Text Input on a Smart Watch

Andreas Komninos and Mark Dunlop
University of Strathclyde

Although smart-watches let users receive many forms of communication, there is usually no direct way of replying. We introduce an interaction design and an optimized alphabetic layout for smart-watch text entry and present an evaluation using a working prototype.

Text entry is a key component of many smartphone applications. The recent release of smart watches has met considerable interest, but without text entry, interaction is frustratingly limited—users can see posts, short messages, and emails but can't reply using the watch. As part of our ongoing work, we outline a text entry approach for smart watches, describe our initial prototype, and discuss the outcomes of our lab-based evaluation of the prototype.

Text Entry on Small Devices

Before the widespread adoption of touchscreen smartphones, 12-key physical-keypad phones were the most common text entry method on small devices. Predictive technologies interpreted the ambiguous keys (usually three or four letters per key) and suggested.[1,2] This approach was shown to achieve speeds of approximately 10 words per minute (wpm) for novices and 20 to 25 wpm for experts in controlled studies.[3]

In a previous study, we investigated this approach using a reduced number of keys for text entry on watches, but we implemented it on a touchscreen handheld device.[4] In theory, 12-key ambiguous predictive text quality could be very high (over 90 percent accurate), but in reality each key sequence could match many different words, and some of these sequences included pairs of common words that caused problems. (For example, on a standard phone keypad, "he" and "if" are typed with the same keys.) The early models of prediction were based on simple unigram dictionary models that suggested the most common word matching a sequence. Nowadays, phones have much more power and memory, so they can easily support more complex prediction models, greatly reducing the impact of ambiguity.

Alternative approaches for input on small devices include handwriting,[5] fast but difficult-to-learn chord keyboards,[6] and specialized alphabets.[7] Many domestic appliances, such as televisions and games, use a date-stamp-inspired method, where the user scrolls through the alphabet and picks letters from a 2D line or 3D grid. However, this has been shown to be a slow entry method.[8] Finally, gesture word-based input techniques have shown great benefits for mobile phones and can provide a fast and more relaxing input method than continuously tapping small on-screen buttons.[9]

There is currently very little work focusing on text entry for smart watches. One such method is Zoomboard, where a full QWERTY keyboard is shrunk to fill the smart-watch screen.[10] Users tap once to zoom into a keyboard area and a second time to select a letter from that area. Although shown to be good enough for input speeds up to 9 wpm, this interaction method lacks suggestion support and increases the number of interactions, because additional input is required for zooming. Furthermore, this method places a cognitive load on users who have to remember the approximate area where the desired key might be located. Minuum* recently demonstrated smart-watch entry using a keyboard that compresses a QWERTY keyboard layout to one line and incorporates word suggestions. This layout's efficacy has not been evaluated in a publication and the layout has not been formally evaluated as being optimal for this size of device, although the keyboard is a direct derivative of the work of F.C.Y. Li and colleagues,[11] who showed this keyboard layout to work efficiently on tablet-size devices.

* http://minuum .com

"Text Input on a Smart Watch", A. Komninos and M.D. Dunlop, IEEE Pervasive, 13(4) 2014.

Smart-Watch Prototype

Based on the literature and our previous experience,[12] we hypothesized that efficient text entry is possible with a wearable device such as a smart watch.

Initial Design

We decided to focus on taps for the primary input method, because this is the quickest simple interaction to perform compared to handwriting and tracing.[12,13] Because the accuracy of taps declines rapidly when buttons are small,[14] we decided to design for large keys, rather than try to squeeze overly small keys onto the device and rely heavily on correction.

We segmented the display into seven zones (see Figure 1a). Zones 1 through 6 form large ambiguous keys while the center zone shows the current input text and also acts as a space bar. For word entry the user will type on keys 1 through 6 with the input being disambiguated by the text entry system (running on the connected smartphone). Figure 1b and 1c show an allocation of the alphabet to the six keys: letters A, B, C, and D share button 1; E, F, G, H, I, and J share button 2; and so on.

Figure 1. Our prototype implementation on a Sony SmartWatch 2: (a) concept, (b) design, and (c) implementation.

For our initial design, we defined interaction as follows:

1. A tap on an ambiguous key entered that key number and updated the current word display to reflect the most likely word from the disambiguation engine based on the current key sequence.
2. A first tap on the central zone added a space with subsequent taps rotating through alternative suggestions that match the ambiguous entry.
3. Swipe gestures included backspace (←), word completion (→), toggle capitalization (↑), and numeric punctuation mode (↓).
4. A long press on the center zone entered edit mode to allow movement of the caret, while a long press on the alphabetic keys showed extended characters for that key (for example, à, á, å, ç, and so on for the ABCD key).

The arrows in element (3) above denote the finger swipe direction that constitutes the gesture. In our prototype implementation we did not use all of these gestures, as will be explained later, but the design here shows that oft-used functions of an input method should be quickly accessible to users and a gestural implementation permits this, without taking up additional screen space for dedicated buttons.

Keyboard Layout

Although there has been considerable work on optimized keyboard layouts,[15,16] here we decided to maintain a standard alphabetical layout to aid initial pick-up usability. There are, however, many ways to split the alphabet across multiple keys, with two competing optimization criteria: ambiguity of the layout and movement distance. To reduce ambiguity errors, the best assignment of letters to keys would separate letters that can commonly cause confusion when in the same location in a word—for instance, putting "a" and "e" on the same key would be problematic, because many common words, such as bed and bad, are

only differentiated by this pair. Arranging the splits can help minimize the distance users have to move their 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 minimize the amount of movement of the fingers while typing, but at a massive cost to ambiguity.

We analyzed all possible alphabetic arrangements over six keys using an ambiguity score based on *badgrams*[15] (bigram frequencies for English of how likely a single letter is substitution is to result in a different word, e.g. the most common badgram *AE* includes substitutions such as *bad->bed*) and weighted distance based on English bigram frequency data (e.g. the most common bigram in English is *TH* so the distance from *T* to *H* is weighted higher than, say, *QI*). The least ambiguous keyboard was *abcd efgh ijklm nop qrs tuvwxyz*, whereas the keyboard with least travel for the finger was *abcdefghijklmnopqrstu v w x y z*. Figure 2 shows the distribution of the layouts (with both axes scaled to the range 0…1, where 0 is the worst we found and 1 the best).

To select a layout, we took a weighted average with disambiguation getting more weight than distance—because distances are small, we felt it more important to minimize ambiguity than movement. The best compromise keyboard was selected as abcd efghi jklmn opqrs tuv wxyz, which is highly ranked for disambiguation quality and received the highest distance score on the plateau in Figure 2 (this keyboard is shown as a red dot at the top center of the figure). For reference, the traditional phone keyboard is shown as an orange dot at the top left. This shows that 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 earlier, prediction technology has improved considerably since predictive text began appearing on physical phone keyboards, so we expect higher prediction accuracy in practice.

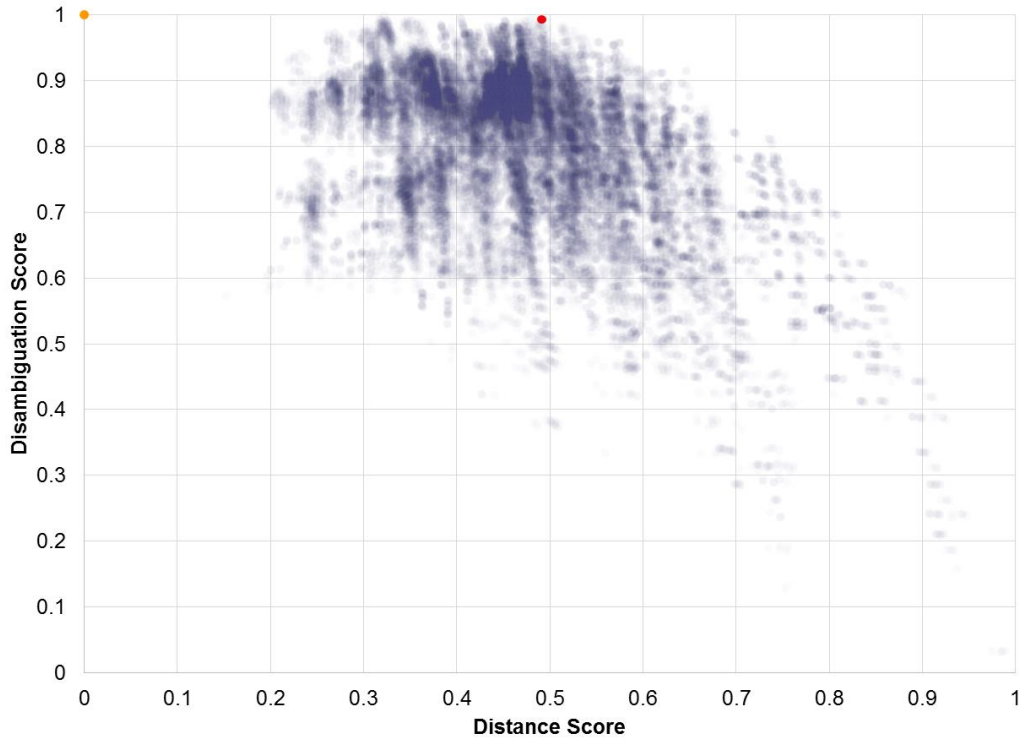

Figure 2. Distribution of keyboard scores. The score of the best compromise keyboard, abcd efghi jklmn opqrs tuv wxyz, *is noted with a red dot at the top center, and the score of the traditional phone keyboard is shown as an orange dot at the top left.*

Initial Implementation

Our implementation was built using OpenAdaptxt[17] running on an Android smartphone paired to a Sony SmartWatch 2. The watch has a 30 × 25 mm screen linked by Bluetooth to the smartphone, where the bulk of processing is done. The OpenAdaptxt framework provided us with a powerful disambiguation engine

that gives contextually based word suggestions, word completion, and next-word suggestions.

For our prototype, we implemented elements 1 and 2 of our interaction design listed earlier, along with the backspace (←) and completion (→) gestures. We also implemented a "symbol" mode, activated by pressing the watch's menu button instead of using the downward swipe gesture. Our test phrase set used the basic Latin alphabet, so we didn't require accented characters for this trial (and thus omitted those from our current implementation). Figure 3 shows a storyboard of entering a short phrase.

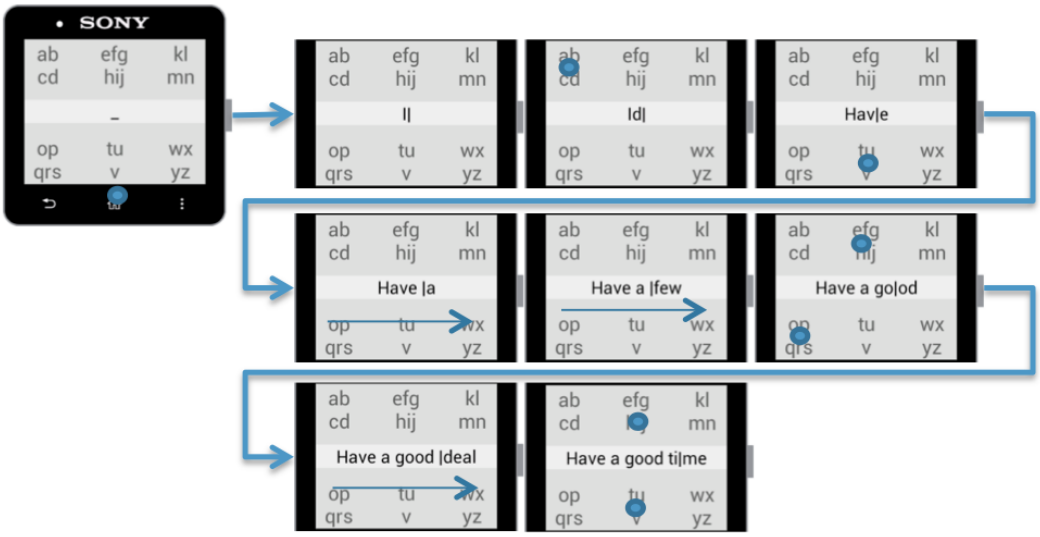

Figure 3. Interaction sequence storyboard to enter "Have a good time." Blue circles represent taps, while on-screen arrows represent ***//correct?//*** *swipe gestures and their direction.*

User Studies

To investigate the usability and performance of our keyboard, we conducted controlled user studies with 20 users (nine female), recruited through mailing lists. The participants were primarily undergraduate and postgraduate students in our University's Computer and Information Sciences Department and were all regular touchscreen smartphone users, but none had prior experience with a smart watch. Participants were given a £10 token for taking part.

Study Design

The study consisted of single-participant sessions, which were composed of four phases:

1. introduce and complete a brief prior-experience form;
2. briefly demonstrate how to enter text using our system;
3. complete formal tasks; and
4. complete final questionnaire and briefly discuss results.

Following the standard text entry approach for evaluation, we asked users to enter a set of short phrases using the smart watch. (Our phrase sets are available at http://personal.cis.strath.ac.uk/mark.dunlop/watchtextentry). We based our formal tasks (phase 3) on the Enron email set.[18] We used the "memorable" phrases from this collection—a set of relatively short phrases that have been shown to be easy to remember in copy tasks. We randomly selected 44 phrases and, to reduce the risk of particular words or phrases excessively affecting results, split them into two sets of 22 phrases. Each set contained two practice phrases followed by four groups of five phrases. Participants were equally and randomly distributed to the two phrase sets. Because the studies were conducted in the UK and we were using a UK-English dictionary, we adjusted the phrase set slightly with minor spelling variants and changed some names to common British names. Because our initial implementation didn't fully

support contractions (such as “won’t”), we also replaced these with full words (“will not”).

To investigate how the length of phrases affects user performance, we sorted the phrases into four groups based on the length of phrases—we focused on phrases of under 160 characters (the traditional SMS limit and higher than the 140 limit on Twitter). The two practice phrases and first group of main phrases were the shortest (average length of 13.0 and 13.1 characters respectively, for example “Are you there?”). In each subsequent group we increased the average phrase length to a maximum average of 52.3 characters for group 5 (for example, “I will follow up with him as soon as the dust settles”). As a result, the five groups had average phrase lengths of 13.0, 13.1, 21.0, 36.2, and 52.3 characters, respectively.

Participants were asked to wear the watch on their non-dominant hand throughout the study and all participants chose to enter text using their dominant hand’s index finger (Figure 4). We asked participants to complete a NASA Task Load Index (TLX) form[19] after completing each group and an exit questionnaire at the end of the session. This form allows participant to self-report on mental, physical and temporal demand of the task, as well as their perceived effort, performance and frustration levels. Completion of the TLX form was followed by a brief discussion about their comments and views.

Figure 4. A participant using our prototype. All participants chose to enter text using their dominant hand’s index finger.

Input Performance

Our prototype also included an automatic logging module. For each input phrase, we captured the time it took participants to type the phrase, the frequency of backspace gestures, the number of word completions, and their computed wpm during the input task (based on the standard five characters per word, including a space). Figure 5 summarizes our results.

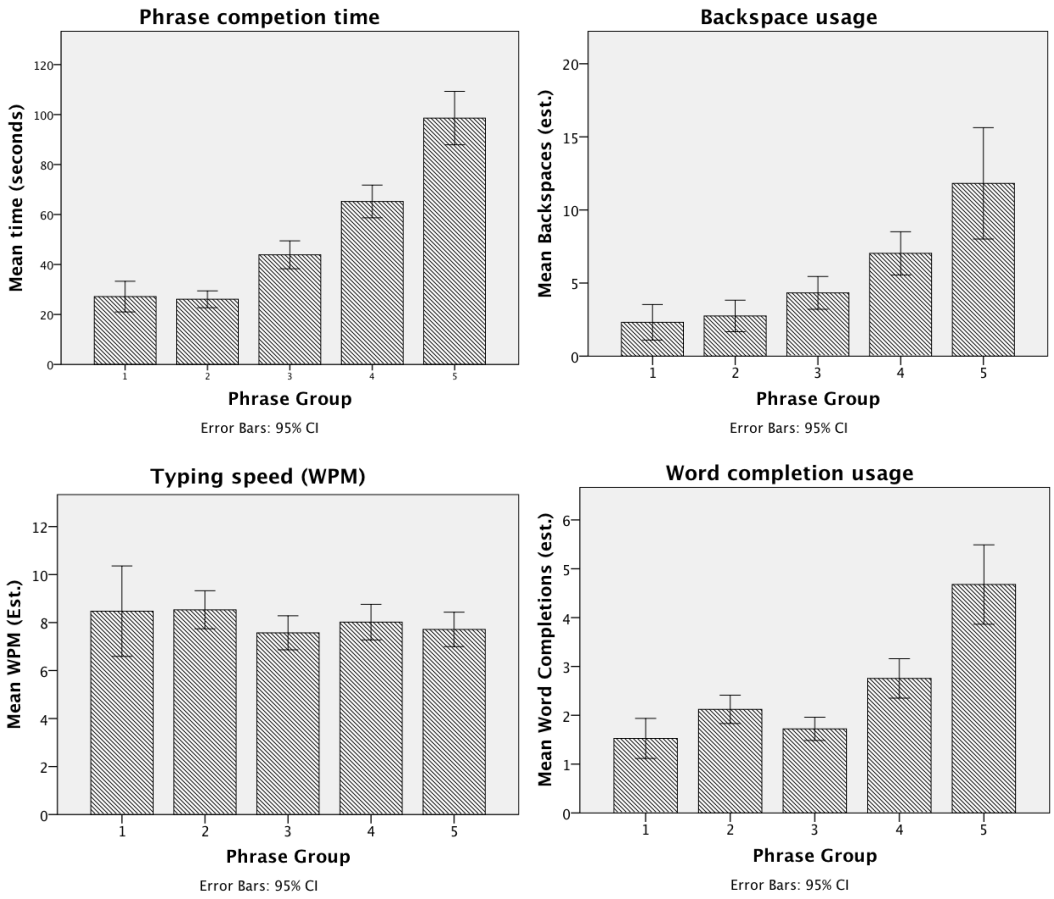

Figure 5. Participants' logged performance metrics as an average per phrase, split by phrase group: (a) phrase completion time, (b) backspace use, (c) typing speed, and (d) word completion use. (The error bars have a 95 percent confidence interval.)

Upon examining the data with a Shapiro-Wilk test, we found it to be not normally distributed in most cases, so in the following report of correlations, we use Spearman's rank correlation coefficient and mean differences using nonparametric tests.

Participants generally took longer to complete phrases as they increased in length. This is confirmed by a statistically significant correlation ($r_s = 0.823$, $p < 0.01$). We also note that the number of backspaces, indicative of typing errors, also shows an increase in line with the increase in task length ($r_s = 0.665$, $p < 0.01$). Although the number of word completions correlates with the size of task length ($r_s = 0.588$, $p < 0.01$), the typing rate achieved by participants was constant during all phrase set tasks ($M_{\text{wpm}} = 8.081$, standard deviation = 2.789), as confirmed by a Friedman (k-independent samples nonparametric) test ($\chi^2 = 4.120$, $p = 0.39$).

Workload Self-Assessment

We were also interested in users' subjective impressions of workload. Figure 6 shows users' self-assessments obtained via the NASA TLX form after each group of phrases. Participants typically ticked one of the gaps in the form, giving a range from 1–20 with the center line being between points 10 and 11. A lower score was good throughout, with 1 being best performance (least load) and 20 being worst performance (highest load). While results were not very low overall, they were on average below the central bar for all dimensions and groups, showing that the watch wasn't particularly demanding to use.

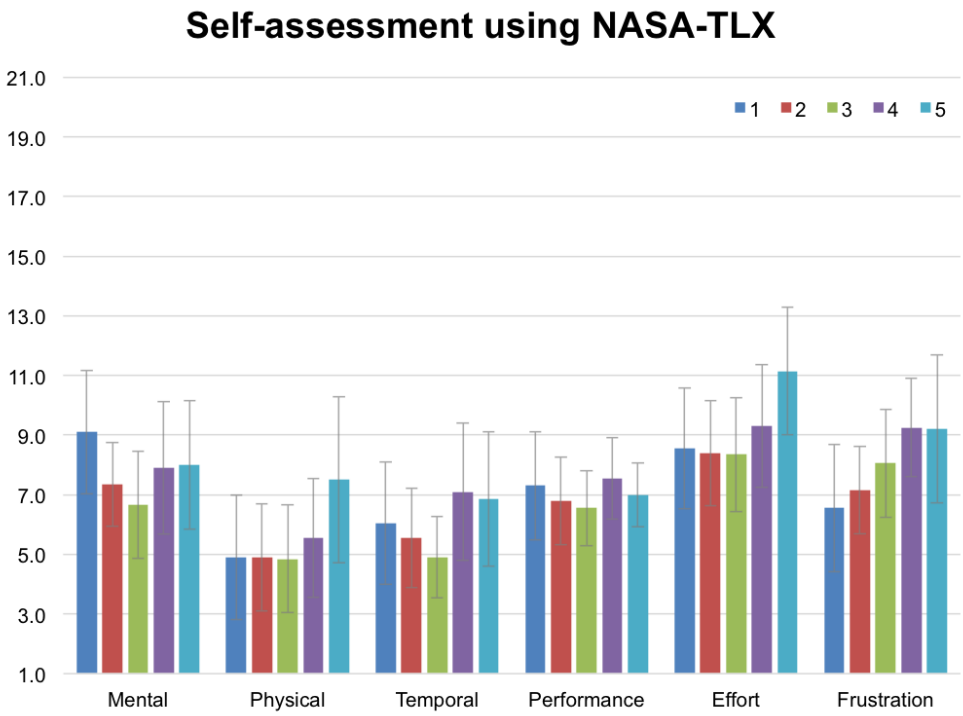

Figure 6. NASA Task Load Index results per phrase group (the error bars have a 95 percent confidence interval).

However, several dimensions showed an increase as participants went through the phrase groups, and none showed an overall drop—as is normal while users are learning a system. This indicates that the increase in length of phrases posed additional load that was not compensated for by increased experience.

Users initially rated their mental load quite high; this fell and then increased toward the end of the session. We found statistically significant differences in the means between phrase groups 2 and 5 (d = 2.970, $p < 0.01$), 3 and 5 (d = 2.558, $p < 0.05$), and finally groups 4 and 5 (d = 2.327, $p < 0.05$), using Wilcoxon signed rank tests.

Users reported that the physical workload was low for the first three phrase groups but higher toward the end of the sessions. We found statistically significant differences in the means between groups 1 and 5 (d = 1.967, $p < 0.05$), 2 and 5 (d = 2.204, $p < 0.05$), and 3 and 5 (d = 2.078, $p < 0.05$) using Wilcoxon signed rank tests, confirming users were finding the physical workload higher in the final group compared to the first three. This reflected some comments from users that they were tiring and over time found the typing position uncomfortable.

Temporal workload measured how much time pressure the participants felt. Following a similar pattern to mental workload, users felt the least temporal pressure in the middle phrase group, and this rose toward the end. A significant difference was shown in the means between groups 3 and 4 (d = 2.086, $p < 0.05$) and groups 3 and 5 (d = 2.207, $p < 0.05$), using Wilcoxon signed rank tests.

Users' rating of their performance in the task didn't vary significantly across the phrase groups (ANOVA with post hoc Bonferroni tests). This is in line with our observation that users tended to focus more on accuracy throughout rather than speed of entry, so they felt no variation in their success in completing the tasks.

The overall effort rating followed the pattern of mental and temporal effort, but it showed a statistically significant increase in effort in pairwise comparisons only between groups 2 and 5 (d = 2.75, $p < 0.05$, ANOVA with post hoc Bonferroni).

Finally, overall user frustration appeared to grow throughout the session on average but with wide variations in reported scores. This was confirmed by statistical tests using the Wilcoxon signed rank test, which revealed a statistically significant difference in the means only between phrase groups 2 and 4 (d = 2.068, $p < 0.05$). Again, this is concerning, because you'd expect frustration to drop with time. This

confirms our view that the increases in phrase and word lengths had a larger impact than learning effects could counter.

Qualitative Feedback

At the end of the session we asked the users several questions about their experience with the watch text entry method. Using 7-point Likert scales, we asked for their overall rating of the keyboard and how likely they would be to use a watch rather than their phone for various tasks. Summarized in Figure 7, this shows that, overall, the watch wasn't particularly easy or hard to use, and participants showed a stronger preference for using the watch for social replies than for typing their own posts. Furthermore, they didn't want to use the watch for longer text such as emails. However, the caveat here is that users were exposed to our solution only for a short time and didn't try it in real-life situations.

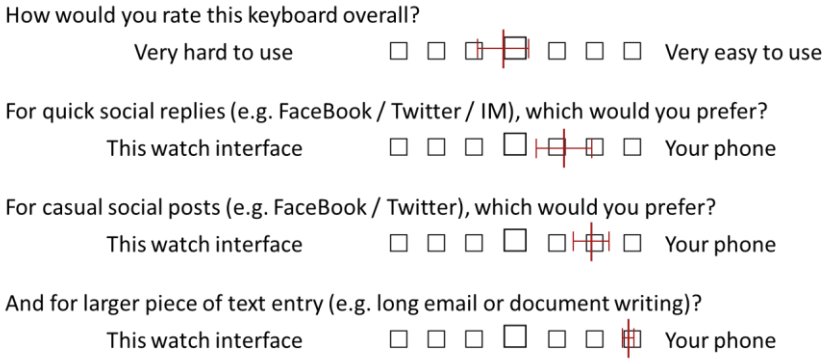

Figure 7. Responses to final questionnaire questions (the error bars have a 95 percent confidence interval).

We also asked users to list the three best and three worst aspects of watch-based text entry. The main strong points were that the prediction quality was high, overall the watch interface was easy to understand and use, the watch interface made good use of the space available, tactile feedback was helpful, and the use of swiping for backspace and completion was helpful and easy to use.

On the negative side, users reported frustration with the watch sometimes being slow to respond and failing to recognize taps. We observed that users didn't clearly understand that the central space bar could also be used to "insert space" and "rotate through suggested words." This was also reflected in users' comments about their confusion regarding how to enter a space (without being given a suggestion) and in having to cycle through the whole suggested word list if they missed the word they wanted. The predictive system also raised problems with editing: if users misspelled or mistapped a word, they often had to backspace the whole word to correct it. As one user explained, "*Getting lost spelling a word in the middle of the word[, it] is sometimes difficult to easily understand which keys have been pressed in the context of your word if another [word] is predicted because each key is mapped to many characters … accidently typing the wrong letter makes me have to delete the whole word*".

Other issues raised were the inability to directly control capitalization or move the cursor. We also had network problems with some users that led to the host application stopping when it failed to save logged data over the network—users commented on these crashes but they weren't frequent enough to impact users' overall feedback. Users rapidly learned the alphabetic layout and were able to quickly tap the right keys. As noted earlier, we observed user confusion with tapping the central area for space and for the next suggestion; this was confounded by the right swipe automatically inserting a space.

We also observed that particular words caused frequent problems. For example, "definitely" wasn't suggested early and the spelling caused difficulty, with any mistake leading to drastically different suggestions. As reported earlier, when users spelled a word incorrectly, they tended to correct with multiple backspaces to the start of the word to start again—reflected in the rise in backspace use through the study groups.

Discussion

Our alphabetic ambiguous-key approach to text entry, based on multiple letters distributed over six keys, worked well with users quickly adapting to the entry process. The only direct impact of the ambiguous-key approach was that users who misspelled a word found it difficult to quickly recover and often backspaced out the whole word. One of the strengths of OpenAdaptxt is its ability to learn the user's individual writing patterns and adjust predictions to their particular patterns. We disabled this feature for this study because it would overlearn the test phrase sets. In reality, a user's regularly used phrases would dominate, reducing the problems with ambiguous entry. Furthermore, integrating a spellchecker would directly address the problem of misspelling words.

Our users liked both prediction and word completion. However, they found our interface design problematic, becoming confused between space and word-complete functions and frustrated when cycling through the suggestion list for rarer words. We now propose removing the overloaded space key as originally proposed in earlier work[4] and instead using the commands presented in Table 1.

Table 1. Various commands and the corresponding gesture.

Command	Gesture
Space (tap on center)	↕
Backspace	←
Word completion	→
Previous word suggestion	↑
Next word suggestion	↓
Symbols and numbers (menu)	⁞
Shift (long press on center)	↥

A few users reported sensitivity problems with the watch. The main problem here appears to be with taps that move slightly during the press—these can erroneously be recorded as swipes, particularly when the user is trying to type carefully (thus pressing hard and slowly). After our initial prototyping, we introduced a time-based threshold for taps and swipes (unfortunately, the Sony SmartWatch API didn't permit distance-based thresholding). In the absence of improved event information from the API, a dynamic thresholding approach could be used to tune the time thresholds to the individual user.

Overall, our users achieved an average of 8.1 wpm, with many phrases being entered at over 10 wpm (in line with novice use of traditional phone predictive text entry).[3] This isn't fast in terms of entry from smartphones, but given the improvements we suggest and the use case of short replies, we see this as positive confirmation that smart-watch text entry speeds can be good enough for short messages. In fact, participants saw value in using the watch to respond to social-network postings without having to retrieve their mobile device. One participant raised the interesting idea that using the watch would allow him to move to a larger "phablet" that could be kept in his bag except for more intense use.

In the longer term, watch APIs will improve to allow more advanced entry methods, such as gesture-based entry, and more dynamic interaction with suggestions and the interface. However, we see strong evidence supporting our ambiguous-key approach, used with simple gestures, and, based on our user study, we've proposed a refined model that applies visual and basic haptic feedback. This method should be suitable for other small touch surfaces, such as fabric-, project-, or skin-conducted surfaces.

Acknowledgments

Our grateful thanks to our experiment participants and KeyPoint Technologies for their help building upon OpenAdaptxt, particularly Krishna Chaitanya Bandatmakuru, Srinivas Chintagunta, Naveen Durga, and Prima Dona. This work was partly supported by the UK EPSRC grant EP/K024647/1.

References

1. M.D. Dunlop and A. Crossan, "Predictive Text Entry Methods for Mobile Phones," *Personal Technologies*, vol. 4, no. 2, 2000, pp.143-143
2. D.L. Grover, M.T. King, and C.A. Kushler, *Reduced Keyboard Disambiguating Computer*, US patent US5818437 to Tegic Communications, Patent and Trademark Office, 1998.
3. C.L. James and K.M. Reischel, "Text Input for Mobile Devices: Comparing Model Prediction to Actual Performance," *Proc. ACM CHI 2001 Human Factors in Computing Systems Conf.* (CHI 01), 2001, pp. 365–371.
4. M.D. Dunlop, "Watch-Top Text-Entry: Can Phone-Style Predictive Text-Entry Work With Only 5 Buttons?," *Proc. 6th Int'l Conf. Human Computer Interaction with Mobile Devices and Services* (MobileHCI 04), LNCS 3160, 2004, pp. 342–346.
5. I.S. Mackenzie et al., "A Comparison of Three Methods of Character Entry on Pen-Based Computers," *Proc. Factors and Ergonomics Soc. 38th Ann. Meeting*, 1994, pp. 330–334.
6. K.M. Lyons et al., *Twiddler Typing: One-Handed Chording Text Entry for Mobile Phones*, tech. report, College of Computing and GVU Center, Georgia Inst. of Technology, 21 Mar. 2003.
7. J.O. Wobbrock, B.A. Myers, and J.A. Kembel, "EdgeWrite: A Stylus-Based Text Entry Method Designed for High Accuracy and Stability of Motion," *Proc. 16th Ann. ACM Symp. User Interface Software and Technology* (UIST 03), 2003, pp. 61–70.
8. T. Bellman and I.S. MacKenzie, "A Probabilistic Character Layout Strategy for Mobile Text Entry," *Proc. Graphics Interface '98*, 1998, pp.168-176.
9. P.-O. Kristensson and S. Zhai, "SHARK2: A Large Vocabulary Shorthand Writing System for Pen-Based Computers," *Proc. ACM CHI 2004 Conf. Human Factors in Computing Systems* (CHI 04), 2004, pp. 43–52.
10. S. Oney et al., "ZoomBoard: A Diminutive Qwerty Soft Keyboard Using Iterative Zooming for Ultra-Small Devices," *Proc. ACM CHI Conf. Human Factors in Computing Systems*, 2013, pp. 2799–2802.
11. F.C.Y. Li et al., "The 1line Keyboard: A QWERTY Layout in a Single Line," *Proc. 24th Ann. ACM Symp. User Interface Software and Technology* (UIST 11), 2011, pp. 461–470.
12. K. Curran, D. Woods, and B.O. Riordan, "Investigating Text Input Methods for Mobile Phones," *Telematics and Informatics*, vol. 23, no. 1, 2006, pp. 1–21.
13. A.L. Smith, "Smartphone Input Method Performance, Satisfaction, Workload, and Preference with Younger and Older Novice Adults," PhD dissertation, Dept. of Psychology, Univ. of Wichita, 2013.
14. P. Parhi, A.K. Karlson, and B.B. Bederson, "Target Size Study for One-Handed Thumb Use on Small Touchscreen Devices," *Proc. 8th Int'l Conf. Human Computer Interaction with Mobile Devices and Services* (MobileHCI 06), 2006, pp. 203–210.
15. M.D. Dunlop and J. Levine, "Multidimensional Pareto Optimization of Touchscreen Keyboards for Speed, Familiarity and Improved Spell Checking," *Proc. ACM SIGCHI Conf. Human Factors in Computing Systems* (CHI 12), 2012, pp. 2669–2678.
16. A. Oulasvirta et al., "Improving Two-Thumb Text Entry on Touchscreen Devices," *Proc. 31st Ann. CHI Conf. Human Factors in Computing Systems* (CHI 13), vol. 38, 2013, pp. 330–334.
17. M.D. Dunlop et al., "OpenAdaptxt: An Open Source Enabling Technology for High Quality Text Entry," *Proc. CHI 2012 Workshop Designing and Evaluating Text Entry Methods*, 2012.
18. K. Vertanen and P.O. Kristensson, "A Versatile Dataset for Text Entry Evaluations Based on Genuine Mobile Emails," *Proc. 13th Int'l Conf. Human Computer Interaction with Mobile Devices and Services* (MobileHCI 11), 2011, pp. 295–298.
19. S.G. Hart and L.E. Staveland, "Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research," *Human Mental Workload*, vol. 1, no. 3, 1988, pp. 139–183.