Honours Report

Mobile Phone Text Entry

Lee Butts Dr. Andy Cockburn

November 8, 2001

ABSTRACT

The rapid growth of Short Message Service (SMS) text messaging has meant that a fast and efficient text input method is a very important aspect of a mobile phone interface. The best way to test a new method is an empirical evaluation, but this is a time consuming and complex task. An alternative would be to predict the performance of a new method using a prediction model. Previous prediction models and evaluations of current text entry methods are discussed. The previous models have been shown to be highly inaccurate. A new prediction technique is presented that uses pilot study data and text analysis instead of the complex mathematical formulas of previous techniques.

Predictions for the multi-press with next and T9 input methods are compared to the actual performance results of an empirical evaluation. The empirical evaluation also compares the performance of the newer T9 system to that of the more common multi-press method. T9 was significantly faster when entering sentences containing only dictionary words. However, the two methods were not significantly different when entering a mix of sentences that contained dictionary and non-dictionary words. As T9 remembers non-dictionary words that a user enters, it was concluded that T9 was the better method. Subjective data reflected this with 60% of subjects stating that they preferred T9 over multi-press.

The prediction technique was found to be too inaccurate to be useful to interface designers. The main failing of this and previous techniques appears to be the inability to accurately predict the mental preparation factor of the text entry actions. Further work to develop a method to calculate such values is needed.

Keywords: Mobile Phones, Text Entry, Performance Prediction, Empirical Evaluation

CONTENTS

1. Introduction 4

2. Background Work 7
2.1 Current Text Entry Methods 7
2.1.1 Two-key 7
2.1.2 Multi-press Methods 7
2.1.3 Predictive Text Entry 8
2.2 Published Work 9
2.2.1 Keystroke Level Modelling Based Prediction 9
2.2.2 Fitts' Law Based Prediction 10
2.2.3 Comparing Predictions to Actual Performance 11

3. Predicting User Behaviour on Mobile Phones 13
3.1 User Data Based Prediction 13
3.2 Pilot Study to Determine Keypress Time 13
3.2.1 Results 15
3.2.2 Discussion 16
3.3 Text Analysis 16
3.3.1 Alternate Key Arrangements 17
3.4 Predictions 17

4. Empirical Evaluation 19
4.1 Experimental Design 19
4.2 Subjects 19
4.3 Procedure 19
4.4 Results 21
4.4.1 Learnability 22
4.4.2 Performance Data 22
4.4.3 Subjective Ratings 23
4.5 Discussion 23
4.6 Quality of Prediction Technique 24

5. Further Work 26

6. Conclusions 27

1. INTRODUCTION

The use of text messaging is growing every month. Between January 2000 and June 2001, Short Message Service (SMS) use grew from 4 billion to 20 billion messages sent per month (Figure 1.1). This is an increase of *400%* in 18 months. The GSM Association also predicts 25 billion messages per month by December 2001 with 200 billion in total for 2001 (GSM Association Press Release 2001).

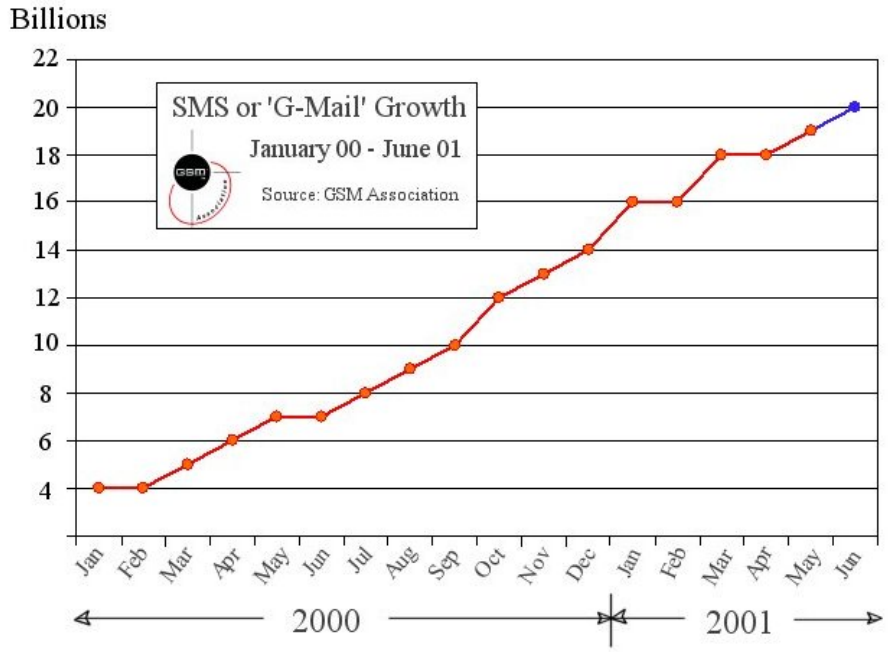


Fig. 1.1: World-wide growth of SMS use.

Mobile phones, however, are not naturally suited to text input. The standard (ISO/IEC 9995-8 1994) layout of a mobile phone uses 12–15 keys to allow basic text input (Figure 1.2). These 12–15 keys must cover the 26 letters, numerous punctuation marks and numbers used in the English language. Each key is overloaded, so that, for example, the 6-key is mapped to M, N and O. Additional special characters and punctuation may also be assigned to each key.

This creates a need for methods that allow the user to specify which letter on a particular key they want when it is pressed. For example, if a user presses

Fig. 1.2: A standard ISO 12-key keypad as found on the Nokia 5110.

the number 2, the mobile phone does not know whether the user wants an A, B or C. Several methods have been implemented to solve this disambiguation problem. Section 2.1 discusses the various approaches.

The best way to evaluate a technique is using an empirical evaluation. This is by no means an easy task. There are a large number of factors to take into account, such as the test sentences used, the apparatus used, how to accurately record results, the number of sessions each subject participates in and the amount of previous experience the subjects have. Any one of these factors could affect the results significantly.

If multiple methods are then added to the evaluation, in order to compare relative performance, complexity increases further. New factors appear, such as which order to present the techniques in, possible bias of the test sentences to a certain method, learning effects between similar methods and more apparatus issues.

In order to avoid a complex and time consuming empirical evaluation, it would be desirable to be to able predict a method's performance using a theoretical model. This would not only allow a quick evaluation of a single method, but also a quick comparison of multiple methods, depending on how generic the prediction model is. Previous models have been based on precise mathematical formulas for movement such as Fitts' Law and Keystroke Level Modelling(KLM) resulting in multiple, method specific formulas. They have been surprisingly inaccurate.

The next chapter details the current text entry methods and covers the previous work in the area of performance prediction and input method evaluation. A new approach to prediction is then presented (Chapter 3) and an empirical evaluation to determine its effectiveness described. The results of the evaluation are then presented, discussed and compared to the model predictions in Chapter 4. This leads to ideas for further research (Chapter 5) and is followed by the

conclusions in Chapter 6.

2. BACKGROUND WORK

This chapter first reviews the current commercial entry methods then discusses previous research in the area of text entry on mobile phones with regard to prediction and empirical evaluation.

2.1 Current Text Entry Methods

2.1.1 Two-key

To enter a character using two-key, the user specifies the position of the desired letter on that key. For example, the letter "N" is on the 6-key in the second position, so the user would press "62" to enter an "N". Likewise, a "G" would be entered using "41" (4-key, first position). This system means that every letter takes exactly two presses to enter.

Two-key is not suited for entering punctuation or special characters, as the user needs to be able to see all the letters mapped to each key in order to determine the position of the key they desire.

2.1.2 Multi-press Methods

A multi-press method works by cycling through letters on a key with each successive press. Each letter can require from one to three presses of a key (or more for certain letters and punctuation). For example, one press of the 2-key enters an "A", a second press of the 2-key changes the "A" to a "B" and a third to "C".

This causes problems when two letters on the same key are entered. The sequence 222 may mean the user wants to enter "ABC" or it can also be interpreted as "C". There are two common solutions to this problem which are described below.

Multi-press with time-out

In order for a user to enter "AB" for example, multi-press with time-out uses a fixed time-out to decide when a user has finished cycling through letters on a key. Once the user presses the 2-key a time-out starts and if the 2-key is pressed before the time-out expires (usually 1–1.5 seconds), the interface will cycle through the letters "A - B - C" until the time-out is left to expire. Once the time-out has expired, pressing the 2-key again will enter another "A" into the string. In this way "ABC" can be entered using the key sequence 2-22-222 where a dash signifies waiting for the time-out to expire.

Multi-press with next button

Another technique replaces the time-out with a 'next' button. Instead of having to wait between successive letters, the user presses the next button to signify that they have finished cycling through letters on that key. To enter the same string "ABC" using the next button method requires "2<next>22<next>222" to be pressed. This method is often combined with a time-out system (e.g. the Phillips *Savvy*) to give users the choice of waiting for the time-out or manually pressing the next button.

2.1.3 Predictive Text Entry

A predictive text entry method aims for one keypress per character. It does this by comparing the words that a sequence of keypresses can represent and guessing which one was intended depending on word probabilities. For example, the key sequence 2-2-5-3 can represent *able*, *cake*, *bald* or *calf*. A predictive method would suggest these words in order from the most to least likely word. These systems store a dictionary of known words and statistical information to use when ranking possible words. They also must handle words that are not in the dictionary. This is usually done by entering the word using a multi-press technique and then adding it to the dictionary.

T9

T9®by Tegic Communications is a word-based predictive method found on many commonly used phones (e.g. the Nokia 3210). After each word has been entered (any number of keypresses followed by a space) the user is presented with the most likely word. If this is not the desired word the user presses the down arrow or similar "next" key to cycle through the possible words. If the desired word is not in the list the user must delete the last group of keypresses, change to multi-press mode and enter the word. The word is then stored in the dictionary. Kober, Skepner, Jones, Gutowitz & MacKenzie (2001) state that when the assumptions used by previous research into T9 are removed (such as the user making no errors), T9's performance degrades significantly.

LetterWise

LetterWise™by Eatoni Ergonomics takes a slightly different approach by using prefix-based disambiguation to make its predictions letter by letter. It predicts which letter is meant by a certain keypress based on the probability that a certain letter will follow the current set of letters. For example, if the text "th" has been entered, the next letter will very likely be an 'e'. If the user does not want an 'e' they cycle through the 3 or 4 possible letters for that key until the correct letter is displayed. The statistical information is stored in a dictionary structure similar to T9. However, because LetterWise works letter by letter, there is no issue of "unknown" words. MacKenzie, Kober, Smith, Jones & Skepner (2001) report a 50% drop in required keystrokes and a 36% increase in entry rate after ten hours use.

WordWise

A second method developed by Eatoni is WordWise™ which takes a rather different approach. It uses a system where the letters 'c', 'e', 'h', 'l', 'n', 's', 't', and 'y' are entered unambiguously using an auxiliary key (like a "shift" key on a standard keyboard). For example, the 5-key is mapped to the letters 'j', 'k' and 'l'. Pressing 5 by itself means the user wishes to enter a 'j' or a 'k', while holding the auxiliary key and pressing 5 will always enter an 'l'. This allows words like "yes" to be entered without the system having to make any predictions. It also reduces the number of possible meanings of a group of key presses as the letters that are entered unambiguously narrow the possibilities. For example, the word "today" would be entered as <aux>8-6-3-2-<aux>9 which tells the system that the user wants a word matching the pattern "t???y". This reduces the amount of searching the system needs to do as well as improving the likelihood that the system predicts the word on the first attempt. The more often a predictive system can guess the right word without the user having to cycle through the possibilities, the closer the system will approach the one keypress per character goal. Entire words (such as "the" and "then") can also be entered unambiguously, which means they do not need to be stored in the dictionary. However, there is still the issue of unknown words, which must be entered using multi-press, as with T9.

2.2 Published Work

2.2.1 Keystroke Level Modelling Based Prediction

Dunlop & Crossan (2000) propose a predictive method very similar to the licensed T9 system. They propose a further extension whereby word completion is offered to speed up text entry. Dunlop & Crossan (2000) use Keystroke Level Modelling (Card, Moran & Newell 1980) (KLM) to create prediction models to compare theoretical multi-press entry speed to that of their proposed system (with and without word completion). They based their models on three factors:

- T_k – the time taken to press a key. This was assumed to be 0.28 seconds. This value was suggested in the original KLM paper as being the typical speed of an "average non-secretary typist".
- T_h – Homing time for the hand to move to the keyboard. This was fixed at 0.4 seconds.
- T_m – Mental preparation time for executing physical actions.

When using multi-press, the time to enter a given phrase (P) involves a homing action followed by the entry of w words. Each word contains k_t keypresses on average and d delays. Average word length is assumed to be 5.98 characters which was derived from experimental text (six months' of newspaper articles). Therefore the model for predicting multi-press performance is:

$$T(P) = T_h + w(k_t T_k + dT_m)$$

Predictive text input is assumed to comprise of one keypress per character, implying that the number of keystrokes per word, k_p is equal to the average

Method	Predicted Speed (wpm)
Multi-press	14.9
Predictive - word based	17.6
Predictive - word completion	7.7

Tab. 2.1: The predictions of Dunlop & Crossan (2000) for Multi-press, Predictive and Predictive with word completion methods.

word length of 5.98. Each word also requires l presses of the end-of-word key, on average, to select the correct prediction. The model for predictive input is:

$$T(P) = T_h + w(k_p T_k + l(T_m + T_k))$$

When word completion is added the predictive model is changed to include k_c, the average number of keys required per word (determined from text analysis), which is 4.60 for the average word length of 5.98. The modified model for word completion is:

$$T(P) = T_h + w(k_p T_k + (k_c - 1)(T_m + T_k))$$

From the assumptions above predictions were made for each method as shown in Table 2.1.

These results predict an 18% improvement from a multi-press to a predictive method. An empirical study using fourteen subjects was also completed to gather data which could be used to evalute the KLM models. The study showed a lower than predicted improvement of 10% when using the predictive system.

2.2.2 Fitts' Law Based Prediction

Silfverberg, MacKenzie & Korhonen (2000) take a different approach to model prediction by applying Fitts' Law (Fitts 1954) to movement on the mobile phone keypad. The paper presents models for four input methods: two-key, multi-press with time-out, multi-press with next button and T9.

The models are built from a series of movement times (from Fitts' Law), MT_i, which are calculated using the following formula:

$$MT = a + b \log_2(A/W + 1)$$

The two constants, a and b, are determined empirically. The log term is known as the index of difficulty and is based on A, the length (amplitude) of the movement, and W the width of the target. The smallest dimension of each mobile phone key was used for W.

Each model is based on a formula for CT which is the time it takes to enter one character. CT is defined as the sum of all the required movements:

$$CT_{ij} = \sum MT_k$$

The first model presented is that for multi-press with time-out. It uses an assumed time-out, $T_{time-out}$ of 1.5 seconds as found on Nokia mobile phones. The model consists of MT_0, the initial movement of a subject's finger or thumb to the desired key. N is the number of key repetitions needed to select the desired character and MT_{repeat} is the time taken to press a key when a subject already has their finger on it. The final formula for multi-press is:

$$CT = MT_0 + N \times MT_{repeat} + T_{time-out}$$

A second multi-press variation, where a next button is used to override the time-out, requires modification of the previous formula by replacing the time taken for the time-out to expire with MT_{kill}, which is the time taken to press the next button. The new formula is:

$$CT = MT_0 + N \times MT_{repeat} + MT_{kill}$$

Two-key is a much simpler formula as all keypresses require two movements, independent of the desired letter. This leads to the following formula for two-key:

$$CT = MT_0 + MT_1$$

The assumption is made that a user would never need to cycle through the predicted words when using T9, so each letter only requires one keypress. This is known as *perfect disambiguation*, a rather unrealistic assumption, and makes the formula for T9 very simple. The formula for T9 is:

$$CT = MT_0$$

In order to obtain predictions from these formulae they need to be combined with a linguistic model to calculate letter-pair probabilities based on frequencies in common English. The combination produces a formula for the average character entry time (using the language l) for a particular method. This is the weighted average of character entry times for all letter-pair combinations:

$$CT_l = \sum \sum (P_{ij} \times C_{ij})$$

This value could then be used to calculate word per minute speeds by taking the inverse, multiplying it by 60 seconds and then dividing it by an average word length of 5 characters per word. Note that the average word length differs from the value used by Dunlop & Crossan.

As mentioned previously, Fitts' Law requires two constants a and b which are specific to each type of input. The authors completed a user study to find values for a and b for one-handed input with a thumb and two-handed input with an index finger. Once these values were known predictions could be made. These predictions, which were not compared to a user study, are shown in Table 2.2.

Method	Predicted Speed (wpm)	
	Index Finger	Thumb
Two-key	25.0	22.2
Multi-press time-out	22.5	20.8
Multi-press next	27.2	24.5
T9	45.7	40.6

Tab. 2.2: The predictions of Silfverberg et al. (2000) for two-key, the two multi-press variants and T9.

Paper	**Method**			
	Two-Key	**MP - Time-out**	**MP - Next**	**T9**
Predictions				
Dunlop & Crossan (2000)	N/A	14.9		17.6
Silfverberg et al. (2000)	25.0	22.5	27.2	45.7
User Performance				
James & Reischel (2001)	N/A	7.98, 7.93		9.09, 20.36

Tab. 2.3: A summary of previous predictions and empirical results.

2.2.3 Comparing Predictions to Actual Performance

A third paper that is of interest appeared in the proceedings of CHI2001. It took the predictions of the previous two papers and compared them to results from an empirical evaluation using twenty subjects. They found that although the models predicted the quickest method correctly, the predicted speeds were significantly higher than the actual performance. They concluded that the models made predictions based on unrealistic expert behavior. They speculated that the errors may be in the mental preparation factor of the model of Dunlop & Crossan and that the models of Silfverberg et al. presume a level of expertise that is unrealistic, such as their presumption that the user makes no errors. Table 2.3 shows the difference between the model predictions of the previous papers and the actual user performance gathered by the empirical evaluation for novice and expert users. The predicted speeds for two-handed input have been used for Silfverberg et al. (2000).

3. PREDICTING USER BEHAVIOUR ON MOBILE PHONES

James & Reischel (2001) showed that previous techniques made predictions for predictive text entry methods based on unrealistic expert behaviour. As such, predictions were vastly different to actual user performance. They stated that the errors were most likely in the mental preparation element, and that they also lacked a component for verifying the input and correcting errors. This chapter presents a predictive model that aims to overcome the unrealistic assumptions and missing components of the previous models, by predicting average user behaviour based on previous empirical data.

3.1 User Data Based Prediction

In order to avoid predicting expert behaviour, this technique bases its predictions on average user performance data gathered from an empirical study. This data is used to find the average time users take to press a key. This time includes any mental preparation time needed. Once this value has been determined, the average number of keypresses required to enter a character with a particular method needs to be calculated. This can be done via text analysis as described in Section 3.3. By combining the average time taken to press a key and the average number of keypresses per character, a words per minute speed can be calculated.

The predictions made should represent the performance of an average user with a new method. This technique is significantly less complex than the previous models.

The following section describes the pilot study used to collect the empirical data and is followed by details of the text analysis process used to calculate the average number of keypresses. Predictions were made for the T9 method in order to test the accuracy of this technique. The procedure for making a prediction and the predictions made are described in Section 3.4. The predictions are compared to the results of the larger empirical study described in Chapter 4. The accuracy of the technique is discussed in Section 4.6.

3.2 Pilot Study to Determine Keypress Time

The goal of the study was to gather data that could be used to determine the average time it takes to press a key using three different methods: two-key, multi-press with time-out and multi-press with next. The study also evaluated these techniques to determine the best method and possible entry speeds.

Experimental Design

The experimental design is a repeated measures two-factor analysis of variance. The two factors are 'interface type' (two-key, multi-press with time-out, multi-press with next) and 'user experience' (novice, intermediate, expert).

Subjects

Eight participants were used in the study. All were male and studied computer science at the University of Canterbury. Their experience with mobile phones and SMS messaging was graded into three levels: novice (no experience), intermediate (0 - 5 messages per week) and expert (5+ messages per week). In total there were three novice, three intermediate and two expert subjects.

Procedure

The subjects were asked to enter five sentences using each input method. They used the numeric keypad of a keyboard relabelled to match the layout of a standard mobile phone keypad. The subject's input appeared on screen via a simple emulator written in Tcl/Tk. Subjects were allowed to practice with each input method until they felt comfortable using it. This usually took about 30 seconds. Any practice sentences sent were not recorded in the results file.

There were three versions of the emulator, one for each input method. Each emulator recorded the elapsed time from the first keypress to the 'Send' key. The timer was reset so that the time taken between sentences was not recorded. There was an error in the multi-press with time-out emulator which caused the time between sentences to be recorded in *some* cases. This was no more than five seconds, but may have affected the results for the multi-press with time-out method. The subject number, input method, elapsed time and entered text were written to a file after each sentence.

The sentences used were taken from James & Reischel (2001) with the aim of reproducing their results. The sentences are listed in Table 3.2. They are conversational sentences intended to mimic real usage. Many subjects, however, commented that the sentences were too long, not realistic and tedious to enter. The use of mobile phones for text input has led to a new language of abbreviations such as "r" instead of "are" and "c u l8r" instead of "see you later". Experienced users suggested how they would have entered the sentences using such common abbreviations. James & Reischel (2001) also use a set of 'newspaper' sentences. These were long sentences similar to ones found in a newspaper article. These were not used in this study because they are not realistic and would have caused each experiment to take twice as long, which would have violated the time constraints.

The order of methods used was counter-balanced in order to minimize any learning effects. This was important as the multi-press variations were very similar.

After the user had completed a set of five sentences, they were asked to complete three questions regarding the learnability, error-rate and efficiency of

the input method they had just used. The Likert scale questions (1 disagree, 5 agree) are shown in Table 3.1.

Q1	I found this input method easy to learn
Q2	I did not make many mistakes with this input method
Q3	Overall, this input method was efficient to use

Tab. 3.1: Subjective questions asked after each input method.

No.	Sentence
1	hi joe how are you want to meet tonight
2	want to go to the movies with sue and me
3	what show do you want to see
4	we are meeting in front of the theater at eight
5	let me know if we should wait

Tab. 3.2: The five test sentences used.

After the third set of sentences, and having used all three input methods, each subject was given the option to modify their previous answers. Several subjects commented that this was useful because their impression of a certain method had changed after using one of the others.

Each subject took about 20-30 minutes to complete the tasks, depending on their experience.

3.2.1 Results

Many subjects found the tasks tedious and frustrating. This is a well-known 'feature' of text messaging. The choice of sentences may have increased subject frustration, with many complaining of their length and American spelling. The subjects, however, understood the reasons for using the sentences.

Performance Measures

Words per minute (wpm) speeds were calculated by dividing the time for each sentence by the number of characters entered, then multiplying this figure by $\frac{60}{5}$, where five is the average length of a word (MacKenzie 2001*a*). Multi-press with next was the fastest method. The mean speed for the multi-press with time-out, multi-press with next button, and two-key were 9.3 (s.d. 3.26), 10.23 (s.d. 3.26) and 7.79 (s.d. 2.35) wpm. There is a significant difference between these means ($F(2, 14) = 12.36$, $p < 0.05$).

As expected, expert users were the fastest, followed by intermediate and then novice users. The mean speeds for novice, intermediate and expert users

were 6.9 (s.d. 1.19), 9.1 (s.d. 2.89) and 12.4 (s.d. 2.18) words per minute. This was a significant difference ($F(2,4) = 26.05$, $p < 0.05$).

Subjective Ratings

The five point Likert scale questions (1 disagree, 5 agree) gave insight into the learnability, error-rate and efficiency of the three methods. The mean response with regard to learnability for multi-press with time-out, multi-press with next and two-key were 4.25 (s.d. 0.46), 4.13 (s.d. 0.83) and 3.75 (s.d. 1.03) respectively. There was no significant difference between the three methods (Friedman Test, $\chi_r^2 = 1.31$, $df = 2$, $N = 8$, $p = 0.52$).

The mean response with regard to error rate was 3.88 (s.d. 0.84) for multi-press with time-out, 4.31 (s.d. 1.03) for multi-press with next and 3.5 (s.d. 1.31) for two-key. This was not significantly different (Friedman Test, $\chi_r^2 = 1.56$, $df = 2$, $N = 8$, $p = 0.46$).

When asked about efficiency the mean response was 3.38 (s.d. 0.9) for multi-press with time-out, 3.75 (s.d. 1.04) for multi-press with next and 3.0 (s.d. 0.93) for two-key. Once again this was not significantly different (Friedman Test, $\chi_r^2 = 2.69$, $df = 2$, $N = 8$, $p = 0.26$).

3.2.2 Discussion

This pilot study confirms the conclusion of James & Reischel (2001) that although a valuable tool to interface designers, mathematical models can be misleading if not compared to actual performance. This is proven by the significant difference between the results of this study and the model predictions of Silfverberg et al. (2000). Expert subjects appeared to have quite strong habits from using a particular input method regularly, which points to the obvious learning effects of regular use. The study found that multi-press with next button was quickest (10.23 wpm), followed by multi-press with time-out (9.3 wpm) and two-key (7.79). Several subjects believed that their performance would improve with regular use of a particular input method, especially two-key. Results showed that experience had a significant effect on input speed. Subjects' preferred method varied, with no significant difference being found.

This empirical data allowed the average time taken to press a key to be derived. The derivation process is detailed in Section 3.4.

3.3 Text Analysis

The aim of this new technique is to provide accurate prediction using a generic formula. Specific values for a particular method can then be applied to this formula. The generic formula for a prediction using the new technique is:

$$Speed = \frac{60}{T_{kp} \times N_{kc} \times N_w} \tag{3.1}$$

where T_{kp} is the time taken to press a key, N_{kc} is the average number of keypresses per character, and N_w is the average number of characters per word.

1	2 zyx	3 wvu
4 tsr	5 qpo	6 nml
7 hfdc	8 aik	9 egjb
	0	

Fig. 3.1: An alternate letter allocation scheme

N_w is assumed to be 5 for all calculations (MacKenzie 2001*a*). The speed of multi-press with next was known from the pilot study and by calculating the average number of keypresses per character for this method it is possible to derive the average time needed to press a key, which could be used to predict the performance of T9.

To calculate the average number of keypresses per character, a small experimental application was written. The application read in a text file and simulated entering the text using the multi-press with next button. It recorded the number of keypresses required and reported the average number of keypresses per character. This was 1.953 using the full text of the book "Alice In Wonderland" from the Canterbury Corpus (`http://www.corpus.canterbury.ac.nz/`).

3.3.1 Alternate Key Arrangements

As an aside, alternate letter allocation when using the multi-press system was investigated. This entailed assigning different combinations of letters to the keys of a mobile phone keypad and recording the effect on the average number of keypresses per character. Unfortunately the simulation process took too long to complete to find the optimum layout but partial results did show significant gains by using alternate layouts to the ISO standard. For example the layout shown in Figure 3.1 reduced the average number of keypresses per character to 1.508.

There could be negative effects to rearranging the letters to find an optimal solution. Firstly, going against a recognised standard would discourage manufacturers from using the new layout, and, secondly, it would require users to re-learn the non-alphabetic layout, which would initially decrease entry speed. There is a significant amount of possible future work in this area, but it was a minor issue with regard to the focus of this report.

3.4 Predictions

Now that N_{kc} (the average number of keypresses per character) is know for the multi-press system, a value for T_{kp} (the time take to press a key) can be cal-

culated. T_{kp} is set at 0.715 to make the prediction match the empirical result from the pilot study. This resulted in a prediction for multi-press with next of 8.59 wpm. The derived value of T_{kp} value was then used for the T9 predictions.

However, in order to predict the speed of T9 a value for N_{kc} was needed. The models of Dunlop & Crossan (2000) and Silfverberg et al. (2000) used 1.204 and 1 respectively. Silfverberg et al. (2000) assumed that T9 guessed the correct word every time and that the user never had to cycle through the list of words. This is an unrealistic assumption, so the value from Dunlop & Crossan that includes having to press the next key on a certain proportion of the words was used. Dunlop & Crossan (2000) calculated this value by recording the position in the possible word list of all words in their experimental text. The average position was found to be 1.03, which equates to 6.03 keys per word, which in turn equates to 1.204 presses per character, assuming 5 characters per word.

Now that N_{kc} was known, the predicted speed for T9 when entering sentences that contained only dictionary words was calculated to be 13.9 wpm. A prediction for sentences that contained non-dictionary words was attempted, but a value for N_{kc} was not easily calculated. It required knowing, on average, how many words entered by users were not in T9's dictionary. As the contents of T9's dictionary are unknown, this was not possible. It was expected that sentences with non-dictionary words would require slightly fewer presses per character than multi-press with next and N_{kc} was set to 1.85 for non-dictionary sentences. This resulted in a speed prediction of 9.07 wpm for T9 when entering sentences that contained non-dictionary words.

Once Equation 3.1 had been used to make predictions for multi-press with next (8.59 wpm) and T9 (13.9 wpm for dictionary sentences and 9.07 wpm for non-dictionary sentences), a larger empirical evaluation was used to measure the accuracy of the method and to gather speed data for the T9 system. It was decided that these predictions could be called accurate if they were within 10% of the actual performance.

4. EMPIRICAL EVALUATION

The aim of the evaluation is to determine whether there are reliable differences between the efficiency of the two text-entry techniques, and to determine the effectiveness of the prediction technique described in Chapter 3.

4.1 Experimental Design

The experimental design is a repeated measures two-factor analysis of variance. The two factors are 'text entry interface' with two levels (multi-press with next and T9) , and 'sentence type' with two levels (dictionary words and non-dictionary words).

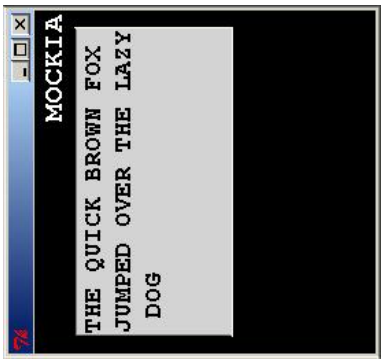
4.2 Subjects

Fifteen participants were used in the study. All were members of the Computer Science Department at the University of Canterbury. The repeated-measures design of the experiment reduces the impact of individual variability on the data analysis.

4.3 Procedure

Each subject was asked to enter five sentences using each input method. The order of interfaces used by each subject was counter-balanced to negate learning effects. The subject's input appeared on screen via an emulator. The multi-press emulator was written in Tcl/Tk by the author, while the T9 emulator (version 5.1) was provided by Tegic Communications. Screen-shots of the two emulators can be seen in Figures 1(a) and 1(b). The subjects interacted with the emulators via a keyboard which had been relabelled to match the ISO standard keypad and keys needed for the T9 emulator. The keyboard layout is shown in Figure 4.2. All unnecessary keys were removed from the right hand side of the keyboard and it was covered with a cardboard cover to clearly show the subjects which keys were available for them to use.

Subjects were initially asked to enter "my name is " and then their name using a particular interface with data logging turned on. The subjects were given no instruction on how the particular interface worked in order to gather data about initial reactions to each interface. This was also to simulate a customer trying a mobile phone text method in a retail store where no instruction would be available. If the subject had not successfully entered the message within two minutes they were told to stop. At this point each subject was given a demonstration of the interface and then, as a training exercise, was asked to enter



(a)

(b)

Fig. 4.1: Screen shots of the multi-press emulator (a) and the T9 emulator (b).

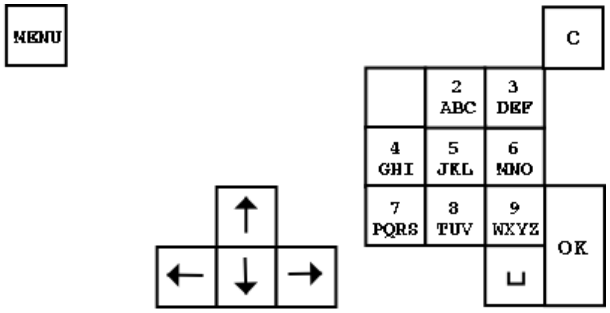


Fig. 4.2: The relabelled keyboard used in the evaluation.

two training sentences. These sentences were chosen to highlight the important features of the two interfaces, such as the use of the next key for multi-press and the need to cycle through the list of possible words using T9. The recorded data from these sentences was not used.

The subject was then asked to enter two sentences: "whats for dinner" and "ill be home at six". The sentences simulate a short, realistic text message conversation. A further two sentences were entered containing some local words that would not be in the dictionary used by T9. Table 4.1 shows the sentences used in the experiment.

Tab. 4.1: Test sentences used in experiment.

	Sentence
Initial Reaction	"my name is <subject name>"
Training	"what are you doing" "im still at mikes"
Dictionary words	"whats for dinner" "ill be home at six"
Non-dictionary Words	"goin to timaru" "im on weka st"

4.4 Results

Once subjects had been instructed on how to use each method, they were able to enter all sentences successfully. The use of short, informal sentences seemed to reduce the frustration of subjects that had occurred in the pilot study. Removing unnecessary keys from the keyboard and using a cardboard cover worked very well and stopped the users hitting keys such as Num Lock which stopped the multi-press emulator from operating correctly.

4.4.1 Learnability

Giving the subjects a chance to work out each method without instruction was a very interesting exercise. Multi-press was easily understood using trial and error, with subjects pressing one key a number of times and realising that they needed to cycle through the letters on a key. T9, in comparison, produced some interesting responses from subjects. Most tried pressing one key multiple times, which produces very confusing output using T9. Because the subjects were asked to enter "my name is..." most would start experimenting with the 6 key which holds the letter 'M'. However, multiple presses of this key resulted in "o", "on", "non" and "noon", which completely confused the subjects. Common comments at this stage where "What the...????" and "Huh?". Subjects then experimented with other keys, with similarly confusing output. A majority of subjects eventually discovered that they only needed to press a key once for the required letter, but very few discovered that they could use the arrow keys to scroll through the list of words. One subject entered the initial sentence by entering a word one letter at a time and then cycling through the possible letters on each key.

Nine (60%) of the subjects entered the initial sentence ("my name is...") successfully using T9, taking a mean time of 62.6 seconds (s.d. 30.4). The six that were unable to either gave up or reached the two minute time limit. This was due to the fluctuating output, as described above, and the complex nature of the method.

All subjects successfully entered the initial sentence using multi-press and took a mean of 42.1 seconds (s.d. 36.0). It was interesting to note that previous experience with text entry appeared to hamper the subjects' ability to learn an unfamiliar method. For example, one subject who had never tried text entry on a mobile phone instinctively used one keypress per letter for T9 and was the quickest to understand the method. In contrast, experienced subjects who use, or have used, a multi-press method found it much harder to use T9.

4.4.2 Performance Data

Once a method had been explained to a subject, most stated that they did not need to enter the training sentences and wanted to start on the recorded set. This had the positive effect of shorter sessions and less subject frustration. The overall mean speeds in words per minute for multi-press and T9 were 7.78 (S.D. 2.71) and 7.72 (S.D. 4.76) respectively. There was no significant difference ($F(1, 14) = 0.09$, $p = 0.93$) between these results.

The mean speeds for multi-press and T9 for the sentences containing only dictionary words were 7.15 wpm (s.d. 2.73) and 10.53 wpm (s.d. 5.16) respectively. This was a significant difference ($F(1, 14) = 11.61$, $p < 0.05$). The mean speeds for the sentences containing some non-dictionary words were 8.40 wpm (s.d. 2.6) for multi-press and 4.90 wpm (s.d. 1.85) for T9. This was also a significant difference ($F(1, 14) = 47.75$, $p < 0.05$). Figure 4.3 shows the mean speeds for each method for the two sentence types.

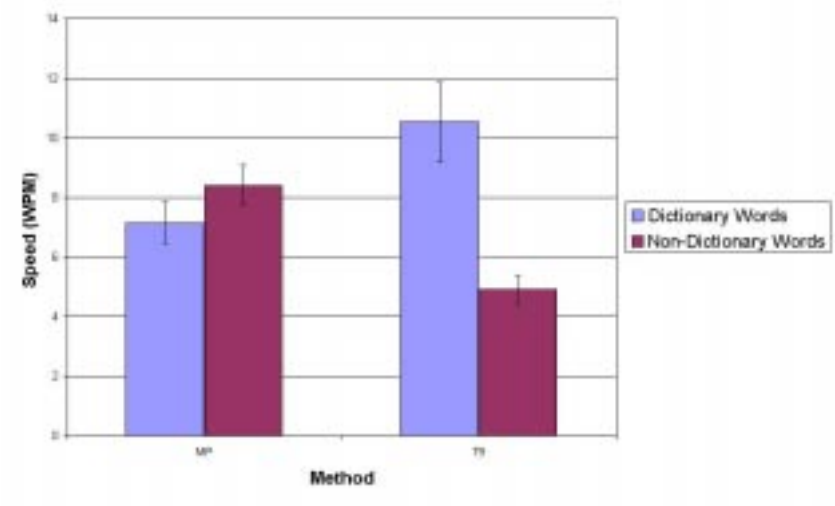


Fig. 4.3: The speed results of the main evaluation.

4.4.3 Subjective Ratings

At the conclusion of each session the subject was asked if they had a preference for either method. Eleven of the fifteen subjects (73%) preferred T9, three (20%) multi-press and one (7%) preferred neither. Discarding the 'no preference' result, this was marginally significant ($\chi^2 = 3.5$, $df = 1$, $p = 0.06$). Several made the comment that it was "annoying" to have to switch to multi-press mode when entering non-dictionary words using T9, and they only preferred T9 for entering dictionary words. However these subjects responded positively when told that T9 would remember words entered in this way and they would only need to be entered once using multi-press. One subject felt that he had to think a lot more when using T9, whereas he described multi-press as being "mechanical" and requiring a lot less cognitive effort. The subject who preferred neither said this was because they felt they needed more practice with both before they could choose one method over the other.

4.5 Discussion

The overall result shows an interesting equivalence between the two methods. However, If the sentences are broken down into those containing only dictionary words and those containing non-dictionary words, such as abbreviations and local place names, there is quite a different result. T9 is significantly faster than multi-press for dictionary sentences, while multi-press is significantly faster than T9 for the non-dictionary sentences. It must be noted, however, that T9 will only perform badly for non-dictionary words the first time they are entered into the dictionary. Therefore it could be assumed that as time passed a user's performance with T9 would approach the dictionary sentence mean speed of 10.53 wpm. Subjective measures show a preference for T9 over multi-press. When using T9, most users also began to anticipate which words would require

them to switch to multi-press mode after only two sentences. This increased their entry times. It should be concluded from these results that T9 is the better method for a user who sends text messages regularly.

4.6 Quality of Prediction Technique

As the predictions were based on empirical data for multi-press with next, the predictions for this method were accurate according to the definition in Section 3.4, which states that they should be within 10% of the actual performance to be acceptable. The predictions for T9 were not accurate. The prediction for dictionary words had an error of 32% and the prediction for non-dictionary sentences had a disappointing error of 85%. However, these predictions using a very simple technique were much closer than those of the complex mathematical models by Dunlop & Crossan (2000) and Silfverberg et al. (2000). Their predictions had errors of 67% and 334%, respectively, when compared to the actual performance on dictionary sentences. It must be noted that Dunlop & Crossan used an average word length of 5.98 charcters whereas Silfverberg et al. and this report have assumed five characters per word. If Dunlop & Crossan had used the same assumption their model would have predicted faster speeds which would increase the error percentage. Tables 4.2 and 4.3 shows the predictions and their accuracy.

Paper	Prediction	Errors
Dunlop & Crossan (2000)	14.9	92%
Silfverberg et al. (2000)	27.2	250%
This report	8.59	10%

Tab. 4.2: The accuracy of the prediction models for multi-press with next.

Paper	Sentence Type		Errors
	Dict.	Non-dict.	
Dunlop & Crossan (2000)	17.6	N/A	67%
Silfverberg et al. (2000)	45.7	N/A	334%
This report	13.939	9.072	32%, 85%

Tab. 4.3: The accuracy of the prediction models for T9.

It appears that the technique described in this paper does not accurately model the mental preparation factor of predictive text entry. This is also the main failing of the other two models. For this model, it is most likely due to the fact that the predictions were made using the average time it takes to press a key using multi-press, which includes the mental preparation needed for this method. As noted by one subject during the evaluation, multi-press is more "mechanical" than T9 and, in that subject's opinion, required less cognitive effort. This would suggest that the value for the time taken to press one key was too quick for using to predict T9's performance. This value could be ad-

justed to correct the predictions, but this would not make the technique generic.

In order to create a generic prediction technique for future text entry methods, an accurate technique for modelling the cognitive effort required for a method is needed. If this were possible then accurate prediction would be possible and very achievable. The addition of a cognitive effort value to a mathematical model such as the one by Dunlop & Crossan (2000), would reduce the current predictions and improve the accuracy.

The prediction technique presented in this paper appears to be too simple to accurately predict the performance of any future methods. However, the accuracy of previous mathematical efforts is also too low to make them useful for evaluating future techniques. More work is needed to develop a method of predicting the required mental effort of a method. Some ideas for such work are discussed in the following section as well as other ideas for future work.

5. FURTHER WORK

As reported in Chapter 4, the main cause of the inaccuracy of current predictions is the failure to account for the mental effort involved in text entry. Investigation is needed to determine what makes a method such as T9 cognitively more expensive than a method such as multi-press. If a relationship could be found between a common feature of all methods and the equivalent cognitive effort, performance prediction would become far more accurate. Once such a feature were identified, it could be combined with a mathematical model for the physical movement of text entry to create an accurate, generic prediction technique. This could then be trialled on new methods such as WordWise and LetterWise, which have yet to be independently evaluated. An empirical evaluation would again be needed to test the accuracy of the new prediction technique.

An empirical evaluation of all current commercial techniques is needed, comparing the multi-press variations with predictive methods, such as T9, and new methods, such as LetterWise and WordWise. Multiple sessions would be necessary to take into account the strong learning effects shown in previous evaluations and subjective results would need to accompany the quantitative speed measures. This would result in a definitive ranking of the techniques to determine the best overall method.

6. CONCLUSIONS

This report has presented a new approach to predicting the performance of a given text entry method. This technique is a simple approach when compared to the mathematical models of previous papers. Predictions were made for multi-press with next and the predictive text entry method T9 using a pilot study and text analysis. An empirical evaluation was completed to test the accuracy of the predictions and to compare the newer T9 method to the multi-press method.

The evaluation showed that despite its simplicity, the presented prediction technique was more accurate than previous prediction models, although it is still not accurate enough to be useful to interface designers. It had an error 10% for multi-press (which was acceptable) and errors of 32% for dictionary sentences and 85% for non-dictionary sentences, which made it inaccurate overall..

T9 was significantly faster when entering dictionary sentences (10.5 wpm vs 7.2 wpm), while multi-press was significantly faster when entering non-dictionary sentences (8.4 wpm vs. 4.9 wpm). Subjectively, a majority (60%) of users preferred T9. As T9 remembers non-dictionary words entered by a user, it was concluded that T9's performance would tend to the 10.5 wpm speed for dictionary words with continued use. As a result, T9 was considered the better method of the two. The evaluation also found that the overall mean speeds of T9 and multi-press were not significantly different (7.77 wpm vs. 7.71 wpm).

The main problem with current prediction techniques (including the one presented in this paper) is the inability to predict the mental effort required for a certain method. Further work is needed to find a way to model this important factor of text entry on a mobile phone. If this were possible it could be combined with the mathematical models previously developed to create a generic accurate prediction technique.

BIBLIOGRAPHY

Card, S. K., Moran, T. P. & Newell, A. (1980), 'The keystroke-level model for user performance time with interactive systems', *Communications of the ACM* **23**(7), 364–410.

Darragh, J. & Witten, I. (1992), *The Reactive Keyboard*, Cambridge University Press.

Dunlop, M. D. & Crossan, A. (2000), 'Predictive text entry methods for mobile phones', *Personal Technologies* pp. 134–143.

Fitts, P. M. (1954), 'The information capacity of the human motor system in controlling the amplitude of movement', *Journal of Experimental Psychology 47* pp. 281–391.

GSM Association (2000), 'Membership statistics'. `www.gsmworld.com/membership/mem_stats.html`

GSM Association Press Release (2001), *More than 200 Billion GSM text messages forecast for full year 2001*, GSM Association, Avoca Court, Temple Road, Blackrock, Co. Dublin, Ireland. `www.gsmworld.com/news/press_2001/press_releases_4.html`

ISO/IEC 9995-8 (1994), *Information systems - Keyboard layouts for text and office systems - Part 8: Allocation of letters to keys of a numeric keypad*, International Organisation for Standardisation.

James, C. L. & Reischel, K. M. (2001), Text input for mobile devices: comparing model prediction to actual performance, *in* 'Proc of CHI2001', ACM, New York, pp. 365–371.

Kober, H., Skepner, E., Jones, T., Gutowitz, H. & MacKenzie, I. S. (2001), Linguistically optimized text entry on a mobile phone. `http://www.eatoni.com`

MacKenzie, I. S. (2001*a*), 'A note on calculating text entry speed', `http://www.yorku.ca/mack/RN-TextEntrySpeed.html`

MacKenzie, I. S. (2001*b*), 'A note on counterbalancing in repeated-measures experiments', http://www.yorku.ca/mack/RN-Counterbalancing.html.

MacKenzie, I. S. (2001*c*), 'A note on phrase sets for evaluating text entry techniques', `http://www.yorku.ca/mack/RN-PhraseSet.html`

MacKenzie, I. S., Kober, H., Smith, D., Jones, T. & Skepner, E. (2001), LetterWise: Prefix-based disambiguation for mobile text input. `http://www.eatoni.com`

MacKenzie, I. S. & Zhang, S. X. (1999), The design and evaluation of a high-performance soft keyboard, *in* 'Proc of CHI1999', ACM, New York, pp. 25–31.

Silfverberg, M., MacKenzie, I. S. & Korhonen, P. (2000), Predicting text entry speed on mobile phones, *in* 'Proc of CHI2000', ACM, New York, pp. 9–16.

Welch, B. B. (1997), *Practical Programming in Tcl and Tk*, 2 edn, Prentice Hall.

Zhai, S., Hunter, M. & Smith, B. A. (2000), 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', ACM, New York, pp. 119–128.