Predictive text entry methods for mobile phones

Draft copy - do not cite or quote - submitted to *Personal Technologies*.

Mark D Dunlop
Centre for Human Machine Interaction
Risø National Laboratory
Denmark

mark.dunlop@risoe.dk

Andrew Crossan
Department of Computing Science
University of Glasgow
Scotland

ac@dcs.gla.ac.uk

Abstract

Mobile phone networks are increasingly supporting the transmission of textual messages between mobile phones and between mobile phones and other services. This paper describes the current text entry method on mobile phones and describes a new text entry method using a single key-press per letter together with a large dictionary of words for disambiguation. This approach, which is similar to technology recently licensed, independently, to several phone companies, is then extended with automatic word completion. The paper reports the results of initial user tests comparing the text entry methods, analysis of word clashes with the dictionary-based methods and keystroke level modelling of the different input methods.

Keywords: mobile phone, text entry, predictive, word completion

1. Introduction

Mobile phone networks are increasingly supporting the transmission of textual messages between mobile phones (e.g. SMS short messages) and through The Wireless Application Protocol (WAP) [WAP Forum 1999] to other systems and services. It is reasonable to expect increased use of these facilities and increased integration with other electronic services such as e-mail and web services. However, the use of textual messages from mobile phones is inherently limited by the very poor text input facilities: mobile phones only have 12 main keys with 5-10 additional function keys (compared to the smallest laptop keyboard of around 65 keys, a problem commonly shared by the smaller palmtop computers).

While the optimal solution would be Englebart's chord keyboard [Englebart and English 1968], this input device is still considered to involve too high an initial learning load to be useful in non-specialised consumer products. This paper investigates an alternative text entry method using a single key-press per letter, on a standard phone pad, together with a large dictionary of words for disambiguation. This method was then extended using word completion. The paper first describes the three methods, then presents a keystroke-level analysis of the input methods, followed by the results of preliminary user tests of the methods. Finally, the paper discusses problems with the new method and possible future directions for research.

1.1. Traditional text entry method

The traditional approach to text entry on a mobile phone is to overload the number pad with characters so that, for instance, the number 2 is mapped to A, B and C (see figure 1 for mobile phone modelled in experiments). When in text entry mode, the end user is required to press 2 once for an A, twice for a B, thrice for a C and four times for a 2 (this is actually a simplification as many phones have more than four characters mapped to each button). This is further complicated by a problem in differentiating some key presses, for example 222 could be one C or the two characters AB. This is usually solved by forcing users to wait around one second before entering subsequent characters on the same key, so AB is keyed by 2-22 where '-' is a one second pause. To enter the phrase *see you in the pub* on a traditional mobile phone would involved the following sequence of keys 777722-22 1 99966688 1 44466 1 84433 1 78822 [1]

Clearly this is a cumbersome method for text entry and likely to be highly error prone. Two forms of error are common: novices can be too slow to multi-click resulting in the cursor moving on to the next character prematurely (resulting in, say, BA instead of C) while

[1] spaces are used around the 1 key to separate words for ease of reading, 1 acts as a space on the modelled mobile phone

Figure 1: Mobile phone modelled in experiments

experts can wait too short a time between subsequent characters on the same key (resulting in, say, C instead of BA).

1.2. Predictive text entry method

The principle behind the two predictive models presented here is that users will only press one key per letter (e.g. for a C the user should just press 2, and not 222). This will remove the multiple key-press and pause problems of the traditional phone but will rely upon use of a large dictionary to disambiguate keystrokes and an additional dialogue for unresolved ambiguities.

We hypothesise that for many combinations of key-presses, there will be only one or two words that match a given keystroke sequence (other combinations being non-sensible). A predictive model of text entry can make use of a large dictionary of words to suggest valid words to the user. Furthermore, statistical information on frequency of word use would allow the most common word to be suggested first when multiple words match the keystroke sequence. Under an ideal implementation of the dictionary model[2], to enter the phrase *see you in the pub* the user would key 733 1 968 1 46 1 843 1 782. The dictionary would disambiguate, for example, 968 to *you* as the most likely word from the possible set of words which can be made from WXYZ as the first letter, MNO as the second and TUV as the third. Of course, this approach faces an inherent weakness: the most likely word from a given sequence of key presses may not be the word the user is wanting to enter. The solution modelled here presents the user initially with the most likely word when (s)he presses the 1/space key. If this is not the required word, the user presses 1 repeatedly until the desired word is shown then (s)he can carry on with the message as normal.

This approach requires a large dictionary of words in the language of usage of the phone. Furthermore, the method also requires all morphological variants which are used in the language together with information on how often each variant is used in the language (i.e. some measure of how common a word is, so that, for example, *care* can be proposed over *acre*).

This method is similar to a text entry method developed independently by Tegic Communications [Tegic WWW] and now licensed to several major mobile phone companies under the name *T9 Text Entry* and is also available on palm-top computers. This study is, however, based on a slightly different technique developed in Glasgow independently. A short comparison will be made between techniques in the discussion section.

1.3. Word Completion Text Entry

An extension of this model would be to give the user the ability to carry out automatic word completion, much like many Internet browsers now support URL completion. When a user starts a word the most likely word given the current prefix could be proposed as a suggested word for auto-completion, thus further accelerating the input process. In this model, using the large dictionary described later, the phrase *see you in the pub* would be entered as 733 1 968 1 4 1 8 1 782 1 (*see you i t pub*). Here the 1 key is used to accept the current suggestion (automatically appending a space) and other characters are used to add additional letters with 'no' being used for alternative suggestions of the same length). While giving some benefit in the short sample text, it is more likely to be of benefit on longer words.

This approach is similar to approaches for word completion for users who cannot use a traditional computer keyboard or are very slow entering on a

---

2 Using the 77K-word dictionary discussed later this keystroke is correct for the given phrase.

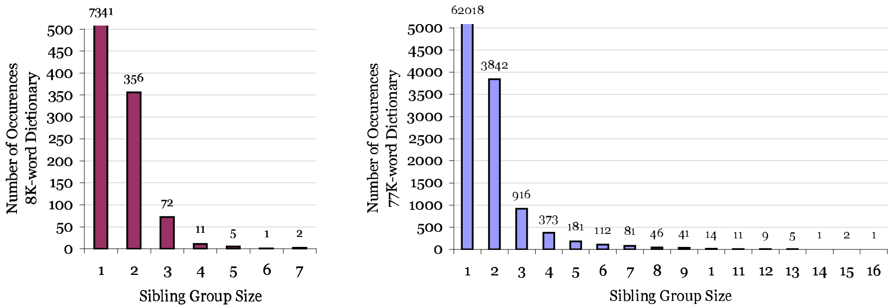


Figure 2: Number of occurrences of sibling groups.

keyboard (e.g. through use of a head pointer) [Gillette and Hoffman 1995]. It is also related to a method developed by Masui [1998] for on-screen, *soft*, keyboards. One problem, however, with the phone-based method is that prefixes are not characters but numeric keys (each representing three or four possible real characters). In an interface this can lead to problems as users enter a sequence of characters the most likely full word can be based on different interpretations of the prefix; for example, entering 34978 gives the word completion suggestions *FOR*, *FIRST*, *FIXED*, and *EGYPT*. This may be confusing for users and is difficult to overcome on the display. This issue is discussed more fully at the end of the paper.

2. Performance analysis

While the predictive method clearly reduces the number of keystrokes required in an ideal implementation, it is not clear whether the number of keystrokes would actually be reduced for actual messages based on real distribution of words in English, because of the additional keystrokes required to select the correct word from a suggestion list. Furthermore, the dictionary based entry methods may involve significantly higher cognitive loads, which would reverse any benefit gained from fewer keystrokes. Finally, it is not clear whether a suitable set of statistics could be derived from usage information. To gain an insight into these questions keystroke level models of the three interfaces were developed and a usability experiment was conducted using students in the university entering messages on a workstation emulation of a mobile phone.

2.1. Dictionary used in studies

Since the user experiments were to be conducted within The University of Glasgow, it was decided to base the dictionary, and statistical estimates of word usage, on a local newspaper. We had on-line access to six months of *The Herald*[3] from 1997 and carried out analysis to extract various lists of the most common words from the collection. The keystroke modelling uses a 77K-word[4] dictionary, while the user experiments were based on an 8K-word[5] dictionary. The 77K dictionary included all word uses in the six month period from *The Herald* whereas the 8K dictionary only included words used 44 or more times. For comparison, Shakespeare had a written vocabulary of 30K-words, a spoken and written vocabulary of 20K is generally considered good with a general population average nearer 10K and an everyday phone conversation requiring around 5K-words. [Reader's Digest 1975]. The Oxford English Dictionary (second edition) contains 231,100 main entries of which 47,100 are for obsolete words and 240 for spurious words. [Oxford English Dictionary WWW].

The use of a newspaper for the dictionary was convenient but not ideal, a large corpus of text is required to gain usage statistics. Ideally these statistics would be derived from messages conveyed by mobile phone users. However, the current text entry method is so slow as to encourage extreme, and often bizarre, abbreviations of words and phrases. We feel, however, that allowing for distortions for newspaper and political language (for example, *published* was the 55th most common word), a newspaper is a good starting point. Interestingly the use of a reasonably local paper does give some benefits: *Scotland*, *Glasgow* and *Edinburgh* all appear as the first suggested predicted word when all characters are entered under the predictive method and can be entered using six, three and three keys respectively using the word-completion method.

---

[3] A Scottish national newspaper published in Glasgow.

[4] 77 317 words

[5] 8 386 words

To investigate how likely users are to have to work down long lists of suggestions for key combinations under the predictive method, we investigated the size of 'sibling groups'. Sibling groups were defined to be groups of words that share the same keystroke sequence (and, under the predictive only method, are thus the same length). Figure 2 shows the number of occurrences for each size of sibling group (i.e. groups of size 1 contain one unique word for a given code, groups of size 2 contain two words of for the same code, etc). This shows that the vast majority of words do not have code siblings and that even for the 77K-word dictionary the worst case is that a word is has 15 siblings choice (in the case of the code 222 which mostly matches acronyms; the top three being *BBC, BAA* and *ABC* and the bottom six ranked combinations occurring less than 4 times over the six month training data).

For the keystroke analysis reported below, a test document was created composed of 17 articles taken from *The Herald On-line* on one day in 1999. This text contained 2481 unique words and 9737 word occurrences, including 34 words not in the large dictionary (43 occurrences). Words not in the dictionary were removed for test purposes, resulting in a text which will be referred to as the *comparative text*.

2.2. Keystroke level modelling

To gain an initial insight into the potential performance of the text entry methods, keystroke-level models [Card, Moran and Newell 1980] were developed for the three entry methods. The keystroke-level models are intended to give estimates of task completion time for prescribed tasks based on error-free usage by an uninterrupted user. The time estimate is based on developing an equation which models the interaction down to individual keystrokes. From the six timing elements contained in the keystroke-level model, three are used here to model the interaction:

T_k Time or button press - the time taken to press a button. The keystroke-level model includes seven times for key-stroking, ranging from 1.20s for the worst typist (one unfamiliar with a keyboard) through to 0.08s for best touch typist (equivalent to 135 words per minute). For text entry on a keyboard where the number of keys is drastically reduced but is typically used with only one finger, we have taken the "average non-secretary typist" rate of 0.28s (40wpm). The discussion section briefly compares performance given different typing speeds.

T_h Homing time for the hand to move to the keyboard (fixed 0.40s)

T_m Mental preparation time for executing physical actions (fixed 1.35s).

There is no drawing (T_d), system response time (T_r) nor pointing time (T_p) as the system is only key-based and gives instant reactions to key-presses.

Traditional phone entry

Traditional phone entry of a given phrase (P) involves a homing time followed by w words. Each word is composed of k_t keystrokes on average[6] and d delays (where d is the number of pairs of consecutive characters in an average word that are on the same key, i.e. require a delay for the cursor to move on to the next character). For example, *see you in the pub*, involves 35 key presses and one pause (on the repeated *e*). For a phrase of w average words this leads to:

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

Analysis of the comparative text yields an average word length (including final space) of 5.98 characters per word. Entering these on a traditional mobile phone would require an average of 11.85 key presses (k_t) and 0.49 multi-press pauses per word. For a ten word sentence this gives a total time of 40.2s (equivalent to 14.9 words per minute).

Predictive text entry method

Predictive text entry of a word requires one key press per letter, k_p, followed by the end-of-word key (here the 1 key is used for end-of-word and automatically appends a space when the user moves on to the next word). The end-of-word key needs pressed l times, where l is the average position in the ranked list of suggestions, e.g. pressed once for the first suggested word, twice for the second and so on. Pressing a non-end-of-word character starts subsequent words. In terms of keystroke equations:

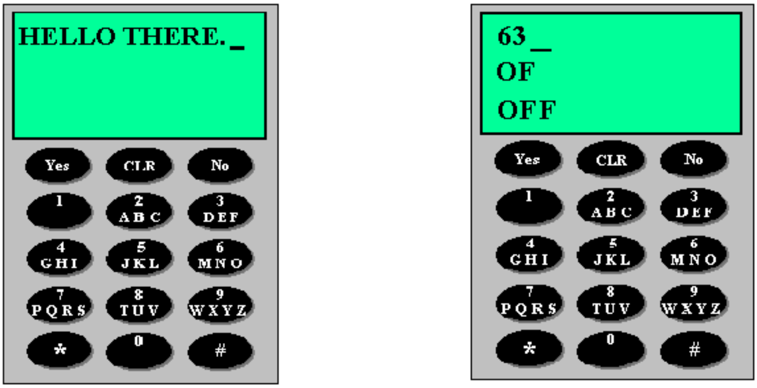
$$T(P) = T_h + w\ (k_p T_k + l(T_m + T_k))$$

Since this model is based on a single keystroke per letter the number of keypresses per word, k_p, equals the average word length of 5.98. The large dictionary yields an average position in the ranked list for words from the comparitive text at position 1.03 (average of 6.02 keys per word). For a ten word sentence this gives a total time of 34.0s (or 17.6 words per minute).

---

[6] For simplicity in modelling, all words are assumed to be entered in a single case, contain no punctuation and be followed by a space.

In keystroke modelling the traditional phone, the 1s delay required to move on to the next character when subsequent letters are on the same key is subsumed by the mental preparation time.



Traditional *Hybrid Dictionary-based*

Figure 3: On screen emulator.

Word-completion text entry

Word completion text entry of a word involves pressing the key corresponding to the first character, followed by:

- space if the suggested word is correct;
- the next character of the word if the suggestion is wrong and there are more letters to go;
- 'no' if the suggestion is wrong and there are no more letters to go (repeated until the correct suggestion is given)

This gives an equation as follows:

$$T(P) = T_h + w\ (\ T_k + (\ k_c\text{-}1)(T_m + T_k)\ )$$

Again using the large dictionary and comparative test, the average number of keys required per word, k_c, is 4.60 for an average word length of 5.98 (including final space), a keystroke saving of approximately 25%. However, once mental preparation operators are taken into account this leads to a time for a 10 word phrase of 7.81s (equivalent to only 7.7 wpm).

Keystroke Modelling Discussion

The Keystroke-level modelling predicts an improvement of 18% in typing speed from traditional mobile phone keyboard to predictive, dictionary-based input. With the relatively small number of repeat key pauses, this improvement is coming mostly from the reduced number of keystrokes required to enter only one key-press per letter. Furthermore, only 1.03 mental operations are required per word as the vast majority of first suggested words are correct.

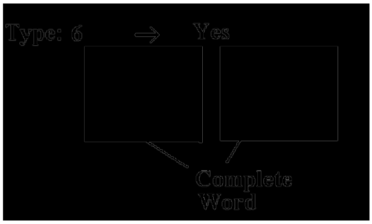
In contrast, the word-completion method is predicted to decrease performance to 58% of that of the traditional method. The prediction model is working fairly well with an average saving of 17% on keystrokes. However, each of these keystrokes requires a mental operation to decide whether the suggested word is correct or whether more letters are required. The averages used here are based on occurrence information and so take into account the saving on frequently occurring words proportionally. The problems with word prediction and reasons for inclusion despite this prediction are discussed later in light of user tests.

Mackenzie, Zhang and Soukoreff [1999] present an interesting comparison of entry speed using soft keyboards. They used Fitt's law for rapid aimed movements and Hick-Hyman's law for choice selection time to analyse 6 keyboards:

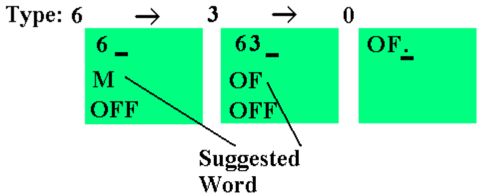
- the standard QWERTY keyboard;
- the Dvorak keyboard (a rearranged standard keyboard, optimised for physical keyboard typing speed);
- the ABC keyboard (a tall thin alphabetical keyboard developed for on-screen use);
- the FITALY keyboard (a square 28-key keyboard optimised for on screen speed)
- the standard Telephone keyboard (assuming perfect word prediction with single key press per character);
- and the JustType keyboard (an optimised 12-key arrangement).

Although their predictions are very promising for predictive entry on telephones they do not address the issue of additional dialogue when disambiguation fails and thus do not include mental processing times. Their predictions also show only a marginal improvement for the JustType layout which would require users learning a new complex key layout, the traditional phone being alphabetic.

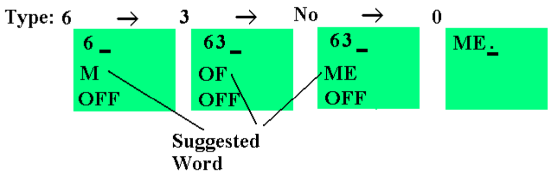
Mackenzie et al. observed that if the vocabulary is set accordingly, ambiguities can be prevented. While this



Entering OFF *(using word completion)*



Entering OF *(prediction only)*



Entering ME *(second predicted suggestion for keys 63)*

Figure 4: Three example words entered with the hybrid dictionary method

is not a general solution it might be possible within specific domains and if possible then the mental preparation operations above would be removed, leading to considerably higher input speeds.

2.3. User tests

In this pilot study 14 users were to enter three sentences on a mobile phone emulator twice: once using the traditional multiple click method and once using a hybrid dictionary based method. The users were split into two groups, half of whom did traditional first, half dictionary-based. All groups were given a short period of training in each method just before using that method. The trials were measured in terms of number of keystrokes, time to complete the task and subjective workload.

Experimental Setting

The small, 8K-word, dictionary was used by 14 subjects to enter the following three training phases and three experimental phrases (all written by a third party under the instruction to write short messages *in the style of messages you would send by mobile phone*). The phrases were:

Training:

- Can you pick us up at the airport
- My dog is called rover
- Hello. How are you doing?

Experimental:

- Hi. In the pub now - where are you?
- At the supermarket. Need anything?
- Missed the train so I'll get the next one.

The tests were conducted using an on-screen emulator[7] (figure 3) written in Java and running on a Sun workstation. In traditional mode, the emulator showed the entered text on the single line display. In dictionary-based mode the top line shows the keys pressed, the second line the most common word given the current keystroke (predictive text entry) and on the third line the most common word with the given keystroke prefix (word completion).

The keystrokes required for the hybrid-dictionary model are highlighted in figure 4 and were as follows:

- buttons 2..9 were used for character input;
- button 1 was used for space and button 0 for punctuation (using repeated clicks for different punctuation marks, both buttons of which accepted the suggested word and moved on);
- the NO button was used to cycle round predicted words after entering all character codes for that word;
- the YES button was used to accept the auto-complete word at any point.

---

[7] The experiments were conducted on an on-screen emulator since reprogramming mobiles phones is prohibitively expensive. However, there were some effects of this, for example double- and triple-clicking with a mouse is considerably quicker and easier than on most phone keypads while users are likely to be more at home with reactive systems on a computer screen than on a telephone.

Results from user study

The results showed that on-average users entered a total 81 key-presses for the traditional phone and 46 for the hybrid-dictionary phone (with all three sentences giving statistically significant differences).

The users took a total of 223s to enter the three phrases on the traditional phone and 204s on the hybrid phone (with only the middle sentence showing a statistically significant result) (see figure 5).

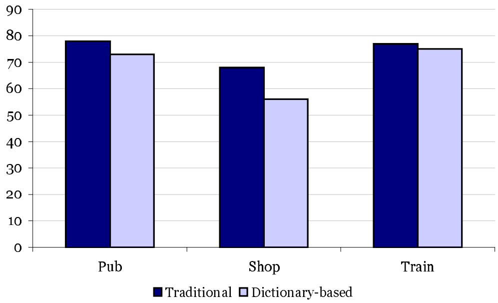


Figure 5: Time taken for each sentence

To assess how much pressure and frustration users felt NASA TLX task workload sheets were completed by each user after each interface [Hart and Staveland, 1988]. The results, shown in figure 6, show statistically significant results for physical demand, frustration and annoyance (all in favour of the dictionary-based method). However, while not statistically significant, the results for mental demand and effort expended are higher for the dictionary-based method.

Although no statistics were kept, users did make use of word completion and claimed to like the feature. However, possibly in line with the keystroke-level predictions, they tended to over use the facility which may have resulted in slower than expected overall performance.

3. Discussion

Although the experiment attempted neutrality by using sentences which were written by a third party to be "in the style of a message you would be likely to send by phone message", there were only three sentences in this provisional experiment and it is not clear that these are the kind of message that users would actually send using a mobile phone with better text entry. A more detailed laboratory experiment has to be conducted with more users and more test sentences to make conclusive design decisions. Furthermore, after the design has been refined in the laboratory longitudinal studies would be required of users using the system implemented on a mobile phone. In particular, it would be interesting to carry out comparative experiments while the users are on the move.

The use of a national newspaper to gain statistical information worked well but was biased by the language used by a newspaper, for example political terms and political names were more prominent than is likely in mobile phone messages. This is likely to be a bootstrapping problem of the dictionary based phone system, ideally the dictionary would be based on the use of English for sending messages by phone but the current method is so slow to be distorting those message (e.g. by use of very short abbreviations between friends).

A related, but somewhat more serious, issue is the flexibility of the dictionary-based phone system, as only words in that dictionary can be entered. There are many solutions to this which would have to be experimented with: the dictionary could be augmented with proper names from, say, telephone directories and birth registrars as well as the users' own

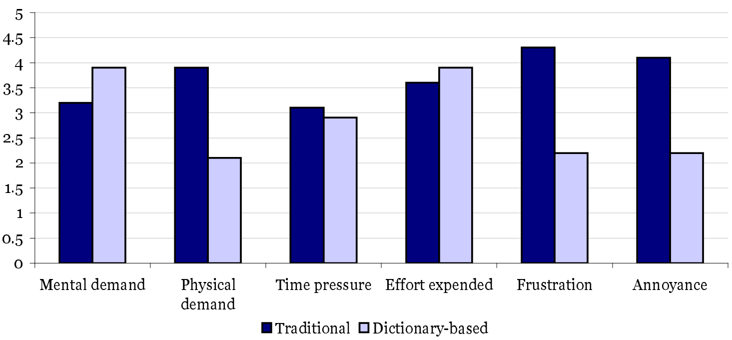


Figure 6: NASA TLX Results

Code	*Predicted word*	*Suggested completion*
3	D	FOR
34	DI	FIRST
349	FIX	FIXED
3497	*no match*	EGYPT
34978	EGYPT	EGYPT

Code	*Predicted word*	*Suggested completion*
7	S	SAID
72	QC	SAID
726	RAN	SCOTTISH
7268	SCOT	SCOTTISH
72685	RAOUL	SCOTLAND
726852	*no match*	SCOTLAND
7268526	*no match*	SCOTLAND
72685263	SCOTLAND	SCOTLAND

Table 1: Sample unstable keystroke combinations

telephone book if this could be imported from another device. However, for full flexibility a mode based system would be required where users could force the phone into traditional mode to learn words not in the dictionary. This is a serious problem: the dictionary method is not a sufficient input method on its own but having two modes is likely to lead to great confusion. The use of users' own telephone directories also introduces the notion of adapting the statistical model of likelihood for each word to take into account previous usage, so that, for example, one's partner's name automatically becomes easier to enter over time. This would, however, alter the keystroke sequence and may reduce the ability of users to convert common words into a learned sequence (or *skill-based* activity).

Both word completion and simple prediction suffer from a problem discussed briefly earlier, the keystroke sequence does not represent characters but a sequence of numbers – each number representing a character from a set of three or four potential characters. This can lead to two problems:

- When a user starts a word by pressing a character, another character may be displayed (e.g. when (s)he enters what (s)he considers an *M*, an *O* might be shown on screen);
- The predicted and completed word suggestions are likely to be unstable under some key combinations, when the most likely word for a given sequence changes drastically as each additional character is entered (for examples see Table 1).

Both these problems are likely to lead to initial resistance from users on their first use of the system (the former being a potential major problem for initial experimental use by users who have not read their manuals). The later problem is likely to persist for experienced users as there is insufficient feedback that the correct key sequence has been pressed. While we cannot suggest a perfect solution to these problems, the use of word completion does make the sequence more stable and could be biased, at least initially, to give suggestions from different possible characters on the first key (e.g. suggest the most common word for A, B, and C when a user presses 2). This would, however, only work on multiple line high resolution phone displays.

For experts users of the predictive model, the mental preparation times modelled in the keystroke analysis are likely to disappear as users learn the complete keystroke sequence for commonly used words (i.e. they move from rule-based to skill-based usage of the system [Rasmussen 1986]). Similarly, some of the mental preparation times will disappear from the word completion method. However, in terms of raw keystroke-level modelling, the faster a user's typing speed the less the benefit of the new model (for typist with 0.5s keystroke the predictive model leads to a speed-up of 34%, while this method is predicted to slow down users with a keystroke speed of under 0.153s). As with many of the issues raised here, further empirical work is needed to reach a better understanding of these effects in practice.

The keystroke level models developed here, in line with all keystroke level modelling, reflect skilled-user error-free interaction. With large dictionaries resulting in the correctly predicted word on the vast majority of cases, it could be argued that the predictive method would lead to less errors and corrections than the traditional phone entry method – this increasing the performance gain for novice users. This is likely to be one of the main reasons for the reduced frustration levels displayed in the user tests.

One double-edged issue with predictive only dictionary-based text entry is that only correctly spelled words can be entered. This is a benefit in that words are automatically spell checked and should massively reduce the amount of editing required after writing text. However, not all mobile phone users will be good spellers and this could lead to problems of the system refusing to suggest words because the keystroke sequence does not match the word the user is wanting (because (s)he has spelled it wrongly). Tegic have solved this to some extent by including common misspellings of words in the dictionary. The hybrid model presented here partly solves the problem differently by allowing users to select common, correctly spelled, words without having to type the whole word in. Spell correction algorithms may also help, as correctly spelled but wrongly typed words could be suggested with correctly typed words.

Finally, we have taken the approach of using only statistical information to predict words. Other techniques, such as lightweight linguistic parsing (simpler than most grammar checkers in modern word processors), could be used to increase the accuracy of the first suggested word. Working within a domain where the choice of words is very limited, by the keys pressed, techniques such as simple sentence structure analysis might be good enough to make the first word predicted 100% correct. If this is the case, then the user can stop worrying about the words and type more freely with considerably less mental effort.

4. Conclusions

This paper has described three methods of text entry using the limited keyboard of a mobile phone: traditional multi-press method, predictive method based on one key press per character and word-completion method based on fewer than one key press per character.

Keystroke-level models predict a 18% increase in typing speed for predictive text entry (but a reduction of performance when using word-completion) and a reduction of 49% in number of keys pressed.

User tests using a hybrid word-completion and word-prediction method showed a 10% increase in performance over the traditional method. Furthermore, users felt significantly less frustrated and annoyed with the interface using the dictionary method and felt they had worked less hard physically. However, non-statistically significant results suggested increased mental demand and effort expended with the dictionary-based model.

These results are encouraging for predictive text entry method, and probably conservatively so – a variation of this method is now being implemented on many mobile phones (independently of this research). It is unclear, however, from the experiments conducted here whether the indicative increase in mental load in the user trial is as a result of the predictive or word-completion elements. Considering the success rates for large-dictionary, we feel the later is more likely as the keystroke modelling highlights the very high number of mental operations required for word-prediction. In favour of word prediction, however, is that it could provide a general solution to bad-spelling and designed as a secondary feature may help users adopt a strategy of only using word-completion when they are unsure about spelling or for long words where the benefit is likely to be high.

Further research needs to be carried out to investigate: the indicative rise in mental effort; the use of the traditional and predictive methods in mobile settings where the users mental effort is shared; the inclusion of word-completion as a secondary feature and longitudinal studies on whether users can develop strategies that allow word-completion where helpful or necessary but such that the mental load does not punish normal text entry; and finally studies could be carried out into adaptive dictionaries methods when a word is not in the dictionary and other improvement techniques discussed here.

Acknowledgements

We would like to thank the subjects in our experiments and those who have given extensive comments on earlier versions of this paper, in particular Morten Hertzum who helped tighten up the paper considerably.

The experimental and emulator development work was carried out by Andrew Crossan, supervised by Mark Dunlop, at The University of Glasgow. The keystroke modelling and analysis work was partly funded by The Danish National Research Foundation through it Centre for Human Machine Interaction..

References

Engelbart, D.C., and English, W.K. "A Research Center for Augmenting Human Intellect," *AFIPS Conference Proceedings of the 1968 Fall Joint Computer Conference*, San Francisco, CA, December 1968.

Gillette, Y., & Hoffman, J.L. (1995, March). "Getting to word prediction: Developmental literacy and AAC". *Proceedings of the Tenth Annual International Conference, Technology and Persons with Disabilities*.

Hart, S. & Staveland, L.E. (1988) "Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research", *Human Mental Workload,* Hancock, P.A. & Meshkati, N. (eds.), Elsevier Science, 139-183.

Mackenzie, I.S., Zhang, S.X., and Soukoreff, R.W. (1999), "Text entry using soft keyboards", *Behaviour and Information Technology*, vol 18 (4), pp 235-244.

Masui, T. (1998), "An efficient text input method for pen-based computers", *Proceedings of ACM*

Conference on Human Factors in Computing Systems (CHI'98), ACM Press.

Oxford English Dictionary (WWW), http://www.oed.com/

Rasmussen, J. *On Information Processing and Human-Machine Interaction: An Approach to Cognitive Engineering*, Elsevier, 1986.

Reader's Digest (1975), *Oddities in words, pictures and figures*, London

Tegic Communications (WWW), http://www.tegic.com/

WAP Forum (1999), *WAP White Paper*, WAP Forum, Mountain View, California, USA. (http://www.wapforum.org/what/whitepapers.htm).