QWERTH: An Optimized Semi-ambiguous Keyboard Design

Mark D Dunlop
Computer and Info. Sciences
University of Strathclyde
Richmond Street
Glasgow G1 1XH. Scotland
Mark.Dunlop@cis.strath.ac.uk

Prima Dona
KeyPoint Technologies Pvt. Ltd.
9Q1A Cyber Towers, Hitec City
Hyderabad, India
pdona@keypoint-tech.com

Naveen Durga
KeyPoint Technologies Pvt. Ltd.
9Q1A Cyber Towers, Hitec City
Hyderabad, India
ndurga@keypoint-tech.com

Varun Medapuram
KeyPoint Technologies Pvt. Ltd.
9Q1A Cyber Towers, Hitec City
Hyderabad, India
vmedapuram@keypoint-tech.com

Sunil Motaparti
KeyPoint Technologies Pvt. Ltd.
9Q1A Cyber Towers, Hitec City
Hyderabad, India
smotaparti@keypoint-tech.com

Figure 1. QWERTH layout

MobileHCI'12, September 21–24, 2012, San Francisco, CA, USA.
ACM 978-1-4503-1443-5/12/09.

Abstract

Mobile phone keyboards can be categorized as unambiguous (one key per letter) or ambiguous (multiple letters per key). In this poster we present QWERTH: a semi-ambiguous keyboard for English in which we have grouped keys together that are less likely to cause prediction problems, while keeping some other keys unique. This has allowed us to increase keys to 1/5 screen width instead of 1/10 while maintaining a near-QWERTY layout. A prototype keyboard built on the OpenAdaptxt text entry engine, results from initial user studies and future study plans are discussed.

Author Keywords

Text entry; keyboard layouts; semi-ambiguous entry.

ACM Classification Keywords

H.5.2 User Interfaces: Input devices and strategies

General Terms

Design; Human Factors.

Introduction

Mobile phone keyboards have always been a compromise between size of keyboard and accuracy of text entry.

With physical keypad phones most devices used the ISO 9995-8 standard 12-key phone pad with either multitap or T9™ like disambiguation (e.g. [7] and [2]). For example using multitap *hood* would be entered as *44666-6663* to cycle round each key and – to wait for the timeout between subsequent keys on the same button. With T9, it would typically be entered as *4663*** with *4663* giving *good*, * changing to next suggestion of *home* then a further * to see *hood*. While, Gong and Tarasewich [6] investigated the best layout of ambiguous keyboards for different sizes, the standard 12-key keypad is still the only widely used ambiguous layout. Alternatively, and mostly popularized by the BlackBerry range, a mini QWERTY keyboard can be provided – however these tend to have both very small keys and increase the overall dimensions of the device.

Touch screen phones give text input designers more freedom – however the two classic input methods still dominate: touch-screen QWERTY or touch-screen 12-key. In the early days of mobile touch screen interaction Parhi et al. [15] recommended that touch screen buttons be 9.6mm in size for thumb-based entry (common, when say, walking). Typical touch-screen phones (e.g. Apple iPhone or HTC Desire) have buttons that are around 5mm in size. This, together with lack of tactile feedback (e.g. [8]), results in increased error rates for touch screen phones (e.g. [10]).

In this poster we present a new keyboard layout for touch screen phones that attempts to combine some keys in order to create bigger keys overall, while maintaining the overall accuracy of entry and reducing prediction problems associated with T9 like entry.

Design criteria

A standard QWERTY keyboard has three rows of letters with 10 keys on the top row, 9 on the central and 7 on the bottom row. On touch screen mobiles, the top row is used exclusively for letters giving one tenth of the screen width to each key. This results in keys of roughly 5mm across which are, basically, too small to be hit accurately. An ideal keyboard should, we argue, be usable by thumbs and by fingers – as such should follow Parhi et al's recommendations [15] and ideally have keys of approximately 10mm size.

Fitts's law [5] states that the nearer and bigger a target is, the quicker it is to tap. This implication and the modeling behind Fitts's law has been used widely to inspire different keyboard layouts (e.g. The Opti [14], Metropolis [16] and matrix [11] keyboards) by attempting to put keys together that frequently follow each other. These are known as *bigrams* (or digraphs), with the most common letter-bigrams in English being *TH* and *HE.* Ideally a keyboard should have common bigrams close together to minimize finger travel time and, thus, increase typing speed.

While T9-like prediction works well overall [6], there are some key combinations that are particularly difficult (e.g. *he* and *if* share the same keystrokes, as do *good* and *home*). Gong and Tarasewich [6] attempted to reduce this ambiguity be rearrangement of the keys on different sized ambiguous keyboards. Dunlop and

Levine [3] introduced the concept of *badgrams*, to address a similar problem with auto-correction on touchscreen QWERTY keyboards. Modern mobile phone spell correction algorithms attempt to correct both users' spelling mistakes and their typing errors – most commonly hitting adjacent keys to the intended ones (e.g. [9]). Dunlop and Levine used letter pair statics for letters that when replaced by each other are likely to result in a valid word, thus making life difficult for the correction algorithms (e.g. on a QWERTY touch screen, having *I* and *O* together can led to many uncorrected typing errors such as *fir* instead of *for*). The most common badgrams in English were given as *AE, AO, EO,* and *ST*. Ideally a keyboard should have common badgrams separated to improve spell checking quality.

Finally, the QWERTY keyboard has very widespread use and designing drastically different keyboards can be a major inhibitor to adoption (e.g. the early failure of the Dvorak keyboard). Bi, Smith and Zhai [1] attempted to overcome peoples' reluctance to new layouts by only allowing keys to move one position from their standard-QWERTY position. Ideally a keyboard should be very close to the standard QWERTY layout.

Our Design

Our first design criteria was to increase button sizes to approximately those recommended by Parhi et al. [15] on standard smartphone portrait screens. In contrast to Li et al's one-line approach for tablets [12] in which they focused on reducing the screen-area of the keyboard, we aimed to keep the same screen-area and thus targeted 5 keys per row to give a 15 key layout for the main alphabetic section of the keyboard.

We ranked the bigrams and badgrams from [3] (excluding bigrams that included the space key). For each set of data we calculated the percentage of best score, for example the bigram *TH* is the most common letter-pair bigram with 612050 occurrences in the analyzed corpus. We gave this a score of 100% and scaled others according, e.g. the second most common bigram is *HE* with 588748 occurrences and got a score of 96%. A similar calculation was done for badgrams where *AE* is the most common badgram with 1 227 442

pair	**bigram**	**badgram**	**bi - bad**
TH	100	0	100
HE	96	0	96
RE	56	0	56
ON	48	0	48
ER	66	20	46
ND	38	0	38
…			
RO	23	0	23
NE	22	0	22
RI	22	0	22
LA	20	0	20
LI	20	0	20
IC	19	0	19
RA	18	0	18
MA	17	0	17
…			
GE	13	0	13
TR	13	0	13
US	13	0	13
…			

Table 1: Ranked letter-pair scores

Figure 2: Final Android keyboard with suggestion bar

occurrences in the corpus, followed by *AO* (1214949 / 99%). Table 1 mainly shows pairs of letters where a substitution did not lead to another word in the corpus (score 0), with the exception of *ER* which has a score of 20% based on 250581 occurrences.

We then calculated a "pairability" score by subtracting the badgram score from the bigram score. This resulted in, for example, a final score of 100 for *TH* which was the most common non-space bigram but got a score of 0 on the badgram table. Table 1 shows a sample from the top of this table. The aim of the *bi-bad* calculation was to pair letters that commonly occur after each other but which would not cause disambiguation difficulty by having common words separated only by an ambiguous key.

Finally we worked down this table combining letter pairs to form a 5x3 semi-ambiguous layout. We tried to keep keys as close as possible to their QWERTY position (at least one key for paired keys) and stopped pairing when we reached a 5x3 layout (on the top row we also paired QW as this was fairly safe and reduced the long row to only 5 keys). Sample selected pairs are shown in bold in table 1 with our final keyboard layout shown in figure 1.

Prototype keyboard

We developed the prototype keyboard using a custom version of OpenAdaptxt™ text entry framework for Android™ [1]. This open-source framework supports powerful text prediction and was used to give high quality prediction and spell correction in line with commercial touch-screen text entry systems [4]. The final keyboard design, with suggestion bar, is shown in figure 2.

[1] http://sourceforge.net/projects/openadaptxt/

Initial user studies

We conducted an initial one-session user study with 8 users to get initial impressions of our keyboard. Users were asked to type 14 short phrases taken from MacKenzie and Soukoreff's standard phrase set [13] standard plus two more general composition tasks of five minutes each. In total studies took just under 30 minutes. Users were aged between 23 and 40, all were fluent English speakers with University level education.

Although the initial study was too short and had too few users to give valid quantitative data on timing and learning rates, we did, however, get qualitative impression data from the users. On a 7 point scale users were asked how easy it was to find keys on the new keyboard. Results are shown in figure 3 and clearly show users found it quite difficult to find keys on first use.

Figure 3: How easy is it to find keys?

We also asked users to rate their overall impression of the usability of the system, again on a 7-point scale. Figure 4 shows more positive responses here.

Figure 3: User testing QWERTH

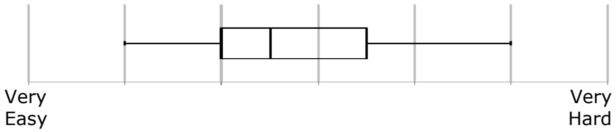

Figure 4: Overall how easy is it to use?

When asked for positive statements about the keyboard users focused on (a) the larger key sizes making it easier to type and (b) the high quality of word suggestions. Negative comments focused on the initial difficulty of finding keys that had moved on the keyboard. Together these outcomes support our design criteria: larger buttons and high quality suggestions are appreciated, but moving away from QWERTY is problematic.

Future work

All new keyboard layouts take time for users to learn. We are currently planning semi-longitudinal studies in which users will use our keyboard over a 2 week period with periodic formal speed and accuracy evaluation over the study.

The keyboard shown in figure 2 included movement of letters to optimize the keyboard. Our initial user studies raised concerns over the initial learning curve because of the moved keys. Gong and Tarasewich [6] investigated both freely optimized and alphabetically constrained keyboards. We are now planning to additionally develop a QWERTY-constrained version of our semi-ambiguous keyboard. Our modeling shows this will perform less well with word suggestions and error corrections but should be easier to use initially. Laboratory studies of this keyboard are also planned.

We are also working on building alternative keyboards for other languages. As both bigram and badgram tables from Dunlop and Levine [3] are based on analysis of English language corpora, we need to develop equivalent analysis for each additional language – however, once a corpus is built the analysis is straightforward.

Conclusions

In this poster we have presented a new keyboard layout that is designed to have large keys that are more suitable for larger fingers and thumbs on touch screen phones. The design is based around making “safe” pairings of letters onto ambiguous keys that are commonly used one after each other, e.g. *TH*, while minimizing the impact of the ambiguous keys on the accuracy of text entry through not-pariing keys that are likely to cause disambiguation difficulty. Initial results confirmed our design critieria with users appreciating the larger buttons and accurate predictions but raised concerns about the movement of keys away from the standard QWERTY layout.

Acknowledgements

We thank users for the time to help us conduct the user studies and their valuable feedback.

References

1. Bi, X., Smith, B. A. and Zhai, S. Quasi-qwerty soft keyboard optimization. In *Proc. CHI 2010*, ACM Press (2010), 283-286.
2. Dunlop, M. D. and Crossan, A. Predictive text entry methods for mobile phones. *Personal Technologies*, 4, 2 (2000).
3. Dunlop, M. D. and Levine, J. Multidimensional Pareto optimization of touchscreen keyboards for speed, familiarity and improved spell checking. In *Proc. CHI 2012*, ACM (2012).
4. 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).
5. Fitts, P. M. The information capacity of the human motor system in controlling the amplitude of movement. *J. Experimental Psychology*, 47, 6 (1954), 381-391.
6. Gong, J. and Tarasewich, P. Alphabetically constrained keypad designs for text entry on mobile devices. In *Proc. CHI '05*, ACM Press (2005), 211-220.
7. Grover, D. L., King, M. T. and Kushler, C. A. *Reduced keyboard disambiguating computer* Tegic Communications, Inc., Patent US5818437 (1998).
8. Hoggan, E., Brewster, S. A. and Johnston, J. Investigating the effectiveness of tactile feedback for mobile touchscreens. In *Proc. CHI '08*, ACM Press (2008), 1573-1582.
9. Kwon, S., Lee, D. and Chung, M. K. Effect of key size and activation area on the performance of a regional error correction method in a touch-screen QWERTY keyboard. *International Journal of Industrial Ergonomics*, 39, 5 (2009), 888-893.
10. Lee, S. and Zhai, S. The performance of touch screen soft buttons. In *Proc. CHI 2009*, ACM Press (2009), 309-318.
11. Lewis, J. R., Kennedy, P. J. and LaLomia, M. J. Development of a Digram-Based Typing Key Layout for Single-Finger/Stylus Input. In *Proc. HFES* (1999).
12. 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. UIST 2011*, ACM (2011), 461-470.
13. MacKenzie, I. S. and Soukoreff, R. W. Phrase sets for evaluating text entry techniques. In *Proc. CHI 2003*, ACM Press (2003), 754-755.
14. MacKenzie, I. S. and Zhang, S. X. The design and evaluation of a high-performance soft keyboard. In *Proc. CHI'99*, ACM Press (1999), 25-31.
15. Parhi, P., Karlson, A. K. and Bederson, B. B. Target size study for one-handed thumb use on small touchscreen devices. In *Proc. MobileHCI 2006*, ACM Press (2006), 203-210.
16. Zhai, S., Hunter, M. and Smith, B. A. The metropolis keyboard - an exploration of quantitative techniques for virtual keyboard design. In *Proc. UIST'00*, ACM Press (2000), 119-128.