

# Investigating five key predictive text entry with combined distance and keystroke modelling

Mark D. Dunlop and Michelle Montgomery Masters

*Computer and Information Sciences  
University of Strathclyde,  
Richmond St, Glasgow G1 1XH, Scotland*

Mark.Dunlop@cis.strath.ac.uk

*Abstract* This paper investigates text entry on mobile devices using only five-key. Primarily to support text entry on smaller devices than mobile phones, this method can also be used to maximise screen space on mobile phones. Reported combined Fitt's law and keystroke modelling predicts similar performance with bigram prediction using a five-key keypad as currently achieved on standard mobile phones using unigram prediction. User studies reported here show similar user performance on five-key pads as found elsewhere for novice nine-key pad users.

Keywords: predictive text entry user modelling

## Introduction

Text entry is vitally important to many applications that are becoming common on mobile devices, for example text messaging, instant messaging and email. The industry has reacted to this requirement by providing devices based around either full or half-qwerty keyboards to improve text entry (see figure 1). However, these devices either have to compromise the overall size of the device or the screen size in order to make space for the larger keypads. In parallel, increased data services and the movement of traditional PDA functionality onto phones, puts pressure on devices to have larger screens at a time when retail markets still show strong pressure for smaller overall device sizes. This paper aims to address these contradicting pressures by investigating predictive text entry using only five keys.



## Text entry methods for mobile phones

Most mobile phones still adhere to the ISO standard 9-key layout for core text entry (see figure 2) with letters spread over eight keys plus a space key. This results in an ambiguity problem: for example if the user types 2, the phone does not know if she wishes an A, B, or C. Traditionally the user manually disambiguated each keystroke using the multitap standard, users pressed keys multiple times to achieve the letter they wished (e.g. pressing 2 would give A, 22 B, 222 C, with a timeout or escape key used to separate subsequent letters on the same key). This is a slow and error prone form of disambiguation (e.g. [1, 2]), often due to the delay around subsequent letters on the same key or missed triple clicks on poor quality keypads. Predictive text entry methods attempt to provide faster text entry by reducing the disambiguation to a word-by-word basis: the user only presses one key per letter then selects from possible matching words after each word. To reduce the effort of disambiguation further, words are presented in decreasing likelihood of being the correct word. There are many techniques for estimating these likelihoods with the most common being to present words that match the ambiguous key sequence in decreasing order of frequency of use in the expected language. This frequency-based approach, also known as unigram approach, is implemented in most standard format mobile phones using technology such as Tegic's T9 [3, 4].



Figure 2: Nokia 5110 ISO standard 9-key keypad

Primarily to support text entry on smaller devices than mobile phones, some work has been conducted on text entry for smaller keypads using approaches such as the date-stamp method (where users use either three-way or five-way joystick to enter text on a letter by letter basis). Bellman and MacKenzie [5] investigated dynamic rearrangements of the letters to reduce the distance from the start point based on probabilistic modelling of the most likely next character, unfortunately the results showed no improvement over a fixed QWERTY layout with an

average entry speed of about 10wpm, rising to about 15wpm at best. Dunlop [6] reported some initial work on text entry on a five-key pad using a watch-like interface. This work used a variation of predictive text entry using only four soft keys for letters plus a combined space/next key. This paper builds on that work to investigate 5-key text entry on small physical keyboards that can be used to optimize screen-space on a phone format device.

An alternative is the growing move away from physical keypads to touch-screens, which are more common on PDAs. MacKenzie et al. [7] compared the three most common input techniques for stylus based text entry on touch screen devices: small on-touch-screen keyboards either in QWERTY or alphabetic layout and hand-printing of letters. Their results showed that a standard QWERTY layout can achieve around 23wpm while hand-printing achieved only 17wpm (and alphabetic soft-keyboard only 13wpm). While the on-screen QWERTY pad achieves decent text entry rates, it has been shown that users can achieve similar rates using standard 9-key pads [2] while not requiring users to use a stylus to operate the device.

## **Evaluation of mobile text entry**

The evaluation of new text entry devices is complex: users take considerable time to reach expert user performance and the nature of the devices makes prototyping difficult. One solution is to use model-based evaluation techniques early in the design to predict performance and complement successful approaches with usability studies later in the process. There are two main schools of interaction modelling for text entry on mobile devices: movement modelling based around Fitt's law [8] for time to press buttons and keystroke modelling of the user action sequence [9]. Both have their advantages and a combined model is used here to investigate five key text entry.

## **Structure of this paper**

The paper first discusses our proposed design for text entry using five-keys. We then present performance analysis of how well a text-entry engine can disambiguate five-key input compared to nine-key input. These disambiguation performance estimates are then used as part of a model of text entry to compare

predicted user performance between five-key and nine-key ambiguous text entry. We introduce a combined model of performance estimation based on keystroke modelling with key timing information from Fitt's law derived movement models and use this model to revise our predictions. Finally, we report on a set of user trials of a prototype five-key system to investigate novice user performance.

## Design

The two main motivations behind the development of a five-key text entry method for mobile devices are to (a) reduce the space used on devices by the keypad, both on current phones to increase screen space and on other smaller devices to permit text entry, and (b) to potentially increase speed by reducing finger movement. Our design is based around a five-key ambiguous keypad with four alphabetic keys and a combined space/next key, and is similar to that developed for watch-top touch screens [6]. Users press one key per letter to enter a word followed by space – the first press of the space key after a word inserts a space (highlighted as \_ on the interface) with subsequent presses of the space key rotating round matching suggestions (in a similar fashion to the next key on many T9 phones). To aid users a small area of the screen is used to give a guide to which letters are on which keys, using visual clues to guide text entry [10] this area dims letters that are not possible given the letters already entered (thus helping users to spot letters they are looking for).

Figure 3 shows a paper prototype of the ideal implementation of our interface on a phone size device: five entry keys above two soft-keys and arranged around a 5-way joystick at the top of a phone with a large screen (using top section of screen for key-guide while entering text). This shows how the new interface can be used to maximise screen space, giving a screen area of approximately  $22\text{cm}^2$  compared to the standard phone screen of  $9\text{cm}^2$  on the same size device, in addition to a more comfortable and stable grasp.



Figure 3: paper prototype of ideal 5-key text phone

Figure 4 shows the current J2ME prototype implementation as used in user experiments discussed later. This prototype uses only the top part of a nine-key traditional phone pad with quarter of the alphabet allocated to each of buttons 1, 4, 3 and 6 with 2 acting as combined space/next key. Figure 4 shows the prototype interface for a user entering the phrase "hello how are you" – note the lower-right (6) button on the key-guide showing that only *u* is valid as the next letter on that key.

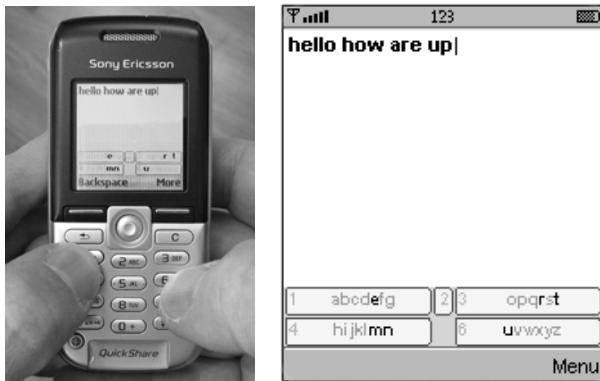


Figure 4: current implementation

While the experiments reported here focus only on alphabetic text entry, we envisage the on-screen display being used to support modal entry of punctuation and numbers (chords could be used for common punctuation, e.g. bottom two keys for a period, while a joystick controlled keyboard could be used for out-of-dictionary word entry and for numeric entry). One of our design aims was to maintain the joystick and soft-keys for editing and application control as normal but these could be used in conjunction with text entry keys as modifiers.

With ambiguous keypads the layout of keys affects the performance of the text entry method: for example, putting all the vowels on one key would mean

common words such as *on*, *in* and *an* would all have the same key sequence. Attempts at optimised keypads (e.g. [6, 11]) have, however, shown only a marginal increase in theoretical performance over alphabetic arrangement of keys and much slower user input due to time spent searching for letters on, to a user's viewpoint, randomly located keys. Thus, the experiments reported here have all been conducted with an alphabetic arrangement of keys shown in figure 4. This is actually slightly sub-optimal compared to the best alphabetically constrained key arrangement, which is collection dependent [11]. We estimate that the results reported below would be marginally affected by changes to optimal alphabetic keyboards and improved by around 2% for fully unconstrained optimised keyboard layout.

## Performance analysis

Standard predictive text entry models (including Tegic's T9 and Dunlop & Crossan's [1]) are based on unigram models of prediction: words are suggested based purely on frequency of occurrence information for words that match the number sequence entered. With predictive text on a standard 9-key pad, on entering 4663 a user is presented with words that can be composed from GHI as the first letter followed by MNO and MNO then DEF – these are typically presented in descending order of frequency in English usage as expected on a mobile phone. For example a standard T9 enabled phone will suggest *good*, *home*, *gone*, *hood*, *hoof* etc. More advanced models can adjust predictions to make them more likely to be correct, in particular bigram models [12, 13] bias predictions based on the previous word – so that ranking is based on frequency of occurrences of word  $w_n$  following word  $w_m$ , for example *home* would be more likely than *good* after the word *at*. To assess the likely impact of reducing the number of keys from nine to five, we calculated best case unigram and bigram predictions using two corpora: a sub-set of The Herald collection used in Dunlop & Crossan [1] and the Singapore SMS corpus [14]. To simplify analysis and focus on core text-entry speeds, both collections were pre-processed to leave only alphabetic characters separated by spaces and newlines.

In both cases the average ranked-list position (ARP) was calculated by learning statistics from the collection then running through the collection

averaging the position of the expected word in suggested ranked list given the ambiguous key-coding of the expected word. An ARP value of 1.0 would indicate that the required word was always in the first position in the ranked list of suggestions, a value of 2.0 that on average the required word was second in the ranked list. This approach to averaging naturally biases the averaging process so that words are taken into account proportionally to their occurrence in the text collection. For unigram prediction all words were ranked in decreasing frequency for each key combination, for bigram the same ranking was used but based solely on frequency of the word occurring after the previous word in the sentence. As an example, given the phrase "are you home" in the text collection the word *home* is converted to its numeric equivalent (4663). A ranked list is calculated to predict suggestions for 4663 with the position of *home* in that list taken as the ranked-list position for that word (position 2 in the case of T9's standard offer list). The suggestions for 4663 are based on the frequency of all words matching 4663 for the unigram model while the bigram model bases the suggestions on frequencies of words occurring after *you*. This approach gives a good approximation to the performance expected in-use once a prediction engine is tuned to the language. However, this approach to modelling avoids two common problems with predictive text entry: out of dictionary words for both approaches and sparse frequency information for bigram models. Predictive text-entry models can only predict known words and alternative input techniques are required for out-of-dictionary words, which are normally stored in the user dictionary – so need only be entered once per device. Bigram models rely on knowing considerable statistical information about words – for rare words some combinations may simply not have been seen before, even though the individual words have. There has been considerable work (e.g. [15, 16]) on adjusting bigram models to, essentially, degrade gently to unigram models when either there is no bigram statistics for the word-code combination or the confidence of those statistics is low. Experiments reported here use a simple bigram model, as the training method ensures usable statistics.

The literature commonly uses three different methods for reporting the performance of text entry methods. Above we have used average ranked list position (e.g. [1]), alternatives are *disambiguation accuracy* (DA) and *keystrokes*

per character (KSPC). Disambiguation accuracy (e.g. [11]) reports the percentage of times the first word suggested by the disambiguation engine is the word the user intended – a DA value of 100% implies the disambiguation engine always give the correct word first, while 50% indicates that it only manages to give the correct word first half of the time. This is an intuitive and very direct measure but does not take into account the performance of words that do not come first in the list. KSPC (e.g. [6]) reports the average number of keystrokes required to enter a character, for example *home* on a standard T9 mobile phone requires 5 keystrokes – 4663\*, where \* is the next suggestion key, giving a KSPC for that word of  $5/4=1.25$ . A KSPC value of 1.0 indicates perfect disambiguation as the user never needs to type any additional letters, while a higher figure reflects the proportional need for the next key in disambiguation. KSPC does take into account ranked list position for all words and compares well with non-predictive models, however it is a rather abstract measure being based on letters for inherently word-based methods. Given a standard word length of 5 letters per word (including space),  $ARP = KSPC \times 5 - 4$ . All methods are normally averaged over a large corpus.

Based on the first million lines of *The Herald* collection (9 141 467 words with average of 4.73 letters per word excluding space), table 1 shows the ARP, DA and KSPC values for unigram and bigram prediction on both five-key and nine-key keypads. This shows that (a) nine-key text disambiguation is considerably more accurate than five-key; (b) that disambiguation is considerably more accurate with bigram modelling and (c) that bigram modelling improves five-key entry proportionally more than nine-key entry.

Herald Collection		
	5-key	9-key
unigram	1.554 ARP 80% DA 1.097 KSPC	1.058 ARP 96% DA 1.010 KSPC
bigram	1.148 ARP 92% DA 1.026 KSPC	1.017 ARP 99% DA 1.003 KSPC

Table 1: average ranked list position of required word in The Herald collection



Table 2 shows the same experiment and results pattern for the Singapore SMS corpus [14] (121 126 words with an average 3.49 letters per word ex-space).

	SMS Collection	
	5-key	9-key
unigram	1.832 ARP 66% DA 1.185 KSPC	1.135 ARP 90% DA 1.030 KSPC
bigram	1.199 ARP 87% DA 1.044 KSPC	1.028 ARP 98% DA 1.006 KSPC

Table 2: average ranked list position of required word in Singapore SMS collection

For comparison: full-sized non-ambiguous keyboards achieve KSPC=1.00, standard date-stamp method for entering text on 3 keys achieves KSPC=6.45, date-stamp like interaction on 5 keys achieves KSPC=3.13 and multitap on a standard 9-key mobile phone achieves KSPC=2.03 [17]. Gong and Tarasewich [11] reported DA for 5-key and 9-key at 85% and 97% respectively for written English corpus and 69% and 92% for SMS messages. Hasselgren et al. [12] used bigram modelling with word completion, where words were suggested before the user had finished entering them (leading to sub-1.0 KSPC figures [17]). Their results report KSPC of 1.01 and 1.08 for T9 using Swedish news and SMS corpora respectively, improving to 1.01 and 0.88 respectively for their bigram model with word completion suggestions. As a comparison for Hallegran et al.'s work, unigram word completion has been estimated to reduce KSPC by around 25% for a 9-key keypad (but to lead to potential cognitive load problems) [1].

## Keystroke level modelling (KLM)

To gain an insight into potential expert user behaviour with different keyboards, different approaches have been taken to modelling interaction in order to predict expert (trained, error-free) performance. Dunlop and Crossan [1] proposed a model based on Card, Moran and Newall's keystroke modelling [9]. The model is based on predicting the time  $T(P)$  taken by an expert user to enter a given phrase  $P$ , performing without errors and containing only alphabetic words. These

restrictions are clearly severe limitations on this modelling approach. However, while more complex modelling approaches have been researched to support novices, model more complete interaction and model error behaviour (e.g. [18, 19]), we believe that this relatively simple performance figure complements user studies and is a reasonably accurate and worthwhile estimate of expected peak performance of a text entry system.

The keystroke model is based on building an equation to represent the user activity by summing a set of small time measurements, in the case of text entry the appropriate times are: the homing time for the user to settle on the keyboard  $T_h$ ; the time it takes a user to press a key  $T_k$ ; and the time it takes the user to mentally respond to a system action  $T_m$ . Dunlop and Crossan [1] modelled predictive text entry on a sentence where disambiguation occurs as extra characters representing moving down the ranked list of suggestions, their overall time equation is as follows:

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

Equation 1: Dunlop and Crossan's KLM model

In equation 1,  $w$  represents the number of words in the phrase,  $k_w$  is the number of letters per word (renamed here from their  $k_p$ ) and  $l$  the ARP measure (average position in the suggested word list of the correct word). In that paper the calculation led to a predicted speed of 17.7 words per minute (wpm). Pavlovych & Stuerzlinger [18] later modified the calculation by correcting double counting of the first space key after a word by changing  $k_w$  to 4.98, leading to a predicted speed of 19.3 wpm<sup>1</sup>. Table 3 shows details of the calculation for both sets of data.

---

<sup>1</sup> The figure of 19.3 wpm represents our revision of Dunlop and Crossan using Pavlovych and Stuerzlinger's observations, which differs numerically from their Pavlovych and Stuerzlinger's.

	Dunlop & Crossan	As revised by Pavlovych & Stuerzlinger
$T_h$	0.4	0.4
$w$	10	10
$k_w$	5.98	4.98
$T_k$	0.28	0.28
$l$	1.03	1.03
$T_m$	1.35	1.35
$T_{10}$	33.9s	31.2s
speed	17.7wpm	19.3wpm

Table 3: nine-key unigram prediction time and speed

James and Reischel [2] carried out focused user experiments on both novice and experienced users of T9. They reported a measured T9 performance averaging at 20.4 wpm for experts, within 10% of the keystroke level modelling prediction from table 3. For comparison handwriting with ink and paper achieves around 20 to 30 words per minute while desktop typists can achieve in excess of 150 wpm, though this drops considerably for non-secretary users and for composition rather than transcription (e.g. Karat et al [20] found speeds of 33wpm transcription dropping to 19wpm for composition).

### Simple unigram and bigram modelling

Equation 1 has constant values for  $T_h$ ,  $w$ ,  $T_k$  and  $T_m$  with other values being dependent on the disambiguation engine and the text collection in use. Based on the performance analysis reported above, table 4 shows the predicted words-per-minute for unigram and bigram predictions using the two test collections, while table 5 gives more details on the parameters of equation 1. This shows that moving to bigram prediction improves performance by 2-6% for a nine-key keypad and 20-34% for the five-key keypad. Furthermore, the improvement in moving to bigram prediction brings the five-key keypad from around 26-40% slower than nine-key down to only around 7-10% slower.

	9-key wpm	5-key wpm
herald unigram	19.42	15.39
herald bigram	19.85	18.54
sms unigram	20.93	14.99
sms bigram	22.28	20.19

Table 4: Summary KLM model predicted results

		$k_w$	$l$	$T_{10}$ (s)	wpm
9-key	unigram	4.73	1.058	30.89	19.42
Herald	bigram	4.73	1.017	30.22	19.85
9-key	unigram	3.49	1.135	28.67	20.93
SMS	bigram	3.49	1.028	26.93	22.28
5-key	unigram	4.73	1.554	38.97	15.39
Herald	bigram	4.73	1.148	32.36	18.54
5-key	unigram	3.49	1.832	40.03	14.99
SMS	bigram	3.49	1.199	29.72	20.19

Table 5: Variable KLM model parameters and predicted results

## Adjusting model for variable keystroke times

Dunlop and Crossan modelled keystroke speed at 0.28s based on a fixed figure from Card et al.'s figure of equivalent to "an average non-secretary typist" on a full QWERTY keypad [9]. As supported by James and Reischel [2], equation 1 works well, however it cannot take into account one of the main motivations for the five-key keypad: reduced finger movement decreasing the time taken to press each key. Mackenzie's group have conducted considerable work on using Fitt's law [8] for calculating the limit of performance given distance between keys and a language model for the movement required between keys to enter text with a given text entry scheme (e.g. [21]). In the basic form their modelling predicts 40.6 wpm for thumb-based T9 input assuming no *next* key operations, with 5 characters per word this equates to an average keying time of 0.30s (without thinking or homing times).

Based on the same phone model used in their paper (a Nokia 5110, figure 4), we have examined the total time for entering a large block of text according to their model using both nine-key and five-key keypads. Our results give a weighted average time per key of 0.26s for the nine-key pad and 0.22s for the five-key pad (the slight difference being due to corpus effects – the movement models are derived from and conditional upon analysis of a large corpus of text). Feeding these figures directly into equation 1 (replacing  $T_k$  as appropriate in table 5) gives the words per minute predictions in table 6.

	9-key wpm	5-key wpm
herald unigram	20.18	17.04
herald bigram	20.64	20.81
sms unigram	21.62	16.29
sms bigram	23.05	22.30

Table 6:revised KLM models adjusting for distance calculations

Table 6 shows that the five-key keypad is predicted to perform roughly equivalently to a nine-key keypad when bi-gram prediction is used and reduced keying time is taken into account (1% better on The Herald collection, 3% slower on the SMS collection). Furthermore, five-key performance with bigram prediction is predicted to marginally out perform standard unigram prediction on nine-key keypads for both test collections.

### Improving these models

The distance model above, while taking into account some aspects of physical keyboard layout, has two inaccuracies that can noticeably affect predictions: repeat keys and parallel finger movements. Repeat keys, where subsequent letters are on the same key, are not modelled correctly with Fitt's law, which tends to underestimate zero distance movements. Soukoreff and MacKenzie [22] conducted user experiments on a modified Fitt's law model that includes a separate estimated time for repeat keys. The standard Fitt's law model also does not take into account one of the main speed gains of touch typing: being able to move fingers on one hand in preparation while still typing with the other hand. The same team also showed separately that this can impact on two-thumbed text

entry where one thumb can be moving to the appropriate key in parallel with the key press on the other thumb [23].

We recalculated our predictions based on modelling both repeat-key timing and two-thumb parallel movements. Again we simplified the models from the distance papers by using them simply to estimate the average key-stroking times for our KLM modelling. Assuming users are using two-thumbs to enter text, our initial modelling gave key times of 0.27s for The Herald collection with a 9-key keypad and 0.25s for the SMS collection, reducing to 0.23s and 0.22s respectively on the 5-key keypad. Table 7 shows the key stroke time for the different modelling and collection combinations for 9-key pad and the resulting predicted text entry speed for both unigram and bigram prediction. Table 8 shows the same data for 5-key pad.

		$T_k$	unigram	bigram
Herald	simple	0.27	19.9	20.4
	repeat-key	0.25	20.9	21.4
	two-thumb	0.18	23.8	24.4
	both	0.17	24.1	24.7
SMS	simple	0.25	22.0	23.4
	repeat-key	0.24	22.9	24.5
	two-thumb	0.18	25.0	26.9
	both	0.17	25.2	27.1

Table 7: Predicted words-per-minute for 9-key pad using different prediction models

		$T_k$	unigram	bigram
Herald	simple	0.23	16.7	20.4
	repeat-key	0.22	17.3	21.1
	two-thumb	0.16	19.1	23.7
	complex	0.15	19.0	23.7
SMS	simple	0.22	16.3	22.3
	repeat-key	0.21	16.7	23.0
	two-thumb	0.15	18.0	25.2
	complex	0.15	17.8	25.0

Table 8: Predicted words-per-minute for 5-key pad using different prediction models

These tables show that, averaged over the two collections, our initial simple predictions show a drop of around 21% in words per minute when moving from a 9-key to a 5-key keypad using simple unigram word prediction. This difference drops to only 3% when using bigram prediction, with simple Fitt's law modelling of keystroke times. When we take into account increased times for repeat-keys and reduced times when swapping thumbs the predicted performance drop for the 5-key keypad increases to 6% when comparing bigram prediction on 5-key and 9-key pads. However, bigram prediction on 5-key pads is still only 2% slower overall than unigram prediction on 9-key pads, showing predicted 5-key performance roughly in-line with the prevalent current mobile phone text entry approach.

## User trials

Keystroke level modelling is useful for predicting expert user performance but is focussed on expert error-free performance and gives little feedback on how new users find a technology to use. To address this we conducted a controlled usability experiment with users entering text phrases in a controlled setting.

## Experimental Equipment and Users

A set of sentences were collected from members of the department who regularly send text messages. They were asked to write some messages that were in the style of those they typically send/receive. The messages were edited to expand

any SMS shorthand (e.g. *ur* converted to *your*), remove punctuation marks, convert numbers into their written form (e.g. replacing 8 with *eight*) and remove capitalisation. Although not natural these edits were consistent between test interfaces, more closely match the earlier technical studies and can be justified as gaining an insight into peak text entry rates that are not affected by modal changes. The final 20 sentences were randomly shuffled to give the trial test phrases.

Twenty staff and students of the Computer and Information Sciences Department at University of Strathclyde took part in the experiment (ages ranged from 24-45, 6 female/14 male, 2 dominant-left-handed/18 right). Fifteen of these users considered themselves to be regular senders of text messages, four rarely sent messages and one user had never sent a text message. Fourteen of the subjects regularly used T9 prediction when messaging.

Experiments were all conducted on a Sony Ericsson K300i handset running a dedicated J2ME editor that was identical for both 5-key and 9-key text entry bar the provision of an on-screen guide to key-mapping for 5-key entry (as shown in Figure 4 and 5). Although not the ideal format for 5-key entry, we felt this was the best experimental compromise between appropriateness and consistency. Both versions of the text engine used the same small dictionary with a unigram prediction model.

## **Experimental Process**

The participants were asked to complete a short pre-experiment questionnaire on demographic information and text message usage. The participants were then given a demonstration of our application and our approach to nine-key text entry. To recap, our approach differed from standard T9 in use of the space key to double as next key. It also varied from some handsets in that only 0 and # were supported as space keys (unfortunately ruling out the normal 1 key for users of certain brands of handsets).

After the demonstration participants were asked to enter 10 sentences using the nine-key layout on the mobile phone handset.

The participants were then given a demonstration of the five-key layout and instructed to enter 10 sentences using this layout.



Finally, the participants were then asked to complete a feedback form to indicate their preference between the keypads and give any comments on either keypad layout.

Using a GPRS data connection, key-strokes were recorded each time the space key was pressed. These records were time stamped in tenths of a second and recorded the display status at that point.



Figure 5: five-key and nine-key prototypes as used in user study

## Results

The results shown in figure 6 summarise the average text entry rate for users over the last eight phrases (the first two are excluded to allow users to settle in with the device and technique).

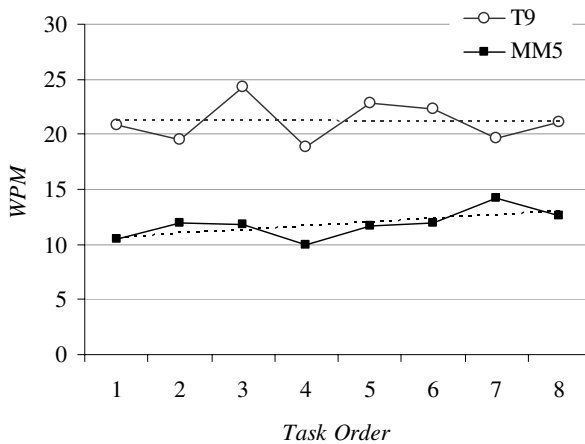


Figure 6: user-averaged text entry speed over last eight phrases

The results show that, averaged over all users and tasks, nine-key entry was approximately 21wpm while five-key is around 12wpm. James and Reischel [2] measured 9-key users entering chat-style message at between 11 and 26 wpm

depending on experience – in line with our studies, given our profile of users and slight differences in prior experience. Furthermore our result for novice five-key users is in line with their novice nine-key user speed of around 11 wpm. The results also show a clear positive trend for 5-key usage compared to a steady performance for 9-key entry.

In discussion with the users several participants stated that they could see the benefit of having fewer keys to allow devices to be made smaller. Many found that the key-guide for 5-key entry acted as an effective spelling guide since letters were greyed out that would not create a possible word in the dictionary, although some mentioned that they felt locked in by this *spelling assistance*. Furthermore, several participants found that this display made it was easier to resume text entry after a distraction – an aspect of the outcomes we are investigating further. Many reported that they felt any difficulties in using 5-key entry would reduce over time since they were trying to break pre-programmed 9-key and T9 habits and that they felt they were performing better towards the end of the experiment. Interestingly, despite requiring more keystrokes in 5-key entry, several participants mentioned that they felt they had to make fewer "clicks" when using 5-key entry. We can only conclude that this impression was due to less key-searching and/or finger movement. Finally, some subjects mentioned that they did not like looking at the screen. Continuous focus on the screen is an intended benefit of 5-key entry and we feel the likely cause of this reaction was the use of the 9-key pad for experiments, where users are used to checking the labels on keys.

## **Future work**

Although often used while stationary, designing for mobile users puts new challenges on the design on interfaces. [24] and [25] have studied the effect of different sized on-screen buttons on walking speed and data-entry errors. The five-key pad proposed here lends itself to reduced vision load compared to soft keypads or printing, with physical keys rather than screen areas and expert users only needing to check each word instead of each letter. However, this checking needs to be done more carefully than with a 9-key pad with the same prediction engine. Studies are planned to investigate 5-key text entry for walkers to assess the impact of this on real mobile use.

We also plan to conduct a longitudinal study of five-key text entry using a fuller prototype system that will support full text entry (including numbers, punctuation and out of dictionary words).

## Conclusions

This paper reported our investigation into predictive text entry using five keys. This work was motivated by three desires: to reduce the space taken up by keyboards on mobile phones, to extend predictive text entry to other formats of mobile devices and to reduce finger movement distances in the hope of improving text entry speed.

An investigation into purely technical performance of both unigram and bigram predictive text entry showed that five-key text entry was considerably poorer than nine-key for unigram modelling for both written English and SMS corpora. However, the quality of predictions was considerably increased and the difference between keypads reduced when bigram modelling was used.

These figures were then used as the basis for keystroke level modelling of users entering text. Simple keystroke modelling with fixed keying times, showed a speed reduction of around 21% for five-key text entry. These models were then refined to take into account the reduced finger movement times for five-key text entry using methods derived from Fitt's law, this resulted in new predictions that five-key text entry would be within approximately 6% of the performance of nine-key when both are using bigram modelling. Furthermore, five-key text entry using bi-gram modelling was predicted to perform very close to the level of standard nine-key unigram modelling.

Finally, users trials were conducted to gain an impression of non-experts usage of five-key text entry. Using a prototype system on a mobile phone handset with unigram prediction, users achieved approximately 12 words-per-minute using five-key entry and 21 words-per-minute for nine-key. While much slower on five-keys, these figures are in line with, respectively, novice and expert speeds reported elsewhere for 9-key pads.

Through our combined keystroke and Fitt's law modelling we have predicted that five-key text entry using bigram word prediction will perform equivalently to current nine-key technology for expert users. Furthermore, our

user trials have shown similar user performance using five-key to novice nine-key users in other trials. Combined, while not achieving one of our aims of faster text entry, we believe that these results show that five-key keypads can be used as a replacement for current nine-key text entry without noticeable loss of text entry speed.

Acknowledgements: Once again we extend our gratitude to our users in our user trials.

## References

1. Dunlop, M.D. and A. Crossan, *Predictive text entry methods for mobile phones*. Personal Technologies, 2000. **4**(2).
2. James, C.L. and K.M. Reischel, *Text input for mobile devices: comparing model prediction to actual performance*, in *SIGCHI conference on Human factors in computing systems*. 2001, ACM Press: Seattle, Washington, United States.
3. Grover, D.L., M.T. King, and C.A. Kushler, *Reduced keyboard disambiguating computer I*. Tegic Communications, Editor. 1998: USA.
4. Kushler, C., *AAC Using a Reduced Keyboard*, in *Technology and Persons with Disabilities Conference*. 1998: USA.
5. Bellman, T. and I.S. MacKenzie, *A probabilistic character layout strategy for mobile text entry*, in *Graphics Interface '98*. 1998, Canadian Information Processing Society: Toronto, Canada. p. 168-176.
6. Dunlop, M.D., *Watch-Top Text-Entry: Can Phone-Style Predictive Text-Entry Work With Only 5 Buttons?*, in *6th International Conference on Human Computer Interaction with Mobile Devices and Services: MobileHCI 04*, S.A. Brewster and M.D. Dunlop, Editors. 2004, Springer Lecture Notes in Computer Science: Glasgow.
7. MacKenzie, I.S., et al., *A comparison of three methods of character entry on pen-based computers*, in *Factors and Ergonomics Society 38th Annual Meeting*. 1994: Santa Monica, USA. p. 330-334.
8. Fitts, P.M., *The information capacity of the human motor system in controlling the amplitude of movement*. *Journal of Experimental Psychology*, 1954. **47**(6): p. 381-391.
9. Card, S.K., T.P. Moran, and A. Newell, *The keystroke-level model for user performance with interactive systems*. *Communications of the ACM* 1980. **23**(7): p. 396-410.
10. Magnien, L., J.L. Bouraoui, and N. Vigouroux, *Mobile Text Input with Soft Keyboards: Optimization by Means of Visual Clues*, in *6th International Symposium on Mobile Human-Computer Interaction – MobileHCI 2004* 2004, Springer LNCS Volume 3160 / 2004 Glasgow, Scotland.
11. Gong, J. and P. Tarasewich, *Alphabetically constrained keypad designs for text entry on mobile devices*, in *CHI '05: SIGCHI Conference on Human Factors in Computing Systems* 2005: Portland, USA.
12. Hasselgren, J., et al., *HMS: A Predictive Text Entry Method Using Bigrams*, in *Workshop on Language Modeling for Text Entry Methods, 10th Conference of the European Chapter of the Association of Computational Linguistics*. 2003: Budapest, Hungary. p. 43-49.
13. Klarlund, N. and M. Riley, *Word n-grams for cluster keyboards*, in *Workshop on Language Modelling for Text Entry Methods, 11th Conference of the European Chapter of the Association for Computational Linguistics* 2003, 51-58.
14. How, Y. and M.-Y. Kan, *Optimizing predictive text entry for short message service on mobile phones*, in *Human Computer Interfaces International (HCII 05)*. 2005: Las Vegas, USA.
15. Witten, I.H. and T.C. Bell, *The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression*. *IEEE Transactions on Information Theory*, 1991. **37**: p. 1085-1094.
16. Katz, S.M., *Estimation of probabilities from sparse data for the language model component of a speech recognizer*. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 1987. **35**(3): p. 400-401.

17. MacKenzie, I.S., *KSPC (keystrokes per character) as a characteristic of text entry techniques*, in *MobileHCI 02: Fourth International Symposium on Human-Computer Interaction with Mobile Devices*. 2002, Springer Berlin, Lecture Notes in Computer Science: Pisa, Italy. p. 195-210.
18. Pavlovych, A. and W. Stuerzlinger, *Model for non-expert text entry speed on 12-button phone keypads*, in *CHI '04: SIGCHI Conference on Human Factors in Computing Systems*. 2004: Vienna, Austria. p. 351-358.
19. Sandnes, F.E., *Evaluating mobile text entry strategies with finite state automata*, in *MobileHCI 05: 7th international Conference on Human Computer interaction with Mobile Devices and Services*. 2005: Salzburg, Austria.
20. Karat, C., et al., *Patterns of Entry and Correction in Large Vocabulary Continuous Speech Recognition Systems*, in *ACM CHI99: Human Factors in Computing Systems*. 1999, 568-575.
21. Silfverberg, M., I.S. MacKenzie, and P. Korhonen, *Predicting text entry speed on mobile phones*, in *SIGCHI Conference on Human Factors in Computing Systems (CHI'00)*. 2000, ACM Press: The Hague, The Netherlands.
22. Soukoreff, R.W. and I.S. MacKenzie. *Using Fitts' law to model key repeat time in text entry models*. in *Graphics Interface 2002*. 2002.
23. MacKenzie, I.S. and R.W. Soukoreff, *A model of two-thumb text entry*, in *Graphics Interface 2002*. 2002.
24. Brewster, S.A., *Overcoming the Lack of Screen Space on Mobile Computers*. *Personal and Ubiquitous Computing*, 2002. **6** (3): p. 188-205.
25. Mizobuchi, S., M. Chignell, and D. Newton, *Mobile text entry: relationship between walking speed and text input task difficulty*, in *MobileHCI 05: 7th international Conference on Human Computer interaction with Mobile Devices and Services 2005*: Salzburg, Austria.