Optimal-T9: An Optimized T9-like Keyboard for Small Touchscreen Devices

Ryan Qin[1]*, **Suwen Zhu**[2], **Yu-Hao Lin**[2], **Yu-Jung Ko**[2], **Xiaojun Bi**[2]

[1]Ward Melville High School
East Setauket, New York, USA
ryanqin15@gmail.com

[2]Department of Computer Science
Stony Brook University
Stony Brook, New York, USA
{suwzhu, yuhalin, yujko, xiaojun}@cs.stonybrook.edu

ABSTRACT
T9-like keyboards (i.e., 3×3 layouts) have been commonly used on small touchscreen devices to mitigate the problem of tapping tiny keys with imprecise finger touch (e.g., T9 is the default keyboard on Samsung Gear 2). In this paper, we proposed a computational approach to design optimal T9-like layouts by considering three key factors: clarity, speed, and learnability. In particular, we devised a clarity metric to model the word collisions (i.e., words with identical tapping sequences), used the Fitts-Digraph model to predict speed, and introduced a Qwerty-bounded constraint to ensure high learnability. Founded upon rigorous mathematical optimization, our investigation led to *Optimal-T9*, an optimized T9-like layout which outperformed the original T9 and other T9-like layouts. A user study showed that its average input speed was 17% faster than T9 and 26% faster than a T9-like layout from literature. Optimal-T9 also drastically reduced the error rate by 72% over a regular Qwerty keyboard. Subjective ratings were in favor of Optimal-T9: it had the lowest physical, mental demands, and the best perceived-performance among all the tested keyboards. Overall, our investigation has led to a more efficient, and more accurate T9-like layout than the original T9. Such a layout would immediately benefit both T9-like keyboard users and small touchscreen device users.

CCS Concepts
•Human-centered computing → Text input; Touch screens;

Author Keywords
Text entry; Touchscreen; Smartwatches; Pareto optimization.

INTRODUCTION
As computing devices become diversified in the modern Post-PC computing era, a tremendous amount of challenge for entering text has arisen, especially on small-sized devices such as smartwatches. Text entry on small touchscreens is notoriously error-prone and inefficient. Since 26 letters are crammed into a small area, each key occupies a very tiny space. Also, because a finger is intrinsically an inaccurate input device (a.k.a., Fat finger problem), it is extremely difficult to accurately land the input finger on the tiny intended key when typing on such a keyboard.

To cope with this challenge, many manufacturers and users have adopted multi-letter key layouts (i.e., each key may correspond to multiple letters), especially on devices with small touchscreen. For example, the default input method on Samsung Gear 2 is T9 [15]; a number of multi-letter key keyboards for wearable devices and smartphones are now available in Google Play Store and Apple App Store, ready for immediate use; the T9 layout has always been an option for almost all major virtual keyboard products such as Google Gboard [13], Microsoft Swiftkey [23], etc.

T9-like layouts possess advantages on small touchscreen devices. Merging tiny keys together enlarges the key size, making the key acquisition tasks easier (according to the Fitts’ law [7]). On the other hand, multi-letter key layouts inevitably introduce word collisions (i.e., words with identical tap sequences), which may cause ambiguation and slow users down.

To address this issue, one option is to rearrange the key positions to reduce the chance of word collision. However, as shown in previous research [2, 3, 31, 37], layout optimiza-

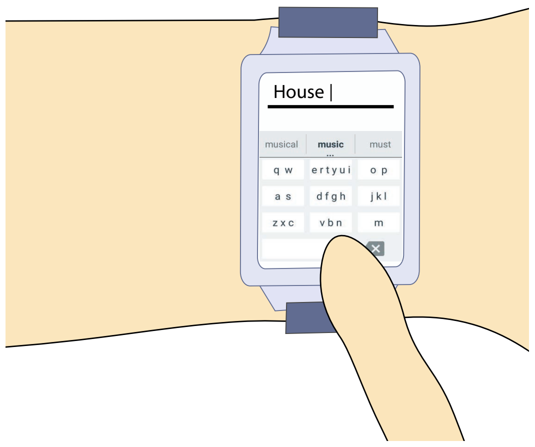


Figure 1. Typing on the Optimal-T9 keyboard on a smartwatch.

*Ryan Qin is a high school student supervised by Suwen Zhu.

tion is a non-trivial task. A number of factors including the finger traveling distance between letter pairs and similarity with Qwerty could all affect the input performance. Additionally, since the modern touchscreen keyboard adopts the statistical decoding technique [4, 8, 12, 34, 35], it is necessary to take into account the decoding principle when rearranging key positions to strengthen the ability to disambiguate input. Although there has been a sizable amount of research on keyboard layout optimization, most previous works have focused on single-letter key layout design; research related to multi-letter key layout design and optimization has lagged behind.

The primary goal of this research is to investigate principles and methodologies for optimizing multi-letter key layouts. To this end, we have proposed a computational approach for designing optimal multi-letter key layouts by considering three important factors: clarity (i.e., reducing the number of words with identical tapping sequences), speed, and learnability. Note that this method is orthogonal to approaches aiming at improving the decoding ability for small-sized keyboards. In this work, we focus on the effects of different keyboard layouts on users' typing performance.

To test the validity of the proposed approach and provide users with optimized layouts, we applied a rigorous mathematical optimization to search for T9-like (a.k.a., 3×3) layouts. Our research has led to *Optimal-T9*, a layout which outperformed the existing multi-letter key layouts, including T9, and also had advantages over the de facto standard Qwerty layout, which is the state-of-the-art text entry method.

The impacts of this research are twofold. The proposed optimization principles and approach will benefit keyboard layout designers and engineers, while the generated optimized layouts, especially Optimal-T9, will benefit multi-letter key keyboard users as well as those who often type on small touchscreen keyboards.

RELATED WORK

This work is relevant to keyboard layout optimization research and text entry techniques on small touchscreen devices.

Keyboard Layout Optimization Research

It has been widely known that the traditional Qwerty layout is sub-optimal for one finger or stylus input. Starting from at least as early as Getschow and colleagues' optimization work [9] for increasing efficiency for the motor impaired, researchers have explored a number of approaches to search for efficient alternatives, first using simple algorithms [9, 22] or heuristics [28], and eventually relying upon more rigorous mathematical optimization [2, 3, 37]. As the understanding of keyboard performance has deepened, the optimization procedure has become more sophisticated, progressing from improving input speed exclusively [3] to considering multiple factors such as speed, accuracy, and learnability [2, 5, 6]. A number of optimization techniques have also been proposed, including the Metropolis algorithm [37], Pareto multi-objective optimization [33], and integer programming [21]. A large portion of previous research examined single-letter key layout design, whereas our work focuses on multi-letter key layout design.

Most of the multi-letter key layout research was carried out on 9 - 12 key keypads [10, 11, 19], prior to the popularity of touchscreen devices. Gong and Tarasewich's work [10, 11] investigated the design of an alphabetically constrained keypad to reduce the number of key strokes needed to enter words. Their research revealed that alphabetically-constrained designs were close to unconstrained designs in terms of performance, and the alphabetically-constrained design was easier to learn than an unconstrained design. Hwang and Lee's work [19] showed that the spatial memory about the Qwerty keyboard helped the user locate correct keys on a keypad. They proposed a Qwerty-like 9-key layout, which had advantages in typing speed and learning over T9. Moving forward, we advance the multi-letter layout optimization research to virtual touchscreen keyboards powered by the statistical decoding technique. Our approach is also more sophisticated, as it performs Pareto optimization on three objectives.

Text Entry Methods on Small Devices

A number of innovations have been proposed to overcome the challenges of typing on small virtual keyboards. One common approach is to expand the keyboard or key size using various methods. For example, ZoomBoard [29] requires the user to first zoom into a region containing the desired key to enlarge the keyboard. DualKey [16] associates each key with two letters, while tapping the key using different fingers selects different letters. DriftBoard [32] allows a user to pan a movable keyboard to position the desired key under a fixed cursor point. SplitBoard [18] supports switching between the left and right halves of a Qwerty keyboard with a flick gesture. MacKenzie [24] proposed the scanning ambiguous keyboard (SAK) with three virtual keys and one SPACE key. Keys are highlighted in sequence ("scanned") and users press a physical key to select an intended key.

Past research has also shown that the modern statistical decoding technique worked reasonably well on small keyboards. Gordon et al. [14] revealed that human motor control adaptability, coupled with modern statistical decoding and error correction technologies developed for smartphones, can enable a surprisingly effective typing performance for both gesture typing and tap typing on a regular Qwerty keyboard on a watch-sized screen. Inspired by Gordon et al.'s research, we coupled multi-letter key layout design with the modern statistical decoding technique and compared the optimized multi-letter key layout with a regular Qwerty keyboard.

OPTIMIZING MULTI-LETTER KEY LAYOUTS

Objectives

Keyboard layout design is a non-trivial task. A number of factors have to be carefully considered and balanced. To gain successful adoption, we believe the following objectives are essential:

1. **Clarity.** The layout should minimize the potential word collisions (i.e., words sharing identical tap sequences).
2. **Speed.** The layout should improve input efficiency.

3. **Learnability.** The layout should be easy-to-learn to facilitate adoption.

Clarity

We have devised a metric, called *clarity*, to quantitatively measure the extent to which layouts are resistant to the word collision problem. We first defined the clarity score of word W on a multi-letter key layout L. Intuitively, such a metric should satisfy the following properties:

1. If W has a collision with more words, its clarity score should be lower.
2. If W has a collision with common words (i.e., words with high frequencies), the clarity score should be relatively low.

Our definition of the clarity score is as follows:

$$clarity(W) = \frac{f_W}{\sum_{i=1}^{N} f_{W_i}}, \tag{1}$$

where f_W is the word frequency of W in a given corpus, W_i is a word that shares the identical tapping sequence with W on the layout L, including the word W itself, and N is the total number of words that share the same tapping sequence. As shown, the expression for clarity satisfies the two desired properties.

In fact, we derived the definition of word clarity (Equation (1)) from the statistical decoding principle [4, 8, 12, 34, 35]. It approximates the probability that the intended word (W) appears as the top suggestion if a user provides the perfect input signals (e.g., tapping the center of the key for each letter). The derivation process for clarity is as follows.

The statistical decoding principle can be described as follows: given a set of tapping sequences on the keyboard $S = \{s_1, s_2, s_3, ..., s_n\}$, the decoder is to find word W^* in lexicon L that satisfies:

$$W^* = \underset{W \in L}{\arg\max} P(W|S). \tag{2}$$

From the Bayes' rule,

$$P(W|S) = \frac{P(S|W)P(W)}{P(S)}. \tag{3}$$

As $P(S)$ is an invariant across words, Equation (2) becomes:

$$W^* = \underset{W \in L}{\arg\max} P(S|W)P(W). \tag{4}$$

$P(W)$ is obtained from a language model (LM) and is often referred to as the language score. $P(S|W)$ is from a spatial model (SM) and is often called the spatial score. $P(S|W)P(W)$ is referred to as the overall score.

If perfect spatial signals were observed for a given word W, the spatial score for the words sharing the same key pressing sequences with W (i.e., the words W has a collision with) will be very high and identical. In contrast, the spatial scores for other words will be very small, close to 0. In this case, the decoder can only consider words sharing an identical input sequence with W for decoding. Assuming W has collisions with N words (including W itself), according to Equation (2) and Equation (4), the probability of the word W being the top suggestion becomes:

$$clarity^*(W) = \frac{P(W|S)}{\sum_{i=1}^{N} P(W_i|S)} = \frac{P(S|W)P(W)}{\sum_{i=1}^{N} P(S|W_i)P(W_i)}. \tag{5}$$

Equation (5) essentially normalizes the overall score of W among words it has a collision with. As explained before, words sharing the identical tapping sequence always share the same spatial score ($P(S|W_i)$), Therefore, Equation (5) becomes:

$$clarity^*(W) = \frac{P(S|W)P(W)}{\sum_{i=1}^{N} P(S|W_i)P(W_i)} = \frac{P(W)}{\sum_{i=1}^{N} P(W_i)}. \tag{6}$$

$P(W)$ is the probability of the word from the language model only. Without knowing any language context, it is the frequency of the word (or the probability from a unigram language model). Equation (6) and Equation (1) are identical. The derivation ends here.

For any given word W, $clarity(W)$ is a value between 0 and 1, indicating how likely the word will avoid word collisions on an arbitrary keypad configuration. For example, if the input sequence of W is not shared by any other words, $clarity(W) = \frac{f(W)}{f(W)} = 1$, meaning that it has no collisions at all. In contrast, if W has a collision with a large number of common words, $clarity(W)$ will be small.

To describe the degree to which a layout (L) is free of word collisions, we define the clarity score of a layout as the sum of all words' clarity scores on that keyboard weighted by their frequencies:

$$C(L) = \sum_{j=1}^{M} f(W_j) clarity(W_j), \tag{7}$$

assuming that the corpus has M words in total.

For the regular Qwerty layout, $C(L)$ equates to 1 because $clarity(W)$ is 1 for any word W. The more likely a layout will cause word collisions, the smaller $C(L)$ will become.

Speed

The typing speed metric estimates how fast expert users will be able to tap type on a keyboard layout. We used the widely known Fitts-Digraph model [3, 38] for speed prediction, which shows that the average time (t) for inputting a letter is:

$$t = \sum_{i=1}^{26} \sum_{j=1}^{26} P_{ij} T_{ij}, \tag{8}$$

where P_{ij} is the frequency of the ordered character pair i, j from 26 Roman characters, and T_{ij} is the movement time for the input finger travelling from key i to key j, which is typically predicted by the Fitts' law:

$$T_{ij} = a + b \log_2\left(\frac{D_{ij}}{W_{ij}} + 1\right), \tag{9}$$

where D_{ij} is the distance from the center of key i to the center of key j, and W_{ij} is the key width. Since each key tap action is essentially a 2-dimensional Fitts' law task, we used $min(W_{ij}, H_{ij})$ (i.e., the minimum of key width or height) as W_{ij} in Equation (9) [26]. Previous research [26] showed that it yielded a fairly successful fit for 2D Fitts tasks. In the context of touchscreen typing, Fitts' law parameters were $a = 0.083s$ and $b = 0.127s$, estimated by Zhai et al. [38]. t has the unit of seconds. t can be converted to input speed (V) in characters per minute (CPM): $V = 60/t$.

Learnability

Learnability is critical to the success of any new layout design. Perhaps the biggest obstacle of any newly optimized keyboard is learning the layout. Consequently, although numerous layouts have been proposed, very few are actually implemented extensively. To achieve superior performance over existing layouts, users likely have to spend a considerable amount of time practicing, and not every user is willing to make such an effort. For an optimal layout to maintain high learnability, we devise a strict Qwerty-bounded constraint: we preserve Qwerty's alphabetical arrangement to ensure that users can immediately use this keyboard fluently. Note that the Qwerty-bounded constraint only works for layouts with 3 rows.

Multi-Objective Optimization

With the two aforementioned objectives and the Qwerty-bounded constraint, designing a multi-letter key layout is essentially a multi-objective optimization problem: searching for a layout optimized for both clarity and speed, subject to the Qwerty-bounded constraint.

As commonly used in layout optimization research, we adopted the Pareto optimization technique [5, 6] to address this multi-objective optimization problem. Instead of generating a single optimized layout, Pareto optimization will lead to a Pareto front, in which each layout is Pareto optimal, meaning that none of its metric scores can be improved without compromising the other scores. The designer selects layouts from the Pareto front after considering the relative weights between metrics or other factors.

COMPUTATIONALLY DESIGNING 3×3 LAYOUTS

Optimization Procedure

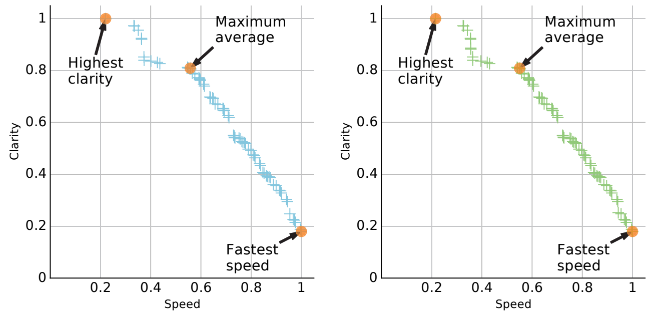
To validate the proposed optimization approach, we applied it to design optimal 3×3 layouts for a watch-sized multi-letter key keyboard. As multi-letter key layouts are traditionally implemented on small touchscreen devices, we investigated typing performance on the watch platform; a salient goal of this study is to produce an effective text entry interface option for users of small electronics. We based our keyboard representation on dimensions of the Apple Watch, with a screen size of 312 pixels $\times$390 pixels (26.15 mm $\times$32.69 mm).

The corpus for optimization was taken from the American National Corpus (ANC) [20] and contains a total of 239,208 unique words with respective frequencies. We carried out the optimization as follows:

1. All 15,120 possible multi-letter key configurations satisfying the Qwerty-bounded constraints were iterated in the algorithm, and we obtained the greatest and smallest values in speed and clarity separately.

2. We normalized each candidate layout's metric scores in a linear fashion to the 0 - 1 scale.

3. We iterated over all the possible multi-letter key layouts again, and using the Pareto optimization approach, generated the Pareto front.

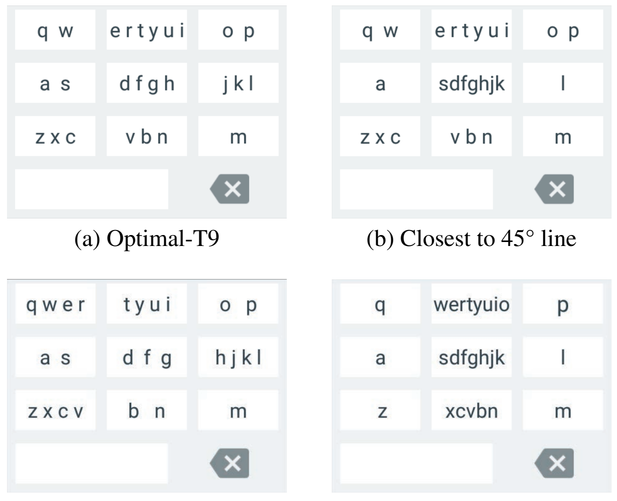


(a) (b)

Figure 2. The Pareto fronts using (a): Fitts' law and (b): FFitts laws for speed prediction.

Optimized Keyboard Layouts

Figure 2a illustrates the complete Pareto front formed by 71 Pareto optimal layouts. As shown, the front approximately forms a curve spanning the top-left and bottom-right corner, indicating that clarity and speed are conflicting metrics: a gain in one of them causes a loss in another. The preference for a particular layout is decided by the relative importance of speed or clarity. However, because typing preferences largely depend on the individual user, the trade-offs between clarity and speed will vary. For instance, a user who tap types very rapidly but erroneously may favor a layout with higher clarity



(a) Optimal-T9 (b) Closest to 45° line

(c) Maximum clarity score (d) Maximum speed score

Figure 3. Examples of optimized multi-letter key layouts generated using the proposed computational approach. The letter positions all followed their Qwerty alphabetical arrangement.

score, while a user who types at a slower pace, but accurately, may wish to choose a layout with a high speed score.

As we are particularly interested in the layouts with the most balanced typing clarity and speed, we closely examined the layouts near the center of the Pareto front. We selected the layout carrying the maximum average of normalized clarity and speed scores as the optimized layout subject to our specific Qwerty constraints. We referred to this configuration as Optimal-T9 (Figure 3a), which lies on the 55.4° line from the origin. Figure 3b shows the keyboard closest to the 45° line on the Pareto front, i.e., has the most balanced speed and clarity scores. Figures 3c and 3d display the layouts at two ends of the front: the one possessing the highest clarity and the one holding the fastest speed. The clarity scores and estimated input speeds are shown in Table 1.

	Optimal-T9	Highest clarity	Fastest speed	T9	Qwerty-like
Clarity	0.8738	0.9412	0.6519	0.9234	0.9468
CPM - Fitts'	309.70	284.27	343.14	278.18	270.55
WPM - Fitts'	61.94	56.85	68.63	55.64	54.11
CPM - FFitts	281.50	257.83	313.31	252.40	244.90
WPM - FFitts	56.3	51.57	62.66	50.48	48.98

Table 1. The clarity and speed (in CPM and WPM) of different 3×3 layouts. Note that the highest clarity layout possesses a lower clarity score than the Qwerty-like keyboard. This is because both highest clarity and fastest speed refer to keyboards with best respective metric scores within the scope of our optimization with the Qwerty-bounded constraint.

Although Fitts' law has been widely used in layout optimization, recent research confirmed that FFitts law [1] has better capability for modeling finger touch target selection. We replaced the Fitts' law in the Fitts-Digraph model with FFitts law, and re-ran the optimization procedure to verify the optimization results. Using FFitts law in lieu of Fitts' law produced little difference in the Pareto front, with the biggest disparity being that the new Pareto front contained 74 Pareto optimal layouts instead of 71 produced with the Fitts' law application. As shown in Figure 2b, the new Pareto front shape closely resembles the one with Fitts' law. Additionally, the layouts with the highest clarity score, highest speed, and maximum average of clarity and speed remained unchanged. However, the respective estimated input speeds generally varied between the two models, as shown in Table 1. Though speeds predicted using the FFitts law were lower than those predicted using Fitts' law, FFitts law predictions more accurately reflected the empirical data, as shown later in the user study.

EVALUATING THE OPTIMIZED MULTI-LETTER LAYOUT

Our investigation up to this point was theoretical. To empirically validate the proposed optimization approach and understand to what degree the optimized layout would improve typing performance in realistic text entry tasks, we conducted a user study to evaluate the performance of the optimized multi-letter key layout.

Experiment Setup

Tasks

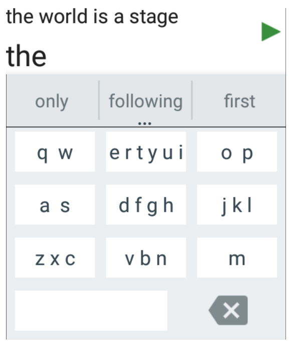
The study used watch-size keyboards to transcribe phrases in which the transcribed phrase was shown on the screen. Participants were seated in front of a desk, and were instructed to type as fast and as natural as possible. Participants were instructed to use the suggestion bar freely. As shown in Figure 4, users pressed the green button to proceed to the next trial.

The phrases were randomly selected from a subset of the Mackenzie and Soukoreff phrase set [27], using the selection procedure proposed by [36]. The same set of phrases was used for all the participants and the order of the phrases was randomized for each participant.

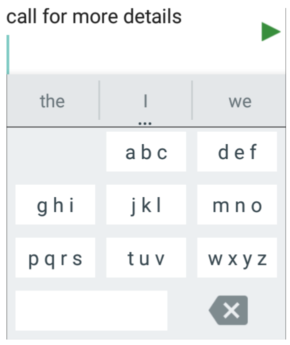
Design

We adopted a within-subject design in which the independent variable was the keyboard type with four levels. In addition to the Optimal-T9 layout (Figure 4a), we included the following layouts as baselines: the traditional ABC-T9 layout (Figure 4b), the Qwerty-like layout [19] (Figure 4c) which was adapted from Qwerty, and the standard Qwerty layout (Figure 4d). We included the Qwerty-like layout because Hwang et al. [19] showed that it outperformed ABC-T9 in their study.

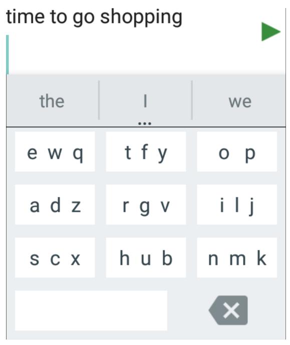
All the keyboards were built based on the statistical decoding principle with the same *n*-gram language model, which has a 60K lexicon and includes unigrams, bigrams and trigrams. All keyboards also used the same predictive model. Since our goal was to evaluate the performance of keyboard layouts, we kept the number of suggestions unchanged across all the conditions. Each keyboard displayed three suggested words on the suggestion bar, which is the number of suggestions commonly used on touchscreen keyboards (e.g., Google GBoard, Microsoft SwiftKey, etc.).



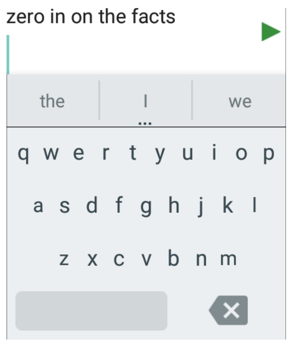
(a) Optimal-T9



(b) T9



(c) Qwerty-like



(d) Qwerty

Figure 4. Keyboard layouts used in the evaluation experiments.

The orders of the four keyboards were balanced across participants. Before the formal study, participants performed a practice session with three phrases. Each task consisted of 5 blocks, and each block contained 6 phrases. Participants were allowed to take a short break after the completion of each block. In total, the study collected: 4 keyboards $\times$ 5 blocks $\times$ 6 phrases $\times$ 20 participants = 2400 phrases.

Participants and Apparatus

20 subjects (5 female) aged from 23 to 34 ($M = 27.1, SD = 3.02$) participated in the study. Participants were asked to rate their familiarity with the Qwerty layout and touchscreen typing before the study. The median familiarity with Qwerty layout (1: not familiar; 5: very familiar) was 4.5. The median familiarity with touchscreen typing (1: not familiar; 5: very familiar) was 4. Participants were instructed to use their preferred input finger throughout the study: 15 participants used the index finger, and 5 participants used one thumb.

An LG Nexus 5X device was used to simulate the watch-size screen. The simulated watch-size screen was 26.15 mm $\times$32.69 mm, identical to the screen dimensions of the Apple Watch. The keyboard size was 26.15 mm $\times$20.18 mm.

Results

Speed

We calculated the text entry speed following Mackenzie [25]:

$$WPM = \frac{|T-1|}{S} \times 60 \times \frac{1}{5}, \qquad (10)$$

where T is the target string and S is the elapsed time in seconds from the first to the last touch in the sentence. Figure 5 shows the average text entry speed for each keyboard. As shown, Optimal-T9 was the fastest among all the multi-letter key layouts. It was 16.6% faster than T9, and 26.2% faster than the Qwerty-like layout.

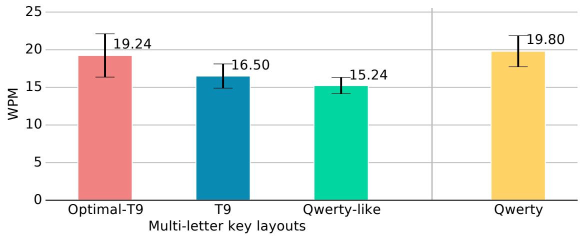


Figure 5. Means (95% confidence interval) of input speed by keyboard.

There was a main effect of keyboard type on the average text entry speed ($F_{3,57} = 8.408, p < 0.001$). Pairwise t-tests showed that the differences were significant between Optimal-T9 vs. Qwerty-like ($p = 0.02$), Optimal-T9 vs. T9 ($p = 0.0396$), Qwerty vs. Qwerty-like ($p < 0.001$), and Qwerty vs. T9 ($p = 0.011$).

Error Rate

We define word error rate as:

$$r = \frac{MWD(S,P)}{LengthInWords(P)} \times 100\%, \qquad (11)$$

where $MWD(S,P)$ is the minimum word-level editing distance between transcribed phrase S and the target phrase P, and $LengthInWords(P)$ is the number of words in P.

As shown in Figure 6, the average word error rate was 1.95% ($SD = 1.69\%$) for Optimal-T9, 2.58% ($SD = 2.15\%$) for T9, 2.46% ($SD = 2.26\%$) for Qwerty-like, and 6.98% ($SD = 15.63\%$) for Qwerty. ANOVA did not show a main effect of keyboard type on the word error rate ($F_{3,57} = 1.863, p = 0.146$).

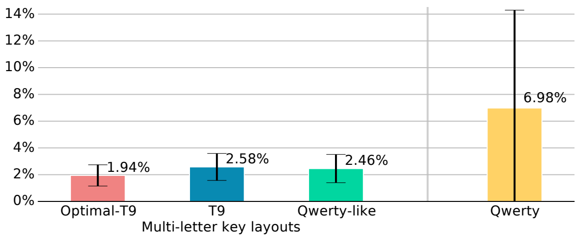


Figure 6. Means (95% confidence interval) of error rate by keyboard.

Learnability

To understand the general learning curve, and the learnability of each specific keyboard type, we analyzed how text entry speeds changed over time in Figure 7. There was a main effect of block number on the average speed of the tested keyboards at large ($F_{4,76} = 3.54, p = 0.0105$). Pairwise t-tests with Bonferroni adjustments showed that the differences were significant between the first block vs. the third block ($p = 0.0037$) and the first block vs. the fourth block ($p = 0.0314$). For individual keyboards, no significant differences were observed between any two blocks for Optimal-T9, Qwerty, or Qwerty-like. Pairwise t-tests with Bonferroni adjustments indicated a significant difference between the first block vs. the fourth block ($p = 0.024$) on the T9 layout.

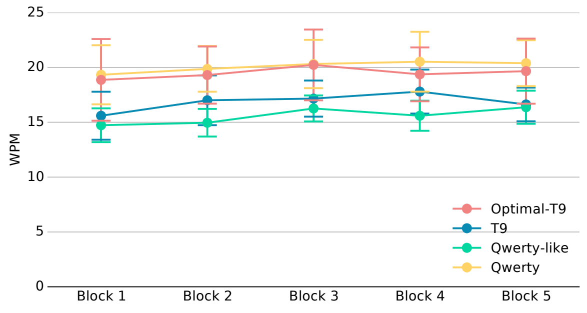


Figure 7. Means (95% confidence interval) of text entry speed by keyboard and block.

Backspace Usage

We measured the backspace usage by dividing the number of backspace key presses in one trial by the total number of words in the target phrase. The mean (SD) backspace usage was 0.33 (0.24) for Optimal-T9, 0.32 (0.33) for T9, 0.29 (0.22) for Qwerty-like, and 0.65 (0.36) for Qwerty. ANOVA

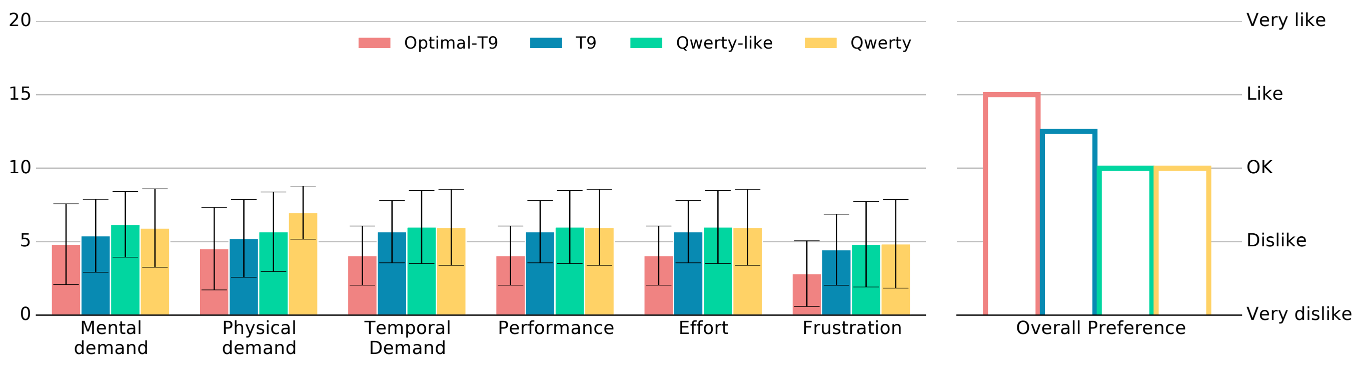


Figure 8. (a): Means (SD) of the NASA-TLX measures in (1-20) scale. 1 is the most positive rating, and 20 is the negative rating. (b): Means (SD) of the overall preference for each keyboard in (1-5) scale. 1 is the most negative rating, and 5 is the most positive rating.

showed a main effect of keyboard type on the backspace usage ($F_{3,57} = 10.66, p < 0.001$). Pairwise t-tests with Bonferroni adjustments showed that the difference was significant between Qwerty and the other three keyboards (all $p < 0.05$).

Subjective Ratings

After the completion of each keyboard type, participants were asked to rate their perceived task load with questions in NASA-TLX [17] in a continuous numeric (1-20) scale. The results are shown in Figure 8a. No significant differences were observed on mental demand, temporal demand, and performance. ANOVA indicated a main effect of keyboard type on the other three measures: physical demand ($F_{3,57} = 5.745, p = 0.00166$), effort ($F_{3,57} = 5.049, p = 0.00359$) and frustration ($F_{3,57} = 3.908, p = 0.0131$). Pairwise t-tests with Bonferroni adjustments revealed that the differences were significant between Optimal-T9 vs. Qwerty in terms of physical demand ($p = 0.027$), and Optimal-T9 vs. Qwerty-like in terms of effort ($p = 0.0016$) and frustration ($p = 0.013$). No significance was observed between other pairs.

At the end of the study, participants were also asked to give an overall preference of each keyboard on the scale of 1 (very dislike) to 5 (very like). As shown in Figure 8b, the median rating for Optimal-T9 (4) was the highest.

Discussion

Overall, Optimal-T9 exhibited better performance than the other 3×3 layouts. Its mean input speed was 17% faster than T9 and 26% faster than the Qwerty-like layout. The differences in typing speeds were statistically significant ($p < 0.05$) between Optimal-T9 vs. Qwerty-like and Optimal-T9 vs. T9. Likewise, the mean error rate of Optimal-T9 was also the lowest (1.94%), while the error rates for T9 and Qwerty-like were all above 2.4%.

Optimal-T9 has an edge over a standard Qwerty layout: although their speeds were similar, the error rate on Optimal-T9 was 72% lower and this difference between them approached significance ($p = 0.0803$).

Subjective ratings revealed substantial advantages of using Optimal-T9. Its mean rating was the best among all the categories, including the overall rating. Moreover, users felt the most relaxed physically and mentally when typing on Optimal-T9. Users also felt the least frustrated and exerted the least effort on Optimal-T9 compared to when typing on the other layouts.

The per-block typing speed exhibited a fast learning curve for typing on watch-size devices for Optimal-T9. We found that Optimal-T9 maintained a consistently high typing speed relative to the other layouts across all blocks. No significant differences were observed between any two blocks on Optimal-T9, which suggested minimal adaption in typing behavior. This finding confirmed our assumption that the Optimal-T9 layout, while preserving the Qwerty character arrangements, has a high user adaptability and makes for a readily usable keyboard.

LIMITATIONS

Though this work has successfully produced a novel keyboard that surpasses traditional T9 among multi-letter key layouts and shows an edge over Qwerty, many questions beyond the scope of this work merit further research. Optimal-T9 is the layout with highest average of clarity and speed. Depending on the preference, some users may prefer clarity over speed or vice-verse. It is interesting to investigate how the weights of clarity and speed differ across users, or whether an optimal weight exists for the general population. The current work focuses on layout optimization only.

Previous research [30, 31] has also shown that the number of suggestions could affect the layout performance because extra visual attention or cognitive effort is needed when searching through a candidate word list. The ultimate keyboard design needs to take into consideration this factor. It is also necessary to conduct larger and more thorough empirical investigations of multidimensional optimization spanning more objectives such as finger or stylus input strategies. More specifically, research is warranted for broader investigation of multi-letter key layout optimization beyond strictly Qwerty-bounded layouts. It may be computationally feasible to model expertise on Qwerty-similar layouts for optimization and lesson the layout learnability constraint of this study.

CONCLUSIONS

The contributions of our research are of both theoretical and practical significance. At the theoretical front, we have proposed a computational approach for designing optimal multi-letter key layouts. Our approach takes into consideration 3 salient factors: clarity, speed and learnability. In particular, we derived a clarity metric to quantify a candidate layout's ability to resolve word collisions, used the Fitts-Digraph model to predict input speed, and introduced the Qwerty-bounded constraint to ensure high layout learnability.

At the practical front, this work has led to an ergonomic and promising interface option that can immediately benefit users of portable electronics. Our research produced Optimal-T9, a 3×3 layout optimized for the three aforementioned factors. A user study demonstrated that Optimal-T9 improved tap typing performance over existing 3×3 layouts, and possessed advantages over the standard Qwerty layout. Our investigation confirmed that the proposed computational approach is both effective and promising in designing a multi-letter key layout, and led to Optimal-T9, a layout that can immediately benefit multi-letter key layout users and small touchscreen device users.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their insightful comments and our user study participants. This work was supported in part by NSF CHS-1815514.

REFERENCES

1. Xiaojun Bi, Yang Li, and Shumin Zhai. 2013. FFitts Law: Modeling Finger Touch with Fitts' Law. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 1363–1372. DOI: http://dx.doi.org/10.1145/2470654.2466180
2. Xiaojun Bi, Barton A. Smith, and Shumin Zhai. 2010. Quasi-qwerty Soft Keyboard Optimization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 283–286. DOI: http://dx.doi.org/10.1145/1753326.1753367
3. Xiaojun Bi, Barton A Smith, and Shumin Zhai. 2012. Multilingual touchscreen keyboard design and optimization. *Human–Computer Interaction* 27, 4 (2012), 352–382.
4. Xiaojun Bi and Shumin Zhai. 2013. Bayesian Touch: A Statistical Criterion of Target Selection with Finger Touch. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. ACM, New York, NY, USA, 51–60. DOI: http://dx.doi.org/10.1145/2501988.2502058
5. Xiaojun Bi and Shumin Zhai. 2016. IJQwerty: What Difference Does One Key Change Make? Gesture Typing Keyboard Optimization Bounded by One Key Position Change from Qwerty. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 49–58. DOI:http://dx.doi.org/10.1145/2858036.2858421
6. Mark Dunlop and John Levine. 2012. Multidimensional Pareto Optimization of Touchscreen Keyboards for Speed, Familiarity and Improved Spell Checking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 2669–2678. DOI: http://dx.doi.org/10.1145/2207676.2208659
7. Paul M Fitts. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology* 47, 6 (1954), 381.
8. Andrew Fowler, Kurt Partridge, Ciprian Chelba, Xiaojun Bi, Tom Ouyang, and Shumin Zhai. 2015. Effects of Language Modeling and Its Personalization on Touchscreen Typing Performance. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 649–658. DOI: http://dx.doi.org/10.1145/2702123.2702503
9. CO Getschow, MJ Rosen, and C Goodenough-Trepagnier. 1986. A systematic approach to design a minimum distance alphabetical keyboard. In *Proc. RESNA (Rehabilitation Engineering Society of North America) 9th Annual Conference*. 396–398.
10. Jun Gong and Peter Tarasewich. 2004. Guidelines for handheld mobile device interface design. In *In Proceedings of the 2004 DSI Annual Meeting*.
11. Jun Gong and Peter Tarasewich. 2005. Alphabetically Constrained Keypad Designs for Text Entry on Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05)*. ACM, New York, NY, USA, 211–220. DOI: http://dx.doi.org/10.1145/1054972.1055002
12. Joshua Goodman, Gina Venolia, Keith Steury, and Chauncey Parker. 2002. Language Modeling for Soft Keyboards. In *Proceedings of the 7th International Conference on Intelligent User Interfaces (IUI '02)*. ACM, New York, NY, USA, 194–195. DOI: http://dx.doi.org/10.1145/502716.502753
13. Google. 2018. Gboard - the Google Keyboard. (2018). https://play.google.com/store/apps/details?id=com.google.android.inputmethod.latin
14. Mitchell Gordon, Tom Ouyang, and Shumin Zhai. 2016. WatchWriter: Tap and Gesture Typing on a Smartwatch Miniature Keyboard with Statistical Decoding. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 3817–3821. DOI: http://dx.doi.org/10.1145/2858036.2858242
15. DL Grover, MT King, and CA Kuschler. 1998. Patent no. US5818437, reduced keyboard disambiguating computer. Tegic communications. *Inc., Seattle, WA* (1998).

16. Aakar Gupta and Ravin Balakrishnan. 2016. DualKey: Miniature Screen Text Entry via Finger Identification. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 59–70. DOI: http://dx.doi.org/10.1145/2858036.2858052

17. Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology*. Vol. 52. Elsevier, 139–183.

18. Jonggi Hong, Seongkook Heo, Poika Isokoski, and Geehyuk Lee. 2015. SplitBoard: A Simple Split Soft Keyboard for Wristwatch-sized Touch Screens. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1233–1236. DOI: http://dx.doi.org/10.1145/2702123.2702273

19. Sunyu Hwang and Geehyuk Lee. 2005. Qwerty-like 3x4 Keypad Layouts for Mobile Phone. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems (CHI EA '05)*. ACM, New York, NY, USA, 1479–1482. DOI: http://dx.doi.org/10.1145/1056808.1056946

20. Nancy Ide and Catherine Macleod. 2001. The american national corpus: A standardized resource of american english. In *Proceedings of Corpus Linguistics (Vol. 3)*. Lancaster, UK, 1–7.

21. Andreas Karrenbauer and Antti Oulasvirta. 2014. Improvements to Keyboard Optimization with Integer Programming. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 621–626. DOI: http://dx.doi.org/10.1145/2642918.2647382

22. James R Lewis, Peter J Kennedy, and Mary J LaLomia. 1999. Development of a digram-based typing key layout for single-finger/stylus input. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 43, 5 (1999), 415–419. DOI: http://dx.doi.org/10.1177/154193129904300505

23. TouchType Ltd. 2018. SwiftKey - Smart prediction technology for easier mobile typing. (2018). https://swiftkey.com/

24. I. Scott MacKenzie. 2009. The One-key Challenge: Searching for a Fast One-key Text Entry Method. In *Proceedings of the 11th International ACM SIGACCESS Conference on Computers and Accessibility (Assets '09)*. ACM, New York, NY, USA, 91–98. DOI: http://dx.doi.org/10.1145/1639642.1639660

25. I. Scott MacKenzie. 2015. A Note on Calculating Text Entry Speed. (2015). http://www.yorku.ca/mack/RN-TextEntrySpeed.html

26. I. Scott MacKenzie and William Buxton. 1992. Extending Fitts' Law to Two-dimensional Tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '92)*. ACM, New York, NY, USA, 219–226. DOI: http://dx.doi.org/10.1145/142750.142794

27. I. Scott MacKenzie and R. William Soukoreff. 2003. Phrase Sets for Evaluating Text Entry Techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems (CHI EA '03)*. ACM, New York, NY, USA, 754–755. DOI: http://dx.doi.org/10.1145/765891.765971

28. I. Scott MacKenzie and Shawn X. Zhang. 1999. The Design and Evaluation of a High-performance Soft Keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99)*. ACM, New York, NY, USA, 25–31. DOI: http://dx.doi.org/10.1145/302979.302983

29. Stephen Oney, Chris Harrison, Amy Ogan, and Jason Wiese. 2013. ZoomBoard: A Diminutive Qwerty Soft Keyboard Using Iterative Zooming for Ultra-small Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 2799–2802. DOI: http://dx.doi.org/10.1145/2470654.2481387

30. Philip Quinn and Shumin Zhai. 2016. A Cost-Benefit Study of Text Entry Suggestion Interaction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 83–88. DOI: http://dx.doi.org/10.1145/2858036.2858305

31. Sayan Sarcar, Jussi PP Jokinen, Antti Oulasvirta, Zhenxin Wang, Chaklam Silpasuwanchai, and Xiangshi Ren. 2018. Ability-Based Optimization of Touchscreen Interactions. *IEEE Pervasive Computing* 17, 1 (2018), 15–26.

32. Tomoki Shibata, Daniel Afergan, Danielle Kong, Beste F. Yuksel, I. Scott MacKenzie, and Robert J.K. Jacob. 2016. DriftBoard: A Panning-Based Text Entry Technique for Ultra-Small Touchscreens. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 575–582. DOI: http://dx.doi.org/10.1145/2984511.2984591

33. Brian A. Smith, Xiaojun Bi, and Shumin Zhai. 2015. Optimizing Touchscreen Keyboards for Gesture Typing. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 3365–3374. DOI: http://dx.doi.org/10.1145/2702123.2702357

34. Keith Vertanen, Haythem Memmi, Justin Emge, Shyam Reyal, and Per Ola Kristensson. 2015. VelociTap: Investigating Fast Mobile Text Entry Using Sentence-Based Decoding of Touchscreen Keyboard Input. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 659–668. DOI: http://dx.doi.org/10.1145/2702123.2702135

35. Daryl Weir, Henning Pohl, Simon Rogers, Keith Vertanen, and Per Ola Kristensson. 2014. Uncertain Text Entry on Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 2307–2316. DOI:http://dx.doi.org/10.1145/2556288.2557412

36. Xin Yi, Chun Yu, Weinan Shi, Xiaojun Bi, and Yuanchun Shi. 2017. Word Clarity As a Metric in Sampling Keyboard Test Sets. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 4216–4228. DOI: http://dx.doi.org/10.1145/3025453.3025701

37. Shumin Zhai, Michael Hunter, and Barton A. Smith. 2000. The Metropolis Keyboard - an Exploration of Quantitative Techniques for Virtual Keyboard Design. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology (UIST '00)*. ACM, New York, NY, USA, 119–128. DOI: http://dx.doi.org/10.1145/354401.354424

38. Shumin Zhai, Alison Sue, and Johnny Accot. 2002. Movement Model, Hits Distribution and Learning in Virtual Keyboarding. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02)*. ACM, New York, NY, USA, 17–24. DOI: http://dx.doi.org/10.1145/503376.503381