Strathprints Institutional Repository

Towards High Quality Text Entry on Smartwatches

Mark Dunlop

Computer and Information Sciences

University of Strathclyde

Glasgow G1 1XH, Scotland UK

Mark.Dunlop@strath.ac.uk

Andreas Komninos

Computer and Information Sciences

University of Strathclyde

Glasgow G1 1XH, Scotland UK

Andreas.Komninos@strath.ac.uk

Naveen Durga

KeyPoint Technologies Ltd

Raja Praasadamu Junction, Botanical Gardens Road

Kondapur, Hyderabad 500081, India

ndurga@keypoint-tech.com

Figure 1: Prototype smartwatch text entry on Sony SmartWatch 2

Abstract

Smartwatches now provide users with access to many applications on smartphones direct from their wrists, without the need to touch their smartphone. While applications such as email, messaging, calendar and social networking provide views on the watch, there is normally no text entry method so users cannot reply on the same device. Here we introduce requirements for smartwatch text entry, an optimised alphabetic layout and present a prototype implementation together with preliminary user feedback. While raising some problems, the feedback gives indicates that reasonable quality and speed is achievable on a smartwatch and encourages our future work.

Author Keywords

Text entry; smart watches; mobile usability

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

Introduction

Text entry is a key component of many smartphone applications, ranging from adding simple diary entries through social network postings to writing complex business emails and documents. The recent release of smartwatches has met considerable interest, but without

text entry the interaction is frustratingly limited. One can see posts, short-messages and emails but one can't reply on the same device. In this work-in-progress we outline a text entry approach for smartwatches and also describe our initial prototype (Fig. 1) and an initial small-scale usability study.

Text entry on small devices

Before the widespread adoption of touch screen smartphones, 12-key physical keypad phones were the most common text entry method on small devices (sometimes much smaller than the current relatively large screen phones). Predictive technologies (e.g. [4, 10, 11]) interpreted the ambiguous keys (usually three or four letters per key) into words. This approach was shown to achieve around 10 words per minute (wpm) for novices and around 20-25 wpm for experts in controlled studies [12]. We investigated using this approach with reduced number of keys - initially targeting watches [3], but due to technological restrictions our system was implemented on a touchscreen handheld, and later, to reduce movement, on physical key phones [6]. While in theory ambiguous predictive text quality was very high (over 90% accurate), each key sequence could match many different words. Some of these sequences included pairs of common words that caused particular problems (e.g. on a standard phone keypad *he* and *if* were typed on the same keys, as were *good* and *home*). The early models of prediction were based on simple unigram dictionary models where the most common word matching a sequence was suggested. Nowadays, phones have much more power and memory so can easily support more complex prediction models, which greatly reduces the impact of ambiguity by taking the context of the words into account.

Alternative approaches for input on small devices include handwriting with a stylus, but this has been shown to be relatively slow at under 20 wpm [16], fast but difficult to learn chord keyboards (e.g. [15]), specialised alphabets (e.g. [8, 19]) and sending handwritten image messages instead of text [13].

One system of particular note is ZoomBoard [17] that miniaturises the QWERTY keyboard onto a watch, with the user first zooming into an area then picking a letter. ZoomBoard achieved approx. 10wpm in studies and our work is essentially investigating whether predictive based approaches can surpass this using single taps per letter. Many domestic appliances such as TV guides and games consoles use a date-stamp inspired method, where the user scrolls through the alphabet and picks letters on a 2D line or 3D grid. In their early work on small device text entry Bellman and MacKenzie [1] showed this to be a slow entry method that was not particularly helped by dynamic optimisation.

Speech input is an obvious alternative but is still prone to problems with background noise, spoken accents and is less private and discrete than typing.

Criteria for smartphone style text entry on smartwatches

For appropriate interaction on a smartwatch we constrained our design by the following criteria, derived from the literature and our own experience:

- Entry must be based on finger use: for casual interaction using a stylus is inappropriate.

- Entry must target a small display (e.g. 25x25 mm) with simple touchscreen interaction.
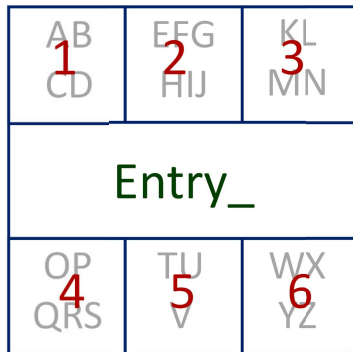
Figure 2: entry layout

← Backspace
→ Word completion
↑ Toggle capitalisation
↓ Numeric punctuation mode

Table 1: Gesture definitions

• Limited dynamic feedback - current smartwatches have limited on device processing with much processing handled on the paired phone. Providing dynamic graphical feedback can challenge the communication between devices.

• Entry should focus on small messages and short texts but should support a wide range of input:

  o Common Western European alphabets in upper and lower case;

  o Numbers

  o Common punctuation;

  o Common symbols (e.g. widely used social networking symbols).

• Should support simple editing of text (e.g. caret movement and backspace).

• For accurate tapping with fingers touch targets should be reasonably large, e.g. >= 7 mm.

• Interaction should include simple gestures that are easy to learn and remember and which users are already familiar with, e.g. on-screen taps and simple short unidirectional swipes that are independent of their location on screen.

Initial Design

We decided to focus on taps for the prime input method, as this is the quickest simple interaction to perform. Although word based gesture input techniques (e.g. [14, 20]) have shown great benefits for mobile phones, we felt that simple taps on large keys are more suited to the watch environment. As such we segmented the display into seven zones as shown in Figure 2. Zones 1 to 6

form large ambiguous keys while the centre zone shows the current input text and acts as a space bar.
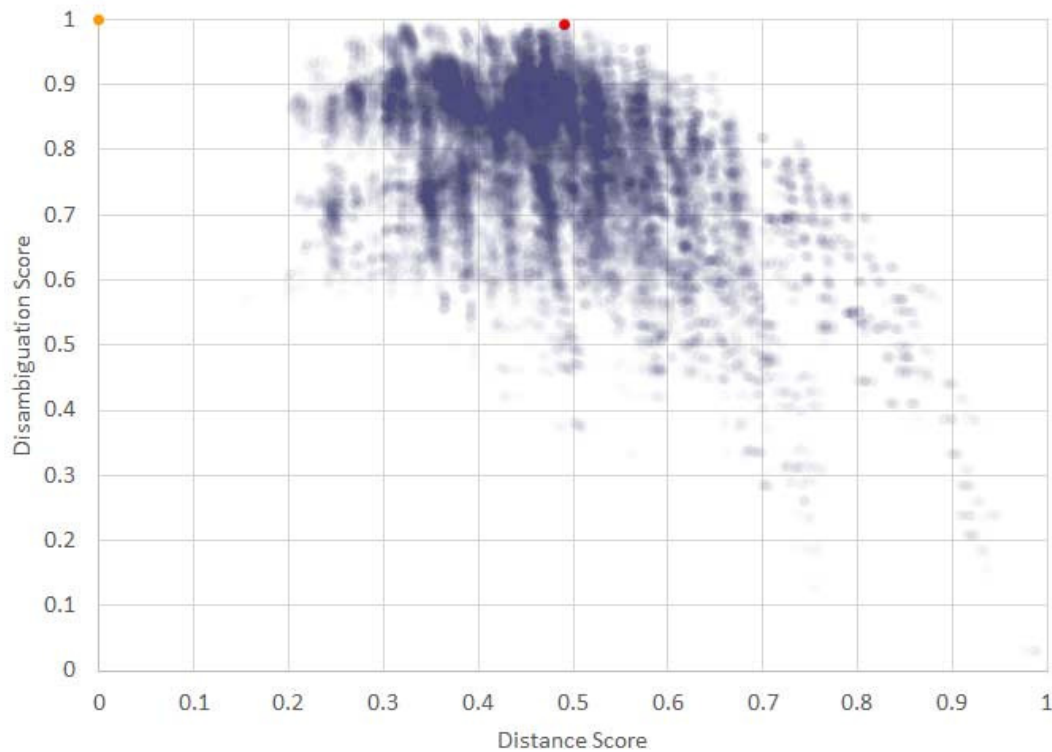
While there has been considerable work on optimised keyboard layouts (e.g. [2, 5, 9, 18]), here we decided to maintain a standard alphabetical layout to aid pick-up usability. For alphabet entry the user will type on keys 1…6 with the input being disambiguated by the text entry system (running on the paired smartphone).

Interaction was defined as follows:

1. A tap on an ambiguous key enters that key number and updates the current word display to reflect the most likely word from the disambiguation engine based on the current number-key sequence.

2. A first tap on the central zone adds a space, with subsequent taps rotating round alternative suggestions that match the ambiguous entry.

3. Swipes are defined as shown in table 1. In punctuation and numeric modes the zones 1…6 are replaced with alternative layouts. Where available, a rightwards swipe gesture can be used for word completion.

4. A long press on the centre zone enters edit mode to allow movement of the caret while a long press on the alphabetic keys will show extended characters for that key (e.g. à, å, ç).

While we decided to restrict the design to an alphabetic ordering, there are, of course, many ways to split the alphabet. There are two competing optimisation criteria here: ambiguity of the layout vs movement distance.

To reduce ambiguity errors the best assignment of letters to keys would result in the lowest ambiguity of keystrokes by separating letters that can commonly cause confusion when in the same location in a word (e.g. putting *a* and *e* on the same key would be problematic as common words such as *bed* and *bad*, for example, are only differentiated by this pair). Arranging the splits can help minimise the distance a user has to move his/her finger when entering text by putting commonly co-occurring letters on the same key. In the extreme case putting all 26 letters on one key would minimise the amount of movement of the fingers while typing, but at a massive cost to ambiguity.

We analysed the 53,131 possible alphabetic arrangements using a normalised ambiguity score based on *badgrams* frequencies for English and distance based on *bigram* data (using same data as [5]). The least ambiguous keyboard was *abcd efgh ijklm nop qrs tuvwxyz* while the keyboard with least travel for the finger was *abcdefghijklmnopqrstu v w x y z*. Figure 3 shows the distribution of the layouts (both axes are scaled to the range 0…1, where 0 is the worst we found and 1 the best). Because distances are small on a watch, we felt it more important to select a keyboard layout that minimised ambiguity rather than movement. Hence, to select a keyboard we took a weighted average giving disambiguation score more weight than distance score. The best compromise keyboard was selected as *abcd efghi jklmn opqrs tuv wxyz* which is very high ranking in disambiguation score and the highest distance scored keyboard on the plateau in Figure 3 (this keyboard is shown in red (top centre)). For reference the traditional phone keyboard is shown in orange in Figure 3 (top left) - showing our 6-letter-key layout performs very close to the 8-letter-key phone layout in terms of raw ambiguity of layout. However, as discussed above, prediction technology has improved considerably since physical phone predictive text so we expect much higher prediction accuracy in practice.

Initial Implementation

Building on the OpenAdaptxt [7] framework provided us with a powerful disambiguation engine that gives contextually based word suggestions, word completion and next word prediction. Our implementation was built

Figure 3: Distribution of keyboard scores
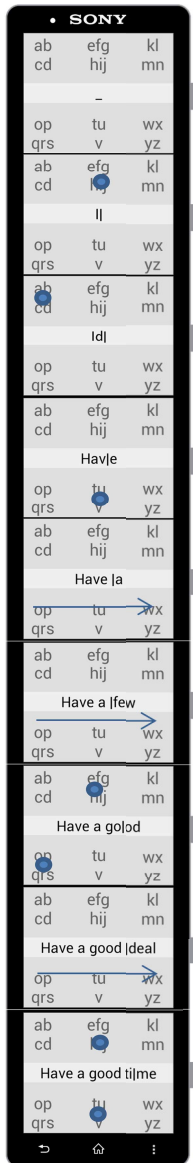
Figure 4:
Interaction sequence to enter *Have a good time*. 🔵 represents tap while ⟶ represents a left to right swipe.

using OpenAdaptxt running on a Sony Xperia Android smartphone paired to a Sony SmartWatch 2. This watch has a 30 x 25 mm screen linked by Bluetooth to the smartphone, where the bulk of processing is done.

For our initial prototype we implemented elements 1 and 2 of our design from above along with the backspace ← and completion → gestures. As users typed, the previous context of their entries was used to steer disambiguation using OpenAdaptxt English dictionaries. Figure 1 shows our initial implementation and Figure 4 a storyboard of entering a short phrase.

Preliminary User Studies

We conducted preliminary user studies with four users entering standard short phrases. While clearly preliminary, our users did appeared to quickly adapt to the layout and had little trouble finding letters in the new layout. They also found the word completion gesture quite natural and found prediction quality high. However, two categories of issues were raised: sensitivity of the watch and screen layout design.

In need of careful investigation before a larger study we found some problems with sensitivity. In particular the watch appears unresponsive to fast repeat taps on the same area. The watch used in the studies also appears to be overly sensitive to movements of the finger while tapping, registering these as swipes instead of taps when using the in-build gesture events. Finally, we give a brief vibration as tactile feedback on taps and swipes being recognised - this appears to be harder to notice for some users when wrist-worn rather than hand-held as normal with a phone. We will conduct tests on these issues before further studies.

The current layout has a fairly narrow central "space bar" - this needs to be vertically expanded in the revised version. Giving some visual indication that this is a space bar may also help, as this was not obvious on first use. The space between the upper and lower letters on each ambiguous key also needs to be reduced and more indication given that the group forms a single button, as users were initially trying to tap the actual letters rather than the large ambiguous buttons. An ability to quickly swap back to (or perhaps show in miniature / watermark) the context of text entry from the original application will also be investigated.

Discussion and Future Directions

In this work-in-progress we have introduced an optimised layout for ambiguous smartwatch text entry that maintains alphabetic sorting while attaining high disambiguation quality and keeping finger movement distance low. In our preliminary studies we have seen that users can adapt easily to the key layout and found the prediction quality high, but we also found some interaction and visual problems that need further investigation.

We are now developing our code into a fully functional smartwatch text entry method following the criteria we have outlined here. Once complete we will conduct extensive user studies to analyse potential speed. One interesting opportunity is to more fully exploit the word completion and next word suggestion features of OpenAdaptxt - these can provide very fast entry on a word-by-word basis for regularly used phrases or language constructions. However, adding controls for these in the tight space on the watch will be challenging.

References

[1]   Bellman, T., and MacKenzie, I. S. A probabilistic character layout strategy for mobile text entry. In *Proc. Graphics Interface '98*, CIPS, 1998.

[2]   Bi, X., Smith, B. A. and Zhai, S. Quasi-qwerty soft keyboard optimization. In *Proc. CHI 2010*, ACM, 2010.

[3]   Dunlop, M. D. Watch-Top Text-Entry: Can Phone-Style Predictive Text-Entry Work With Only 5 Buttons? In *Proc. MobileHCI 04*, Springer LNCS, 2004.

[4]   __ and Crossan, A. Predictive text entry methods for mobile phones. *Personal Technologies*, 4(2), 2000.

[5]   __ and Levine, J. Multidimensional Pareto optimization of touchscreen keyboards for speed, familiarity and improved spell checking. In *Proc. CHI '12*, ACM, 2012.

[6]   __ and Masters, M. M. Investigating five key predictive text entry with combined distance and keystroke modelling. *Personal Ubiquitous Computing*, 12(8), 2008.

[7]   __, Montaparti, S., Dona, P., Durga, N. and Meo, R. D. OpenAdaptxt: An Open Source Enabling Technology for High Quality Text Entry. In *Proc. CHI Workshop on Designing & Evaluating Text Entry*, 2012.

[8]   Goldberg, D. and Richardson, C. Touch-typing with a stylus. In *Proc. INTERCHI93*, ACM, 1993.

[9]   Gong, J. and Tarasewich, P. Alphabetically constrained keypad designs for text entry on mobile devices. In *Proc. CHI '05*, ACM, 2005.

[10] Grover, D. L., King, M. T. and Kushler, C. A. *Reduced keyboard disambiguating computer*, Tegic Communications, Inc., Patent US5818437, 1998.

[11] Hasselgren, J., Montnemery, E., Nugues, P. and Svensson, M. HMS: A Predictive Text Entry Method Using Bigrams. In *Proc. Workshop on Language Modeling for Text Entry Methods at EACL*, 2003.

[12] James, C. L. and Reischel, K. M. Text input for mobile devices: comparing model prediction to actual performance. In *Proc. CHI'01*, ACM, 2001.

[13] Kienzle, W. and Hinckley, K. Writing handwritten messages on a small touchscreen. In *Proc. MobileHCI'13*, ACM, 2013.

[14] Kristensson, P. O. and Zhai, S. Learning shape writing by game playing. In *Proc. CHI '07 Extended Abstracts*, ACM, 2007.

[15] Lyons, K. M., Starner, T. E., Plaisted, D., Fusia, J. G., Lyons, A., Drew, A. and Looney, E. W. *Twiddler Typing: One-Handed Chording Text Entry for Mobile Phones*. Georgia Institute of Technology, 2003.

[16] MacKenzie, I. S., Nonnecke, R. B., McQueen, C., Riddersma, S. and Meltz, M. A comparison of three methods of character entry on pen-based computers. In *Proc. Factors & Ergonomics Soc. 38th Meeting,* 1994.

[17] Oney, S., Harrison, C., Ogan, A., and Wiese, J. ZoomBoard: a diminutive qwerty soft keyboard using iterative zooming for ultra-small devices. *Proc. CHI '13*, ACM, 2013.

[18] Oulasvirta, A., Reichel, A., Li, W., Zhang, Y., Bachynskyi, M., Vertanen, K. and Kristensson, P. O. Improving two-thumb text entry on touchscreen devices. In *Proc. CHI '13*, ACM, 2013.

[19] Wobbrock, J. O., Myers, B. A. and Kembel, J. A. EdgeWrite: a stylus-based text entry method designed for high accuracy and stability of motion. In *Proc. UIST'03*, ACM, 2003.

[20] Zhai, S. and Kristensson, P. Shorthand writing on stylus keyboard. In *Proc. CHI '03*, ACM, 2003.