on mobile devices is essential as more and more people are using mobile phones for the purpose of writing text messages. The need to make mobile devices as easy to use and as efficient as possible is more acute than ever. New methods of interaction are proposed but the old ones still stay as long as they provide users with some comfort and benefit which others lack.

I conducted an experiment to compare the two well-known, and till recently, most widely used text entry methods for mobile phones, the multitap and T9. Sixteen participants in two groups, experienced and novice, transcribed phrases, and the speed and error rate were measured.

Statistically significant effects of text entry method, experience, and their interaction were found for text entry rate. T9 users were faster than multitap users and experienced users faster than novice. No statistically significant effects were found for error rate. In comparison to earlier data on the performance of the multitap and T9, novice and experienced users of multitap and T9 performed better with overall 10.67 wpm vs 7.95 wpm for multitap group and 18.39 wpm vs 14.72 wpm for T9 group.

Table of Contents

Acknowledgments

First of all, I thank my thesis supervisor, Prof. Poika Isokoski, whose guidance and support was always available. Among other things, not only did he offer me a topic, but also the software to run the experiment. He also helped in recruiting the participants for my experiment.

I also thank all the participants who volunteered for the experiment and helped me towards the completion of my thesis.

Another thanks goes to my family, especially my parents, for making me what I am. Their support and constant guidance are what I needed most and I got most. I would not have been able to do anything without their tireless effort for everything they have done for me.

Last, but not least, I thank my partner for his love, help in working with software, and patience which made my work as easy as it could get.

1 Introduction

Mobile phones have gone through enormous changes during the last two decades, not only in shape, size and other aspects of their physical appearance but also in their function. They are no longer used only for making and receiving calls but also for sharing pictures and videos using MMS (Multimedia Messaging Service), playing games, browsing websites, taking pictures or making videos through the built-in digital camera, playing music, GPS navigation and most of all, for textual communication using SMS (Short Message Service).

Text entry on mobile phones started in the 1990's, and in 1995 on average only 0.4 messages per GSM customer per month were sent. The situation changed, however, with the average number of messages reaching 35 per user per month by the end of 2000, [Wikipedia, 2012], and in 2010, 6.1 trillion text messages were sent around the world [Itu.int, 2010], which makes almost 96 messages per user per month. The enormous increase in the amount of text messaging on mobile phones opened a new area of research and focused the attention of researchers on developing better text entry methods for mobile phones.

However, it was not easy to devise new methods of text entry on keypads small enough for mobile phones. This also gave rise to the need for inventing new and different types of keypads and key arrangements for mobile phones for the purpose of improved text entry, along with using speech and gestures for the same purpose. Despite advances in these new methods of text entry on mobile phones, physical touch-based, manual text entry techniques still dominate and are used on almost all mobile phones.

When developing new text entry methods, there is a need to identify which methods are better than others, hence requiring a comparison between the existing and newly developed methods. Without any data available about the efficiency of a particular method, it is hard to know if it is better, or good at all, to be adopted by users, as only experimental work can reveal many of the problems in user interfaces.

The most common and adopted keypad used for mobile phones is the 12-key keypad, and the most widely used methods of text entry are multitap and T9 (the dictionary-based disambiguation method from Tegic Communications, Inc). Although there have been several other methods in use, or at least developed, such as *LetterWise, TiltText and Twiddler Typing*, which have been compared and evaluated empirically with either or both of these methods, to our surprise, there has not been much of a comparison between the two dominant, well known, and widely used text entry methods themselves. The only study evaluating the two methods against each other was conducted by James *et al.* [2001]. It obtained the performance data for text entry on mobile phones using the two methods, and compared it with the performance predictions based on two different mathematical models namely “keystroke level model” and “movement and linguistic model”.

The result of the study concerning the performance data, however, showed some unexpected results. Two user groups were involved in the study, a group of novice and a group of experienced users. The mean overall text entry speed with the multitap method was unexpectedly lower for the expert group (7.93) than for the novice group (7.98). Moreover, the total number of errors committed by expert users using the multitap method was higher than that of the novice users, 116 and 65 respectively. These observations raise some doubts about the validity of the results which poses a need for further evaluation of the two methods between the two groups.

Another motivation for the work in this thesis was the fact that old multitap and T9 data may not represent modern use cases. Users have gained experience in the course of 11 years, and any comparison based on these old data can be misleading. In order to have a better benchmark data to compare to, conducting a new experiment was essential.

In this thesis, I report a comparative, empirical evaluation to obtain new performance data for multitap and T9, with 'experienced' and 'novice' users. The purpose of the study was to validate the old data and find new data for future comparisons.

In chapter 2, the focus is on the current text entry methods on mobile phones and different types of keypads and keyboards developed so far. Chapter 3 contains the comparative evaluation of the multitap and T9. It includes the experiment setup, procedure, data collection and the results obtained from the experiment. In chapter 4, the results are discussed against the previous studies. Chapter 5 contains a summary of the work done in this thesis with conclusions on the results obtained.

2 Background

During the last 15 years, the rapidly growing text entry activity on mobile phones has motivated researchers and developers to find new ways of text entry. The work has led to many new text entry methods and different keypad or keyboard layouts. When discussing the keypad layout and the text entry methods, it is hard to distinguish between the layout and text entry method as some of the text entry methods are developed particularly for a specific layout, and most of the time, they are discussed so that the terms seems to be interchangeable. However, in this thesis keyboard layout and text entry method are not synonyms.

A layout or a design is how the things are arranged. A mobile phone keypad layout includes the number of keys, placement of letters on the keys and other physical features of the keypad. A layout is tested and evaluated against another using any text entry method. A text entry method, however, is a technique or mechanism for entering text which might be pressing a key once or more to enter a letter, using a combination of keys, using dictionary and others. Multitap and T9 are examples of text entry methods that use the same keyboard layout but do so in very different ways. On the other hand, soft keyboard layouts are the examples of designs where text entry method and layout are synonyms.

2.1 Mobile Phone Keypad Layouts

Different mobile phone keypads have come into existence since the fast growth in use of SMS technology in mobile phones by consumers. These mobile phone keypad variants differ in the number of keys, starting from the normal 12-key keypad layout to keypads bearing the alphabets on more than 12 keys, and from using the standard alphabetical arrangement to adopting the standard keyboard's Qwerty-like arrangement in the same or slightly different manner. I will first discuss the commonly used 12-key keypad layouts, looking in particular at the layouts different from that of the standard 12-key layout next.

2.1.1 The 12-key Keypad

The 12-key keypad is the most widely used tool for text entry on mobile phones. A standard 12-key keypad consists of ten number keys, from 0 to 9, and two special keys bearing the characters * and #. The characters from ‘a’ to ‘z’ are distributed on the number keys 2 to 9 in alphabetical order where each key contains three characters with keys 7 and 9 having four characters each. Key 0 is usually used for inserting the space character while keys bearing * and # serve a different purpose when writing text. This layout is standardized under the name ITU E.161 [Itu.int 2001], also known as ANSI T1.703-1995/1999 or ISO/IEC 9995-8:1994 .

Fig 2.1 The standard 12-key telephone keypad [MacKenzie and Tanaka-Ishii, 2007]

2.1.2 Variants of the 12-key Keypad

Different variants of the 12-key keypad exist which either have the characters arranged in alphabetical order, in Qwerty-like arrangement or differently, on eight keys or more. The purpose of all these variants is to provide more efficient, easy to learn and use, and less error-prone layouts for the users. Some of these are discussed briefly here.

LessTap

LessTap is a phone keypad layout which, like a standard 12-key keypad, assigns the same set of letters to the same number keys on the keypad. The difference, however, lies in the arrangement of the letters on the keys. The letters are not arranged in alphabetical order but arranged based on the frequency of the letters in the English language. For example, the number key 2 consists of the letter arrangement 'acb' instead of 'abc' and so forth, with the aim of reducing the number of keystrokes for most frequent letters.

Pavlovych and Stuerzlinger [2003] tested LessTap with 12 novice participants of the LessTap layout in three sessions, using a Nokia 5190 handset with the multitap method, and outputs displayed on a tablet. LessTap was concluded to be 9.5% faster than the standard keypad (7.82 wpm vs 7.15 wpm), less keystrokes per character (1.6215 vs 2.1505) with no statistically significant difference in error rate except the third session being eye-free.

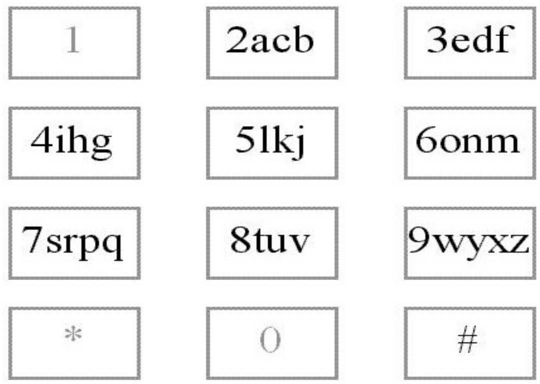

Fig 2.2 The LessTap keypad layout [Pavlovych and Stuerzlinger, 2003]

Alphabetically Constrained Design (ACD)

Alphabetically Constrained Design (ACD) is a mobile phone keypad design that follows the tradition of alphabetic constraints for designing a new keypad layout, i.e. it arranges the characters on the keys in the alphabetical order but not necessarily assigning the same set of characters to the same keys as in the traditional 12-key layout, with a varying number of characters on each key, ranging from one to four.

The design is the result of exhaustive work by Gong and Tarasewich [2005] to find out the best possible alphabetically constrained layout that would provide the users with efficient text entry on mobile phones.

Fig 2.3 Alphabetically Constrained Design (ACD) [Gong and Tarasewich, 2005]

The developers tested the design together with unconstrained design and the standard 12-key mobile phone keypad which were implemented on a Compaq iPAQ Pocket PC. Eight novice users of the ACD and the unconstrained design transcribed phrases using the predictive text entry method. Stylus was used to press the virtual keys on the interface. Results showed that typing speed with constrained keypad design was higher with 6.74 wpm than that of unconstrained design with 5.50 wpm while the error rates were not significantly different, being 0.62 and 0.64 respectively. It proves the developers' hypothesis that alphabetically constrained designs are easier to learn for novice users than the unconstrained designs.

When comparing the speed and error rate of constrained keypad design with standard 12-key keypad, no statistically significant differences were found for speed and error rate between the two designs.

Fewer-letters-per-key (FLpK)

Fewer-letters-per-key, as the name explains, places fewer letters per key on the standard 12-key keypad compared to the standard three to four letter assignment to each key. The design was proposed by Ryu and Cruz [2005]. It assigns only two letters to each key in the same alphabetical order as in the standard mobile phone keypad, with the exception of three letters to two of the keys hence making use of all 12 keys of the 12-key keypad.

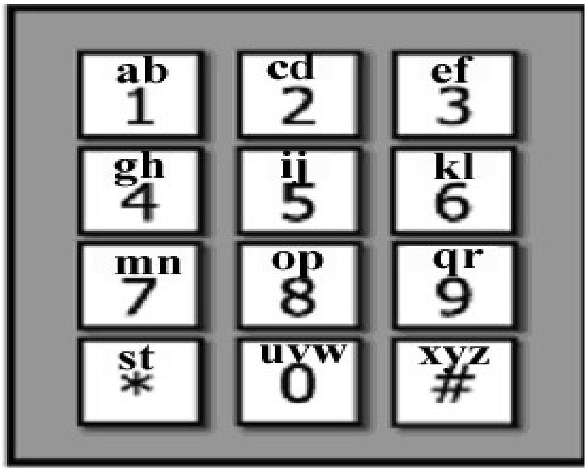

Fig 2.4 Keypad layout for Fewer-letters-per-key [Ryu and Cruz, 2005]

The developers tested the design against the standard mobile phone keypad using the simulation of the layouts on a Pocket PC. Five participants for each layout entered test phrases into the PDA using the multitap method. Words per minute was 6.67 for FLpK and 7.61 for standard mobile phone layout. In addition, mean number of errors for FLpK and standard layout were 3.74 and 2.53 respectively. The reason for FLpK resulting in lower text entry rate and higher error rate may be the users' unfamiliarity with the FLpK layout.

LetterEase

LetterEase, another mobile phone keypad layout presented by the designer of FLpK, utilizes the same 12-key keypad but with different letter assignments to different keys, where each key is assigned two or at most three letters, hence utilizing all the 12 keys for the text entry process. The difference between the two designs, FLpK and LetterEase, however, lies in the letter assignment and arrangement on the keys.

The letter-to-key assignment in LetterEase keyboard layout is based on the frequency of the occurrence of each letter collected randomly from 115 SMS messages of eight users, placing the most frequent letter as the first one on the key, followed by the second most frequent letter and so on. As this design places mostly two letters per key, it adds to the efficiency of the text entry technique compared to the traditional, more common 12-key keypad layout discussed before.

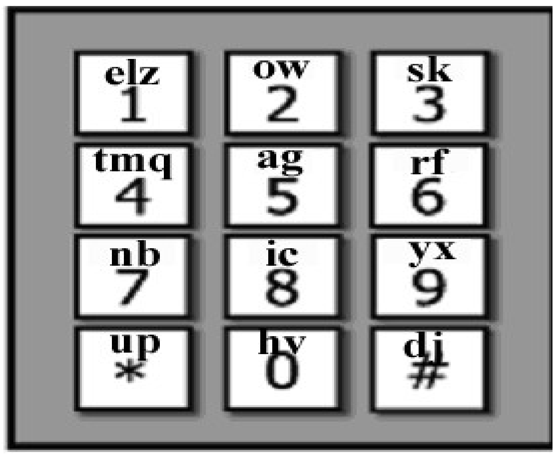

Fig 2.5 LetterEase layout [Ryu and Cruz, 2005]

Ryu and Cruz tested the design against the standard mobile phone keypad using the simulation of the layouts on a Pocket PC. Five participants for each layout entered test phrases into the PDA using multitap method. Participants were not familiar with the LetterEase layout. Text entry rate was 5.49 wpm for LetterEase and 7.61 wpm for standard mobile phone layout. Mean number of errors, however, is less for LetterEase than the standard layout, being 2.21 and 2.53 respectively. The lower text entry rate for LetterEase is caused because of the layout being new for the participants. Text entry speed could be higher if participants had prior experience with the LetterEase layout.

QP (Qwerty-Phone)

Qwerty-Phone (QP) is a mobile phone keypad layout designed by Mackenzie [MacKenzie and Read, 2007], assigning the characters to the nine keys, in an order similar to the Qwerty keyboard. The letter grouping is based on keeping the purpose in mind of reducing the Keystrokes Per Character (KSPC) for text entry.

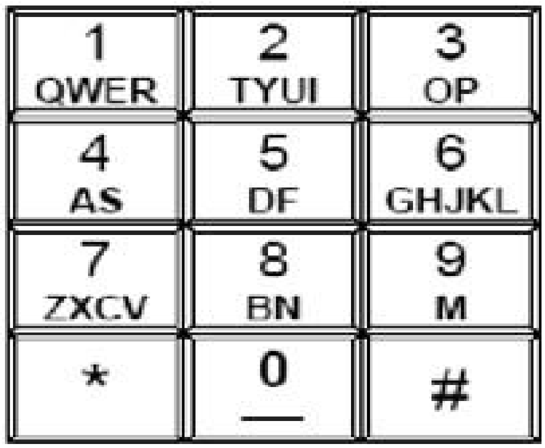

Fig 2.6 QP (Qwerty-Phone) layout [MacKenzie and Read, 2007]

This design has been compared against the standard 12-key layout in a classroom setup using the paper mockups of the layouts. 22 users entered the text “quick brown fox jumps over the lazy dog” by tapping on the layouts with a stylus. Each key was tapped only once assuming that the automatic disambiguation worked. The standard layout resulted in considerably higher text entry rate (26.0 wpm) than the QP layout (12.3 wpm). The keystrokes per character of QP layout was, however, lower (1.0043) than that of the standard layout (1.0072). Since the QP layout was new for the participants, it may have been the cause for lower entry rate for this layout than the standard layout.

2.1.3 Mobile Keypads/Keyboards with Different Layouts

There are different keypad designs suggested for mobile phones, that do not follow the standard 12-key layout. They use more or fewer keys in different arrangements. The purpose of all these variations is to improve efficiency and decrease ambiguity. Some examples of these designs are discussed here. These designs have been either adopted in commercial products or been empirically compared to some other layouts by researchers.

Fastap

Fastap, developed by Digit Wireless Corporation, has the standard 12-key layout for the number keys with one or two rows of extra keys at the bottom. The characters are inserted using the space between the number keys. Each character is assigned to one of the four corners of the number keys and raised high, hence giving the impression of imposing one keyboard on the other. The characters are arranged alphabetically.

Considering the spaces between the number keys as separate keys, it has more than 45 keys and yet has about the same size as a normal 12-key keypad.

Fig 2.8 Fastap keypad layout [Hanlon, 2012]

Fastap layout has been evaluated by Amal Sirisena [2002] against two well-known text entry methods, multitap and T9. Different mobile phones were used for evaluating the multitap, T9, and Fastap. Two user groups, novice and experts, entered four different types of sentences: traditional, non-dictionary, abbreviated and numeric. Fastap novice users had the highest text entry rate (7.43 wpm) , followed by T9 (6.1904 wpm) and multitap (4.56 wpm). However, expert users of Fastap had lower text entry rate (8.86 wpm) than T9 (9.23 wpm) , but higher text entry rate than multitap (5.19 wpm).

SureType

SureType, introduced by RIM in 2004, maps the letters to keys in Qwerty-like arrangement using 14 keys. Each key holds two letters, with the exception of L and M being the only letters on two keys.

Keystrokes per character measure (KSPC) is 1.404 using multitap as the text entry method and almost 1 using T9 [Mackenzie *et al.*, 2011], proving it to be a competitive replacement for the standard keypad.

Fig 2.9 SureType Keypad [AllBlackBerry, 2007]

Stick Keyboard

Stick keyboard is a reduced Qwerty keyboard designed by Green *et al.* [2004] for mobile text entry. This design places all the letters in Qwerty arrangement in one row with nine keys in total, assigning three letters to each key. The letter P, however, is an outlier in this arrangement and is placed last along with the letters O and L.

The stick keyboard layout has been tested and compared against the standard 12-key keyboard with ten novice participants of the deign. Participants entered text to an on-screen text editor on a PC, using multitap and lexicon-based disambiguation. Participants reached almost half of their normal typing speed when using the disambiguation method (49.5 wpm vs 22.5 wpm). Using the multitap method, however, the entry speed was 10.4 wpm.

The design is not publicized as a replacement for cell phone keypads, but considering its efficiency with disambiguation method, it might be a possible option in the future.

Fig 2.10 Stick Keyboard layout [Green *et al.*, 2004]

EQ6

Eatoni’s EQ6 is another mobile phone keypad with the Qwerty-like arrangement of letters. The main characteristic of this keypad is the different number of letters assigned to each key, having 18 keys in total.

This design has been claimed to reduce word collisions in predictive text entry by a factor of 60 [Eatoni.com, 2009]. It has a KSPC very close to 1 [Castellucci, 2007].

Fig 2.11 EQ6 layout [Eatoni.com, 2009]

Other Examples

The keypad layouts described above are not the only ones in this evolutionary industry. There are many other layouts differing in a number of aspects, including the number of keys containing the letters or letter groups and the placement of letters/letter groups. Some examples of these layouts include Qwerty-like [Hwang and Lee, 2005], TouchMeKey4 [Tanaka-Ishii *et al.*, 2002], keyboards/keypads with fewer than 12 keys including letters on four keys, letters on two keys and so forth.

2.2 Mobile Text Entry Methods

As the amount of text entry on mobile devices has increased enormously, the need to improve the text entry methods on these devices is greater than ever. Along with designing different keypad and keyboard layouts to improve text entry, research has also aimed to improve the text entry methods. Results include more efficient methods of text entry, supported by all keypad layouts and device specific methods.

Text entry methods, in general, are divided into two categories by Mackenzie *et al.* [2002]: Key-based text entry methods and Stylus- and finger-based text entry methods.

2.2.1 Key-based Text Entry Methods

Key-based text entry methods comprise of two types of text entry:

1. Methods in which physical keys have the letters mapped to them and user presses those keys to enter text on mobile devices. This method is also referred to as Direct method of text entry.
2. Methods where letters are present on the display screen of the mobile device like a virtual keyboard along with few real, physical keys to select those letters, known as cursor keys and selection key.

I focus on direct text entry methods followed by indirect methods of text entry. Not all keyboards support all text entry methods. I will start from more general and common text entry methods followed by device specific that are less common.

2.2.1.1 Mobile Phone Keypad

As discussed before, a normal mobile phone keypad consists of fewer keys than there are letters in the English language. This results in assigning more than one letter to each key. This arrangement introduces ambiguity in text entry.

A standard mobile phone supports usually two methods of text input: Multitap, also known as Non-predictive, and Dictionary-based disambiguation method, also known as Predictive method. Besides these, there are many other text entry methods developed for mobile phones, some of which will be discussed briefly later.

Multitap

As the name implies, multitap method deals with the ambiguity by requiring users to press a key multiple times to enter some letters. For example, on a standard 12-key mobile phone keypad, the user has to press key number 2 once to enter 'a', twice for 'b' and three times to enter 'c', with a, b and c being the 1^{st}, 2^{nd}, and 3^{rd} character on key 2 respectively. Text entry using this method also introduces segmentation problem.

The segmentation problem occurs when the user wants to enter two consecutive letters that are on the same key. For example, in order to enter 'de', user presses the key 3 two times consecutively without a gap or break, assuming that the first press would result in 'd' and the second press in 'e'. He will be wrong as he will get only letter 'e' as the result.

To resolve the problem of segmentation, there are two common approaches used: ‘Timeout’ and ‘Timeout Kill’.

Using the Timeout method, user has to wait for a short period of time, usually one to two seconds, before he enters the next letter on the same key. In the case above, the user has to press key 3 once to get 'd', wait for a short time, and then press the key 3 again twice to have 'e' in order to get 'de' as the end result. The user comes to know when the timeout is elapsed by noticing the cursor blinking next to the last entered character. The timeout for Nokia phones is 1.5 sec [MacKenzie and Soukoreff, 2002].

Timeout Kill, as the name implies, is a method to skip the timeout period, hence killing the timeout, by pressing a special key commonly known as Next key. It is used by entering the first character, pressing a Next key which is commonly 'Down' or 'Right' arrow key in the mobile phones, and then entering the same letter or next letter on the same key.

In the example discussed above, user presses key 3 once to get 'd', presses the Next key, and then presses the key 3 again twice for 'e', in order to have 'de' as final result. Most of the mobile phones provide users with both strategies so that the user can choose whatever method they prefer.

Keystrokes per character (KSPC) with multitap method is calculated to be 2.0342 using 'Next' button [MacKenzie and Soukoreff, 2002].

Since multitap was the first widely available text entry method for mobile phones, it has become a standard to compare new methods with it. However, since many new text entry methods have been developed and are used successfully in the mobile phones, it might be good to evaluate new methods against the best known system.

The reason for this is the fact that many users now a days use dictionary-based disambiguation methods or predictive methods because of their being faster and easier. Multitap, though easy, is comparatively slower than many other methods in use in mobile phones.

TwoKey

TwoKey is an alternative to multitap method on mobile phones which allows users to enter characters by using exactly two key presses. First press of the key selects the group of letters while second press of a key selects one of the letters in that group.

For example, if the user wants to enter 'd', he presses key 3 once where 'd' is located. Then he presses key 1 to enter 'd' as 'd' is the first character on key 3. Similarly, the user has to press second time key 2 to enter 'e' or key 3 for 'f' as they are the second and third characters respectively in the selected group.

KSPC for this method is calculated to be 2 as each character requires exactly two key presses.

Despite the KSPC figure being more than the multitap method, the method is easy and simple to use and requires no prior practice or experience to enter text efficiently on the mobile phones.

Dictionary-based Disambiguation

Dictionary-based disambiguation methods make use of a built-in dictionary in the mobile phones to help users enter the text efficiently and more accurately by reducing the number of key strokes compared to the non-predictive methods. Following two methods are examples from this category.

One-key with disambiguation

One-key with disambiguation is a commonly used disambiguation method on mobile phones along with multitap. T9, developed by Tegic Communications Inc, is the most common example of this method.

This method requires users to press each key once for each character and matches the resulting key sequence with words in the dictionary that have the same key sequence. The most common or most probable word is displayed as the final output. If the displayed word is not the word intended by the user, user presses a Next key to select from the word list provided by the mobile phone that match the same key sequence. Pressing the space key marks the end of the word. For example, if user wants to enter “with”, he would press the following keys in the order given 9484-, where '–' indicates the space character.

As each character requires only one key press, the method is presumably more efficient than multitap, but this is not always so.

A problem in T9 occurs when the user’s intended word is neither the word proposed by the system nor is in the list of words so that user can choose from. In this case, user has to enter the word using the multitap method. Most mobile phones add the users' words into the dictionary and presents the next time the same sequence is entered, eliminating the need to enter it with multitap every time.

Another problem that users have with dictionary-based disambiguation is the instability of the displayed word while writing is still in progress. As the user is writing the word, each additional key press may result in a different word being displayed. This instability does not demand from the users to pay continuous attention to the screen while writing is in process. Users can check in the end if the final word is the desired word.

KSPC for T9 has been calculated to be 1.0072 [MacKenzie and Soukoreff, 2002]. T9 has been compared against the multitap method in a study conducted by James *et al.* [2001], with novice and experienced users typing two sets of different types of messages. Although the direction of result was as expected, T9 being faster than multitap, novice users in some cases performed better than experts, making the results rather ambiguous and unsatisfactory. In this thesis I try to gain further insight into the performance of multitap and T9.

Text entry methods similar to T9 have been available for almost as long as T9 but they have not been implemented in the mobile phones as widely as T9. Examples include eziText by Zi Corp and eziTap by iTap.

Shift-WordWise

Shift-WordWise, developed by Eatoni ergonomics Inc (www.eatoni.com), falls in the category of one-key with disambiguation method. This method also allows text entry using one key press for each character. In addition, it provides an unambiguous character on each key which can be entered by holding a shift key and pressing the key that has the unambiguous character. For example, to enter 'h', user will press key 4 once while holding down the shift key. If the user does not hold down the shift key while pressing key 4, the result will be either 'g' or 'i' depending on the preceding key sequence.

Key 1 or the volume control button on the side of the mobile phone is usually used as the shift key. The unambiguous characters which will be entered while holding the shift key are c, e, h,l,n,s,t,y on each key respectively as shown in the figure.

Fig 2.12 WordWise keypad with unambiguous characters highlighted [Eatoni.com, 2005]

As there is at least one unambiguous character on each key, the overall ambiguity of a word when entering text decreases considerably since the number of possible combinations from resulting key sequence decreases. The unambiguous characters on each key is chosen so that the remaining characters on that key have the least chance of overlapping with words that have the same key sequence.

LetterWise: Prefix-based disambiguation

LetterWise, developed by Eatoni Ergonomics, is another disambiguation technique that, instead of using a dictionary of stored words and matching the user's entered words, uses a database of letter-prefix probability to guess the user's intended letter. It simply means that the probability of next letter occurring in a word depends on the letter/letters entered before the current key stroke, hence the prefix.

For example, if a user has already entered 'th', then pressing key 3 would result in 'e' rather than 'd' or 'f' as 'the' is more common word and has higher probability of occurrence in English compared to 'thd' or 'thf'. However, if the output character is not the desired one, user has to press a 'Next' key to select the right letter from the list of letters mapped to the pressed key.

In this method, every new keystroke results in the letter that is the most probable in relation to the prefix. This results in a stable output on the display, reducing the confusion for the users. Moreover, as this method uses the probability of a letter occurrence instead of using a dictionary, any non-dictionary word can be entered as efficiently as a word in dictionary, an advantage over the dictionary-based disambiguation methods.

A longitudinal study consisting of 20 sessions has been conducted in order to compare LetterWise with the multitap method by MacKenzie *et al.* [2001] with 20 participants. After 20th session, the WPM figures were 21.0 and 15.5 for LetterWise and multitap respectively, showing 36% faster entry rate with LetterWise than with multitap.

The calculated KSPC value for LetterWise is 1.1500 [MacKenzie and Soukoreff, 2002].

TiltText

TiltText, developed by Wigdor and Balakrishnan [2003], is also a text entry technique to be used on standard 12-key mobile phones. This technique, along with requiring users to press a key on the keypad to enter a letter, uses another feature to disambiguate the result i.e. tilt, letting the users press each key once to enter a letter.

The mobile phone is equipped with sensors which determine the right letter as a result of key-press and tilting the phone in one of the four directions. To input first letter, user has to press the particular key and tilt to left, for second letter, tilt forward (away from user’s body), for third letter user has to tilt to right and in case of fourth letter, tilt towards the body of the user. For example, to input 'p', user has to press key 7 and tilt the phone to left, tilting forward displays 'q', right tilt results in 'r', tilting towards one’s body would result in 's'.

The method has been compared against the standard multitap method with ten participants using a Motorola i95cl phone. Use of TiltText resulted in 11.76 wpm compared to 10.11 wpm for multitap. In other words, TiltText was 16.3% faster than multitap, but with higher error rate (11%) than multitap (3%) [Wigdor and Balakrishnan, 2003].

2.2.1.2 QWERTY and Other Small Keyboards

Text entry on Qwerty and other small keyboards depends on the keyboard design. Some of these are briefly explained here.

Reartype

Reartype is a text entry mechanism which places a full Qwerty keyboard on the back of the mobile device, such as tablet PC, by splitting it in two halves. Reartype saves the space on the front of the device. Users can reach two halves of the keyboard with their hands easily while gripping the device.

Fig 2.13a Front of Reartype prototype

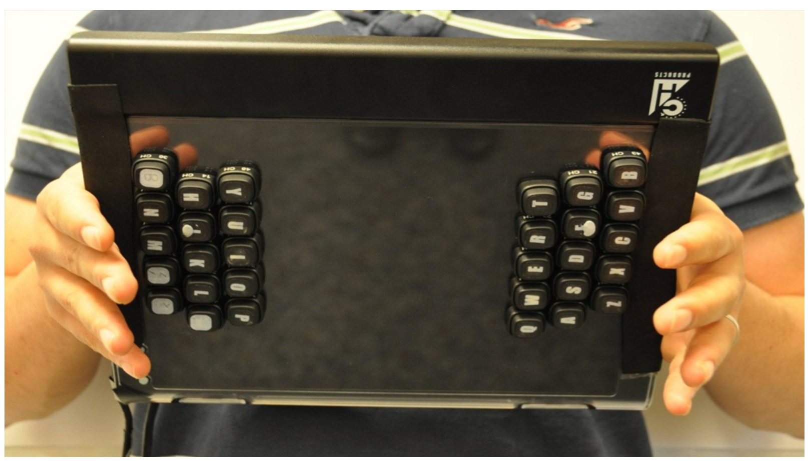

Fig 2.13b Back of Reartype prototype [Scott *et al.*, 2010]

This technique, developed by Scott *et al.* [2010], makes use of the user's typing skill with the standard Qwerty keyboard by placing the two halves of the keyboard in the same relative position with each other as would be in a normal full keyboard, hence each finger reaching the same keys as in Qwerty keyboard.

Reartype was compared against a normal touchscreen soft keyboard and normal desktop Qwerty keyboard. After one hour of practice, typing speed with Reartype was 15.1 wpm which was not significantly different from the touchscreen soft keyboard with mean typing speed of 21.2. As was expected by the developers of the method, the typing speed was much lower than that of standard keyboard with 72.1 wpm. This is, however, result of long hours of practice and work with the keyboard, compared to only one hour of practice with Reartype.

Since a full-sized keyboard is placed on the back side of the device, it can be a substitute for mini and optimized keyboards or touchscreen soft keyboards, saving screen space. The technique might not be difficult to learn for expert keyboard users as the design allows the same key assignment to the same hands and fingers as with standard keyboard. However, the method can be complicated and difficult to learn in the beginning for novice users.

Half-Qwerty

Half-Qwerty, another keyboard design as well as writing technique, developed by Matias *et al.* [1994], also takes advantage of user's typing speed and familiarity with the Qwerty keyboard while reducing the size of the keyboard to half by splitting it in two parts, making it usable with small mobile devices.

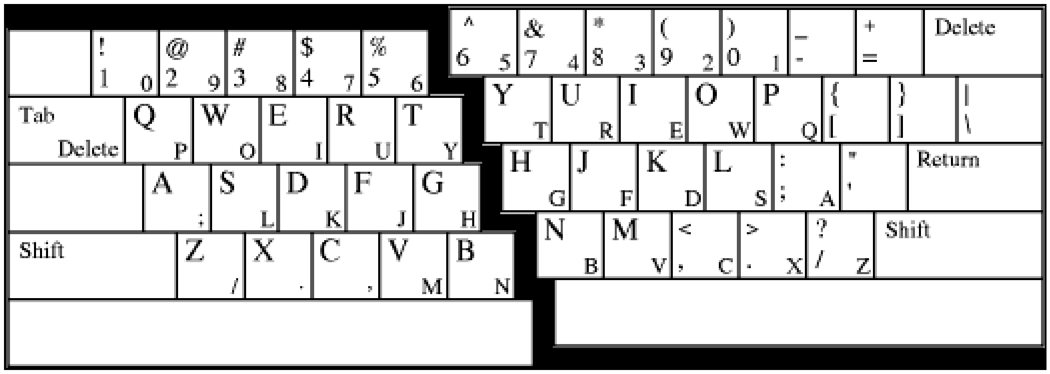

Fig 2.14 Half-Qwerty keyboard layout [Matias *et al.,* 1994]

The keyboard is split in two halves having the standard alphabet assignment to the respective keys so that user can use any of the two halves, right or left, for text entry with one hand by keeping the other hand free for other tasks .

If a user uses left half of the Half-Qwerty for text entry, he can enter the alphabets of the left half by simply pressing the desired key. To enter the letters of the right half of the keyboard, the missing letters, user has to press a modifier key i.e. space key, in combination with the same finger pressing relatively same key that was done with the right hand before. For example, to enter 'O' on the left Half-Qwerty, user has to press the modifier key and key 'W' with his left hand's ring finger as it would be the same when entering with right Half-Qwerty.

The technique has been tested with ten users of standard keyboard, using one and two hands, resulting in average of 34.7 wpm and 64.9 wpm respectively after 10th session [Matias *et al.,* 1993]. The technique, though simple to use, requires practice as the skill transfer from one hand to the other requires time. However, once learned, it is a good replacement to full-sized keyboard on small devices with each key dedicated to only one letter.

Twiddler

Twiddler is a one-handed chord keyboard. The text entry technique for the device, named as Twiddler typing, requires users to press one or more keys simultaneously to enter the letters.

Fig 2.15 Twiddler: One-handed Chord keyboard [Lyons *et al.*, 2004]

The keypad is held in one hand so that back of the keypad faces the user while the face containing the keys is away from the user. User can enter text using one of his four fingers for each row of the keys. As the keyboard contains individual keys for only few of the English alphabets, the missing letters can be typed by pressing two keys simultaneously.

The twiddler typing has been tested on the twiddler against the multitap with ten participants for 20 sessions where the test phrases and output phrases were displayed on a computer screen. Twiddler typing resulted in mean entry speed of 26.2 wpm compared to 19.8 wpm for the multitap after 20 sessions, although the initial entry rates for twiddler were lower than the multitap technique. The KSPC value calculated for the twiddler method using Soukoreff's diagrams [2002] was 1.4764. The technique can be an alternative to the multitap method on mobile devices in the light of the results.

Chord keyboards such as twiddler are not so popular among the users. The reason may be that a key on a chord keyboard is used to enter a number of different characters when used in combination with other keys. Since users have to memorize and learn all the combinations, the interest in using the chord keyboard becomes lower.

Text entry with five or fewer keys

An other category of text entry on mobile devices is to use the navigation/cursor keys to move along the letters displayed on the screen and to press another key to select. The total number of physical keys may range from one to five which include one selection key and others as navigation keys.

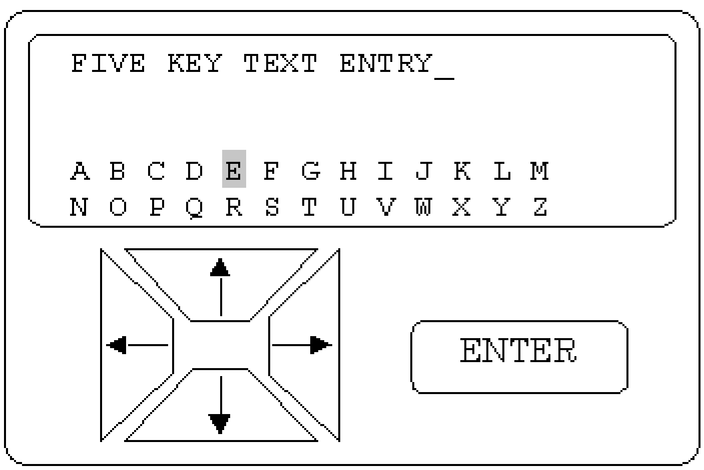

Fig 2.16 The Five-key text entry method [MacKenzie and Soukoreff, 2002]

The reason for this kind of text entry is mainly the small size of the mobile device which makes it difficult to include a full-sized keypad with keys for all the letters. An example of this kind of device is a pager. Text entry time increases as the number of keys decrease.

For example, in a setup using the five key text entry method, if user has entered 'a' and the next letter he intends to write is 'y', he has to move a long distance as shown in Fig 2.16.

To overcome this problem, to speed up the entry, and to decrease the KSPC, Bellman and Mackenzie [1998] proposed the 'Fluctuating Optimal Character Layout (FOCL)'. This method re-arranges the letters based on the user's last entered letter by placing the next most probable letter close to the cursor position. This method reduces the KSPC but introduces extra work for the users. Since the positions of the characters change after every entered character, it imposes the need of constant visual search upon the users to locate the next desired letter.

The FOCL method was compared against a fixed Qwerty layout with 11 participants over ten sessions. The text entry rate was found to be around 10 wpm for both methods after the 10^{th} session. There was no statistically significant difference in entry speeds.

4-key text entry methods place letters on four keys and allows predictive text entry. The 'SnapKeys 2i invisible keyboard' [Snapkeys Ltd, 2011] is an example of this. Similar methods have been proposed allowing text entry with even fewer keys.

The important thing to notice is that as the number of keys decreases, ambiguity on remaining keys increases which increases the key strokes per character (KSPC). However, the increase in KSPC value is still smaller for one key with disambiguation method than the multitap method.

Other Examples

The text entry techniques described above are few of the large number of key-based text input techniques developed and proposed by the researchers. Many more of these methods and devices exist. Examples include TouchMeKey4 [Tanaka-Ishii *et al.*, 2002], Rollpad [Oniszczak and MacKenzie, 2004], TiltType [Partridge *et al.*, 2002], An enhanced multitap text entry method with predictive next-letter highlighting [Gong *et al.*, 2005], ChordTap [Wigdor and Balakrishnan, 2004], and SHRIMP [Wang *et al.*, 2010].

2.2.2 Stylus-based and Finger-based Text Entry Methods

The 2nd category of text input methods is stylus-based and finger-based text entry. Stylus-based and finger-based input methods have been developed for the same reasons that many other text entry techniques have been developed: to gain accuracy and speed while minimizing the size of the mobile device. Stylus- and finger-based text entry is divided usually in three categories: soft keyboards, handwriting recognition, and gesture input.

2.2.2.1 Soft Keyboards

Soft keyboard, also known as the virtual keyboard or the on-screen keyboard, is an image of a keyboard on the screen. As the keyboard is not a real physical one, keys on it are just a designated areas. These areas are tapped by the user with a stylus or with a finger to enter the text.

The standard Qwerty layout for soft keyboards takes advantage of users' familiarity with the layout from desktop keyboards. The alphabetic arrangement of letters on soft keyboards is a simple and easy to use layout by users familiar with the alphabet.

Fig 2.17 A Qwerty keyboard layout [MacKenzie and Zhang, 1999]

The benefit of using a virtual keyboard is that the size of the display on a mobile device can be increased because physical keyboard is not needed. Virtual keyboard can be displayed on demand whenever needed.

Despite the small size of the mobile phones and even smaller soft keyboards on the screens, text entry using soft keyboards is the dominant method of text entry on small handheld mobile devices. An example of soft keyboard is OPTI.

OPTI

OPTI is an optimized soft keyboard layout by Mackenzie and Zhang [1999]. The design was optimized using trial and error method and was chosen as the final one after testing it with a prediction model. The prediction model consisted of Linguistic data, Fitt's law, shortest path model, and key-repeat-time measure. It was used to compute text entry rate for different designs of the keyboard.

The linguistic data used was information on frequencies of the occurrence of the bigrams in the English language. Fitts' law predicted the time to tap a key given any previous key. The shortest-path model provided information about the shortest possible path that can be taken from several discrete paths between two or more consecutive keys. The last part of their predictive model, key-repeat-time measure, tells the time between two consecutive presses of the same key.

The model for the OPTI layout gives the maximum theoretical entry rate of 58.2 wpm in comparison to 43.2 wpm for the Qwerty layout.

Fig 2.18 OPTI layout [MacKenzie and Zhang, 1999]

In addition to modeling, the design was tested with five users over 20 sessions. As was expected, OPTI initially lagged in performance with 17 wpm in comparison with Qwerty with 28 wpm. However, entry speed improved over the sessions and by the 20^{th} session, OPTI reached to 45 wpm leaving the Qwerty layout behind with 40 wpm. The error rates were 4.18% and 4.84% for OPTI and Qwerty respectively at the end of the 20^{th} session.

Other Examples

OPTI is not the only soft keyboard proposed recently. There have been many other designs. Examples include the Fitaly keyboard (www.fitaly.com), Metropolis keyboard and Hooke's keyboard [Zhai *et al.*, 2000].

2.2.2.2 Gesture Recognition

Gesture-based text entry techniques require users to make certain movements, called gestures, on the touchscreen. The pattern is captured and recognized as a character. Unlike soft keyboards, where user has to tap each character, gesture-based systems are easy to use once learned. The reason is their flexibility as the pattern made by the user need not be exactly the perfect match of the original pattern.

Gesture-based text entry techniques help users avoid using soft keyboards where tapping each letter is tedious as well as tiresome, especially for longer text entry tasks. Many gesture-based text entry systems have been developed of which one is discussed briefly.

Quikwriting

Quikwriting, designed and suggested by Perlin [1998], is a gesture-based text entry method for mobile devices. The interface for Quikwriting is a 3*3 grid. All the characters are placed in the eight peripheral zones.

Quikwriting has four modes of writing, in other words, four sets of characters: the lowercase letters, uppercase letters, numeric, and symbols or punctuations. Default mode of writing is the lowercase characters.

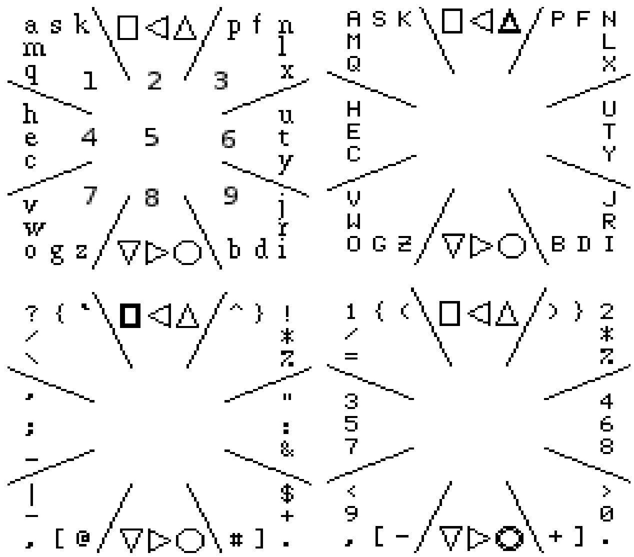

Fig 2.19 The four modes of Quikwriting with zone numbers [Perlin, 1998]

To write a character, user starts from central zone (5) by placing the stylus on the surface and moving it to the one of the eight zones where the intended character is placed. Once the stylus is in that zone, known as the major zone, the user has to select the character from that zone. This happens by moving the stylus to minor zone indicated by the position of that particular character in the major zone. For example, to enter 'c', the user drags the stylus from central zone to zone 4 and makes a downward drag gesture entering the minor zone 7 and then moves back to the central zone.

To enter the most frequent letters, like a, e,i, o, and few others, users have to just move from central zone to the particular major zone and then back to the central zone as these characters are placed in the middle of each zone minimizing the gesture required for their entry.

Unlike its counterparts which allow only one character entry in a gesture, users can enter words or sentences in one sequence of gestures without lifting the stylus.

Other Examples

There are many gesture-based text entry systems. Cirrin, which is a word-level gesture-based text entry system [Mankoff and Abowd, 1998], T-cube using the flick gesture of the stylus [Venolia and Neiberg, 1994], Dasher, that combines user gesture with a language model which calculates the probability of the next possible letter [Ward *et al.*, 2000] are some examples.

2.2.2.3 Handwriting Recognition

Handwriting is the most common way of written communication. For a long time, it was the only way of documentation and communication over distance. With the advent of mobile devices and need for text entry systems, it has become one of the methods for text entry on mobile devices. Despite being slower than a keyboard when it comes to text entry on mobile devices, it is still present and researched, and faster and more accurate handwriting systems are being developed.

To enter text using handwriting recognition, user writes a character or combination of characters (word, sentences) with a stylus on a digital surface. The movement of stylus on the surface from pen down to pen up is called a stroke. A character consists of one or more strokes. Also, more than one character can be entered with one stroke. This results in segmentation problem where the recognizer has to identify if the stroke(s) resulted in one character or multiple characters.

English language consists of a large number of characters and symbols that need to be recognized accurately in order for the user to be satisfied and for the system to be accepted. The addition of numerals to the symbols puts more challenge for the handwriting recognition technologies. Moreover, the different writing styles adopted by different users make the recognition task even more challenging.

For example, user can write a word in different ways: **1.** Each letter separate from each other and distinct, known as discrete characters, **2.** By joining the characters with each other where they can overlap, known as Run-on discrete characters, **3.** Pure cursive writing where characters are joined with each other, and **4.** Combination of all these different writing styles [MacKenzie and Tanaka-Ishii, 2007].

Besides the large number of characters and different writing styles, many characters resemble either another character or a numeral. Examples are the uppercase I and 1, the character o and number 0 and others. As current handwriting recognition technologies are not so efficient, it is harder for these systems to be accepted widely.

Experiments have been conducted evaluating the accuracy of different recognizers. Mackenzie and Chang [1999] compared two recognizers resulting in 87%-93% of accuracy. In a more recent study by Kristensson and Denby [2009], a handwriting recognition system is evaluated against a soft Qwerty keyboard with 12 participants over ten sessions. Participants entered text on a Tablet PC using a capacitance-based pen to write directly on the screen. After the 10th session, the average text entry rate was 24.1 wpm for handwriting recognition system and 24.9 wpm for soft keyboard, with no statistically significant difference between the speeds and the error rates of both methods.

Unistrokes

Unistrokes, as the name explains, is a single stroke alphabet replacement for Roman letters developed by Goldberg and Richardson [1993]. The technique was developed to achieve higher entry speed and accurate recognition along with allowing the users eyes-free text entry.

Using Unistrokes, each letter can be written with one stroke, eliminating the problem of segmentation for the recognizer, resulting in more accurate recognition. Secondly, the method makes the text entry task easier for the users by minimizing the need to look at the stylus or digital surface all the time when writing. Lastly, text entry speed becomes faster when writing each letter with single stroke.

Like any alphabet, the problem with Unistrokes alphabet is that users have to learn the letters prior to using. The learning task required is time consuming, causing reluctance towards adopting the method.

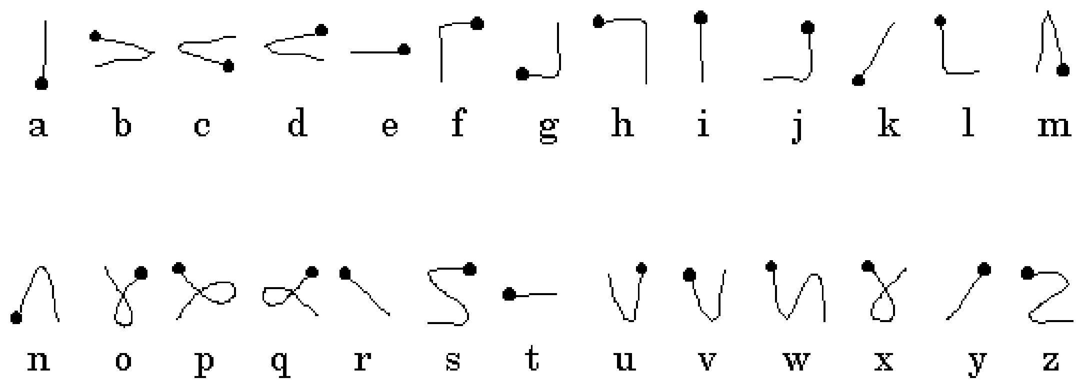

Fig 2.20 Unistrokes letters [MacKenzie and Soukoreff, 2002]

There have been some comparative studies to evaluate Unistrokes against other methods. Goldberg and Richardson conducted an experiment where participants sent email messages using an email sender application developed particularly to test Unistrokes in real use. They reported a text entry rate of 2.8 letters/second by suggesting maximum achievable speed of 3.4 characters/second.

Other Examples

Unistrokes alphabet does not contain numbers, symbols or punctuations. Many other systems have been developed that support the same single stroke input method like Unistrokes and also may support writing numbers, punctuations and symbols. Examples include Graffiti by Palm Inc (www.palm.com), EdgeWrite by Wobbrock *et al.* [2003], and MDTIM: Minimal Device-Independent Text Input Method by Isokoski [1999].

The MDTIM is unique in sense that it is device-independent, allowing users to use it with any input device including mouse, joystick, stylus and others, making it a possible choice of text entry method with higher priority than others. EdgeWrite also has some of these properties.

2.3 Summary

In this chapter, I described different keyboard layouts and text input methods for mobile devices, starting from systems in common use and also covering some that are not in common use.

Among the keypad layouts for mobile phones, the most common is the 12-key keypad layout which might be the most ergonomic, witnessed by its widespread use by majority of the mobile users. The Qwerty keyboard might be a potential candidate and possible replacement for the standard mobile phone keypad because of its familiar layout to the users and excessive use by the users of computers and other mobile devices.

From text entry methods discussed so far, the dominant ones are multitap and T9. They have been in use for a long time despite many other text entry methods having been developed. The reason is their ease of use and users' familiarity with them. Moreover, these methods are compatible with each other and the standard mobile phone keypad.

Users have plenty of choices as newer keyboard designs and text entry methods appear. This has encouraged the users to become more demanding than before. Therefor, it is not worthwhile to spend time and effort redeveloping something which does not provide advantage over the existing ones.

The question however is still there, “can any other method replace the dominant ones?”. For a method to be accepted by the user community, it has to meet the needs of the users as easily and as accurately, without putting much pressure on the users for learning. However, there are some other issues which also need to be considered. For example, software manufacturers prefer simplicity, operators are reluctant to support the users who have problems with text entry, and security issues related to text entry methods. Considering all these, it is not easy to predict about the future of text entry methods.

3 Multitap vs T9

Despite all the data available on performance of multitap and T9 in comparison to other text entry methods for mobile phones, the need for a comparative, empirical evaluation between them still exists. Because of their widespread use, their relative performance is worth knowing in detail.

Moreover, the theoretical performance prediction for the two text entry methods by Dunlop and Crossan [2000] based on the 'keystroke level model' and the prediction based on 'movement and linguistic model' by Silfverberg *et al.* [2000] are quite different from each other and require an experimental data to identify which of the two models is a better representation of users' performance.

To find the answers to these questions, I conducted an experiment similar to the study by James *et al.* [2001].

3.1 Method

3.1.1 Participants

16 participants, eight male and eight female, aged between 21 and 47 years were recruited from the university of Tampere. Ten participants were computer science majors while six were of different majors such as translation studies and history. All the participants were asked to fill-in a questionnaire prior to experiment with questions such as whether or not they wrote text messages on their mobile phones, and if yes, how many

per week and using which method. Based on their answers, they were grouped into four groups: novice and experienced users of multitap and T9.

All participants used SMS, some on daily basis while others occasionally. The minimum number of text messages that the participants reported sending per week was 3 and maximum 100. Some participants used only multitap, some used only T9 while some had switched from multitap to T9 and then to Qwerty keypad and touch screen mobile phones. None of the participants had English as their first language, however they were all, except one, good speakers of English as they studied in the English programs of study in their respective departments.

The number of messages sent by each user per week and the main text entry method on the mobile phone are shown in Fig 3.1.

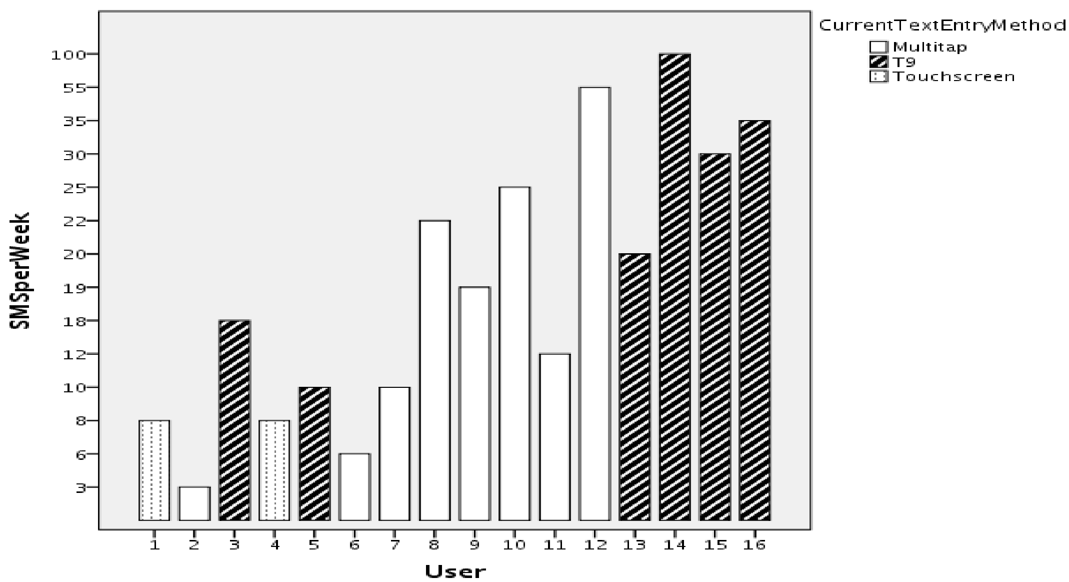

Fig 3.1 Number of messages sent per user per week

Those participants who used multitap as the only text entry method on their mobile phones and had used this method since they owned their first mobile phone, in other words, for many years, and also sent 10 or more messages per week are put in the experienced multitap user group and consequently, in the novice T9 group. The number of years they have been using this method was between 7-10.

Details of number of messages sent per week and the number of years using multitap method for each participant is given in Fig 3.2.

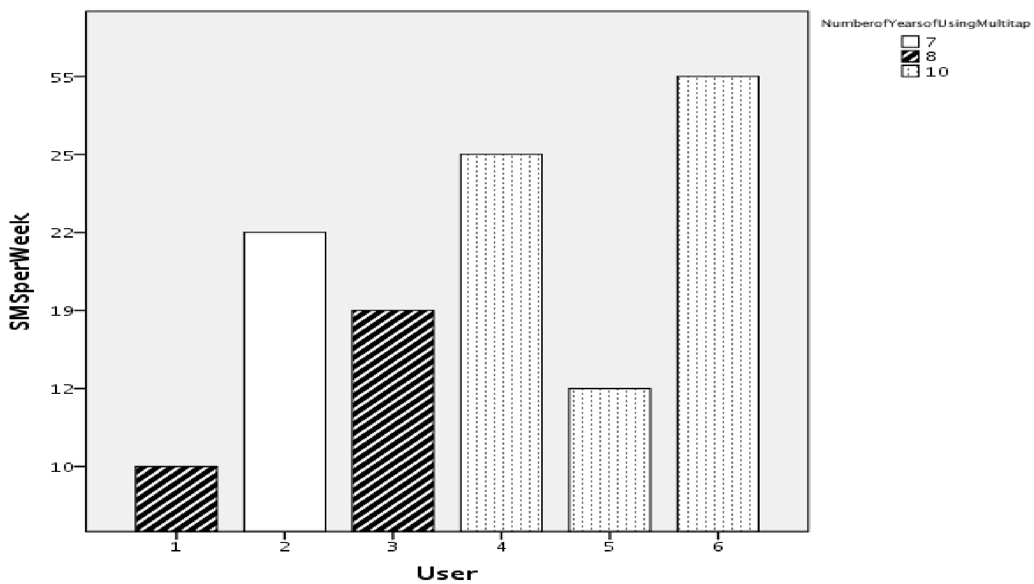

Fig 3.2 Number of messages sent per user per week with the number of years using multitap for the experienced multitap and novice T9 group

Those who used T9 as the only text entry method on their mobile phones, either from the beginning or switching from multitap to T9 (as was the case with almost all the participants in this group), and had used it for long time i.e. for many years, and sent 10 or more message per week are put in the experienced T9 group. The number of years they have been using this method was between 2-12. Two of the users from this group met the criteria of being novice multitap users, and are therefore put in the novice multitap group as well. Details of number of messages sent per week and the number of years using T9 method for each participant is given in Fig 3.3.

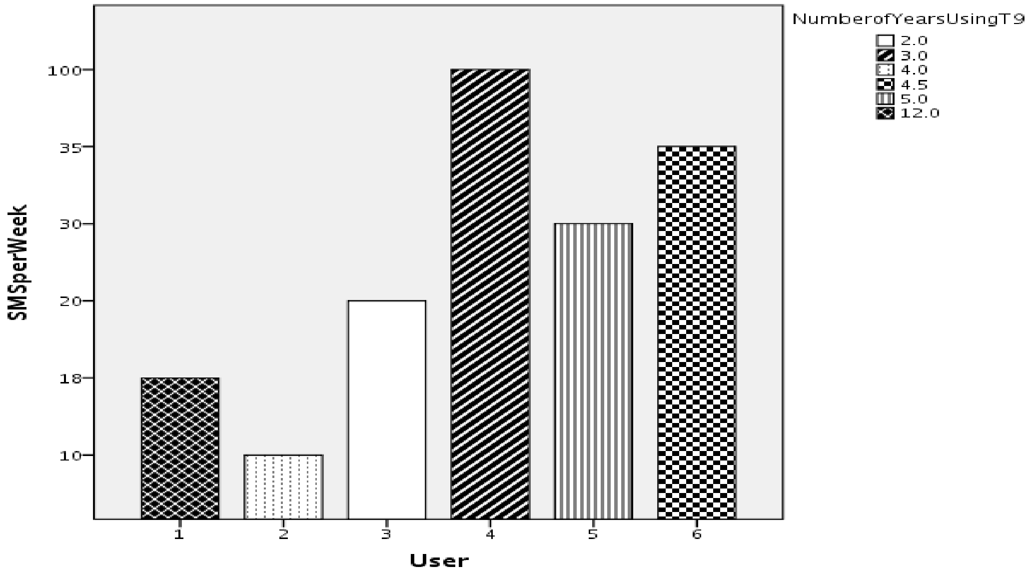

Fig 3.3 Number of messages sent per user per week with the number of years using T9 for the experienced T9 group

Those who used multitap or T9 as the text entry method for writing text messages and sent less than 10 messages per week are put in novice multitap group. Those, who had used multitap before switching to T9 and/or then switching to Qwerty keypad and have been using the new methods for more than an year are also put in the same group. Details of number of messages sent per week and the number of years using either text entry method for each participant is given in Fig 3.4.

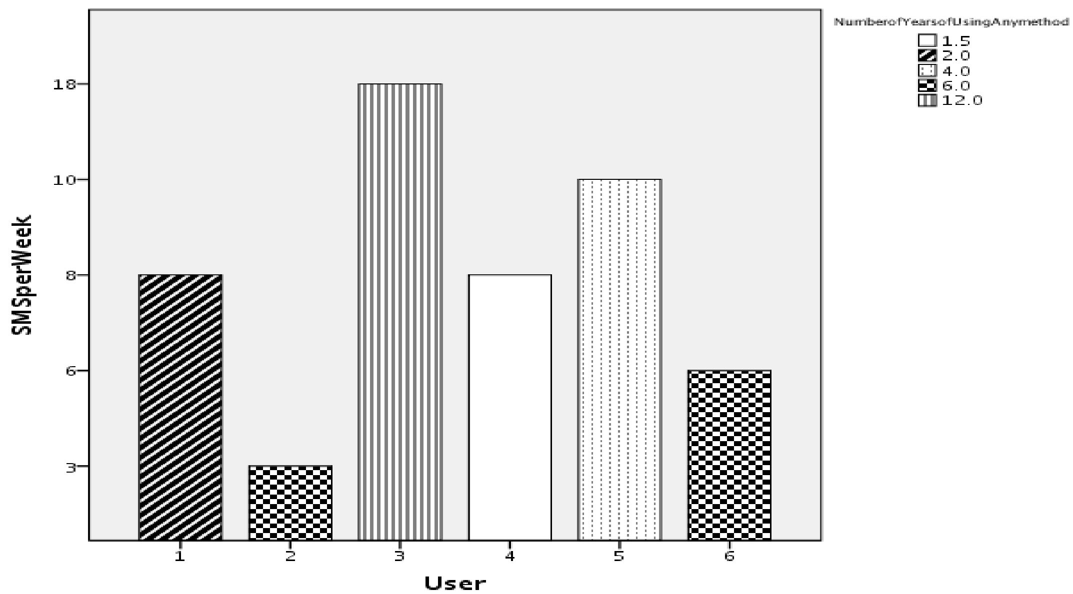

Fig 3.4 Number of messages sent per user per week with the number of years using multitap, T9 or Qwerty/Touchscreen for the novice multitap group

3.1.2 Apparatus

The hardware consisted of a Nokia N95 mobile phone with a standard 12-key keypad for the purpose of text entry with the both methods under study. The phone was set to compose mode in lower case and users were free to hold it in any hand they were comfortable to write the phrases.

The software that provided the users with the test sentences and recorded the experiment data in log files for analysis was written in Java language using Java ME (Micro Edition) platform by the thesis supervisor. 175 phrases were chosen randomly (every third phrase) from the 500 phrases of the phrase set described by MacKenzie *et al.* [2003]. No symbols, capital letters or punctuation were included in the phrases.

The reason for using only a third of the phrases in the set was that all the words in the phrase set were not present in the mobile phone dictionary. To ensure realistic user behavior, we wanted all words to be in the T9 dictionary. Using only a third of the phrases made the work of entering the words into the dictionary faster.

The analysis software, for the purpose of analyzing the log files, was also provided by the thesis supervisor in Perl programming language and it calculated the text entry measures including 'characters per second (cps)'. 'words per minute (WPM)', 'keystrokes per character (KSPC)', and 'minimum string distance (MSD)'. I tested all the software to ensure that it worked properly and analyzed the data correctly.

3.1.3 Procedure

The experiment was performed in the “Gaze laboratory” of the School of Information Sciences of University of Tampere. At first, the purpose of the experiment and the experiment procedure were explained to the participants. It was emphasized that it was the text entry methods that were being examined, not the typing skills of the participants. Participants were then briefed with the procedure by providing them with the mobile phone for five to ten minutes so that they could familiarize themselves with the keys, such as backspace key on the mobile phone and the transcription task.

First, the user name and session number was entered. By pressing the “Done” button, participants were presented with the phrases one by one, for two minutes. They were asked to enter the sentences as fast and accurately as possible. When the time expired, they were given a short break of five minutes and after setting the mobile phone on compose mode for T9, they were asked to repeat the same procedure.

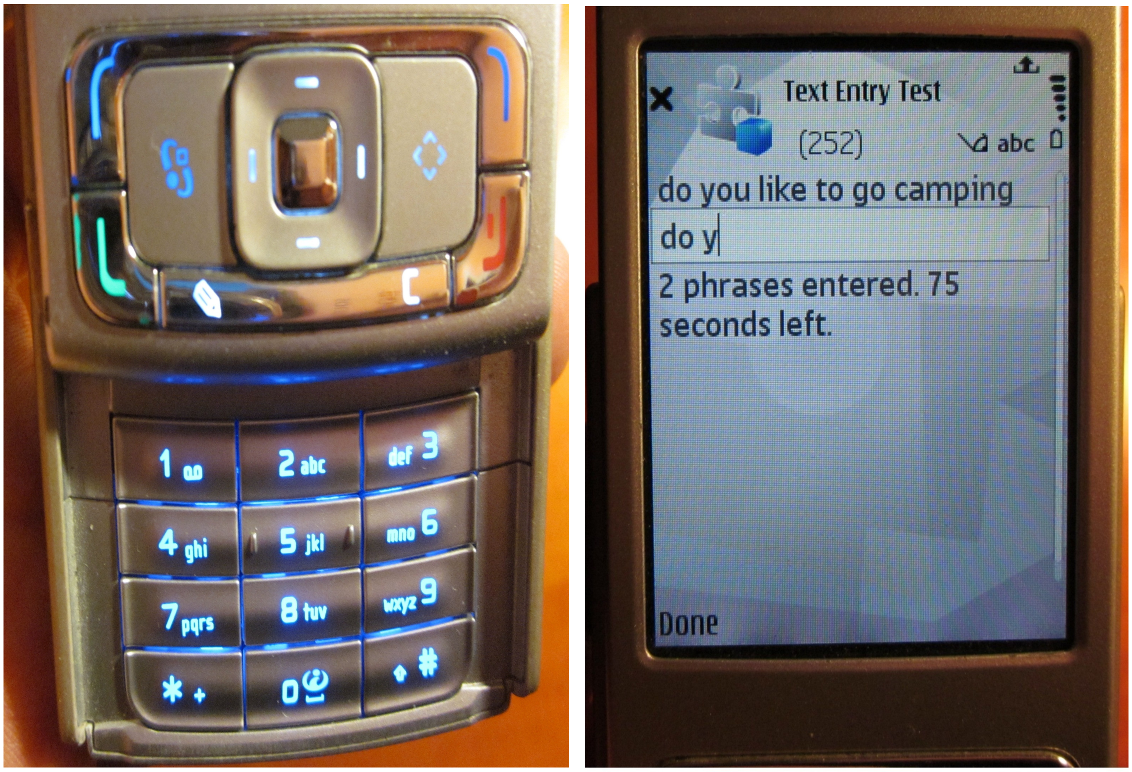

Fig 3.5 Images of the mobile phone keypad and the experiment software

Each participant performed two sessions of two minutes for each method. To balance transfer effects, equal number of participants performed the test with multitap method first and T9 as the second and vice versa. Participants were allowed to correct the mistakes as they wrote.

As the participants pressed the 'Done' button after entering their user name and session number, the text writing mode in the mobile phone changed from the set lower case to upper case for the first character. As all the test phrases were in lower case, participants were asked to press the case mode button once quickly before writing the test phrases to set the system back to lower case mode.

“Begin” time for each phrase was recorded when the first key was pressed by the users i.e. when the first letter appeared on the screen, with the "end" time recorded with the press of "Done" button at the end of the phrase. If the last sentence was incomplete when the time expired for the session, the participants needed to finish it before the session ended.

3.1.4 Dependent Measures

Text entry rate (measured in words per minute (WPM)) for each phrase was calculated with the standard definition of word being five characters including space. I computed the wpm metric using the given formula:

$$WPM = \frac{|T| - 1}{S} \times \frac{60}{5}, \qquad (3.1)$$

where $|T|$ is the length of the transcribed string and S is the time (in seconds) taken to write that string. *60* is the seconds per minute and *1/5* is words per character.
Total WPM for each session was calculated using the following formula:

$$Total\,WPM = \frac{Total\,|T|}{Total\,S}, \qquad (3.2)$$

where *Total* $|T|$ is the total number of characters in transcribed strings in one session (2 min) and *Total S* is the total time taken to write the phrases.

Another way of calculating the total words per minute is to take the average of words per minute figure for each phrase. However, by calculating this way, both short and long phrases will get the equal weight which is not, in my opinion, correct.

To measure accuracy during the text entry process, Keystrokes per character (KSPC) measure is calculated using the formula:

$$KSPC = \frac{|IS|}{|T|}, \tag{3.3}$$

where *IS* is the total number of entered characters (including backspaces) and *T* is the final number of characters in the transcribed string.

Total KSPC for each session was calculated using the following formula:

$$Total\ KSPC = \frac{Total\,|IS|}{Total\,|T|}, \tag{3.4}$$

where *Total* |*IS*| is the total number of entered characters in one session (including backspaces) and *Total T* is the total number of characters in the final transcribed strings. When calculating keystrokes per character (KSPC), the 'Done' button presses (after each phrase) were not included in the calculation.

In this experiment, KSPC measure includes only characters received by the text field, not the actual keypresses.

MSD i.e. Minimum string distance is calculated to measure the accuracy of the final transcribed string in terms of 'uncorrected error count'.

3.1.5 Design

A 2×2 factorial experiment design was used. The two independent variables were the input methods with two levels as multitap and T9, and user experience level with two levels as novice and experienced. The dependent variables were text entry speed measured in words per minute (WPM) and accuracy measured in terms of minimum string distance (MSD) and key strokes per character (KSPC).

One of our concerns was that in earlier work by James *et al.*, the user groups were not really distinct. Because of this we wanted to test statistically whether our groups followed the pattern to be expected i.e. experts should beat novices. In addition, we expected T9 to be faster than the multitap, as predicted by the modeling work in earlier work.

3.2 Results

3.2.1 Speed

For multitap users, minimum text entry rate was recorded to be 7.3 wpm and maximum 15.6 wpm, and for T9 users, 10.3 wpm and 29.6 wpm respectively. The average text entry rate for each group are shown in table 3.1.

Method	Novice	Experienced	Mean
Multitap	10.93 (SD = 2.79)	11.29(SD = 3.06)	11.11
T9	14.12 (SD = 3.72)	22.68 (SD = 6.70)	18.4
Mean	12.52 (SD = 3.55)	16.98 (SD = 7.75)	

Table 3.1 Mean text entry rate in WPM for both user groups

Result show that there is an increase of 29.2% for novice T9 users and 100.9% for experienced T9 users over the respective multitap groups.

One-Way ANOVA showed that experienced T9 users were faster than novice T9 users ($F(1,10) = 7.489$, $p = .021$), but there was no difference between experienced and novice multitap users. When comparing the input methods, experienced T9 users were faster than experienced multitap users ($F(1,10) = 14.357$, $p = .004$), but there was no difference between the performance of novice T9 and multitap users.

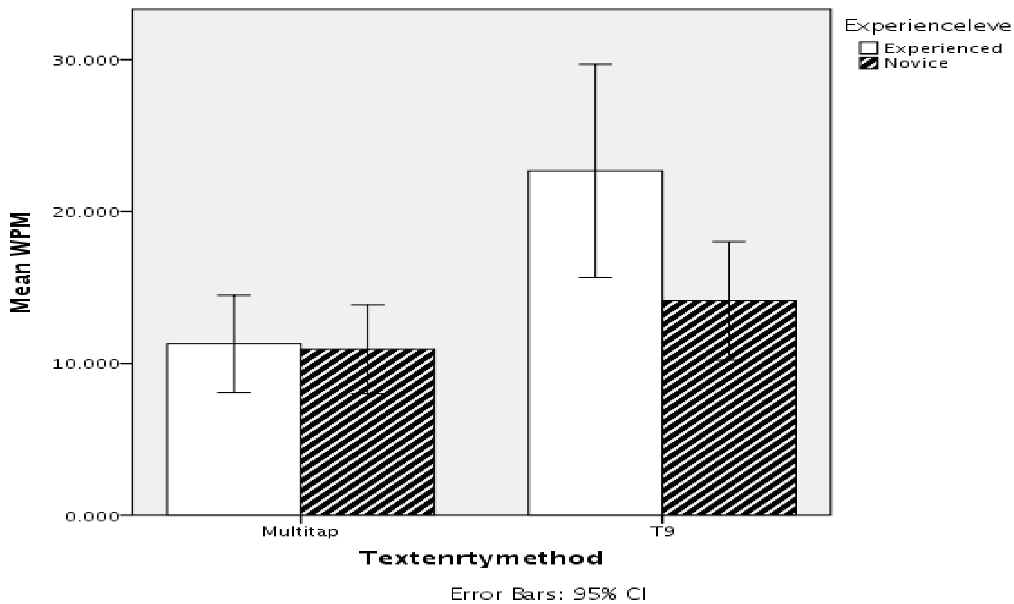

Fig 3.5 Interaction between Text entry method and Experience level for wpm

3.2.2 Accuracy

Average key stroke per character (KSPC) for each group is shown in table 3.2.

Method	Novice	Experienced	Mean
Multitap	1.10 (SD = .06)	1.11 (SD = .10)	1.1
T9	1.05 (SD = .04)	1.07 (SD = .06)	1.06
Mean	1.07 (SD = .06)	1.09 (SD = .08)	

Table 3.2. Mean KSPC for both user groups

Result show that there is a decrease of 4.5% for novice T9 users and 3.6% for experienced T9 users from respective multitap groups.

No statistically significant differences were found between the experienced and novice users of the multitap and T9.

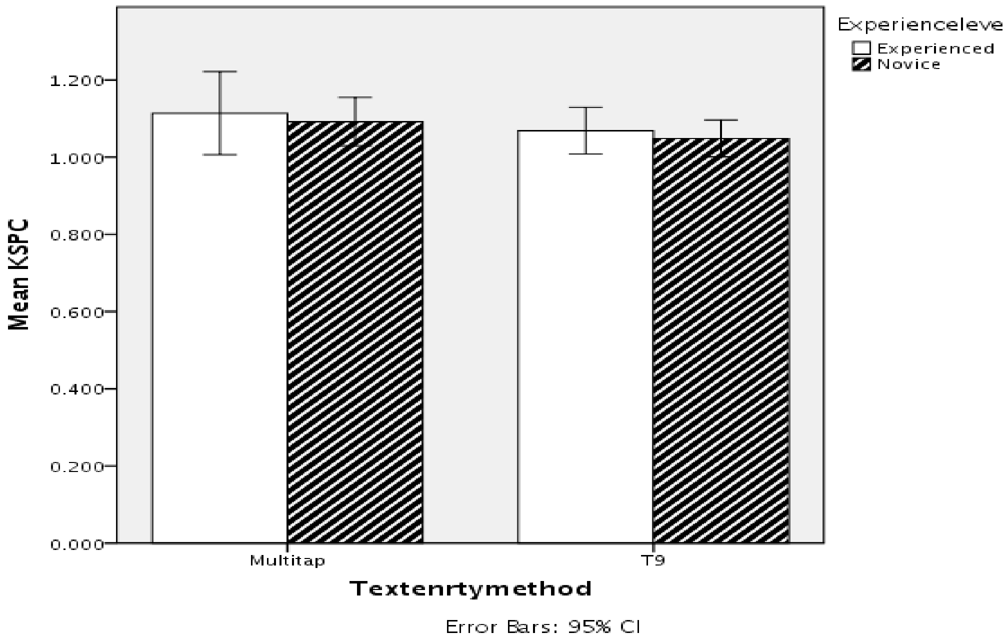

Fig 3.6 Interaction between Text entry method and Experience level for KSPC

Minimum number of 'uncorrected error count' was 0 for both text entry methods with maximum being 4 and 7 for multitap and T9 respectively. Average minimum string distance (MSD) calculated in terms of uncorrected error count for each group is shown in table 3.3.

Method	Novice	Experienced	Mean
Multitap	0.67 (SD = .82)	1.5 (SD = 1.76)	1.08
T9	2.5 (SD = 2.88)	0.83 (SD = .41)	1.67
Mean	1.58 (SD = 2.23)	1.17 (SD = 1.27)	

Table 3.3. Mean MSD for both user groups

Result show that there is a decrease of 44.7% for experienced T9 users from the respective multitap group. However, there is an increase of 273.1% for novice T9 users over the respective multitap group.

No statistically significant differences were found between the experienced and novice groups of the multitap and T9.

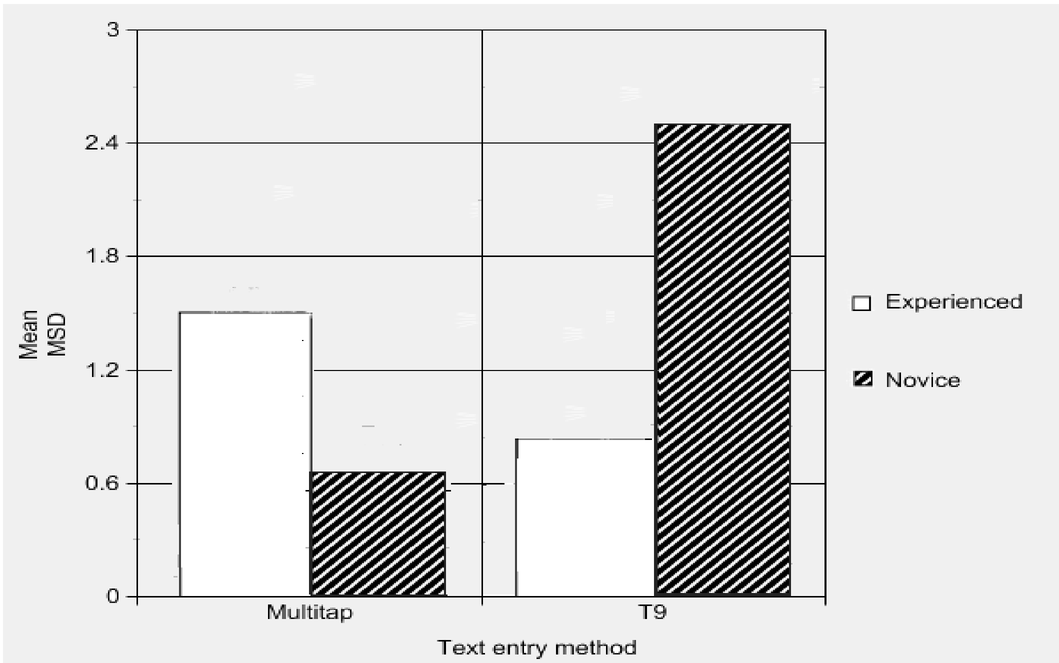

Fig 3.7 Interaction between Text entry method and Experience level for MSD

4 Discussion

The direction of the results of this experiment is the same as in the previous study by James *et al.* [2001] with experienced users being faster than novices and T9 being the faster text entry method in comparison to multitap. However, with a gap of 11 years between the two studies, the experience level of the participants and the usage of SMS messages has increased considerably.

The number of messages sent per week by the participants on average (23.8) was twice as large as reported by James *et al.* (11). Moreover, the criteria for being experienced and novice has also changed for the fact that there is no real novice user of the multitap method now. All the participants in our experiment had used the multitap method.

In addition, none of our participants had English as their first language compared to the previous study, where some of the participants had English as their native language, which also contributed towards the performance of the participants and hence the results. There could have been more difference in the results between the two studies had my participants been native speakers of English.

Table 4.1 shows the text entry rate for my study and the study by James *et al.*

	Novice		Experienced	
Method	Current study	Previous study	Current study	Previous study
Multitap	10.07	7.98	11.28	7.93
T9	14.11	9.09	22.67	20.36

Table 4.1 Mean WPM for both studies

The text entry rates have increased in comparison to the previous study for both user groups and both text entry methods. There is not much difference between the performance of the experienced and novice multitap groups in both the studies, although the novice group in the previous study were those who had never entered SMS messages on mobile phones. The possible explanation to this is that their experienced multitap users were not so experienced. Regarding the current study, the answer lies in the fact that our novice multitap users were not so novice as they all had used or were using multitap as the text entry method.

Comparing the results with the two models referred to in the previous study, our results are also closer to the predicted WPM figures in table 4.2 by Dunlop and Crossan based on the 'keystroke level model'. The predicted figures are compared with the performance of experienced users.

Method	Text Entry Rate
Multitap	14.9
Predictive Text Input	17.6

Table 4.2 Predicted wpm for Multi-tap and predictive text input i.e. T9 based on keystroke level model

The performance of our experienced multitap users is quite close to that predicted by the model (11.28 vs 14.9 respectively), however, our experienced T9 users performed comparatively better than predicted by the model (22.67 vs 17.6 respectively).

Our WPM figures are lower than predicted in the 'movement and linguistic model 'in table 4.3, however, the percentage increase in WPM for T9 over multitap is similar to the model prediction (101% vs 66-103%).

Method	Index Finger	Thumb
Multitap – time out	22.5	20.8
Multitap – time out kill	27.2	24.5
T9	45.7	40.6

Table 4.3 Predicted wpm for Multi-tap and T9 based on movement and linguistic model [Silfverberg *et al.*, 2000]

When comparing the error rates, the numbers are not directly comparable to the previous study due to the fact that in the previous study there were six different types of error recorded which are “misspelled word”, “missed next”, “over next”, “overshot”, “failed to pause”, and “other” . In this study, however, the error rate is calculated using MSD in terms of “uncorrected error count”.

When looking at the direction of the results, our novice T9 users have committed more errors than the novice multitap group, whereas it is otherwise in the previous study. In both studies, experienced T9 users have made less errors than experienced multitap users. The percentage difference of the results between both studies, however, is quite big for the novice and experienced users of both methods (273% increase vs 47.7% decrease for novice group, and 44.7% vs 70.7% decrease for experienced group).

	Novice		Experienced	
Method	Current study	Previous study	Current study	Previous study
Multitap	0.67	65	1.5	116
T9	2.5	44	0.83	34

Table 4.4 Mean error rate for both studies

The reason for novice T9 users committing more errors than novice multitap group is the fact that the output on the screen changes continuously when using T9 method. Since users are not familiar with the method, they consider it as if they have entered the wrong letter and so are constantly trying to correct it by pressing other letters, resulting in wrong output.

The KSPC figures for both the methods in our study is quite close to 1.

All novice users of T9 had tried to use T9 but had given up. The reason was that they did not understand how it worked because of the changing output. Therefor, while using T9 in this experiment, they were frustrated with the changing output on the screen. This may have contributed towards lower text entry rate and higher error rate.

This leads to the conclusions that any new method requires learning before using and many users avoid adopting it. Hence, the very first problem that any new text entry method faces is to convince the users that the writing method is simple, easy to learn, and requires less effort from the users. If it is not so, users might give it up after trying once, as was the case with T9.

5 Conclusions

I conducted an experiment to get insight into the performance of two well-known text entry methods, multitap and T9, with respect to the two user groups, novices and experienced users. Users were asked to enter short phrases for two minutes as fast and as accurately as possible. The results were then analyzed and compared to the results from a previous study.

Results showed that T9 users were faster than multitap users and experienced users were faster than novice users. However, there was not much difference found between the performance of the novice and experienced multitap groups. The reason was that all the users had had previous experience with the multitap method. Hence, there were no real novice multitap users.

My initial thought was that it might be comparatively easier to find novice multitap users because of it being slow, and difficult to find novice T9 users, which proved to be otherwise. Although T9 is one of the oldest method of text entry for mobile phones, it was discovered that there were many users who had not used it or had given up trying to learn it after using once. Since T9 is a text entry method which needs to be learned, the novice users of T9 were found easily.

References

[AllBlackBerry, 2007] AllBlackBerry, BlackBerry typing tips. Mobile Nations*, available at:* http://crackberry.com/blackberry-typing-tips *(accessed 01.01.2012)*.

[Amal Sirisena, 2002] Amal Sirisena , *Mobile Text Entry* . Honours report, University of Canterbury, November 2002. *Available at:* www.cosc.canterbury.ac.nz/research/reports/HonsReps/2002/hons_0201.pdf *(accessed 23.05.2012)*.

[Bellman and MacKenzie, 1998] Tom Bellman and I. Scott MacKenzie, A probabilistic character layout strategy for mobile text entry. In: *Proceedings of Graphics Interface '98*, pp. 168-176. Toronto: Canadian Information Processing Society.

[Castellucci, 2007] Steven John Castellucci, *Mobile text entry using ambiguous keypads: New method in a new toolkit*. Master's thesis, York university, May 2007. *Available at:* http://www.cse.yorku.ca/~stevenc/research.shtml *(accessed 01.01.2012)*.

[Dunlop and Crossan, 2000] Mark D Dunlop and Andrew Crossan, Predictive text entry methods for mobile phones. *Personal Technologies* **4** (2-3), 2000, 134-143.

[Eatoni.com, 2005] Eatoni.com, Image: CehlnstyKeypad.gif. Eatoni Ergonomics Inc, *available at:* http://www.eatoni.com/wiki/index.php/Image:CehlnstyKeypad.gif *(accessed 08.01.2012)*.

[Eatoni.com, 2009] Eatoni.com, Eqx- Converging the QWERTY keyboard with a telephone keypad. Eatoni Ergonomics Inc, *available at:* http://www.eatoni.com/wiki/index.php/EQx *(accessed 01.01.2012)*.

[Goldberg and Richardson, 1993] David Goldberg and Cate Richardson, Touch-typing with a stylus. In: *CHI '93 Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems.* New York: ACM.

[Gong and Tarasewich, 2005] Jun Gonag and Peter Tarasewich, Testing predictive text entry methods with constrained keypad designs. In: *Conference proceedings on Human factors in computing systems.* New York: ACM.

[Gong *et al.*, 2005] Jun Gong, Bryan Haggerty and Peter Tarasewich, An enhanced multitap text entry method with predictive next-letter highlighting. In: *CHI '05 extended abstracts on Human factors in computing systems*. New York: ACM.

[Green et al., 2004] Nathan Green, Jan Kruger, Chirag Faldu and Robert St. Amant, A reduced QWERTY keyboard for mobile text entry. In: *proceedings of extended abstracts on Human factors in computing systems.* New York: ACM.

[Hanlon, 2012] Mike Hanlon, Innovative fastap keyboard. Gizmag, *available at:* http://www.gizmag.com/go/1617/ *(accessed 01.01.2012).*

[Hwang and Lee, 2005] Sunyu Hwang and Geehyuk Lee, Qwerty-like 3x4 keypad layouts for mobile phone. In: *proceedings of extended abstracts on Human factors in computing systems.* New York: ACM.

[Isokoski, 1999] Poika Isokoski, *A minimal device-independent text input method.* Master's thesis, University of Tampere, November 1999. *Available at:* www.cs.uta.fi/reports/A-1999-14.ps.Z *(accessed 30.05.2012)*.

[Itu.int, 2010] itu.int, The world in 2010- The rise of 3G. ITU 2011, *available at:* http://www.itu.int/ITU-D/ict/material/FactsFigures2010.pdf *(accessed 01.01.2012).*

[Itu.int, 2001] itu.int, ITU-T: Telecommunication standardization sector of ITU: E.161, a*vailable at:* www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-E.161-200102-I!!PDF-E&type=items *(accessed 15.03.2012).*

[James and Reischel, 2001] Christina L. James and Kelly M. Reischel, Text input for mobile devices: comparing model prediction to actual performance. In: *Conference proceedings on Human factors in computing systems.* New York: ACM.

[Kristensson and Denby, 2009] Per Ola Kristensson and Leif C. Denby, Text entry performance of state of the art unconstrained handwriting recognition: A longitudinal user study. In: *Proceedings of the 27th ACM Conference on Human Factors in Computing Systems - CHI '09*. ACM Press: 567-570.

[Lyons *et al.,* 2004] Kent Lyons, Thad Starner, Daniel Plaisted, James Fusia, Amanda Lyons, Aaron Drew and E. W. Looney, Twiddler typing: One-handed chording text entry for mobile phones. In: *Proceedings of the SIGCHI conference on Human factors in computing systems- CHI '04.* New York: ACM.

[MacKenzie, 2009] I. Scott MacKenzie, The one-key challenge: searching for a fast one-key ext entry method. In: *Proceedings of the ACM Conference on Computers and Accessibility – ASSETS 2009*, pp. 91-98. New York: ACM.

[MacKenzie and Chang, 1999] I. Scott MacKenzie and Larry Chang, A performance comparison of two handwriting recognizers. *Interacting with Computers*, **11**, 1999, 283-297.

[MacKenzie and Read, 2007] I. Scott MacKenzie and Janet C. Read, Using paper mockups for evaluating soft keyboard layouts. In: *conference proceedings of the center for advanced studies on Collaborative research.* New York: ACM.

[MacKenzie and Soukoreff, 2002] I.Scott MacKenzie and R.William Soukoreff, Text entry for mobile computing: models and methods, theory and practice. *Human-computer interaction*, **17**, 2002, 147-198.

[MacKenzie and Tanaka-Ishii, 2007] I. Scott MacKenzie and Kumiko Tanaka-Ishii, (Ed.) *Text Entry Systems: Mobility, Accessibility, Usnaiversality.* Morgan Kaufmann, Boston, Amsterdam, 2007.

[MacKenzie and Zhang, 1999] I. Scott MacKenzie and Shawn X. Zhang, The design and evaluation of a high-performance soft keyboard. In; *Proceedings of the ACM Conference on Human Factors in Computing Systems – CHI '99*, pp. 25-31. New York:ACM.

[MacKenzie *et al.*, 2001] I. Scott MacKenzie, Hedy Kober, Derek Smith, Terry Jones and Eugene Skepner, Letterwise: prefix-based disambiguation for mobile text input. In: *Proceedings of the ACM Symposium on User Interface Software and Technology - UIST 2001,* pp. 111-120. New York: ACM.

[Mackenzie *et al.*, 2011] I. Scott MacKenzie, R. William Soukoreff and Joanna Helga, 1 thumb, 4 buttons, 20 words per minute: design and evaluation of H4-writer. In: *Proceedings of the ACM symposium on User interface software and technology,* New York: ACM.

[Mankoff and Abows, 1998] Jennifer Mankoff and Gregory D. Abowd, Cirrin: A word-level unistroke keyboard for pen input. In: *Proceedings of the 11th annual ACM symposium on User interface software and technology* - UIST '98. New York: ACM.

[Matias *et al.*, 1994] Edgar Matias, I. Scott MacKenzie and William Buxton, Half-

QWERTY: Typing with one hand using your two-handed skills. In: *companion proceedings of the ACM conference on Human Factors in Computing Systems - CHI '94*, pp. 51-52. New York: ACM.

[Matias *et al.*, 1993] Edgar Matias, I. Scott MacKenzie and William Buxton, Half-QWERTY: A one-handed keyboard facilitating skill transfer from QWERTY. In: *Proceedings of the ACM Conference on Human Factors in Computing Systems - INTERCHI '93*, pp. 88-94. New York: ACM.

[Oniszczak and MacKenzie, 2004] Aleks Oniszczak and I. Scott MacKenzie, A comparison of two input methods for keypads on mobile devices. In: *Proceedings of the third Nordic conference on Human-computer interaction - NordiCHI '04.* New York: ACM.

[Partridge *et al.*, 2002] Kurt Partridge, Saurav Chatterjee, Vibha Sazawal, Gaetano Borriello and Roy Want, TiltType: Accelerometer-supported text entry for very small devices. In: P*roceedings of the 15th annual ACM symposium on User interface software and technology* - UIST '02. New York: ACM.

[Perlin, 1998] Ken Perlin, Quikwriting: Continuous stylus-based text entry . In: *Proceedings of the 11th annual ACM symposium on User interface software and technology* - UIST '98. New York: ACM.

[Ryu and Cruz, 2005] Hokyoung Ryu and Katrina Cruz, LetterEase: Improving text entry on a handheld device via letter reassignment. In: *Conference proceedings on Computer-Human Interaction: Citizens Online: Considerations for Today and the Futur,* Canberra, Australia, 2005. CHISIG.

[Scott *et al.*, 2010] James Scott, Shahram Izadi, Leila Sadat Rezai, Dominika Ruszkowski, Xiaojun Bi and Ravin Balakrishnan, RearType: Text entry using keys on the back of a device. In: *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services - MobileHCI '10.* New York: ACM.

[Silfverberg *et al.*, 2000] Miika Silfverberg, I. Scott MacKenzie, and Panu Korhonen, Predicting text entry speed on mobile phones. In: *Proceedings of the ACM Conference on Human Factors in Computing Systems* - *CHI 2000*, pp. 9-16. New York: ACM.

[Snapkeys Ltd, 2011] snapkeys.com, Snapkeys 2i overview. Snapkeys limited, *available at :* http://snapkeys.com/technology/overview *(accessed 12.04.2012).*

[Soukoreff, 2002] Robert William Soukoreff, *Text entry for mobile systems: models, measures, and analyses for text entry research*. Master's thesis, York university, June 2002. *Available at:* soukoreff.com/academic/thesis.pdf *(accessed 12.04.2012).*

[Tanaka-Ishii *et al.*, 2002] Kumiko Tanaka-Ishii, Yusuke Inutsuka and Masato Takeichi, Entering text with a four-button device. In: *Proceedings of international conference on Computational linguistics* - *Volume 1*, Association for Computational Linguistics, *S*troudsburg, USA, 2002.

[Venolia and Neiberg, 1994] Dan Venolia and Forrest Neiberg, T-Cube: A fast, self-disclosing pen-based alphabet. In: *Proceedings of Conference companion on Human factors in computing systems* - CHI '94. New York: ACM .

[Wang *et al.,* 2010] Jingtao Wang, Shumin Zhai and John Canny, SHRIMP: Solving collision and out of vocabulary problems in mobile predictive input with motion gesture. In: *Proceedings of the 28th international conference on Human factors in computing systems* - *CHI '10.* New York: ACM.

[Ward *et al.,* 2000] David J. Ward, Alan F. Blackwell, and David J. C. MacKay, Dasher—

A data entry interface using continuous gestures and language models. In: *Proceedings of the 13th annual ACM symposium on User interface software and technology* - UIST '00. New York: ACM.

[Wigdor and Balakrishnan, 2003] Daniel Wigdor and Ravin Balakrishnan, TiltText: Using tilt for text input to mobile phones. In: *Proceedings of the 16th annual ACM symposium on User interface software and technology - UIST 2003*. New York: ACM.

[Wigdor and Balakrishnan, 2004] Daniel Wigdor and Ravin Balakrishnan, A comparison of consecutive and concurrent input text entry techniques for mobile phones. In: *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '04.* New York: ACM.

[Wikipedia, 2012] en.wikipwedia.org, SMS. Wikimedia Foundation, Inc. *available at: http://en.wikipedia.org/wiki/SMS (accessed 15.05.2012)*.

[Wobbrock *et al.*, 2003] Jacob O. Wobbrock, Brad A. Myers, and John A. Kembel, EdgeWrite: a stylus-based text entry method designed for high accuracy and stability of motion. In: *Proceedings of the 16th annual ACM symposium on User interface software and technology -UIST '03*. New York: ACM.

[Zhai *et al.*, 2000] Shumin Zhai , Michael Hunter and Barton A Smith, The metropolis keyboard – an exploration of quantitative techniques for virtual keyboard design . In: *Proceedings of the 13th annual ACM symposium on User interface software and technology* – UIST '00. New York: ACM.

Questionnaire

Please fill in this questionnaire and return it if you agree to participate in our experiment.

Name: ______________________

Program of study (if student): ______________________

Age: ___________

Native language: ______________________

Email address: ______________________

1. Do you write SMS messages on your mobile phone?

   Yes ☐ No ☐

2. How many messages, approximately, do you send in a day?

   ____

3. How many messages, approximately, do you send in a week?

   ____

4. What kind of mobile phone(s) do you own now or what keypad it has?

   12-key keypad phone ☐ Full Qwerty keypad phone ☐ Touch-screen phone ☐

   other, please name ______________________

5. Have you recently switched from 12-key keypad mobile phone to any other type?

Yes ☐ No ☐

5.1 If Yes, how long it has been since you switched and to which type of mobile phone?

____

6. Have you recently switched from any other type of mobile phone to 12-key keypad mobile phone?

Yes ☐ No ☐

6.1 If Yes, how long it has been since you switched and from which type of mobile phone?

____

7. During the last 1-3 months with a phone with a 12-key keypad:

7.1 How many messages, approximately, did you send in a day?

____

7.2 How many messages, approximately, did you send in a week?

____

8. Which method of text entry you use for composing the messages?

Multitap ☐ T9 ☐ Both ☐ other dictionary-based method, please name ________

8.1 If 'Both', which method you use more often for writing the messages?

Multitap ☐ T9 ☐

9. How long (in years) have you used these text entry methods in your primary mobile phone?

Multitap: __________ years

T9: __________ years