공학석사학위논문

Comparing QWERTY and Dvorak Keyboard Speed: a Pilot Study

QWERTY와 드보락 키보드의 속도 비교: 시험 연구

2014년 8월

서울대학교 대학원

전기컴퓨터공학부 컴퓨터공학전공

조 항 준

Abstract

Comparing QWERTY and Dvorak Keyboard Speed: a Pilot Study

Hanggjun Cho
Department of Electrical Engineering and Computer Science
College of Engineering
The Graduate School
Seoul National University

The comparison between QWERTY and Dvorak layouts has been a controversial for many years. It is well known that QWERTY layout was not designed for efficient English typing. Conversely, Dvorak layout was developed by considering the characteristics of English. Nevertheless neither layout has been demonstrated empirically to be superior. The main reason is that it is almost impossible to find people who are equally skilled in both layouts. Comparative studies measuring the typing speed on Dvorak against QWERTY have generally failed to remove biases arising from the unequal familiarity of users.

In preceding research, we suggested an efficient methodology for screening ultimate typing speed for different key layouts. Because it uses the users' familiarity with the generally-used layout, it naturally also has biases. However these biases can be easily quantified and corrected. We had shown this method's validity and effectiveness from the relatively large-scale experiments in mobile keypads with a wider range of users.

We apply this screening method to QWERTY and Dvorak to figure out which layout is faster and how much which layout is more efficient. We built a web based system to gather a large amount of detailed typing data in small cost. This enabled the objective comparison between two layouts and also discovering characteristics of English typing. This information can be beneficial to develop new input devices or methods for various computing devices these days.

Keywords: keyboard, layout, speed, comparison, QWERTY, Dvorak

Student Number: 2012-23240

Contents

List of Figures

List of Tables

Chapter 1

Introduction

People use various types of computers together in daily life. Many of these receive character input. Although mobile devices with touch screens come into wide use in recent, the dependency for keyboard is even getting increased. A few years ago, when most people used feature phones, 12-key numeric keypads were widely used to type characters. But people just use the standard keyboard layout these days because it is the most familiar way of typing and it is usually faster than others.

QWERTY is the 'de facto standard' keyboard layout for English input devices. This layout was originally designed for a typewriter in the early 1870s. At that time, the main concern of the layout was to prevent jamming of arms of the typewriters. In other words, it was somewhat purposed to slow down typing speed. This approach was fairly successful to type English safely with less problems. Due to this reason QWERTY layout became very popular in a short period [1].

But the problem was that the QWERTY layout was adopted as the layout of computer keyboards. It has been the default layout for keyboards from the beginning of the history of computers in that way. It is apparent that we do not need to care about jamming. Therefore, many alternative layouts had been developed to type English faster. Dvorak Simplified Keyboard is the representative of those alternatives. Its inventor, August Dvorak researched characteristics of English use and then designed a new layout for efficient English typing [2]. Although it is used by a very small number of people, it has been used by the fastest typists in the world [3]. And it is also registered as an ANSI standard.

Nonetheless, new layouts including Dvorak have been not accepted among the public. Simply because the difference in typing speed is not significant. It cannot overwhelm the cost of changing the layout. This cost also includes the public interest. Not many people pay attention to the better keyboard layouts. Moreover the typing speeds are difficult to be measured and compared because typing environments are too diverse. And it is very difficult to find people who can type two or more keyboard layouts in similar fluency. Therefore, researchers have tried to provide sufficient time for participants to learn new layouts. However this learning period can affect the speed on other layouts and it is also very expensive to train and observe the participants' learning process. For these reasons, the comparison between QWRETY and Dvorak is still a controversial issue for a long time.

In this thesis, we will try to show which layout is fundamentally and objectively faster than the other. For this comparison, we use a novel method which is suggested in our preceding research. The approach of this method is completely different from other research. It estimates typing speeds on different layouts from typing on the familiar layout. It means learning new layouts or changing layouts is not required. Thus it can provide a good estimation of the ultimate typing speed and useful information of a new layout quickly and economically.

In the following background chapter, previous research on keyboard comparisons will be presented. Then the new method and the experimental design will be explained in detail in the methodology chapter. And we can see the preliminary measurements which layout is faster and other characteristics from experiments in the result and discussion chapters.

Chapter 2

Background

Over the last two decades, many authors have suggested new layouts for various kinds of character input device. They usually evaluated its validity through human testing. Asking a group of people to type some standard texts with both a standard and the new layout, and comparing the elapsed time was the most widely-used methodology. While many authors made efforts to deal with the difference in familiarity, the extent to which they were able to eliminate it remains open to serious question.

In order to evaluate previous testing protocols, we set several conditions which an ideal testing methodology should satisfy:

1. As far as practicable, the effects of any difference in familiarity between the two layouts should be eliminated.

2. Comparisons should be at as close-to-expert level as is achievable.

3. The evaluation methodology should not make it difficult to recruit a large number of participants from a wide variety of backgrounds.

4. In particular, the testing regime should not require a large investment of time on the part of the participants (since this is likely to conflict with the previous criterion).

Previous strategies for dealing with the difference in familiarity can be roughly categorized into four groups:

- Ignoring the familiarity issue
- Restricting testing to novices
- Providing training session
- Observing learning progress

Unfortunately, none of these strategies can meet all our criteria, as discussed below.

2.1 Ignoring the Familiarity Issue

The easiest way to deal with the familiarity issue is to construct participant groups independent of their familiarity with the current layout. In the analysis phase, one simply ignores the issue, and compares performance without regard to familiarity (thus unfairly biasing the analysis toward the more familiar layout). Unfortunately this may be difficult to avoid when comparing physically dissimilar keyboards. A number of researchers have used this approach when their primary concern was the initial acceptance of new design, rather than the ultimate speed. In this case, we may be able to conclude that the new design is acceptable even if it is significantly slower than the control.

MacKenzie et al. [4] used such a method for comparing three text input methods on a pen-based computer (hand printing, QWERTY-tapping, and ABC-tapping). They tested 15 participants, without providing any training time. They argued that it was acceptable to not provide training sessions as they were interested in the walk-up acceptance.

Lewis et al. [5] used a similar strategy for evaluating key layouts for stylus input. A total of 12 participants, well-experienced with QWERTY, participated in the experiments. In the discussion section, the authors acknowledged the unfairness of the comparison, commenting that "typing speed results indicate that participants lowered their typing speed when using unfamiliar layouts."

Green et al. [6] used this method to evaluate stick keyboards (both multi-tap and lexicon types) against a QWERTY layout. Green gathered 10 participants, who were familiar with the QWERTY keyboard, but not with multi-tap methods. After a small number of experiments, they merely commented that stick keyboards were acceptable in comparison with QWERTY. Since they were trading off portability against speed, no deeper comparison was needed.

This approach, of simply ignoring familiarity issues, is sometimes adopted because of lack of data on the control input method. Mittal and Sengupta [7] tested their improvised mobile phone keypad layout with 6 participants. They noted that it was impossible to compare learning curves with existing layouts due to the absence of studies on their learning curves.

With this strategy, it is relatively easy to conduct experiments in a short time. Thus, it makes it easy to invite a large number of participants, although the majority of previous studies have tested only a small number of participants (5 to 20). That is, criteria 3 and 4 are satisfied. However criteria 1 and 2 are explicitly ignored through disregarding any difference in familiarity between layouts, so we would just be able to conclude that our new design is acceptable, not that it will ultimately be faster.

2.2 Restricting Testing to Novices

When we are unable to test the competing layouts at expert level, using novices for both layouts can be the fairest way of comparison. Unfortunately, however, it may be extremely difficult to find novice users for the current standard layout because of its ubiquity. This method may also result in serious bias because novices with the standard layout may be atypical users. Nevertheless, some previous research tried this approach, mainly due to its simplicity and fairness.

Hirsch [8] used 55 non-typists in his experiment comparing typing speeds between QWERTY and Griffith's Minimotion [9] layouts. After a pre-trial of the experiment, only 40 of the novices were selected for the actual trial. Hirsch anticipated that participants would be faster with the Minimotion layout due to its easily-recognizable alphabetical order. Nonetheless, the results indicated that novice users were good at QWERTY. Hirsch concluded that the participants might have had some experience on QWERTY standard, so he acknowledged that his intention of inviting complete novices at QWERTY was not fulfilled.

The learnability evaluation of Dvorak layout conducted by Harnett [10] used a similar method. Harnett reported that a novice user can reach up to 50 WPM (words per minute) in 5 hours of training. Another participant who could type 50 WPM with QWERTY was able to reach 35-40 WPM with Dvorak after 4 hours of training.

As there is no familiarity difference between layouts for novice participants, this approach satisfies criterion 1. It also does not require too much time and resources, so satisfies criterion 4. However, criterion 2 (expert-level comparison) is explicitly ignored. Criterion 3 is also difficult to fulfil, because it is generally difficult to find perfect novices for a widely-used standard layout.

2.3 Provision of Training Sessions

A minor variant of the first strategy provides a little time for participants to familiarize themselves on the new layout before evaluation. Although participants may experience a large difference in familiarity between the two layouts at first, they may narrow the gap before the actual test. Previous research in this category generally allowed a very short time for pre-training, so it does not really breach criteria 3 and 4, although it may require a little more time than the preceding strategies.

Butts and Cockburn [11] evaluated multi-tap mobile phone keypads against the two-key method. They provided sufficient time for participants to practice on each keyboard until they felt comfortable. They noted that this training time was usually less than one minute, even for those without any prior experience. Strictly speaking, providing training sessions cannot practically eliminate the difference in familiarity between keyboards. They defined "familiarity" by the subjective feeling of each participant, rather than a measurable index. There was no real attempt to measure whether each participant was sufficiently familiar with each layout to ignore the familiarity difference. Indeed, given the wide spread of multi-touch layouts at that time, one might reasonably be skeptical whether participants were really estimating their equal-familiarization time, or perhaps their boredom threshold.

Gong and Tarasewich [12] provided a short training time for evaluating their alphabetically constrained mobile phone layout in comparison with an unconstrained optimized one and with ABC-layout. The training session provided two sample sentences to be typed. They did not check whether the training time was enough for each participant, whether by subjective feeling or an objective index. Nevertheless, this minimal training time may have reduced the large errors occurring in the very first use of a layout.

In short, this approach has both pros and cons. It satisfies criteria 3 and 4, since it does not require much extra time or other resources for practice. It may not perfectly fulfill criterion 1, but does try to reduce the gap. Criterion 2, however, is not attainable, as it is impossible for the participants to practice sufficiently within the short time.

2.4 Detailed Observation of the Learning Progress

The final option is to provide a sufficiently long training time for participants to make substantial learning progress. Once the learning rate starts to tail off, the speed is plotted against training time, and this graph is used to estimate the optimum typing speed that each participant will eventually attain. Progress through time can be compared between the old and new layout even though participants may have some experience in the old design. At the start, we can generally expect a higher speed from the old layout, but the new design may overtake it in the long run. From such data, we may (if we are bold) conclude that the new layout will eventually be superior to the current standard based on this plot.

Strong's experiment is widely known in this QWERTY-Dvorak comparison and clearly shows the limitation of traditional methods. Strong compared typing speeds for QWERTY and Dvorak [13]. For fair comparison, he trained 10 typists with Dvorak until they attained their previous QWERTY speed. Then, he invited 10 more typists who are familiar with QWERTY layout. Each group was trained with each layout for further 100 hours, and Strong recorded the learning curves. Surprisingly, the Dvorak group did not show better performance than the QWERTY group with further training, even though Dvorak is reported to very easy to learn in a short time [10] and participants in Strong's experiment also learned it quickly. Thus we cannot simply assume that initial stage data can be a good estimator for ultimate speed.

Michaels compared an alphabetically-ordered computer keyboard layout with the QWERTY standard [14]. He tested 30 participants over 25 sessions for both layouts, concluding that alphabetical order has no advantage over QWERTY.

Thomas et al. [15] used a similar approach in comparatively evaluating three kinds of wearable computer (forearm mounted keyboard, virtual keyboard, and Kordic keypad). They provided an hour of training time, conducting 6 sessions over 3 weeks.

MacKenzie and Zhang [16] compared a new mobile keypad against the QWERTY layout. Five participants took part in 20 experimental sessions, spread over a week, and each lasting 45 minutes. They plotted the results, concluding that the new design was much easier to learn, and that it would be faster after a sufficient amount of practice (around 15 sessions).

Ingmarsson et al. [17] applied this strategy in testing television-based appliances. Five participants were involved, over ten experimental sessions.

Hwang and Lee conducted a similar experiment on a QWERTY-like mobile phone keypad layout [18]. In evaluation, they conducted 5 sessions of experiments with 20 participants over 5 days. Based on similar plots, they argued that the QWERTY-like layout was not only easy to learn, but also ultimately faster than the more common ABC-layout.

We also previously employed this method in [19], with 10 participants, to evaluate a Personalized Multigram (PM) layout against the ABC 12-key layout. We conducted sufficient repetitions of training with the PM layout for each participant to overtake their initial ABC speed, and drew the conclusion that the PM keypad is more efficient through comparing progress in typing speed.

Although these examples appear successful in validating the new designs, there are a number of limitations. Most important is the experimental cost – around a week of participants' time. Thus condition 4 is not satisfied. Partici-

Table 2.1 Evaluation of Traditional Keyboard Comparison Approaches

Method	Fairness	Expert-level	Scalability	Cost
Ignoring the familiarity issue	×	×	○	○
Restricting testing to novices	○	×	×	○
Providing training session	×	×	○	○
Observing learning progress	○	○	×	×

pants are required to dedicate themselves to the experiment for a substantial amount of time, while the three earlier strategies require from each participant only about an hour of effort. Moreover, in many scenarios, we need a specific machine or equipment for testing. This results in a resource limitation, restricting the pool of participants to very small numbers of people. For these reasons, the preceding experiments used only very narrow pools of participants – mostly laboratory members. This limitation, however, can cause serious problems in statistical reliability. In [16], the number of participants was limited to 5. All were CS (Computer Science) majors, and only one was female. Although the results seemed to show enhanced performance from the new design, it may be reliably extended only to CS-major male graduate students, covering a very small segment of society. In other words, condition 3 is not satisfied.

As discussed above, none of the four strategies widely-used for testing can guarantee to satisfy all of our criteria. In order to conduct effective and reliable experiments, we need a new protocol which can meet all of these criteria.

Chapter 3

Methodology

3.1 Familiarity Issues

There are two main kinds of familiarity issue in comparing keyboard layouts. First, participants may have different levels of recollection of where each key is located. An expert user can immediately and unconsciously locate each key to be typed. A medium-experience user may be able to recollect where each key is located quickly, and be able to find it in around a second. A novice may have little or no idea about the layout, and so waste a long time just looking for each key. Such variation is common for the current standard layouts such as QWERTY. For a new design, on the other hand, no one knows where to find target characters.

If we are to undertake fair comparisons, we need to remove this gap, and to standardize the level of recollection to a higher level, because keyboard efficiency is only highly relevant when it is based on sufficiently trained participants. Of course, layouts may have different levels of learnability, taking different amounts of time for people to become expert. This issue is beyond the scope of this paper.

A second category of familiarity is finger memory. Frequently used phrase or multigrams are easy to remember not only because of the memory of individual key locations, but because of finger memory of the sequence on the keyboard. For example, most experienced QWERTY users do not need to separately locate the ‘t’, ‘h’, and ‘e’ on the keyboard to type ‘the’ in English. The more experienced is the user, the more such words may be remembered through finger memory of the sequence. Thus we also need to eliminate bias due to this kind of memory.

To deal with such familiarity issues, and to remove the bias effect, there are two alternatives: standardization to the higher level of an experienced user (equilibrating the less-known layout to the memory level of the familiar layout); or to the lower level of a novice, (penalizing the memory of the expert down to the same level as the new user). For the first disparity, memory of where the characters lie on the keyboard, we propose a compensation approach, applying a character mapping method which transfers the user’s knowledge of the familiar layout to the new layout. For the latter problem, finger memory, we use a penalty strategy, eliminating any benefit from finger memory from both layouts.

3.2 Handling Location Memory through Character Mapping

In order to estimate the expected time required to correctly type a given text, we need to remove the unnecessary delay of looking for keys. It is not difficult to find experimental participants who are sufficiently familiar with the current layout, but it is practically infeasible to bring them to the same level of familiarity with a new design in a short period of time. Thus, we need a methodology to transfer their proficiency with the current keyboard to the new design. We detail the proposed methodology here, but note that this is just a general description, so it may need further adaptation to specific cases. To explain the method, we need to define some terminology. First, we need to assign a unique number to each key, the keystroke code. This can be coded in various ways, depending on the physical device. For a computer keyboard, an integer value may be assigned to each of around 100 keys.

Formally, we define the function

$$f_{\text{layout}}(l) : L \to C \tag{3.1}$$

which maps a letter l to a keystroke code in the given layout, where L is the set of letters in the target language and C is the set of keystroke codes. This mapping f_{layout} is bijective: that is, each alphabet letter in the target language L corresponds to exactly one keystroke in C. It therefore has an inverse bijection

$$f^{-1}_{\text{layout}}(c) : C \to L \tag{3.2}$$

which, conversely, maps a key code c to the corresponding character in the target language, for the specific layout.

Through iterative application, f_{layout} specifies the sequence of keystrokes corresponding to a character string, and f^{-1}_{layout} specifies the character string that a given sequence of keystrokes will produce. Since no ambiguity will result, we use the same notation (f and f^{-1}) to denote mappings between strings of characters and the corresponding sequences of keystrokes.

Suppose we have an 'old' layout and a 'new' one that we wish to compare; we assume that our experiment participants are familiar with the 'old', in the sense that they know f_{old} instinctively – that given a string s, they can immediately type $f_{\text{old}}(s)$. On the other hand, they cannot immediately find $f_{\text{new}}(s)$. Of course, this leads to unfair comparisons. But suppose that, instead, we ask them to type $f^{-1}_{\text{old}}(f_{\text{new}}(s))$ using the old keyboard. That is, they are asked to apply the mapping f_{old} (which they know) to the string $f^{-1}_{\text{old}}(f_{\text{new}}(s))$. What they will actually do is to produce the string

$$f_{\text{old}}(f^{-1}_{\text{old}}(f_{\text{new}}(s)) = f_{\text{new}}(s)$$

which is what we desired – but using their current knowledge of the 'old' layout.

To make this more concrete, assume that we want to compare Dvorak layout (assuming that this is a generally unfamiliar layout) with familiar QWERTY layout, using "computer" as the test word. We may use this string "computer" directly to measure speed in QWERTY layout, but we need the transformation described above for Dvorak. Thus we apply the nested function:

$$f^{-1}_{\text{qwerty}}(f_{\text{dvorak}}(\text{"computer"})) \tag{3.3}$$

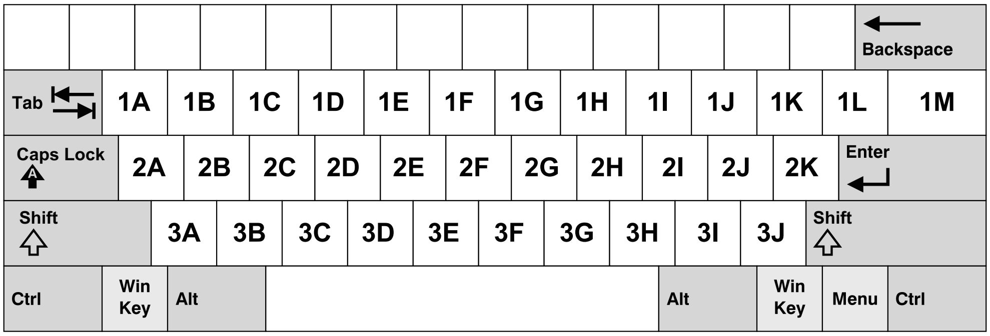

Figure 3.1 Example of Key Coding

using, for example, the key coding shown in Figure 3.1, and the layouts shown in Figure 3.2. f_{dvorak}(”computer”) returns the sequence of <1H, 2B, 3G, 1D, 2D, 2H, 2C, 1I>. Applying f^{-1}_{qwerty} then returns the string “ismrfkdo”. So measuring the time to type “ismrfkdo” using QWERTY layout tells us about the physical time required to type “computer” in Dvorak layout – because they are exactly the same physical sequences – without requiring our participants to know Dvorak. Of course, there are remaining issues relating to the relative familiarity of “computer” and “ismrfkdo”, and finger memory of sequences such as “put” but not “rfk”; we deal with these issues next.

3.3 Removing Finger Memory Bias using Layout Transformation

It seems unfair to compare typing speeds for “computer” and “ismrfkdo” directly, because of the greater familiarity of “computer” than “ismrfkdo”. As we can easily recognize the word “computer”, and as we have a lot of experience in typing “computer” in our familiar layout, it is likely to be fast. For “ismrfkdo”,

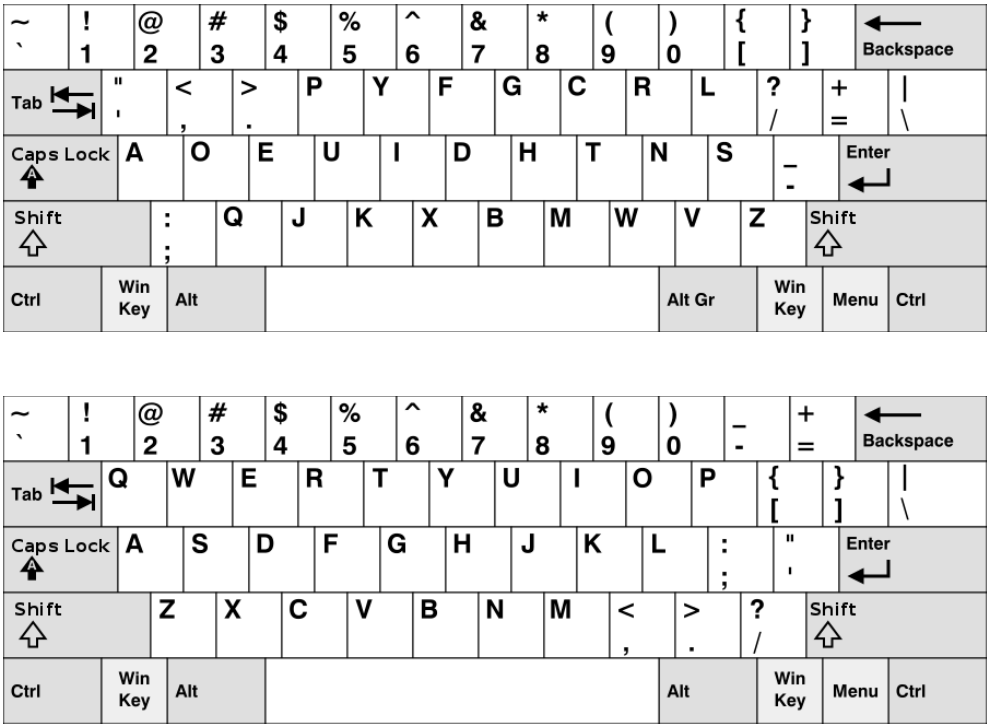

Figure 3.2 Character Mapping Example: Dvorak and QWERTY

however, there will be some recognition time to confirm its spelling and map it to finger movement. Even if the word "ismrfkdo" is the result of transformation from a wonderful keyboard layout, far faster than QWERTY, the result of this experiment may not demonstrate this advantage. For impartial comparison, we need to remove this familiarity advantage from the more familiar layout.

To achieve this, we define a layout transformation

$$t : C \to C \tag{3.4}$$

The aim of this transformation t is to re-map the keyboard layout in such a way as to remove the effects of word familiarity and finger memory from the current layout, while not affecting the physical speed of the layout.

Each layout differs in the frequency of use of each hand, finger, or row/column. Thus [20] Dvorak [21] and Griffith [9] designed their new layouts based on the observation that QWERTY over-uses the left hand, especially the index finger. These traits should not change during the test.

In order to minimize this change, we base our transformation on symmetries of the layout: most layouts have physical symmetries that do not affect (or more accurately, only very weakly affect) typing speed. For example, it is reasonable to assume that the ultimate (post-training) typing speed of QWERTY layout would be only very weakly affected if the layout were horizontally reflected about the mid-line (of course, the accuracy of this assumption will need to be separately assessed). Why do we think so? Because the same fingers (but on the opposite hand) would be used to form the characters, and all required movements would be of the same magnitude, simply reversed in direction. That is, the symmetry (reflection about the mid-line) preserves some invariants (specific finger, movement distance) that we think determine typing speed, while varying other properties (hand, horizontal direction) that we assume have little effect (in specific cases – highly "handed" individuals, or people with specific injuries – these assumptions may not be valid, hence the corresponding conclusions will need to be suitably qualified).

More generally, we need to define a set of invariants that should be preserved, and then define a transformation – generally, using symmetries – that preserves these invariants [22]. The specific symmetries and invariants involved may depend on the specific physical keyboard and layout. For example, the full QWERTY keyboard also has a vertical symmetry, but we should probably not make use of it, because the top row (numbers) is less familiar to most users than the "letters" part of the keyboard, so that typing speed will be affected if numbers are mapped to letters and vice versa. Vertical reflection about the mid-line of the letter part of the keyboard would be feasible, but will result in no change to the middle row ("asdf..."), and thus may not completely remove familiarity bias (a small number of words, such as "glad", would be completely unchanged, and a larger number may be only slightly changed). Other fairness considerations also intrude.

3.4 Layout Transformation for QWERTY vs. Dvorak

Before applying a transformation, we need to look at both layouts. First, we discarded the top row of numbers to focus on typing speed on the alphabet keys. Therefore we used leftmost 10 keys of 3 rows for the test. We will call this area as a 'region'. And it is reasonable to exclude 1K('[' in QWERTY), 1L(‘]’), 1M(‘\’), and 2K(‘’’) keys in the QWERTY-Dvorak transformation. Only special characters are coded in those keys in both layouts. This exclusion, however, makes the layout transformation of ‘/’ paired with ‘?’ problematic. In QWERTY ‘/’ and ‘?’ are coded to the key 3J in the region, and it is okay to apply any transformation or reflection. But in Dvorak, ‘/’ or ‘?’ will be transformed to ‘[’ or ‘{’ in QWERTY which are outside of the region. If we expand the region to include 1K(‘[’ and ‘{’), it will trigger other expansions of the region because ‘[’ and ‘{’ is in the top number row in Dvorak. Therefore, we decided to exclude all ‘/’ and ‘?’ in the test set.

Another problem is quotation marks, ‘’’ and ‘"’ which are in the region of Dvorak (1A), however, not in QWERTY (2K). It is also difficult to expand the region as ‘/’ and ‘?’, but we decided to include this key in the test set. As we will present about characters’ occurrences in the 3.6, it is because quotation marks are highly used as alphabets. Quotation marks remain in QWERTY test, while they are transformed in Dvorak test.

In addition, we did not apply vertical reflection. Both layouts have four special character keys in the region. In QWERTY, those keys are in the right bottom (2J, 3H, 3I, and 3J). On the other hand, those keys are on the left side, especially left top of Dvorak (1A, 1B, 1C, and 3A). Therefore, if we apply horizontal and vertical reflection together, participants should use unfamiliar reason heavier than usual. To save each layout’s characteristic we decided to apply only a horizontal reflection with layout transformation.

After all, 2A(‘a’) maps to 2J(‘;’) and 3A(‘z’) to 3J(‘/’).

Now, to compare the two layouts using our chosen string “computer”, we

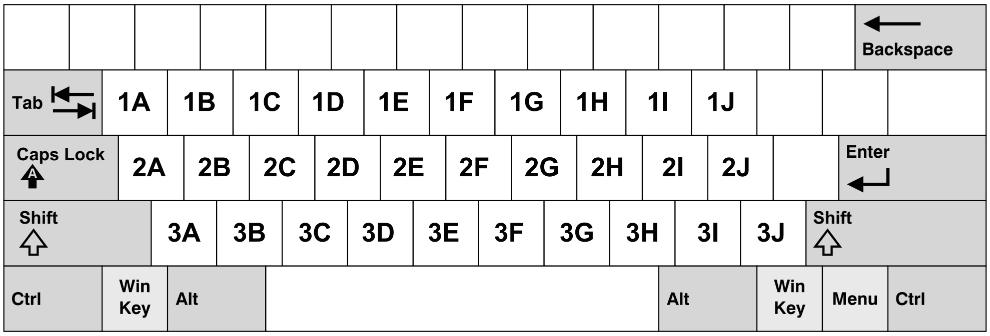

Figure 3.3 Key Coding Used for QWERTY vs. Dvorak

apply the transformation t to the requested keycodes. In our case, for testing QWERTY, we ask the user to type the string

$$f_{\text{qwerty}}^{-1}(t(f_{\text{qwerty}}(\text{"computer"}))) \tag{3.5}$$

that is, the string “,wvqryiu", while for Dvorak, we request the string

$$f_{\text{qwerty}}^{-1}(t(f_{\text{dvorak}}(\text{"computer"}))) \tag{3.6}$$

that is, the string “elvujdkw”. More abstractly, in testing the “old" layout, we use the string $f_{\text{old}}^{-1}(t(f_{\text{old}}(s)))$ and in testing the “new" layout, the string $f_{\text{old}}^{-1}(t(f_{\text{new}}(s)))$ (in all cases, we ask the user to type using the old, familiar layout).

Before starting the actual experiment, it is important to check whether the layout transformation leads to biases. These should be checked, as far as possible, at two levels:

1. Are the intended invariants (finger frequencies, row frequencies etc.) in fact preserved in the target language or corpus? (This can be checked computationally.)

2. Does the transformation significantly affect typing speed? (This requires experimental verification.)

The former is simply a matter of sampling the test corpus in both original and transformed forms, collecting statistics of the invariants, and confirming whether they are substantially changed. It is easy to do, but relies on the assumption that we have correctly identified the invariants. What if our assumptions are wrong? For example, handedness, or the cramping effect of typing on the bottom row, may be greater than we have assumed. To identify this, we need to carry out experiments, measuring the extent to which our chosen invariants preserve typing speed (specifically, we may ask our test participants to type the same phrases before and after application of the transformation). Of course, in most cases with human experiments, we won't get perfect invariance – there probably will be some effect from swapping hands or exchanging rows. When we find that typing speed is not perfectly preserved under some invariant transformation, we have three main options:

1. Where the bias is large relative to the performance differences between the layouts, we may need to discard the invariant (and the corresponding symmetries and transformations) and find new invariants that better preserve typing speed.

2. Where the bias is small relative to the performance differences, we may safely ignore it, on the basis that the performance differences could not be the result of the biases.

3. Where the resulting bias is comparable in size to the performance differ-

ences, another option is to correct the (by now, known) bias in comparing the typing speeds mathematically.

3.5 Overall Experimental Protocol

Putting all these together, we propose a protocol for comparing a new key layout for a keyboard with a more familiar "old" layout.

To illustrate, we briefly explain how to apply it in comparing Dvorak layout with QWERTY. We first prepare a set of words from an appropriate text corpus. These words need to be filtered by relevant criteria such as word length, and (depending on the application) the occurrence of special characters etc. We also need to define the letter stroke coding, and the two functions f_{QWERTY} and f_{Dvorak}. We use the mapping shown in Figure 3.1. We must also define the transformation t. Here, we make use of the left-right hand symmetry only. Thus we interchange columns A-J, B-I etc. as described earlier. We then convert each word into two test inputs. In parallel, we apply the following two nested procedures to each word:

$$f^{-1}_{\text{qwerty}}(t(f_{\text{qwerty}}(s))) \tag{3.7}$$

$$f^{-1}_{\text{qwerty}}(t(f_{\text{dvorak}}(s))) \tag{3.8}$$

3.6 Test Set Generation

The preceding explanation shows how we can transform a test corpus to remove sources of bias in comparatively evaluating the physical usability of two keyboard layouts. But where does this corpus, used to generate the test pairs, come from? What restrictions should we impose to ensure statistical stability?

The first requirement is clear: the corpus should reflect the intended use of the layout. For example, many users will use a computer keyboard primarily for typing text in their primary language. For these uses, a corpus of representative text will be most appropriate. In the preceding research, we used a corpus of SMS messages because a feature phone's keypad layout is primarily intended for typing short messages. If we test smart phones' keyboard layouts, corpora of relatively longer messages or SNS postings can be also considered. Whatever the keyboard, these considerations apply whether using the methodology proposed here, or more traditional comparison methodologies. We thus do not address the choice of underlying corpus further.

We used the Brown Corpus [23] for the test set generation. It might be considered too old because it was the first corpus issued in the 1960s. The vocabulary distribution of this corpus may not mirror the current English and of course, it does not contain newly coined words. Nonetheless, it has distinctive merits in contrast to latest corpora.

1. It collected a million words from 500 samples. It guarantees we can give participants test queries of continuous sentences. This could make participants feel like typing a paragraphs.

2. It compiled text from 15 different fields such as press, religion, and popular lore. This characteristic can prevent a participant from typing most of the sentences from familiar topics.

3. All text was written by native speakers of American English.

4. It also has a Part-of-Speech (PoS) tagged version. We can easily exclude unnecessary components, for example, special characters. It makes preprocessing easy.

From the tagged corpus, we removed PoS tags and the characters located outside of the keyboard's region except ‘’’ and ‘"’. Then 25 test queries from 30 characters to 100 characters long are equally sampled from the corpus. Among 25 queries, first 5 queries are presented without any transformation to test a participant's actual typing speed on QWERTY keyboard. Next 20 queries, the half will be transformed from Dvorak to QWERTY, then reflected horizontally. The other half will be just horizontally reflected to test the hypothesis that this reflection does not affect the typing speed much.

3.7 Web-based Experiment

In this experiment, we want to find out which layout is ultimately faster than the other. Thus our attention is mainly focused on typing of very fast typists. And we also assumed that skilled typists can recollect positions of keys on their keyboards. This is why we do not control the experimental environment tightly. Using the same machine and the same keyboard is not effective to collect ultimate speeds of typists. Unless the layout is different from the standard, allowing to use one's own keyboard can help reduce the time for moving fingers and clicking. The size of key caps and the depth of the keys can affect the speed of typing. That effect can be more influential for skilled typists. This experiment is not for testing a keyboard. Because we need the best and optimal performances on the familiar layout, it is reasonable to let participants decide their typing environment.

For the convenient deployment of the testing program, we built a website for this experiment. Since we decided not to restrict typing environments, we did not need to invite people to participate in our experiments. Web technologies fitted to this situation. Instead of building, transferring, and installing a program, it just required to send an address of the website. It has enabled us to recruit people regardless of the platform, location, and time. And it has also helped us collect results easily.

Another important advantage of the web-based system was safety. Anyone can start or quit an experiment when they want. And there are no external factors affecting it. For example, people do not feel any pressure or burden for participating because no one can observe the experiment. Our system only collects the minimum personal information which is not identifiable and related with typing speed. Hence it is almost impossible to check the specific person's participation.

Chapter 4

Results

4.1 Distribution of Test Set

We analyzed the Brown Corpus to confirm our experimental design. Total 56,936 sentences, 4,939,148 characters excluding 966,650 white spaces were in the corpus. 4.1 supports the claim that Dvorak placed most frequently used keys in the home (middle) row. It is worth noticing that, according to 4.1, comma, period, double quotes, dash, single quote, and semi-colon are relatively used many times. Except the dash key only, those special characters are included in the keyboard region.

Basically, we generated test queries whose lengths are from 30 to 100 characters long. But 4.2 shows there are lots of sentences in the corpus shorter than 30 characters. One of our design's goal is to give participants feeling like typing a real paragraph. To achieve this goal and not to waste 6,973 (12.25%) short sentences, we concatenated continuous sentences to make a longer test query,

Table 4.1 Top 10 Most Appeared Characters in the Brown Corpus

Character	Occurrences	Percentage	Location	
			QWERTY	Dvorak
e	589,538	9.982%	1C	2C
t	423,037	7.163%	1E	2H
a	371,150	6.285%	2A	2A
o	356,706	6.040%	1I	2B
i	332,945	5.638%	1H	2E
n	332,700	5.633%	3F	2I
s	300,223	5.084%	2B	2J
r	287,104	4.861%	1D	1I
h	249,023	4.217%	2F	2G
l	192,778	3.264%	2I	1J

if a sampled sentence is shorter than 30 characters. Conversely, if a sampled sentence is longer than 100 characters, we extracted a substring randomly. It applied also for concatenated queries.

4.2 Research Participant Recruitment

Before we start the recruitment process, we had submitted the detail of our research to the Institutional Review Board at Seoul National University and it was carefully reviewed by the committee. This master's thesis had tried small number of experiments as a pilot study. We tested our method and system before starting the actual crowd sourcing and we could have figured out minor

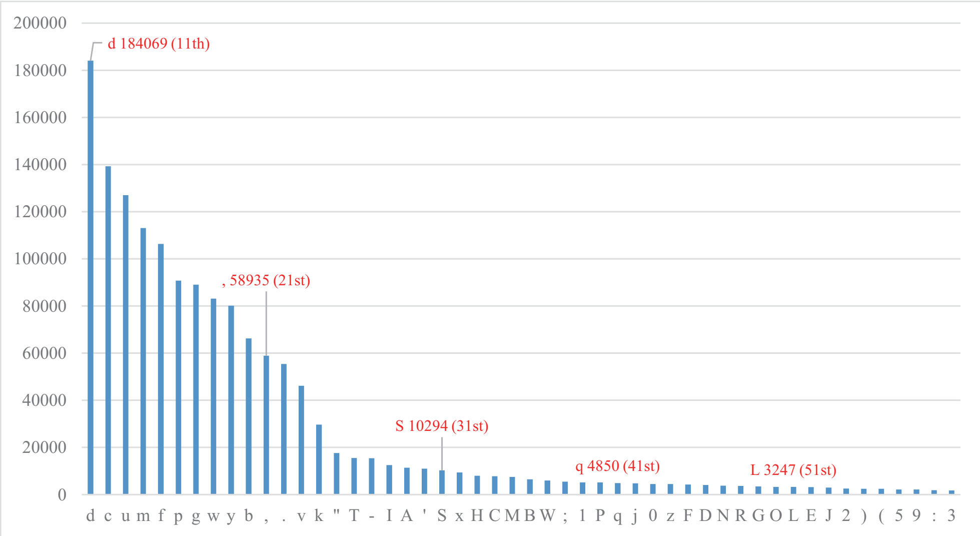

Figure 4.1 Occurrences of characters except Top 10

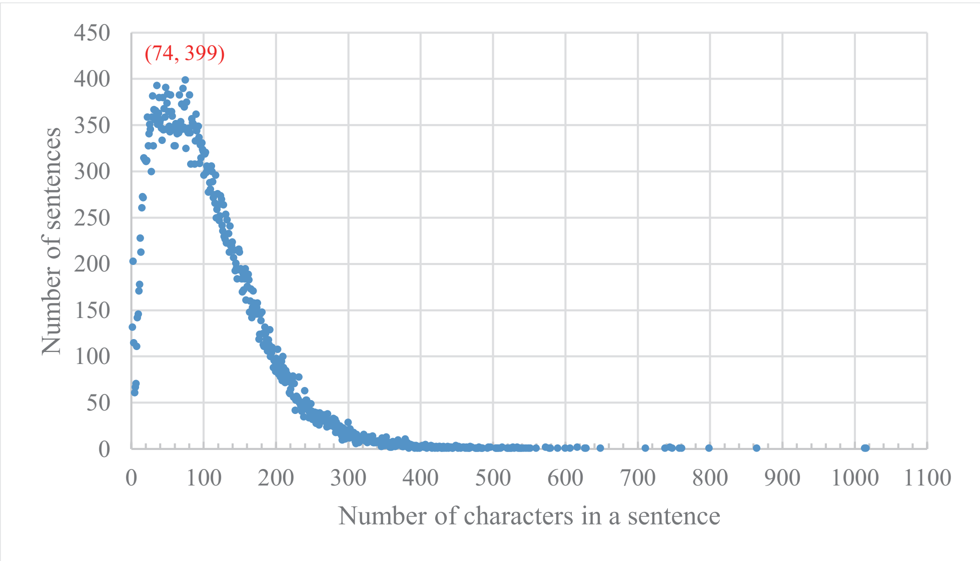

Figure 4.2 Distribution of sentence lengths

problems of our system and improved it. We just recruited total 15 research participants for this purpose.

We collected basic information from research participants. And their distribution is:

- Gender: 9 were male and 6 were female. There was no special tendency of the result from the gender.
- Age: The average was 30 and the standard deviation was 9.83. Most participants were in their twenties.
- Major: About a half was computer related major and their result were relatively higher.
- Native Language: Most of the participants were native Korean speakers.
- Occupation: Over two thirds belonged to universities. Others were office jobs.

Generally, our participants can be summarized as a native Korean in the late twenties and familiar with computers. Except computer related majors, there were no specific trend in our result data.

4.3 Overall Results

It can be roughly said that these results support the Dvorak layout's efficiency. As expected, typing speeds on the untransformed QWERTY was faster than the others except only one participant. And horizontal reflections made typing speeds slower in half. It was more drastic than we expected because we

Table 4.2 Typing Speeds and Accuracies Ordered by Original Speeds

Participant	Speed			Accuracy		
	Original	QWERTY	Dvorak	Original	QWERTY	Dvorak
1	503.569	237.924	239.337	98.85%	94.93%	96.65%
2	465.367	234.254	246.092	99.34%	96.87%	98.91%
3	362.813	222.432	211.612	96.59%	95.51%	90.94%
4	359.762	212.107	197.502	97.81%	93.79%	94.35%
5	333.946	132.553	153.540	97.52%	92.65%	94.70%
6	311.879	169.038	181.454	97.61%	89.63%	93.31%
7	311.713	166.221	172.065	98.86%	96.12%	96.53%
8	307.863	179.122	202.246	97.23%	92.99%	95.69%
9	306.175	138.703	141.422	98.60%	97.88%	98.37%
10	288.536	161.508	172.778	84.49%	96.46%	95.20%
11	242.461	104.254	118.160	96.50%	86.31%	90.04%
12	219.941	146.586	164.108	89.25%	87.35%	91.13%
13	203.579	106.474	103.063	97.59%	98.77%	96.53%
14	199.609	110.877	127.857	94.01%	91.15%	92.73%
15	144.877	126.032	145.411	92.51%	88.59%	93.23%
Average	304.139	163.206	171.776	95.78%	93.27%	94.55%
SD	96.683	45.872	42.166	4.15%	3.91%	2.66%

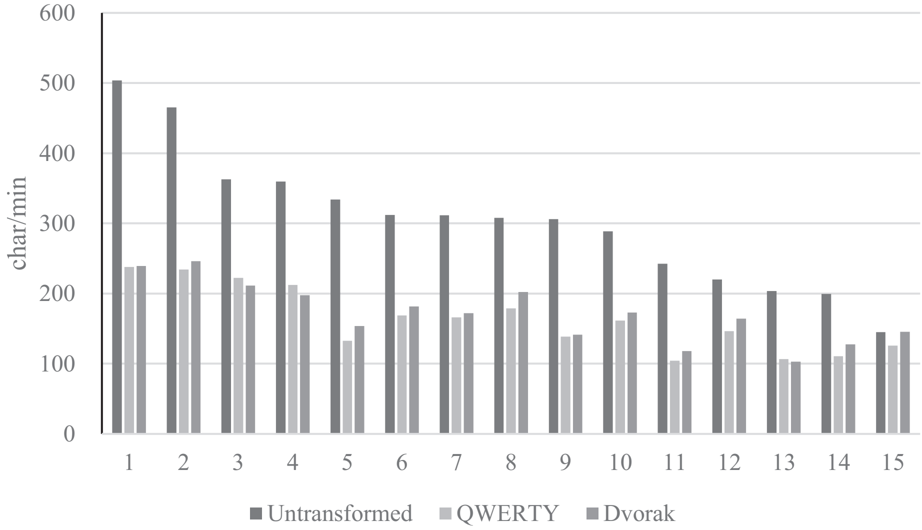

Figure 4.3 Typing Speeds on Different Layouts

thought as more skilled a participant is, the effect of reflections would be more insignificant. The effect of reflections in accuracy, however, was relatively low. The accuracy was probably more dependent on the use of delete keys, not on reflections. We need to investigate the influences of disabled delete keys by A/B testing in a follow-up study.

The difference of accuracy was not that conspicuous. In both transformed experiments, reduced amounts of accuracy were small. This can be interpreted our experiments were done in the expertise level.

That 15th participant's result was exceptional because the original speed on QWERTY was very slow and there were no significant differences of speeds on both layouts. The recollection speed of this participant might be good considering his results on transformed layouts. This result might be caused

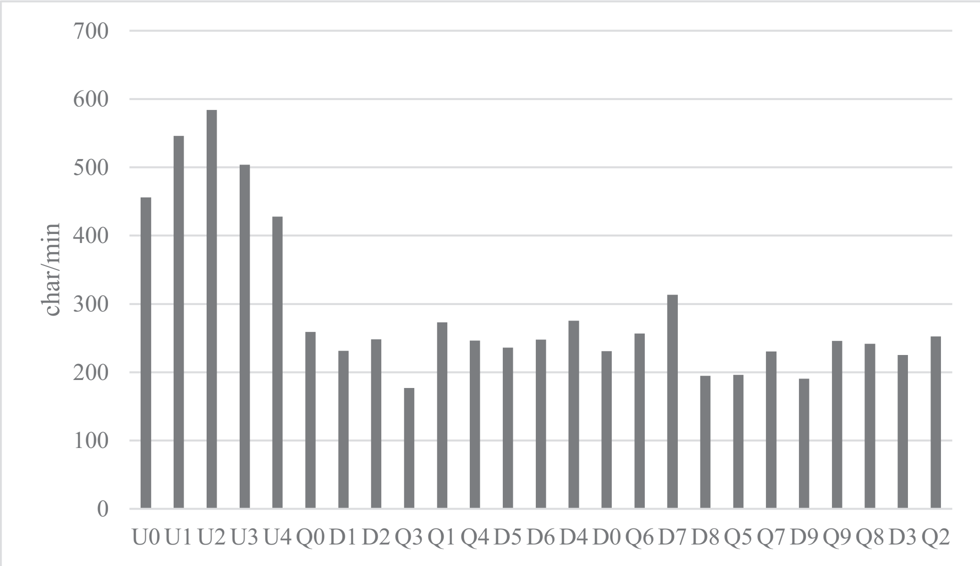

Figure 4.4 Typing Speed of the Fastest Participant

(U: Untransformed, Q: QWERTY, D: Dvorak)

from the problems in typing environment. We can analyze this participant typing data key by key to figure out reasons later.

When we see the typing speed of the fastest typist in our experiment group 4.4, his typing speed was relatively steady in each layout. It is clear he typed almost twice as fast in the first five untransformed queries. Then speed became slower in the next twenty transformed queries. But it was not varied considerably as the test progressed. It might be the evidence the whole experiment's fatigue is not that significant.

4.5 shows typing speeds of originally same text. We sampled 10 strings from the corpus and generated two transformed strings for each. This fastest typist had recorded steady speed on both layouts. As we saw in 4.4, this result was

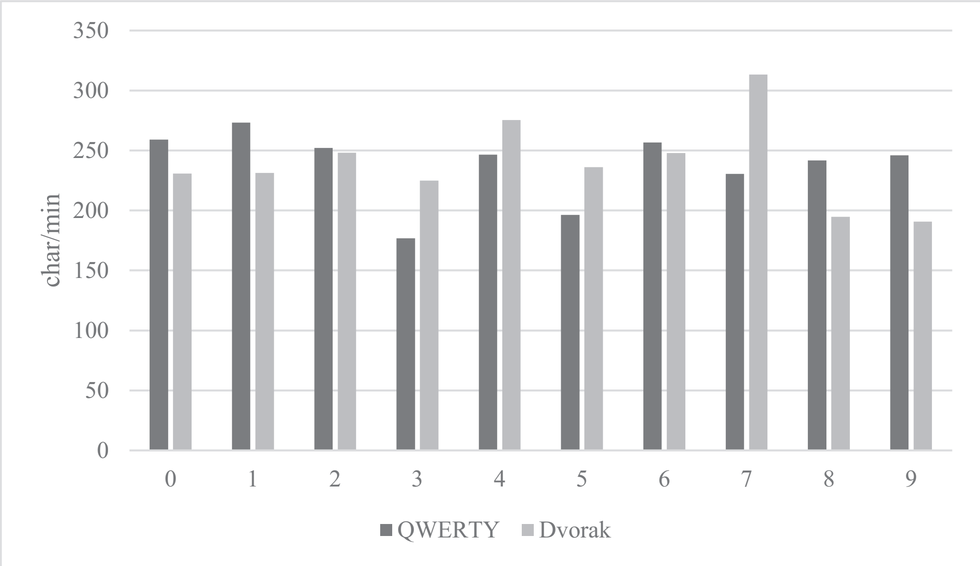

Figure 4.5 Difference of Typing Speed of Same Text

not affected by the total order because his typing speed was not diminished in later queries. Therefore, it can be said that the layout difference could not make a noticeable difference in typing speed.

4.4 Key-level Analysis

4.3 shows the 10 fastest transitions in our experiments. We collected total 16,189 key transitions including special characters including Space bar and Enter key. It means our participants totally typed this amount of keys. Then there were 151 transitions occurred over 30 times. As we can see, generally transitions like English took shorter time. For example, non-English transition from 'l' to 'j' took 479 ms. It is $3\tilde{4}$ times longer than those 10 faster transitions. If we can collect much larger data of transitions, we could compensate this

Table 4.3 Fastest Transition Times between Keys

Transition	Time (ms)
t-h	106.616
r-e	122.946
n-g	127.526
n-d	127.735
e-n	137.625
a-n	141.094
e-r	146.431
i-n	148.667
t-i	159.306
o-n	160.625

effect from finger memory for fairer experiments.

Chapter 5

Discussion

The principal aim of the new screening methodology is to measure the differences in expected physically-limited typing performances between layouts, but to do so with lower per-participant experimental effort, enabling a much wider sample of participants to be used, and thus eliminating potential major sources of bias.

In preceding research [24], we compared feature phone's keypad layouts and discussed the usefulness of the new methodology. In this research, we applied this methodology to long-time controversial keyboard layout comparison. We expected Dvorak might be more effective because it places most frequently used keys in the home row of the keyboard in the corpus analysis. Then we observed that the typing performance on Dvorak is slightly better than that of QWERTY in this methodology although our experimental group is rather small. Though there are some issues we still need to investigate in this comparison and the methodology.

5.1 Effect of Reflections

First of all, we found that reflections cut off typing speed about in half. We experienced similar decreases in the previous experiments. It mainly came from the difference of cognitive loads. Unlike other approaches asking to type real English strings, the test queries look random in our method. During the "character recognition" phase – the first phase in the process of typing [25] – the eye does not read with its maximum achievable speed (the required speed for comprehension). Instead, the eye reads the characters just fast enough to feed the copy to the hand as it is needed [26]. In the next phase, remembering the characters to be typed in short-term memory, we can buffer just 4 to 8 letters, preventing further look-ahead [27]. Overall, randomly appearing strings may be much more difficult to remember because of unfamiliarity.

In this keyboard layout comparison, we gave participants much longer sentences to type. It might increase the influence of the difference in cognitive loads. In addition, typing environment is definitely different in mobile devices and computers. Typing speed on the computer is much faster than in the feature phone. The short-term memory problem could be more critical in faster typing because it could shorten the mechanical typing period. The longer distance between the eyes and the screen also can be a reason.

We could see how much reflection affect typing speed in this research. It is possible to test the effect solely on Drovak layout by not applying reflections after transforming Dvorak into QWERTY. And we might conduct this experiment in mobile devices which support two-thumb typing. In this, we can see the effect of the distance between eyes and an input device.

5.2 Statistical Issue

Another obvious issue is the statistical stability. The results were based on a sample of only 15, apparently too small for statistical reliability (an almost universal problem with this kind of experiment). Moreover many participants majored computer science or closely related field of study, whose performance may be atypical.

In addition, to test ultimate speed on keyboard, it is required to observe participants who can type the maximum speed on QWERTY. Generally, we assume a word is 5 characters long when we say words per minute (WPM). The world record typing speed is 150 WPM. It means it is about 750 characters per minute. According to the [3], this world record is achieved even on Dvorak. Thus it might be difficult to observe this high speed on QWERTY. Therefore we need to handle this insufficiency in the future. By collecting statistically significant size of experimental result, it would be possible to estimate the highest speed on QWERTY.

Chapter 6

Conclusion

6.1 Cautious Forecast: Dvorak's Slight Win is Not a Victory

Dvorak's excellence over QWERTY in design is widely known and statistically reasonable as we saw in 4.1. In this research, we want to compare these two layouts in the practical typing environment. Thanks to the advantages of the new screening methodology, it was able to compare two layouts at the expertise level of the familiar layout. And the cost of experiments was very low considering other similar research in this field. Moreover, by the web based experiment, participants can use their own typing environment and their familiarity to the layout could be maximized.

We observed some evidences supporting Dvorak was slightly faster, although experiments were conducted in a small number of participants. However, it is difficult to conclude this 5.25% faster speed is meaningful. Even if we had supposed horizontal reflection of keyboard layout cut down typing speed in about half, Dvorak's advancement stayed in around 10%. Assuming a person type 1,000 words a day, typing will solely take about 15 minutes a day. If that person can save 10% of typing time by changing to Dvorak, he or she can save 1.5 minutes a day, 45 minutes a month, and finally 9 hours a year. It means this time saving might be overwhelmed by the cost of changing the layout. That is why we should investigate factors not covered in this pilot study.

6.2 Assumptions and Limitations

The most important assumption in this pilot study was the insignificant effect of symmetric reflections. It had been also questioned in preceding research and it clearly exposed at this time. We might be able to test only Dvorak-to-QWERTY transform additionally to check the actual effect of reflections in Dvorak.

Next, we assumed the typing in predefined region is enough to test ultimate speed of layout. Since numbers and special characters did not appear many times in the Brown corpus, this assumption might be reasonable for this research. On the other hand, if those characters' occurrences were limited, it might be also reasonable to include those in the test set. QWERTY and Dvorak share same number keys, including special characters over numbers. It is needless to say for other function keys, such as Enter, Space bar, or Shift. Maybe testing full keyboard would reflect real typing faithfully.

6.3 Epilogue

These days, the cost of changing the layout is continuously diminishing. The period for learning new layout is the only big thing. Other points are already considered by good software. We can 'softly change our devices' layout and even simulation of other layout is available. Therefore, using 'better' layout only depends on our decision.

By the way, when it comes to education, it might be different. If a young child could learn other layout than QWERTY from the start, there is no cost to save hundreds or even many hours in his or her life. It might be easier to learn other layouts because they were carefully designed considering the language's characteristics. No one can explain why we lay our index fingers on 'f' and 'j'.

Our methodology can help finding the best keyboard layout economically. Applicability is one of our methodology's powerful advantages. And it is easy to add or change test queries in the web based experiment system. This flexibility is useful to analyze characteristics of typing on a specific layout.

Bibliography

[1] S. J. Liebowitz and S. E. Margolis, "The fable of the keys," *Journal of Law and Economics*, pp. 1–25, 1990.

[2] R. C. Cassingham, *The Dvorak keyboard: The ergonomically designed typewriter keyboard now an American standard.* Freelance Communications, 1986.

[3] R. Ranger, "Barbara blackburn, the world's fastest typist," 2013. [Online]. Available: http://rcranger.mysite.syr.edu/famhist/blackburn.htm

[4] I. S. MacKenzie, R. B. Nonnecke, J. C. McQueen, S. Riddersma, and M. Meltz, "A comparison of three methods of character entry on pen-based computers," *Human Factors and Ergonomics Society Annual Meeting Proceedings*, vol. 38, no. 5, pp. 330–334, 1994. [Online]. Available: http://www.ingentaconnect.com/content/hfes/hfproc/1994/00000038/00000004/art00030

[5] J. R. Lewis, M. J. LaLomia, and P. J. Kennedy, "Evaluation of typing key layouts for stylus input," *Human Factors and Ergonomics*

Society Annual Meeting Proceedings, vol. 43, no. 5, pp. 420–424, 1999. [Online]. Available: http://www.ingentaconnect.com/content/hfes/hfproc/1999/00000043/00000005/art00006

[6] N. Green, J. Kruger, C. Faldu, and R. St. Amant, "A reduced qwerty keyboard for mobile text entry," in *CHI '04: CHI '04 extended abstracts on Human factors in computing systems.* New York, NY, USA: ACM, 2004, pp. 1429–1432.

[7] A. Mittal and A. Sengupta, "Improvised layout of keypad entry system for mobile phones," *Computer Standards and Interfaces*, vol. 31, no. 4, pp. 693–698, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/B6TYV-4TTMNDB-2/2/3e6f06007e9e334e2ae8778d3f00907a

[8] R. S. Hirsch, "A human factors comparison of two keyboards, proposed for special purpose and limited application," *IBM Report No. RJ151*, 1958.

[9] R. T. Griffith, "The minimotion typewriter keyboard," *Journal of the Franklin Institute*, vol. 248, pp. 399–436, 1949.

[10] J. Harnett, "Qwerty lives another day," *Office Equipment News*, pp. 15–57, 1972.

[11] L. Butts and A. Cockburn, "An evaluation of mobile phone text input methods," *Aust. Comput. Sci. Commun.*, vol. 24, no. 4, pp. 55–59, 2002.

[12] J. Gong and P. Tarasewich, "Alphabetically constrained keypad designs for text entry on mobile devices," in *CHI '05: Proceedings of the SIGCHI*

conference on Human factors in computing systems. New York, NY, USA: ACM, 2005, pp. 211–220.

[13] E. P. Strong, *A Comparative Experiment in Simplified Keyboard Retraining and Standard Keyboard Supplementary Training.* Washington D.C, USA: US. General Services Administration, 1956.

[14] S. E. Michaels, “Qwerty versus alphabetic keyboards as a function of typing skill,” *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 13, no. 8, pp. 419–426, 1971. [Online]. Available: http://www.ingentaconnect.com/content/hfes/hf/1971/00000013/00000005/art00004

[15] B. Thomas, S. Tyerman, and K. Grimmer, “Evaluation of three input mechanisms for wearable computers,” in *Proceedings of the IEEE International Symposium on Wearable Computers.* Los Alamitos, CA, USA: IEEE Computer Society, 1997, pp. 2–9.

[16] I. S. MacKenzie and S. X. Zhang, “The design and evaluation of a high-performance soft keyboard,” in *CHI ’99: Proceedings of the SIGCHI conference on Human factors in computing systems.* New York, NY, USA: ACM, 1999, pp. 25–31.

[17] M. Ingmarsson, D. Dinka, and S. Zhai, “Tnt: a numeric keypad based text input method,” in *CHI ’04: Proceedings of the SIGCHI conference on Human factors in computing systems.* New York, NY, USA: ACM, 2004, pp. 639–646.

[18] S. Hwang and G. Lee, "Qwerty-like 3x4 keypad layouts for mobile phone," in *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*. New York, NY, USA: ACM, 2005, pp. 1479–1482.

[19] J. Lee and R. I. B. McKay, "Optimizing a personalized multigram cellphone keypad," Seoul National University, Seoul, Republic of Korea, Tech. Rep. TRSNUSC:2010:001, 2010.

[20] D. Ferguson and J. Duncan, "Keyboard design and operating posture," *Ergonomics*, vol. 17, no. 6, pp. 731–744, 1974.

[21] A. Dvorak, "There is a better typewriter keyboard," *National Business Education Quarterly*, vol. 12, no. 2, pp. 51–88, 1943.

[22] P. Olver, *Equivalence, invariants, and symmetry*. Cambridge, UK: Cambridge University Press, 1995.

[23] W. N. Francis and H. Kucera, "Brown corpus manual," *Brown University Department of Linguistics*, 1979.

[24] J. Lee, H. Cho, and R. I. B. McKay, "Rapid screening of keyboard layouts," in *Proc. of the Conference on Systems, Man and Cybernetics*, 2012.

[25] e. William E. Cooper, *Cognitive Aspects of Skilled Typewriting*. New York, USA: Springer-Verlag, 1983.

[26] R. Butsch, "Eye movements and the eye-hand span in typewriting," *Journal of Educational Psychology*, vol. 23, no. 2, pp. 104–121, 1932. [Online]. Available: http://www.sciencedirect.com/science/article/B6WYD-4NKXYRX-4/2/742f96ffc5f15f9ef7affea822d72cf4

[27] P. Buzing, "Comparing different keyboard layouts: aspects of qwerty, dvorak and alphabetical keyboards," Delft University of Technology, Delft, Netherlands, Tech. Rep. 60, 2003.

요약

사실상의 표준 영문 자판인 쿼티 자판은 효율적인 영어 입력 보다는 타자기의 고장을 줄이기 위해 만들어졌다. 반면 가장 빠른 자판으로 알려진 드보락 자판은 영어의 사용 특성을 고려해 최적의 타자가 가능하도록 고안되었다. 그럼에도 불구하고 오랜 시간 동안 어느 자판 배열도 상대적 우수함을 확실하게 보여주지 못 하고 있다. 이는 근본적으로 여러 자판 배열에 동등하게 숙련된 사람을 찾기가 불가능하기 때문이다. 지금까지의 수많은 타자 속도 비교 연구들은 이러한 익숙함의 차이에서 오는 결과의 편향을 효과적으로 제거하지 못하였다.

본 논문은 익숙한 자판에서의 실험을 통해 새로운 자판에서의 궁극적인 속도를 예측하는 방법론을 활용하여 쿼티와 드보락 자판을 비교한다. 이 방법론은 사용자의 쿼티 자판 배열에 대한 익숙함을 활용하기 때문에 결과에 자연스럽게 편향이 발생하나, 이러한 편향은 쉽게 측정되어 보정 가능하다. 이 방법론의 간편함과 효율성 덕택에 적은 비용으로 대규모 타자 관련 자료를 수집할 수 있는 웹 기반 시스템을 구축할 수 있었으며, 영문 타자 입력 시 발생하는 특징적인 현상들을 볼 수 있었다. 수집된 실험 결과들은 자판 배열의 우수함을 평가하는 것을 넘어, 여러 가지 형태의 컴퓨팅 기기, 특히 많은 모바일, 웨어러블 기기들이 개발되고 있는 시점에 새로운 효율적인 입력 기기를 개발하는 데에 도움이 될 것으로 기대한다.

주요어: 자판, 배열, 속도, 비교, 쿼티, 드보락

학번: 2012-23240