Komninos, Andreas and Dunlop, Mark (2014) Text input on a smart watch. IEEE Pervasive Computing, 13 (4). pp. 50-58. ISSN 1536-1268 , http://dx.doi.org/10.1109/MPRV.2014.77

Using a smart-watch as a input device for text

Andreas Komninos, Mark D Dunlop
University of Strathclyde

A. Komninos and M.D. Dunlop, Using a smart-watch as an input device for text, IEEE Pervasive, in press 2014.

Abstract

Smart-watches 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 lab evaluation of an implemented prototype using the OpenAdaptxt engine on a Sony SmartWatch 2. While raising some problems, the feedback from our participants indicates that reasonable quality and speed is achievable on a smart-watch and encourages our future work.

Introduction

Text entry is a key component of many smartphone applications. 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 cannot reply on the same device. As part of our on-going work, we outline a text entry approach for smartwatches, describe our initial prototype (Figure 1) and discuss the outcomes of our lab-based evaluation of the prototype.

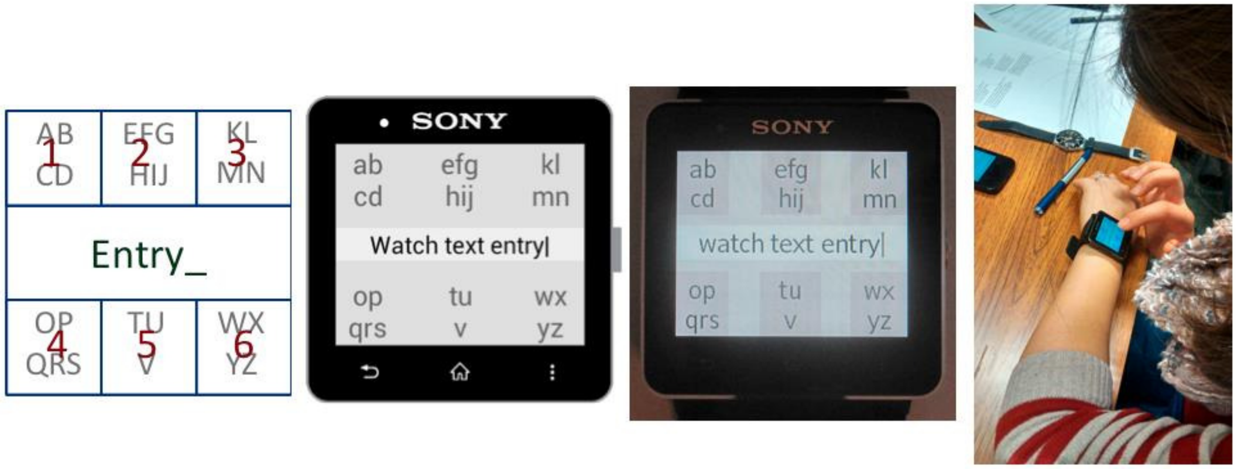

Figure 1: Evolution of our prototype implementation on a Sony SmartWatch 2 (from left: concept, design, implementation, use)

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. Predictive technologies (e.g. [4, 7]) 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 [9]. We have previously investigated using this approach with a reduced number of keys for text entry on watches [3], but implemented on a touchscreen handheld at the time. While in theory 12-key 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). 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 that greatly reduce the impact of ambiguity.

Alternative approaches for input on small devices include handwriting [13], fast but difficult to learn chord keyboards [12] and specialised alphabets [19]. 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 [1]. Finally, gesture word-based input techniques [10] have shown great benefits for mobile phones and can provide a fast and more relaxing input method than continuously tapping very small on-screen buttons.

There is currently very little work focusing on text entry for smart watches. One such method is Zoomboard [14], where a full QWERTY keyboard is shrunk to fill the smart watch screen. 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, as additional input is required for zooming. Further, 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 smartwatch entry using a keyboard that compresses a QWERTY keyboard layout to one line and incorporates word suggestions. Its 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 Li et al. [11] who have shown this keyboard layout to work efficiently on tablet size devices.

Based on literature and our previous experience [2], we hypothesise that efficient text entry is possible with a wearable device such as a smartwatch. The remainder of the paper discusses our examination of this hypothesis, starting with how we derived our keyboard layout, followed by the interaction design and the results of a lab-based experiment. We conclude with suggestions for improvements in our design.

Initial Design

We decided to focus on taps for the prime input method, as this is the quickest simple interaction to perform, compared to handwriting and tracing [2, 17]. As the accuracy of taps declines very rapidly when buttons are small [16] we decided to design for large keys, rather than trying to squeeze overly small keys onto the device and rely heavily on correction.

We segmented the display into seven zones (Figure 1 left). Zones 1…6 form large ambiguous keys while the centre zone shows the current input text and also acts as a space bar. For word entry the user will type on keys 1…6 with the input being disambiguated by the text entry system (running on the connected smartphone). In Figure 1 we have shown an allocation of the alphabet to the six keys with letters ABCD sharing button 1, EFGHIJ sharing button 2 etc.

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 were defined as follows:

← Backspace

→ Word completion

↑ Toggle capitalisation

↓ Numeric punctuation mode

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, á, å, ç etc. on the ABCD key).

Keyboard Layout

While there has been considerable work on optimised keyboard layouts (e.g. [5, 15]), 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 optimisation 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 (e.g. putting a and e on the same key would be problematic as many common words, such as bed and bad, 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 all possible alphabetic arrangements over six keys 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 (this figure shows 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 – as distances are small, we felt it more important to minimise ambiguity than movement. The best compromise keyboard was selected as abcd efghi jklmn opqrs tuv wxyz which is very highly ranked for disambiguation quality and the highest distance score keyboard on the plateau in Figure 2 (this keyboard is shown in red (top centre)). For reference the traditional phone keyboard is shown in orange in Figure 2 (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 higher prediction accuracy in practice.

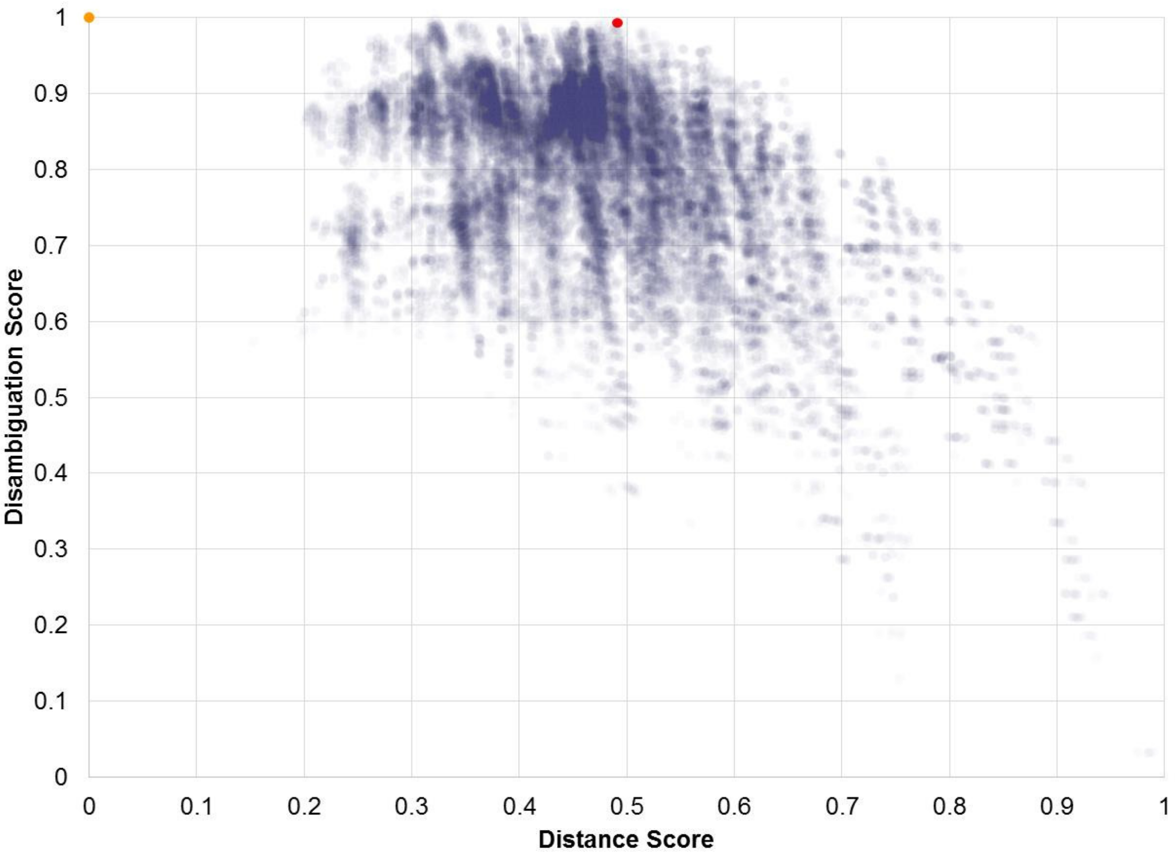

Figure 2: Distribution of keyboard scores

Initial Implementation

Our implementation was built using OpenAdaptxt running on an Android smartphone paired to a Sony SmartWatch 2. The watch has a 30 x 25 mm screen linked by Bluetooth to the smartphone, where the bulk of processing is done. The OpenAdaptxt [6] 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 design from above along with the backspace ← and completion → gestures. We also implemented a "symbol" mode, activated by pressing on the menu button of the watch, instead of a downwards swipe gesture. As our test phrase set was basic Latin alphabet we did not require accented characters for this trial so omitted those from our current implementation. Figure 3 shows a storyboard of entering a short phrase.

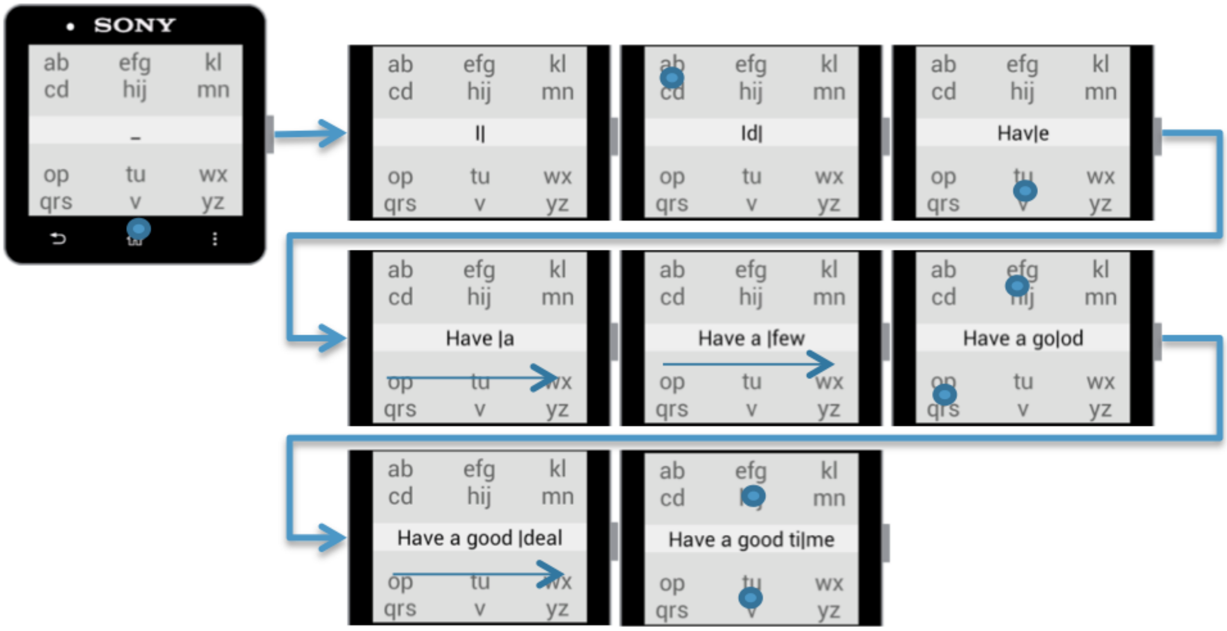

Figure 3: Interaction sequence storyboard to enter “Have a good time”. Blue circles present taps, while on-screen arrows present swipe gestures and their direction.

User Studies

Study design

To investigate the usability and performance of our keyboard we conducted controlled user studies with twenty users (9 female), recruited through mailing lists. The participants were primarily undergraduate and postgraduate students in Computer and Information Sciences and were all regular touch-screen smart-phone users, but none had prior experience of a smart-watch. Participants were given a £10 token for taking part. The sessions were composed of four phases:

1. Introduction and brief prior-experience form completion;
2. Brief demonstration of how to enter text using our system;
3. Formal tasks;
4. Completion of final questionnaire and brief discussion of results.

Following the standard text entry approach to evaluation, we asked users to enter a set of short phrases using the smart-watch. 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 composed of two practice phrases, followed by four groups of five phrases. Participants were equally distributed to each phrase set in a random manner. As 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 changing some names to common British names. As our initial implementation did not fully support contractions (e.g. won't) we also replaced these with full words (e.g. will not)[1].

[1] Our phrase sets are available at personal.cis.strath.ac.uk/mark.dunlop/watchtextentry/

In order to investigate how the length of phrases affects user performance, we sorted the phrases into four groups based on the length of phrases - we focussed 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 13.1 characters, e.g. "Are you there?"). In each subsequent group we increased the average phrase length to a maximum average of 52.3 characters for group 5 (e.g. "I will follow up with him as soon as the dust settles"). As a result, the rest of the groups (1-5) 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 1 right). We asked participants to complete a NASA TLX form [8] after completing each group and an exit questionnaire at the end of the session, followed by a brief discussion about their comments and views.

Input performance

Our prototype also included an automatic logging module. For each input phrase we captured the time it took participants to complete it, the frequency of backspace gestures, the number of word completions and finally their computed words-per-minute during the input task (based on the standard 5 characters per word including space). The outcome is summarised in Figure 4.

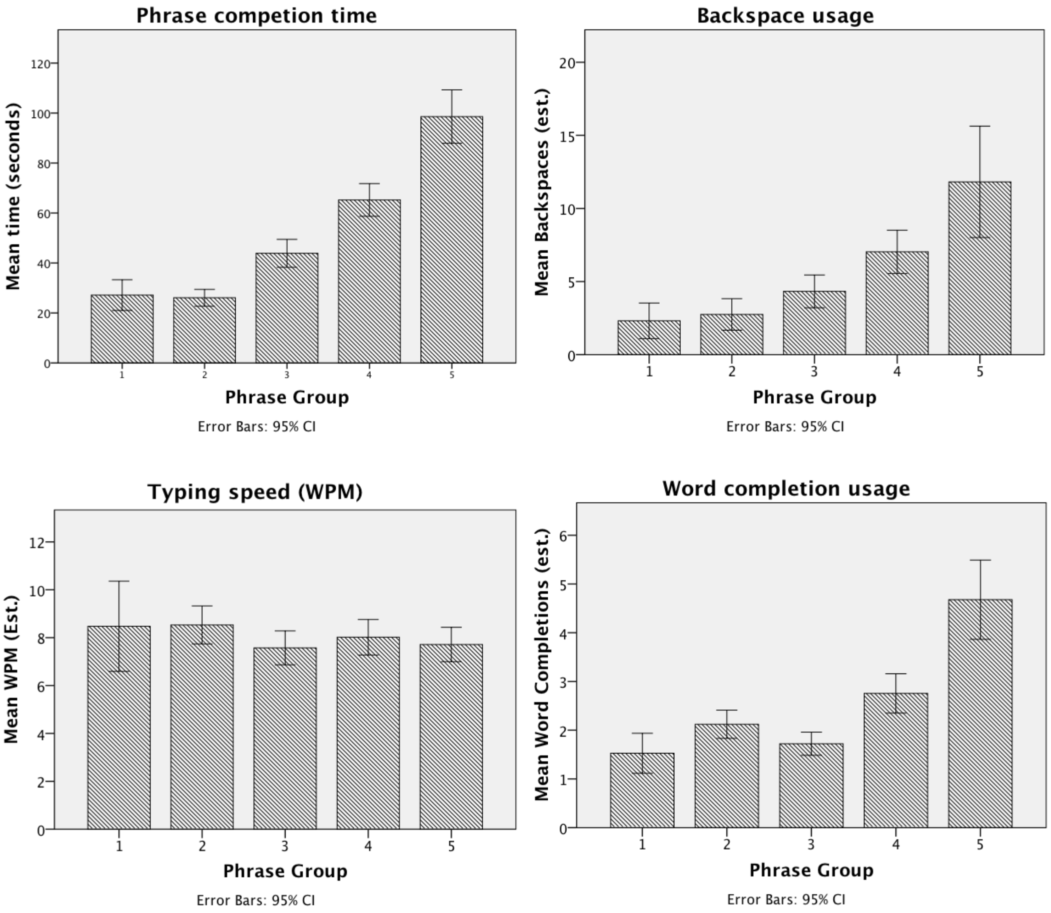

Figure 4: Participants' logged performance metrics as an average per phrase, split by phrase group

Upon examining the data with a Shapiro-Wilk test, we found it to be not normally distributed in most cases therefore all correlations below are reported using Spearman's rho and mean differences using non-parametric tests.

Participants generally took longer to complete phrases, as these 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), we observe that the typing rate achieved by participants was constant during all phrase set tasks (M_{wpm}=8.081, SD=2.789), as confirmed by a Friedman (k-independent samples non-parametric) test (χ^2=4.120, p=0.39).

Workload self-assessment

We were also interested in users' subjective impressions of workload. After each group of phrases the users were asked to complete a NASA TLX form. The results are summarised in Figure 5. Participants typically ticked one of the gaps in the form giving a range from 1-20 with the centre 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 was not particularly demanding to use. However, several dimensions showed an increase as participants went through the phrase groups and none 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.

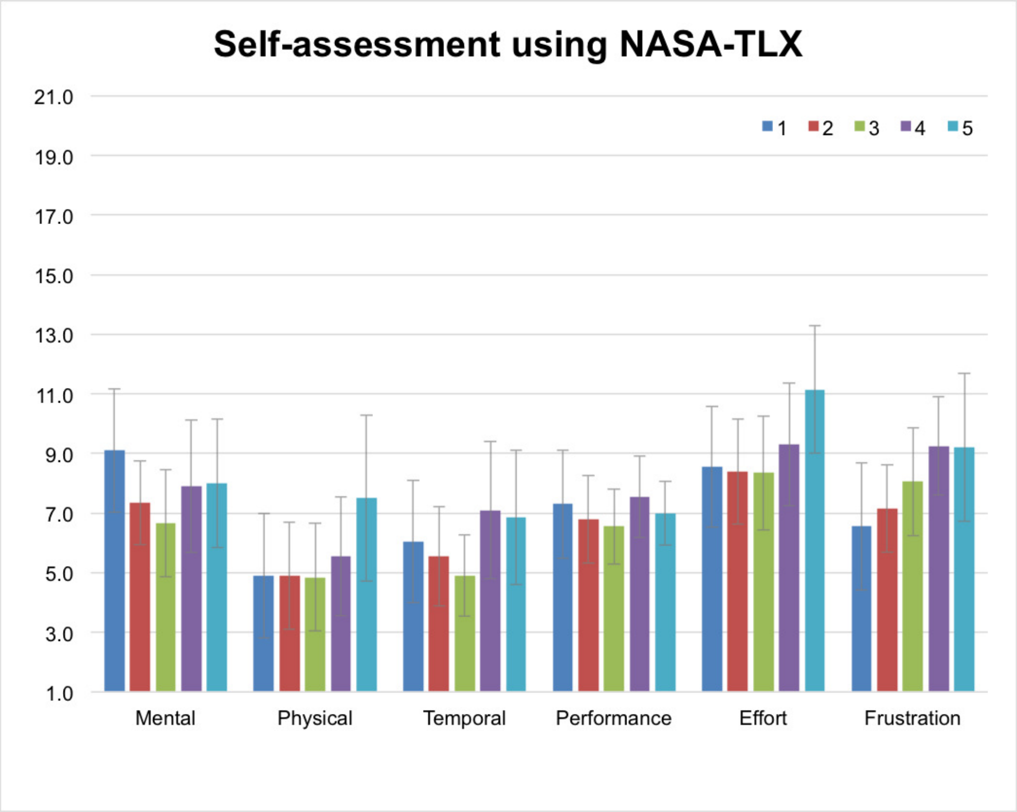

Figure 5: NASA Task Load Index results per phrase group (error bars 95% C.I.)

Users initially rated their mental load quite high, this fell and then increased towards 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.

Physical workload was low for the first three phrase groups but rose towards the end of the sessions. We found statistically significant difference 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 measures how much time pressure the participants felt under. Following a similar pattern to mental workload, users felt under least temporal pressure in the middle phrase group and this rose towards 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 did not 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 only showed a statistically significant increase in effort to be present in pairwise comparisons 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 that confirmed a statistically significant difference in the means only between phrase groups 2 and 4 (d=2.068, p<0.05, Wilcoxon signed rank). Again this is concerning, as one would expect frustration to drop with time and confirms our view that the phrase and word lengths increasing 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 than their phone for various tasks. Summarised in Figure 6, this shows that overall the watch was not particularly easy nor hard to use and that participants showed a stronger preference for using the watch for social replies than posts themselves and a strong preference against using the watch for longer text such as emails, although the caveat here is that users were only exposed to our solution for a short time and have not actually tried it in real-life situations).

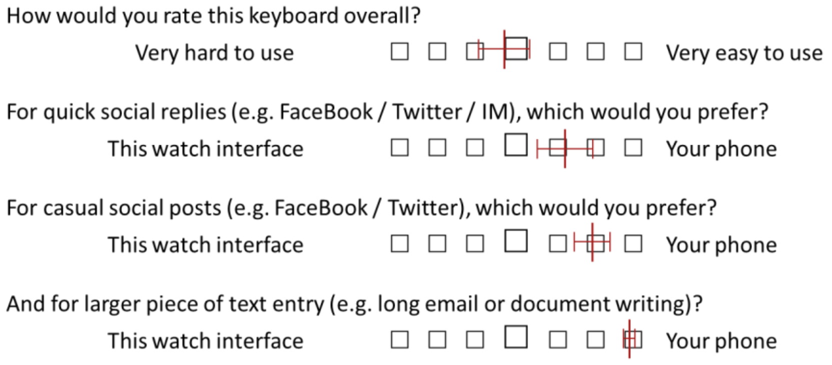

Figure 6: Responses to final questionnaire questions (error bars 95% C.I.)

We also asked users to give the three best and three worst aspects of the watch text entry. The main good points were that the prediction quality was high, that overall it was easy to understand and use, that it made good use of the space available, tactile feedback on touch was helpful, and the use of swipe 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 recognise taps. In observation we noted that users did not clearly understand the overloading of the central space bar as “insert space” and “rotate round suggested words” and this was also reflected in comments with users commenting on confusion of how to enter a space without the suggestion and in having to cycle round 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 in order to correct. 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 capitalisation and no ability to move the cursor. Finally we had some network problems with some users that led to the host application stopping as it failed to save logged data over the network – users commented on these crashes but they were not, we believe, frequent enough to impact on their overall feedback.

In observations we noted that users rapidly learned the alphabetic layout with users quickly tapping on the right keys. As noted above, we observed a confusion with tapping the central area for space and for next-suggestion and that this was confounded by the right swipe automatically inserting a space. We also observed particular words caused frequent problems, for example definitely was not suggested early and the spelling caused difficulty, with any mistake leading to drastically different suggestions. As reported above, 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 and Future Directions

We took an alphabetic ambiguous key approach to text entry based on multiple letters distributed over six keys. This 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. For this study we disabled this feature as 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 round the suggestion list for rarer words. We now propose to remove the overloaded space key as originally proposed in Dunlop’s early work [3] and use the following commands:

↕	space (tap on centre)
←	Backspace
→	Word completion
↑	Previous word suggestion
↓	Next word suggestion
⁞	Symbols and numbers (menu)
↨	Shift (long-press on centre)

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 did not 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.

Conclusions

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 [9]). This is not fast in terms of entry from smart-phones but given the improvements we suggest and the use-case of short replies, we see this is positive confirmation that smart-watch text entry speeds can be good enough for short messages. In the longer term the watch APIs will improve to allow investigation of more advanced entry methods, such as gesture based entry and more dynamic interaction with the suggestions and interface. However we see strong evidence that our ambiguous key approach together with simple gestures is usable and we have proposed a refined model based on our user study experiences.

In conclusion: Our study has shown that text entry is possible on a smart-watch but is better suited to short input tasks. Furthermore, participants saw the value in using the watch to directly respond to social network postings without having to always 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.

The input method we developed was based around segmenting a touch surface into seven relatively large areas combined with simple gestures that were location independent. We paired this with some visual and basic haptic feedback and believe that this method would also be suitable for entry on many small touch surfaces such as fabric, project or skin conducted surfaces.

Acknowledgements

Our grateful thanks to our experiment participants and KeyPoint Technologies for their help building upon OpenAdaptxt, particularly K. C. Bandatmakuru, S. Chintagunta, N. Durga and P. Dona. This work was partly supported by the UK EPSRC grant EP/K024647/1

References

[1] Bellman, T., and MacKenzie, I. S. A probabilistic character layout strategy for mobile text entry. In Proc. Graphics Interface '98, Canadian Information Processing Society (1998).

[2] Curran, K., Woods, D., & Riordan, B. O. Investigating text input methods for mobile phones. Telematics and Informatics, 23(1), 1-21 Elsevier (2006).

[3] Dunlop, M. D. Watch-Top Text-Entry: Can Phone-Style Predictive Text-Entry Work With Only 5 Buttons? In Proc. MobileHCI 04, Springer Lecture Notes in Computer Science (2004).

[4] Dunlop, M. D. and Crossan, A. Predictive text entry methods for mobile phones. Personal Technologies, 4:2 (2000).

[5] Dunlop, M. D. and Levine, J. Multidimensional Pareto optimization of touchscreen keyboards for speed, familiarity and improved spell checking. In Proc. SIGCHI Conference on Human Factors in Computing Systems, ACM (2012).

[6] Dunlop, M. D., Montaparti, S., Dona, P., Durga, N. and Meo, R. D. OpenAdaptxt: An Open Source Enabling Technology for High Quality Text Entry. In Proc. CHI 2012 Workshop on Designing and Evaluating Text Entry Methods (2012).

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

[8] Hart, S. G. and Staveland, L. E. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. Human mental workload, 1:3 (1988), 139-183.

[9] James, C. L. and Reischel, K. M. Text input for mobile devices: comparing model prediction to actual performance. In Proc. SIGCHI Conference on Human Factors in Computing Systems, ACM (2001).

[10] Kristensson, P.-O. and Zhai, S. SHARK2: a large vocabulary shorthand writing system for pen-based computers. Proc. SIGCHI Conference on Human Factors in Computing Systems ACM (2004).

[11] Li, F. C. Y., Guy, R. T., Yatani, K. and Truong, K. N. The 1line keyboard: a QWERTY layout in a single line. In Proc. 24th annual ACM symposium on User interface software and technology (pp. 461-470). ACM (2011).

[12] 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).

[13] 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 and Ergonomics Society 38th Annual Meeting (1994).

[14] Oney, S., Harrison, C., Ogan, A., and Wiese, J. ZoomBoard: A diminutive QWERTY soft keyboard using iterative zooming for ultra-small devices. In Proc. SIGCHI Conference on Human Factors in Computing Systems (pp. 2799-2802). ACM (2013).

[15] 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. SIGCHI Conference on Human Factors in Computing Systems, ACM (2013).

[16] Parhi, P., Karlson, A. K. and Bederson, B. B. Target size study for one-handed thumb use on small touchscreen devices. In Proc. MobileHCI, ACM (2006).

[17] Smith, A. L. Smartphone input method performance, satisfaction, workload, and preference with younger and older novice adults, Ph.D. Thesis, University of Wichita, (2013).

[18] Vertanen, K. and Kristensson, P. O. A versatile dataset for text entry evaluations based on genuine mobile emails. In Proc. MobileHCI, ACM (2011).

[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. 16th annual ACM symposium on User interface software and technology, ACM (2003).