Watch-top text-entry: Can phone-style predictive text-entry work with only 5 buttons?

Mark D Dunlop

Computer and Information Sciences, University of Strathclyde, Glasgow G1 1XH, Scotland
Mark.Dunlop@cis.strath.ac.uk

Abstract This paper presents an initial study into the viability of text entry on a watch face using four alphabetic buttons and a central space key. The study includes a technical evaluation of likely error rates using a large text corpus and user studies on palmtop emulated mobile phone and watch. The results, though in favour of the phone pad, are encouraging and show such a method is feasible.

Introduction

Predictive text-entry on mobile phones, as standardised by Tegic's T9 software [1], has proven extremely effective for mobile phone keypads [e.g. 2, 3]. However, this method still requires a keypad of 9 buttons (8 alphabetic and 1 space for plain text entry). In this paper we report our initial investigation into using a 5 key pad for predictive text entry targeted at watch-top text-entry. The pad used here consists of four soft alphabetic keys around the periphery of a touch screen and a central space key (see fig 1). The motivation is to allow relatively high speed text entry on very small device using an approach familiar to mobile phone users (c.f. very small keypad designs such as [4]) and without the need for a stylus (c.f. handwriting (e.g. Graffiti), many-key soft-keyboards (e.g see [3]), or gesture input (e.g. T-Cube or Cirrin [6]).

Predictive text-entry is based around a large dictionary of word senses with occurrence information, users press one key per letter from multiple-letter keys and the system suggests possible matches to the key sequence in descending occurrence frequency. The simplified text entry approach used here overloads the space key: on first press a space is entered, on subsequent consecutive presses the suggested word cycles. For example to enter *LUNCH* using the interface in figure 1, the user would press *NMLKJIH*, followed by *UVWXYZ*, *NOPQRST*, *ABCDEFG* then *GHIJKLM* at which point *HUMAN* would be suggested as the most common word from those five keys, the user would press space to enter a space followed by another space to cycle words resulting in *LUNCH*. Predictive text-entry methods inherently have a level of errors – there are often more than one word possible from a given key sequence. While presenting words in decreasing order of occurrence frequency reduces the commonality of errors, they still occur. When reducing from eight to four alphabetic keys it is expected that the number of errors will increase. To assess how much the error rate increases a technical experiment was

Figure 1: 5-key text entry

conducted and is report here. Having fewer keys also implies users have fewer, larger, targets to hit and, in fig1, these are centred around *space* making a very close set of relatively large targets. Following Fitt's law, we may expect faster interaction these buttons. To assess use of the keypad, user experiments were run measuring input speed and error rate and are reported later.

Technical Experimental Setup

The technical experiments were based around a dictionary of 77 317 word senses, with frequency information, extracted from six months of *The Herald* newspaper (same as in [2]). The performance of encoding an individual word is dependent both on the keypad layout and the dictionary and was measured as follows:

$$p_{w,d,k} = \frac{|k(w,d)|}{|w|}$$

where $|k(w,d)|$ is the length of the encoding word w by keypad k using dictionary d.

The performance for the top n words was calculated using a weighted average, by frequency of occurrence, of each word in the top n, as follows:

$$P_{n,d,k} = \frac{\sum_{n} p_{w_n,d,k} . f(w_n,d)}{\sum_{n} f(w_n,d)}$$

where $f(w_n,d)$ is the frequency of occurrence of word n in dictionary d.

Using *The Herald* dictionary P_{200} was calculated for the six possible balanced alphabetic keypad layouts using four buttons, to assess the best keypad layout for alphabetic ordering. This analysis resulted in the keypad: *ABCDEFG*, *HIJKLMN*, *OPQRST* and *UVWXYZ* being used as the alphabetic ordered keypad.

Of course, letters do not need to be distributed alphabetically and a separate study was conducted to estimate the best possible key layout from the 4^{26} possible keypads. All 2, 3, and 4 letter words in the dictionary were evaluated to assess the pairwise confusion of individual letters based on one letter error per word, i.e. a measure of how likely swapping one letter for an other would result in a valid word. This resulted in a table[1] of 325 confusion weights, which were sorted into decreasing confusion occurrence to give *AI*, *ST*, *NS*, *NT* and *IO* at the top. Each of the four alphabetic keys was initially assigned one letter from *AIST* and their running total of confusion weights set to zero. For each subsequent letter from the list of pairs that had not already been assigned, a potential confusion weight was calculated as the sum of all confusion weights for combinations of letters currently on the key plus the new letter. The new letter was then added to the key with the smallest resulting total confusion score to minimise the total confusion weight per key (e.g. *N* is added to the *I* key as the confusion weight between *NA*, *NI*, *NS*, and *NT* is lowest for *NI*). This process resulted in the *GORSUV* keypad with the following four keys (rearranged alphabetically): *GORSUV*, *AFKMWXY*, *BDILNQZ* and *CEHJPT* (see figure 2). The *GORSUV* keypad was then used as an estimated optimal keypad.

[1] See http://www.cis.strath.ac.uk/~mdd/research/files/confusionscores.html

Finally, for comparison a similar scheme was used for the traditional mobile phone keypad using both predictive text entry and multi-click text-entry (using the multi-click encoding instead of *k(w,d)* but weighting similarly to the dictionary methods).

Figure 2: GORSUV key-pad

keypad	P_{200}	*top 50 as 1st hit*	*top 200 as 1st hit*
multi-click phone	2.101	n/a	n/a
alphabetic watch	1.060	45	162
GORSUV watch	1.041	46	166
predictive phone	1.009	50	191

Table 1: Weighted keys per letter for different keypads and number of top 50/200 words that appeared as first choice on list of suggested words when entered

Table 1 shows that, on a weighted average over the top 200 words in *The Herald*, the predictive phone keypad achieves an impressive average of 1.009 keys per letter. The *GORSUV* and alphabetic keypads perform significantly worse than the phone pad with 1.041 and 1.060 keystrokes per letter, while multi-click entry averages to over twice as many keystrokes per letter. Table 1 also shows how many of the top-50 and top-200 most common words were suggested as first match when keyed in.

While performing worse than a mobile phone, the suggested error rates for both *GORSUV* and alphabetic four-key pads are encouragingly good and not as bad as may be expected from halving the number of alphabetic keys. While on both measures, *GORSUV* is better than alphabetic it is not clear whether the much longer training time for *GORSUV* would be worth the effort.

Usability Experimental Setup

Usability experiments were conducted on a touch sensitive iPAQ handheld computer with phone (fig 3) and watch (fig 1) simulations written in Java using the same dictionary. Due to memory limitations of handheld Java the dictionary was limited to the top 9000 words from the 77k dictionary used above (augmented with 6 out-of-dictionary words).

Fig 3: Phone Emulation

The experiment followed a within-subject design with two training and two timed task-sets per subject. Each of the four task-sets was composed of entering 3 sentences, from an independent list of humorous short phrases[2], on one interface. The experiment was balanced for first-use system and first-use task-set. Subjects were timed and errors recorded. Twelve subjects carried out the test in total, mostly MSc and PhD students in Computer Science plus two lecturers. The interfaces

[2] http://www.pbbt.com/Directory/Jokes/681.html

deliberately did not include a backspace, to remove correction time from timings, instead users were instructed to hit space and move on to the next word.

Table 2 shows the times for the whole timed task sets averaged over all users, together with the times for just the last two sentences (timing varied more over the first sentence as the user settled with the device). Table 2 also shows the number of words incorrectly entered for each device.

	Watch		Phone	
	mean	stdev	mean	stdev
3 sentences	**3.87**	0.89	**2.75**	0.59
2 sentences	**2.18**	0.60	**1.41**	0.47
Errors	0.75	0.87	1.17	1.08

Table 2: Timing and total error count results from user trials (significant results in bold)

Not surprisingly, the results show statistically significant faster performance with the phone keypad over the watch for both 3 and 2 sentence statistics (at 1% one-tailed correlated t-test). The table also shows no significant difference in error rate between Phone and Watch interfaces. Errors were generally very low, with most errors being caused by a misspelling of a word resulting in wrong suggestions. When asked all users stated that the interface response was suitably fast and did not hinder their interaction.

Over the three sentences the watch was on average 40% slower. Given that many subjects commented that they would expect to get better over time as they still felt they were learning the keypad, this is not a surprising result and shows that the watch keypad, while not reaching the performance of a phone keypad, would be usable for text entry. All subjects stated that they would use the phone in preference to the watch, but that (in all bar two cases, where the subject did not wear a watch) they would sometimes use the watch if given one. One subject highlighted that if holding the watch, two-thumbed text entry could be extremely fast and comfortable.

Discussion

The study reported here was on a short timescale (around 30 mins per subject), a longer trial would be needed to fully assess the speed of entry as it is clear users had not reached a comfort level with the watch interface (and many were very fast phone texters). Ideally the system for subsequent trials would be implemented on a real touch-sensitive watch to assess long-term usage. The use of a newspaper also biased the language somewhat differently to that of normal text messaging, e.g. *lunch* is likely to be more popular than *human* in text messaging. However, the dictionary was used comparatively throughout so this is unlikely to affect results here but would need to be replaced for a long-term study.

The current implementation of watch-face text-entry does not support capitalisation, punctuation, error correction or menu commands. These would have to be implemented using a combination of gestures, two-finger chords, long presses or physical buttons on the side of the watch. Investigations are planned to develop and test a full text entry method for small screens based around the interface presented here. The use of overloading space, almost required for the watch interface, did not cause any usability problems even for very regular texters. However, this might not be

the case when complex schemes are needed to replace the automatic space with punctuation marks etc.; again further investigation is required. The current watch interface does not have “dead-zones” between keys, which may be explain some common misspellings (e.g. users attempting to enter *g* and hitting the *H-N* key instead); while dead-zones would reduce the target zone size it may increase accuracy and requires investigation.

One final improvement that will be investigated is a variant of key-blanking techniques often used on scanning keyboards for people with severe motor control difficulties. These soft-keyboards often omit letters that do not occur in next position of a sequence. In the watch interface greying out the letters on the watch face that are not valid would not change the functionality or timing directly, as there are a fixed number of keys, however it might help users search time for the right letter.

The results presented here show that the use of a five-key keypad does increase the number of times a user needs to scroll down a list of suggested words for predictive text entry. However, this increase is not as great as may be expected by reducing the alphabet to only four keys. The paper presented the *GORSUV* keypad, a pseudo-optimal key arrangement of four keys. While this keypad does have better performance, results here show this improvement to be small and thus unlikely to be of benefit to all bar very frequent users given the extra time needed for users to find the correct key. User trials confirmed that the watch keypad was slower than the phone keypad, though again by not as much as might be expected (approx 40% slower than a touch screen emulation of a phone, however, this itself likely to be slower than using a physical keypad on a real phone). Furthermore, many users stated that they would expect to improve with regular use.

Overall, the results are encouraging and while the watch interface is confirmed to be slower than a phone interface for text entry, the results show that text entry speed on a watch-face by a frequent user can be expected to be reasonably close to that on mobile phone keypad. Furthermore, users were all comfortable with the text entry method after very little training, satisfying the need for a method similar to mobile phones.

Acknowledgements

My gratitude is extended to the subjects for giving their time to these experiments.

References

1 Kushler, C., “AAC Using a Reduced Keyboard”, *Proceedings of Technology and Persons with Disabilities Conference*, California, 1998

2 Dunlop, M. D., and Crossan, A., "Dictionary based text entry method for mobile phones", *Proceedings of 2nd Workshop on HCI with Mobile Devices*, August 1999.

3 Silfverberg, M., MacKenzie, I. S., & Korhonen, P. “Predicting text entry speeds on mobile phones”. *Proceedings of the ACM CHI 2000*, 2000.

4 MacKenzie, I. S. “Mobile text entry using three keys”. *Proc. NordiCHI 2002*, 2002.

5 Venolia, D., Neiberg, F., "T-Cube: a fast, self-disclosing pen-based alphabet", Proc. ACH CHI 94, 1994

6 J. Mankoff and G. Abowd. Cirrin: A World-Level Unistroke Keyboard for Pen Input. In *Proceedings of UIST'98*, 1998.