

Engram: a systematic approach to optimize keyboard layouts for touch typing, with example for the English language

Arno Klein¹, PhD

¹MATTER Lab, Child Mind Institute, New York, NY, United States

Corresponding Author:

Arno Klein, PhD

MATTER Lab

Child Mind Institute

101 E 56th St

NY, NY 10022

United States

Phone: 1-347-577-2091

Email: arno@childmind.org

ORCID ID: 0000-0002-0707-2889

Abstract

Most computer keyboard layouts (mappings of characters to keys) do not reflect the ergonomics of the human hand, resulting in preventable repetitive strain injuries. We present a set of ergonomics principles relevant to touch typing, introduce a scoring model that encodes these principles, and outline a systematic approach for developing optimized keyboard layouts in any language based on this scoring model coupled with character-pair frequencies. We then create a keyboard layout optimized for touch typing in English by constraining key assignments to reduce lateral finger movements and enforce easy access to high-frequency letters and letter pairs, applying open source software to generate millions of layouts, and evaluating them based on Google's N-gram data. We use two independent scoring methods to compare the resulting Engram layout against 10 other prominent keyboard layouts based on a variety of publicly available text sources. The Engram layout scores consistently higher than other keyboard layouts.

Keywords

keyboard layout, optimized keyboard layout, keyboard, touch typing, bigram frequencies

Practitioner summary

Most keyboard layouts (character mappings) have questionable ergonomics foundations, resulting in preventable repetitive strain injuries. We introduce ergonomics principles for touch typing and a system for developing optimized keyboard layouts in any language. We created the Engram layout for touch typing in English, which scores consistently higher than other layouts.

1. Introduction

Almost every conceivable way for a human to enter information into a computer has been attempted: manipulation of peripheral devices of all kinds, motion and gesture tracking, voice commands, dictation, even brain-computer interfaces. Through it all the humble keyboard remains the principal human-computer interface, dominated in form by the conventional, flat, diagonal array of keys that come with computers, laptops, and accessed virtually on mobile device screens. These conventional keyboards are not optimized to accommodate the ergodynamics of the human hand or upper body (Ferguson and Duncan, 1974; Serina, et al., 1999; Amell and Kumar, 2000; Fagarasanu and Kumar, 2003; Garza et al., 2012). Less conventional physical keyboard designs reposition the keys or are shaped in ways intended to reduce the strain of repetitive typing. For

example, some keyboards are split into left- and right-hand sides and angled to reduce bending of the wrists (Zipp et al., 1983; Çakir, 1995; Zecevic et al., 2000; Rempel et al., 2007; Rempel et al., 2009), are rounded to conform to the shape of the hand (Gerard et al., 1994; McLoone et al., 2009), position high-access keys in the middle for easy reach by the thumbs (Gerard et al., 1994), and arrange keys into perpendicular rows and columns to reduce diagonal finger movements (see Figure 2). Some also permit a choice of key switches, whose force–displacement characteristics can impact strain and fatigue (Rose, 1991; Rempel et al., 1997; Radwin et al., 1999; Bufton et al., 2006; Lee et al., 2009). The Kinesis Advantage (Gerard et al., 1994) and the Ergodox are two examples of commercial keyboard designs that also permit remapping of characters to individual keys, and therefore enable completely customizable keyboard layouts.

Since the vast majority of people simply use the keyboard bundled with their computer or physically integrated into their laptop, adopting a better keyboard layout has the greatest potential to significantly improve comfort and reduce strain for the greatest number of people who do touch typing. And to counter the resigned statement “It’s too difficult for people to switch keyboards,” it is important to recognize: (1) there are millions of people for whom it would not be a switch, including every new generation, (2) many languages do not yet have a well-established keyboard layout, and (3) people who suffer or do not wish to suffer repetitive strain injuries from typing but need to type have vested interest in improving the ergonomics in their lives.

Developing an optimal keyboard layout is another challenge altogether. There are over four hundred septillion, or four hundred trillion trillion ($26! = 403,291,461,126,605,635,584,000,000$, or $4.03 \text{ E}+26$) possible ways to arrange a sequence of 26 letters, an NP-complete problem that is currently computationally intractable to optimize. Attempts to solve the “keyboard arrangement problem” have been ongoing (Ladany, 1975), with more recent contenders applying simulated annealing (Light and Anderson, 1993; Krzywinski, 2015; Salvo et al. 2016), ant colony optimization (Eggers et al., 2003; Wagner et al., 2003), Cyber Swarm (Yin and Su, 2011), and genetic algorithms (Deshwal and Deb, 2003; Walker, 2003; Goettl et al., 2005, Deshwal and Deb, 2006; Malas et al., 2008; Khorshid et al. 2010; Onsorodi and Korhan, 2020; Pacheco et al. 2020).

1.1. Prominent keyboard layouts

Since the invention of the typewriter, the Sholes (“Qwerty”) keyboard layout has been the preeminent layout, despite being generally acknowledged as inferior to many of its competitors (Amell and Kumar, 2000). For a thorough account of the history of keyboard layouts, see (Martin, 1981; Noyes, 1983). The Qwerty layout and its cousins were designed without apparent regard for efficiency to touch type common bigrams (two-letter sequences) in English. New layouts have been introduced to allow for more efficient touch typing in English over the last century (see Table 1), but most did not attempt to optimize their layouts. Indeed, for the sake of making it easier to switch from touch typing with a Qwerty layout, most retain elements of Qwerty to make them more efficient for Qwerty users to learn. Exceptions include layouts generated by the CarpalX’s simulated annealing of typing effort models (Krzywinski, 2015) and Halmak’s genetic algorithm (Nemshilov, 2016), both of which are included in our analysis. In the present study, we are not concerned with what a user has already learned and used in the past, but focus on designing new, optimal keyboard layouts for standard and orthonormal keyboard designs. Our principal concern is comfort, but we hypothesize that careful, systematic choices that account for basic ergonomics principles of finger length, strength, and movement that reduce strain can also improve efficiency, in terms of typing speed and accuracy.

Layout	Year	Website
Engram	2021	https://engram.dev
Halmak 2.2	2016	https://github.com/MadRabbit/halmak
Norman	2013	https://normanlayout.info/
Workman	2010	https://workmanlayout.org/
QGMLWB	2009	http://mkweb.bcgsc.ca/carpalx/?full_optimization
Colemak Mod-DH	2017	https://colemakmods.github.io/mod-dh/
Colemak	2006	https://colemak.com/
ASSET	2006	http://millikeys.sourceforge.net/asset/
Capewell-Dvorak	2004	http://michaelcapewell.com/projects/keyboard/layout_capewell-dvorak.htm
Dvorak	1936	https://en.wikipedia.org/wiki/Dvorak_keyboard_layout
QWERTY	1873	https://en.wikipedia.org/wiki/QWERTY

Table 1. Example keyboard layouts.

Dvorak (Dvorak et al. 1936; Dvorak and Dealey, 1936) undertook one of the most comprehensive reevaluations of the keyboard layout without attempting to optimize, and defined eleven criteria for the design and evaluation of keyboard layouts, which can be summarized as follows:

- (1) Alternate between hands.
- (2) Balance finger loads, and avoid using the same finger.
- (3) Avoid the upper and lower rows, and avoid skipping over the home row.
- (4) Avoid tapping adjacent rows with the same or adjacent fingers.

Most keyboard layouts that purport to improve typing efficiency, including Dvorak’s, demand undue strain on tendons, particularly lateral extension of the index and little fingers. The index fingers must reach to the center columns to access letters, and the right little finger must reach to the periphery of the keyboard to access various punctuation. Following Dvorak, many layouts over-emphasize alternation between hands and under-emphasize same-hand, different-finger transitions. However, same-row, adjacent-finger transitions are arguably easier, more comfortable, and faster than alternating hands to type keys that are far removed from one another. Many layouts also ignore the ergonomics of the human hand: different finger lengths and strengths, roundedness of the hand, relative ease of little-to-index finger roll-ins vs. index-to-little finger roll-outs, etc. What is of greatest concern is that most proposed layouts are not based on established open access data, do not provide reproducible evidence for their superiority, and are not published in a peer-reviewed journal for scrutiny by the scientific community. When evaluations are conducted, they are often restricted to comparisons against the Qwerty layout and not to layouts that are known to be more efficient or redesigned based on ergonomics principles.

1.2. Data on typing strength and speed

Data on finger strength and typing speed can be useful for informing the design of a keyboard layout, if the data is relevant and applied appropriately. Table 1 of an article describing the design of a keyboard layout for the Filipino language (Salvo et al. 2016) presents the "average finger strength of Filipinos [n=30, ages 16-36] measured in pounds": the index, middle, ring, and little fingers on the right hand were measured to produce 6.09, 6.37, 5.08, and 4.27 pounds of force,

respectively, and on the left hand 6.57, 5.65, 4.54, and 3.77 pounds. However, these measurements probably don't represent relative strength relevant for typing: "Respondents were asked to sit in upright position, with their wrists resting on a flat surface. A pinch gauge was placed within each finger's reach. The respondents were asked to exert maximum pressure on the device." Martin (Martin et al. 1996) used finger flexor electromyograms to measure finger strengths. From Table 4 of their publication, the index, middle, ring, and little fingers were measured to produce 2.26, 2.36, 2.02, and 1.84 Newtons, respectively. These were based on peak keyboard reaction forces, however, and may not reflect the amount of force a typist would normally or should apply when typing. We factor in strength data in our study primarily for initial placement of the most frequent letters.

Likely the most relevant study regarding typing speed for keyboard layout optimization was conducted by İşeri and Ekşioğlu (2015). While the study generated data on typing speed, care must be taken when using these measurements, as they were collected from right-handed typists on a conventional QWERTY keyboard, without regard to the order in which pairs of letters were typed. We factor in speed data in our study for early experimentation and validation, and we averaged data from the left and right hands to compensate for right-handedness of their study participants.

1.3. Letter, n-gram, and punctuation frequency

Perhaps the most extensive analysis to date of n-gram frequencies (where an n-gram is a string of n adjacent letters) on the largest general English text corpus (maintained by Google) was conducted by Peter Norvig (2012). The order of letters by frequency in English according to this 1-gram analysis is: E, T, A, O, I, N, S, R, H, L, D, C, U, M, F, P, G, W, Y, B, V, K, X, J, Q, Z, and the 10 most commonly used bigrams in English are TH, HE, IN, ER, AN, RE, ON, AT, EN, and ND.

Table 2 aggregates the results of different studies that measured the frequencies of punctuation marks in publications and communications in English and in software programs. The table includes the 12 most frequent punctuation marks from each of the studies described below.

	Sun: /1M	Malik: N-gram %	Ruhlen: /10,000	Cook: /1,000	Xah:		
					All%	JS%	Py%
.	42840.02	1.151	535	65.3	6.6	9.4	10.3
,	44189.96		556	61.6	5.8	8.9	7.5
"		2.284	44	26.7	3.9	1.6	6.2
'	2980.35	0.200	40	24.3	4.4	4.0	8.6
-	9529.78	0.217	21	15.3	4.1	1.9	3.0
()	4500.81	0.140	7		7.4	9.8	8.1
;	1355.22	0.096	22	3.2	3.8	8.6	
z		0.09		-	-		
:	3221.82	0.087	11	3.4	3.5	2.8	4.7
?	4154.78	0.032	14	5.6	0.3		
/		0.019			4.0	4.9	1.1
!	2057.22	0.013	3	3.3	0.4		
-		0.001			11.0	2.9	10.5

Table 2. Frequency of punctuation according to different studies.

Sun, et al. (2018) published statistical values of punctuation frequency in 20 English-speaking countries from large-scale text corpora. The data were acquired through GloWbE, "a large English corpus collecting international English from the internet, containing about 1.9 billion words of text

from twenty different countries. For further information on the corpora used, see <https://corpus.byu.edu/>." Table entries are average frequencies per one million characters.

Malik and Findlater (2013) analyzed the frequency of punctuation input on touchscreen keyboards using Twitter as compared against Google N-grams. They found that punctuation in mobile tweets comprised 7.5% of characters versus only 4.4% in the Google corpus. Only six punctuation symbols appeared more frequently than the letter Q in the Google corpus, and the comma was not included. Table entries are in percentage of total characters from the Google N-gram corpus (version 1 containing 472,764,897 characters in English books published between the years 1538 and 2008), and from 173,876 mobile tweets (uniformly sampling 1% of the public tweet stream from June 2012).

Cook (2013) published on the frequencies of English punctuation based on a corpus of about 459,000 words, including three novels (276,000 words), selections of articles from two newspapers (55,000 words), one bureaucratic report (94,000 words), and assorted academic papers on language (34,000 words). Table entries are average frequencies per 1,000 characters.

Ruhlen and Pressey (1924) published a statistical study of what was then current usage in punctuation. Table entries are in frequencies per 10,000 words, drawn from 38,638 words from 100 business letters, 50 professional letters, and excerpts from one issue each of several newspapers and magazines. While it is limited in scope, it provides confirmation of later and larger studies.

Lee (2013) published online an analysis of the frequencies of punctuation in different software programming languages. Table entries are percent frequencies across all programming languages they included in their study, as well as the popular C and Python programming languages. While some of the punctuation is biased toward C (19.8%) and Python (18.5%), which make considerable use of the underscore, there is some consistency across languages for the most frequent punctuation.

1.4. A new layout

In this work, we outline a set of ergonomics principles related to touch typing, apply a systematic approach to designing optimal keyboard layouts that follow these ergonomics principles, and provide open source software for generating optimal keyboard layouts based on this systematic approach. To generate an example layout for optimal touch typing in English, we made use of publicly available, massive bigram frequency data for the English language (Norvig, 2012). The result is the "Engram" layout (Figures 1 and 2), available for anyone to use on multiple platforms via the Keyman application. We compare this layout with the layouts in Table 1. (The name "Engram" is a pun, referring both to "n-gram", letter permutations and their frequencies that are used to compute the layout, and "engram", or memory trace, the postulated change in neural tissue to account for the persistence of memory, as a nod to the attempt to make this layout easy to remember.)



Figure 1. Engram layout’s default layer (top) and Shift-key layer (bottom).

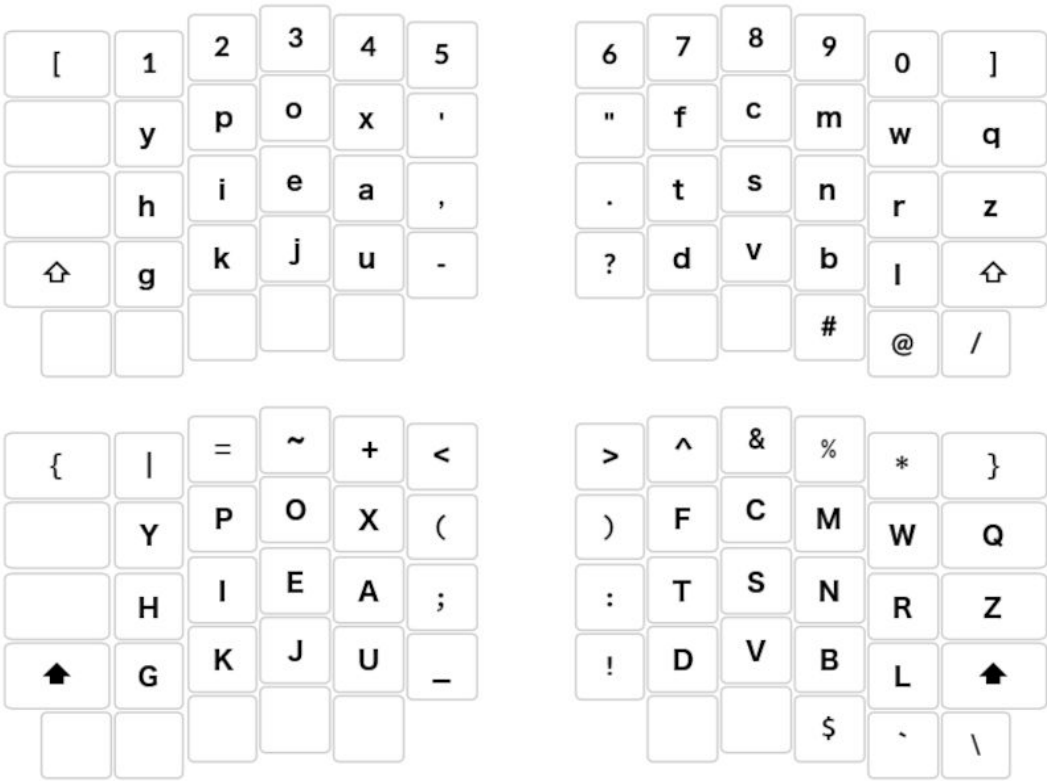


Figure 2. Engram layout’s default layer (top) and Shift-key layer (bottom) on an “orthonormal” keyboard (compressed view without thumb keys).

2. Methods

In this section, we will introduce a set of ergonomics principles for touch typing, define the general organization of a keyboard layout to minimize lateral finger movements, and suggest arrangements for the most frequent letters in English to keys on a keyboard based on comfort and bigram frequencies. We then present our Engram scoring model to optimize the arrangement of the remaining letters. After assigning non-letter characters to the remaining keys, we describe stability tests and a comparison of our resulting layout against other layouts listed in Table 1.

2.1. Ergonomics principles for touch typing

Inspired by Dvorak, but determined to overcome its drawbacks, we introduce a set of 12 ergonomics principles for touch typing, interspersed with the design decisions and scoring model parameters that we use in the present study and detailed below.

2.1.1. Ergonomics principles with design decisions (-) and scoring model parameters (+):

1. Assign letters to keys that don't require lateral finger movements.
 - 24 letters constrained to home columns (2.2.1)
2. Promote alternating between hands over uncomfortable transitions with the same hand.
 - vowels and most common consonants separated to either side (2.3.1, 2.3.3)
 - + scoring model parameters do not penalize bigrams that cross sides (2.4.1)
3. Assign the most common letters to the most comfortable keys.
 - most common letters in the home, top-center, and bottom corner keys (2.2.2)
 - most common vowel in the strongest left finger position (2.3.1)
 - most common consonants in the strongest right finger positions (2.3.3)
 - + implicit in the scoring model (2.4.1)
4. Arrange letters so that more frequent bigrams are easier to type.
 - highest-frequency bigrams arranged on the most comfortable keys (2.3.4)
 - + basis of the scoring model (2.4.1)
5. Promote little-to-index-finger roll-ins over index-to-little-finger roll-outs.
 - highest-frequency bigrams ordered for roll-ins from little to index finger (2.3.1, 2.3.3)
 - + scoring model penalizes roll-outs ("rollout" parameter in 2.4.1)
6. Balance finger loads according to their relative strength.
 - + scoring model optionally weights keys according to normalized finger strength (2.4.2)
7. Avoid stretching shorter fingers up and longer fingers down.
 - + scoring model explicitly penalizes this ("out_hi_in_lo1", "out_hi_in_lo2" in 2.4.1)
8. Avoid using the same finger.
 - + scoring model penalizes same-finger bigrams ("2x" in 2.4.1)
9. Avoid the upper and lower rows.
 - + scoring model penalizes upper and lower rows ("not_home" in 2.4.1)
10. Avoid skipping over the home row.
 - + scoring model penalizes bigrams with an upper and lower row letter ("skip" in 2.4.1)
11. Assign the most common punctuation to keys in the middle of the keyboard.
 - placement accounts for frequency, logical grouping, and ease of recall (2.7)
12. Assign easy-to-remember symbols to the Shift-number keys.
 - placement accounts for logical grouping and ease of recall (2.7)

2.2. Defining the shape of the keyboard layout to minimize lateral finger movements

Before addressing the arrangement of characters on a keyboard, it is important to define the shape of clusters of characters so that finger movements are constrained in beneficial ways. To clearly describe our approach, we will refer to the numbering scheme below. The first nine column numbers equal the number keys 1 through 9 at top, and continue increasing from left to right, and row numbers increase from bottom to top. For example, the left index finger on the home row is situated above the key in the 4th column on the 2nd row (42), and the right little finger on the home row is situated above the key in the 10th column on the 2nd row (102). The numbering scheme is portrayed below in the orthonormal (perpendicular) layout of some keyboards for clarity, but this numbering scheme is perfectly applicable to standard diagonal keyboards as well. The eight “home columns” (1 through 4 and 7 through 10) contain the eight fingers’ “home” keys (12, 22, 32, and 42 on the left and 72, 82, 92, and 102 on the right).

2.2.1. Key numbering scheme, highlighting left and right home columns:

1	2	3	4	5	6	7	8	9	0	

13	23	33	43	53	63	73	83	93	103	113
12	22	32	42	52	62	72	82	92	102	112
11	21	31	41	51	61	71	81	91	101	

We assign 24 of 26 letters to the keys in the 8 home columns, separated by two middle columns reserved for punctuation. These 8 home columns require no lateral finger movements when touch typing on an orthonormal keyboard, and minimal lateral movements on a standard, diagonal keyboard, since each column is accessed by one finger. Due to the natural roundedness of the hand, the most easily accessible keys from the home row (2) include the home keys, the top-center keys for each hand (23 and 33 on the left, 83 and 93 on the right) that allow the longer middle and ring fingers to uncurl upwards, as well as the bottom corner keys (11 and 41 on the left, 71 and 101 on the right) that allow the shorter fingers to curl downwards. We will assign the two least frequent letters, Z and Q, to the 11th column (keys 112 and 113).

2.2.2. The most easily accessible keys from the home keys:

--	23	33	--	--	--	--	83	93	---	---
12	22	32	42	--	--	72	82	92	102	---
11	--	--	41	--	--	71	--	--	101	

2.3. Arranging the most frequent letters based on comfort and bigram frequencies

Given that there are over 600 sextillion (24! = 6.204484017 E+23) possible arrangements for the 24 letters in the home columns, which is currently computationally intractable, we will arrange the letters in stages, based on letter frequency and letter-pair (bigram) frequency in the English language, according to Norvig’s analysis (Norvig 2012). To apply the following approach to optimize the layout for a different language, or optimize according to a different corpus than the one used by Google, you would run the open source software used in this study on your preferred bigram frequency data.

In prior experiments using the same methods as below but with fewer constraints applied, over 200 million different layouts were evaluated, and all vowels consistently automatically clustered together to the left side of the keyboard. Here we will initialize with vowels on the left side and the most frequent consonants on the right side to encourage balance and alternation across hands. We will reevaluate the layout and its mirror image when we consider keys 112 and 113, which break the bilateral symmetry of the layout of the letters.

2.3.1. Letters in descending order of frequency in English, highlighting vowels:

E T A O I N S R H L D C U M F P G W Y B V K X J Q Z

We first assign the five vowels (E, A, O, I, U) to the most easily accessible keys from the home keys on the left side (see 2.2.2). Given that the middle and index fingers are the strongest (Martin et al. 1996), we assign the letter E, the most frequent in the English language, to the middle or index finger position in the left home row (keys 32 and 42). Bigrams with more than 1 billion instances in Peter Norvig's analysis of Google data are: **OU**, **IO**, **EA**, **IE**, AI, IA, EI, UE, UA, AU, UI, OI, EO, OA, OE (bigrams with more than 10 billion instances are in bold). We will initialize the arrangement of the vowels so that the top four bigrams read from left to right (for example, IO with 23.5 billion instances vs. OI with 2.5 billion instances, and IE with 10.8 billion instances vs. EI with 5.2 billion instances). This will prioritize left-hand roll-ins from little to index finger over the slower, less comfortable roll-outs from index to little finger. These initial constraints result in six comfortable and efficient layouts (2.3.2).

2.3.2. Vowel layouts:

Left-hand keys:
-- 23 33 --
12 22 32 42
11 -- -- 41

- 0 U -
I - E A
- - - -

- - U -
I 0 E A
- - - -

- - - -
I 0 E A
- - - U

- 0 - -
I - E A
- - - U

- - 0 -
I - E A
- - - U

- - 0 -
- I E A
- - - U

Next, to populate the home row on the right side of the keyboard, we examine all possible sequences of four letters from the eight most frequent consonants. Each of these consonants has at least 100 billion (at least 3% of) instances in Peter Norvig's analysis, and with the vowels, they cover half the alphabet.

2.3.3. Letters in descending order of frequency in English, highlighting high-frequency consonants:

E **T** A O I **N S R H L D C** U M F P G W Y B V K X J Q Z

These eight consonants are also included among the top-frequency bigrams, with more than 10 billion instances: TH, ND, ST, NT, CH, NS, CT, TR, RS, NC. To facilitate typing these bigrams with 4 fingers on the right-hand home keys, we select 4-consonant sequences that contain at least four top-frequency bigrams, such as NRST (NS, NT, RS, RT, ST). As with the letter E above, we assign the letter T, the second-most frequent letter in the English language, to the middle or index finger position in the right home row (keys 82 and 72) . The resulting 4-consonant sequences are: NSTH, NSTR, NCTH, NCTR, and NRST. The resulting six arrangements of five vowels on the left and five arrangements of four consonants on the right gives us 30 initial layouts, each with nine letters assigned to keys and 15 unassigned keys. In 2.3.4 below, the three rows on the left and right side of the keyboard are represented as a linear string of letters, with unassigned keys denoted by “-”. All sequences on the right side will be reversed so that they read from right to left for ease of typing (right-hand roll-in from little to index finger vs. roll-out from index to little finger).

2.3.4. 8 sequences of 9 letters with 15 unassigned (-) keys:

-OU-	I-EA	----	----	HTSN	----
-OU-	I-EA	----	----	RTSN	----
-OU-	I-EA	----	----	HTCN	----
-OU-	I-EA	----	----	RTCN	----
-OU-	I-EA	----	----	TSRN	----
--U-	IOEA	----	----	HTSN	----
--U-	IOEA	----	----	RTSN	----
--U-	IOEA	----	----	HTCN	----
--U-	IOEA	----	----	RTCN	----
--U-	IOEA	----	----	TSRN	----
----	IOEA	---U	----	HTSN	----
----	IOEA	---U	----	RTSN	----
----	IOEA	---U	----	HTCN	----
----	IOEA	---U	----	RTCN	----
----	IOEA	---U	----	TSRN	----
-O--	I-EA	---U	----	HTSN	----
-O--	I-EA	---U	----	RTSN	----
-O--	I-EA	---U	----	HTCN	----
-O--	I-EA	---U	----	RTCN	----
-O--	I-EA	---U	----	TSRN	----
--O-	I-EA	---U	----	HTSN	----
--O-	I-EA	---U	----	RTSN	----
--O-	I-EA	---U	----	HTCN	----
--O-	I-EA	---U	----	RTCN	----
--O-	I-EA	---U	----	TSRN	----
--O-	-IEA	---U	----	HTSN	----
--O-	-IEA	---U	----	RTSN	----
--O-	-IEA	---U	----	HTCN	----
--O-	-IEA	---U	----	RTCN	----
--O-	-IEA	---U	----	TSRN	----

2.4. Engram scoring model

To optimize keyboard layouts, we need to score and evaluate different layouts using a scoring model. We have developed a scoring model that formalizes our ergonomic principles (2.1) for use in an optimization algorithm. Our optimization algorithm finds every permutation of a given set of letters (or, more generally, characters), maps these letter permutations to a set of keys, and scores these letter-key mappings. The score for a given layout is the average of the scores for all possible bigrams in the layout. The score for each bigram is a product of the frequency of occurrence of that bigram and flow (and optional strength and speed) factors. The flow factors only need to be computed once to optimize or evaluate any layout, assuming a consistent keyboard design. The optional strength and speed factors likewise only need to be computed once, but since they are empirically derived, are subject to change based on the representativeness of the data. As mentioned in the introduction, we factor in strength data (Martin et al., 1996) in this study for initial assignment of the most frequent letters to the home keys, and we factor in speed data in this study for early experimentation and validation based on averaged left- and right-hand data from İşeri and Ekşioğlu (2015). The Engram layout is based primarily on flow factors and bigram frequencies for key pairs. Since pairwise comparisons naturally lend themselves to a matrix representation, we will present matrix heatmap visualizations to indicate higher values as darker colors for pairs of keys in column-by-row positions in the heatmap. Density histograms present the number of key pairs for each value (out of 576 pairs for the 24 letters in the home columns). When computing scores for keyboard layouts in this study, we multiply a matrix of bigram frequencies with a flow matrix. We describe below optional strength and speed matrices that we only use for early testing and for validation.

2.4.1. Flow

Flow is a measure of the ease of a finger transition from the first in a pair of keys to the second. All of the difficult key positions and key transitions listed below lower the score by the same factor (0.9), except for typing two different keys in succession with the same finger (0.81). In one of our stability tests, we remove each individual parameter to evaluate how sensitive the winning layout is to that parameter. These parameters are not mutually exclusive, and accumulate as key positions and transitions become progressively more difficult (see below for examples).

Difficult key positions:

- out_hi: shorter (index or little) finger on top row
- in_lo: longer (middle or ring) finger on bottom row

Difficult key transitions:

- rollout: roll out from index to little finger (or other similarly outward directed pair)
- 2x: use same finger twice for a non-repeating letter
- out_hi_in_lo1: rindex above middle finger, or little above ring finger
- out_hi_in_lo2: index above ring finger, or little above middle finger
- ring_hi_mid_lo: ring above middle finger
- not_home: at least one key not on home row
- skip: one key on top row, the other on bottom row

Example Flow values for left-hand key pairs:
Left-hand keys in the examples below:

13 23 -- --
-- 22 -- --
11 21 31 --

Flow values (21=>13: key 21 to key 13):

22=>22 = 1
23=>23 = not_home = 0.9
13=>13 = not_home * out_hi = 0.9^2 = 0.81
23=>13 = not_home * out_hi * rollout = 0.9^3 = 0.73
22=>13 = not_home * out_hi * out_hi_in_lo1 * rollout = 0.9^4 = 0.66
31=>23 = not_home * ring_hi_mid_lo * in_lo * rollout * skip = 0.9^5 = 0.59
21=>13 = not_home * out_hi_in_lo1 * out_hi * in_lo * rollout * skip = 0.9^6 = 0.53

Figure 3 shows a heatmap visualization of the flow values for all 576 pairs of the 24 keys in the home columns, and Figure 4 shows a frequency histogram of these flow values.

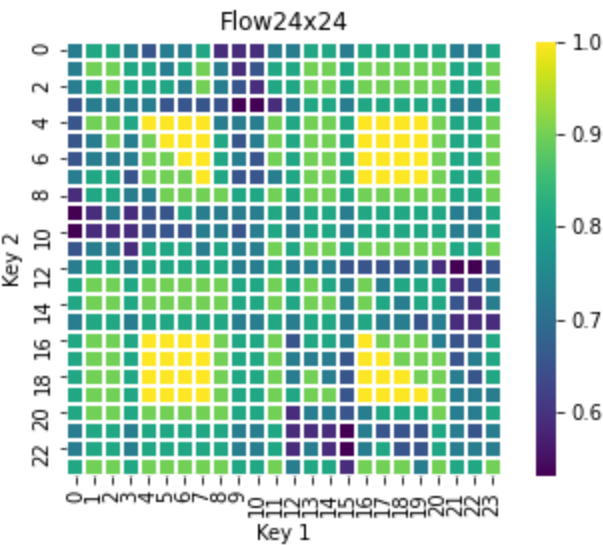


Figure 3. Flow value for every pair of 24 keys in the home columns, shown as a heatmap.

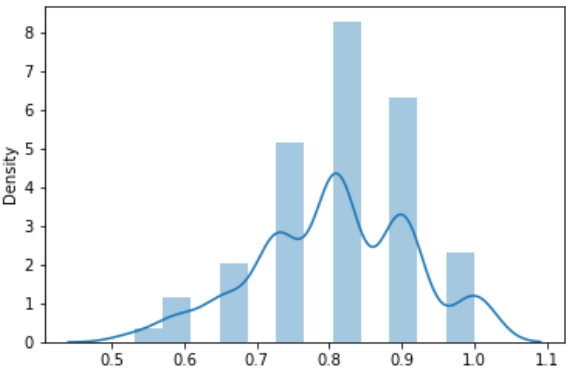


Figure 4. Frequency histogram of flow values for all pairs of 24 keys in the home columns.

2.4.2. Strength

We define strength simply as the normalized empirical measurements of finger strengths from Martin, et al. (1996) without regard for the positions of these fingers. Therefore, the key pairs with the highest strength values are predictably all pairs that use either or both of the strongest (middle) fingers, and pairs with the lowest strength values are all pairs that use either or both of the weakest (little) fingers.

2.4.3. Speed:

We define speed as the left-right averaged measurements of interkey timings from İşeri and Ekşioğlu (2015) without regard for the order of the key presses. Interkey speed values decrease roughly in order from index to little fingers starting from the home keys. Not surprisingly, the key pairs with the lowest interkey speed values (21 to 13, 91 to 103) are the same key pairs with the lowest flow values. Figure 5 shows a heatmap visualization of the speed factors for all 576 pairs of the 24 keys in the home columns, and Figure 6 shows a frequency histogram of these speed values.

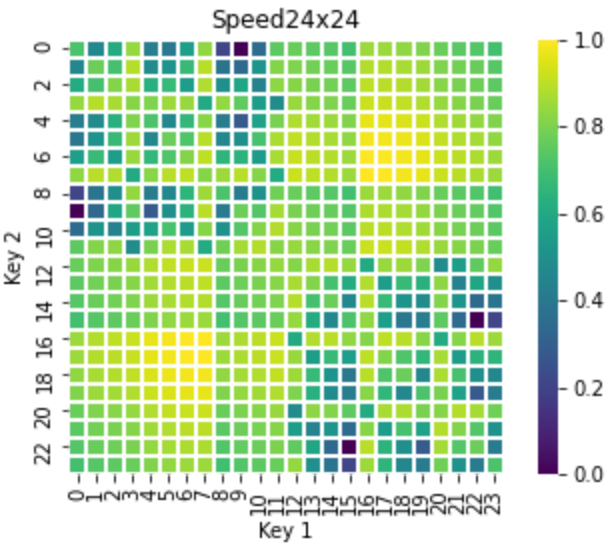


Figure 5. Speed value for every pair of 24 keys in the home columns, shown as a heatmap.

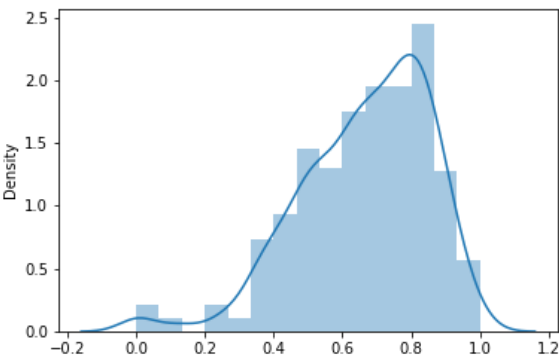


Figure 6. Frequency histogram of speed values for all pairs of 24 keys in the home columns.

2.5. Optimizing assignment of the remaining letters

Using the above scoring model, we can evaluate different letter assignments for the 15 missing letters in each of the 30 fixed initialized layouts from 2.3.4. Since there are over 1.3 trillion possible ways of arranging 15 letters ($15! = 1,307,674,368,000$), we will break up the assignment into two stages. First we will compute scores for every possible arrangement of the seven most frequent remaining letters (2.5.1) in the seven most comfortable remaining key positions (2.5.2, from 2.2.2, including either 22 or 23 depending on the layout). Second, we will compute scores for every possible arrangement of the 8 least frequent letters (aside from Q and Z; 2.5.3) in the 8 least comfortable key positions (2.5.4), after substituting in the results from the first stage.

2.5.1. Letters in order of descending frequency, highlighting the most frequent remaining letters:

E T A O I N S R H L D C U M F P G W Y B V K X J Q Z

2.5.2. The most easily accessible remaining keys:

-- 23 -- -- -- -- 83 93 ---
-- 22 -- -- -- -- -- -- ---
11 -- -- 41 71 -- -- 101

For the 7 most frequent remaining letters, there are $7! = 5,040$ possible permutations, and we have 30 initial layouts, so we need to score and evaluate 151,200 layouts. As mentioned above, to score each layout, we construct a frequency matrix of each ordered pair of letters based on the key assignments, and multiply this frequency matrix by a precomputed flow matrix.

2.5.3. Letters in order of descending frequency, highlighting the 8 least frequent letters (aside from Q and Z): E T A O I N S R H L D C U M F P G W Y B V K X J Q Z

2.5.4. The 8 least accessible remaining keys:

13 -- -- 43 73 -- -- 103
-- -- -- -- -- -- -- --
-- 21 31 -- -- 81 91 ---

For the 8 least frequent remaining letters (aside from Q and Z), there are $8! = 40,320$ possible combinations, and we have 30 layouts, so we need to score and evaluate 1,209,600 more combinations.

2.6 Further optimization by exchanging letters

If we relax the above fixed initializations and permit further exchange of letters, then we can search for even higher-scoring layouts. As a final optimization step we exchange letters, eight keys at a time, to score a total of 21,772,800 more combinations:

- 1-6. Exchange letters between each pair of non-home rows.
 - 1. Allow top and bottom rows to exchange letters on the left.
 - 2. Allow top and bottom rows to exchange letters on the right.
 - 3. Allow bottom rows to exchange letters.
 - 4. Allow top rows to exchange letters.
 - 5. Allow top left and bottom right rows to exchange letters.
 - 6. Allow top right and bottom left rows to exchange letters.
- 7. Exchange letters in corners.
- 8. Exchange letters in middle columns.
- 9. Exchange letters in the home rows.
- 10. Repeat 1-9.

2.7. Non-letter characters

In addition to the 26 letters, we will address the key assignment of non-letter characters, taking into account frequency of punctuation, logical grouping, and ease of recall.

We will assign the most frequent punctuation (from 1.3) to the six keys in the middle two columns:

13	23	33	43	'	"	73	83	93	103	113
12	22	32	42	,	.	72	82	92	102	112
11	21	31	41	-	?	71	81	91	101	

The Shift key will access related punctuation marks (shown as left and right pairs below). The left middle column will contain punctuation for separating and joining text and the right middle column will contain punctuation for closing text.

13	23	33	43	'(	"	)	73	83	93	103	113
12	22	32	42	,;	.	:	72	82	92	102	112
11	21	31	41	-_	?	!	71	81	91	101	

2.7.1. Separating marks (left middle column):

The comma separates text in lists; the semicolon can be used in place of the comma to separate items in a list (especially if these items contain commas); open parenthesis sets off an explanatory word, phrase, or sentence.

2.7.2. Joining marks (left middle column):

The apostrophe joins words as contractions; the hyphen joins words as compounds; the underscore joins words in cases where whitespace characters are not permitted (such as in variables or file names).

2.7.3. Closing marks (right middle column):

A sentence usually ends with a period, question mark, or exclamation mark. The colon ends one statement but precedes the following: an explanation, quotation, list, etc. Double quotes and close parenthesis closes a word, clause, or sentence separated by an open parenthesis.

2.7.4. Number keys:

We reserve the entire number key row to mathematical and logic symbols (Table 3). We enclose the numbers with [square brackets], and the Shift-key symbols with {curly brackets}:

{		=	~	+	<	>	^	&	%	*	}	\
[	1	2	3	4	5	6	7	8	9	0	]	/

1:		(vertical bar or "pipe" represents the logical OR operator: 1 stroke, looks like the number one)
2:	=	(equal: 2 strokes, like the Chinese character for "2")
3:	~	(tilde: "almost equal", often written with 3 strokes, like the Chinese character for "3")
4:	+	(plus: has four quadrants; resembles "4")
5 & 6:	< >	("less/greater than"; these angle brackets are directly above the other bracket keys)
7:	^	(caret for logical XOR operator as well as exponentiation; resembles "7")
8:	&	(ampersand: logical AND operator; resembles "8")
9:	%	(percent: related to division; resembles "9")
0:	*	(asterisk: for multiplication; resembles "0")

Table 3. Number key symbols.

The three remaining keys in many common keyboards (flanking the upper right-hand corner Backspace key) are displaced in special keyboards, such as the Kinesis Advantage and Ergodox. For the top right key, we will assign the forward slash and backslash (/ \). For the remaining two keys, we will assign two symbols that in modern usage have significance in social media: the hash/pound sign (#) and the "at sign" (@). The hash or hashtag identifies digital content on a specific topic (the Shift key accesses the dollar sign). The "at sign" identifies a location or affiliation (such as in email addresses) and acts as a "handle" to identify users in popular social media platforms and online forums.

```
[ { 1| 2= 3~ 4+ 5< 6> 7^ 8& 9% 0* ]} /\
  13 23 33 43 ' ( " ) 73 83 93 103 113 # $ @ `
  12 22 32 42 , ; . : 72 82 92 102 112
  11 21 31 41 - _ ? ! 71 81 91 101
```

2.8. Stability Tests

We will run three stability tests on the winning layouts:

1. Compare score of the winning layout after rearranging random letters
2. Compare ranking of all final layouts based only on interkey speed
3. Compare ranking of all final layouts after removing each scoring parameter

The first test is to see if allowing random sets of letters to rearrange in every possible combination improves the score of the winning layout. We repeat this test 1,000 times, randomly selecting eight of the 24 letters, and another 1,000 times, randomly selecting eight of the 16 letters in the non-home rows, for a total of 80,640,000 additional layout tests.

In the second test, we rescore all of the final layouts, replacing the flow matrix with the inter-key speed matrix to see if this affects their ranking. In the third test we remove each Engram scoring parameter one at a time and rescore all of the final layouts to see if this affects their ranking.

2.9. Comparing the resulting layout against other layouts

In all of the optimization and evaluation of layouts in the service of developing the Engram layout, we score layouts using Engram's scoring model and Google's bigram frequency data. However, as mentioned previously, we can use any bigram frequency data, including such data derived from any text corpus. To compare the winning layout against the layouts in Table 1, we have compiled a data set of large, representative, publicly available text data, with sources listed in Table 4 and available from a public GitHub repository (https://github.com/binarybottle/text_data). We will score each layout on Google's N-gram data and on each of the text sources in Table 4 using the Engram scoring model, and generated by the online Keyboard Layout Analyzer (<http://patorjk.com/keyboard-layout-analyzer/>) using its own much more simple scoring method: "The optimal layout score is based on a weighted calculation that factors in the distance your fingers moved (33%), how often you use particular fingers (33%), and how often you switch fingers and hands while typing (34%)."

Text source	Information
"Alice in Wonderland"	KLA analysis of Alice in Wonderland (Ch.1), a standard text used for comparing layouts
"Romeo and Juliet"	KLA analysis of Romeo and Juliet
"Bhagavad Gita"	KLA analysis of Bhagavad Gita
"Memento screenplay"	KLA analysis of the screenplay for Memento
"100K tweets"	100,000 tweets from: Sentiment140 dataset training data
"20K tweets"	20,000 tweets from Gender Classifier Data
"MASC tweets"	KLA analysis of MASC tweets (cleaned of html markup)
"MASC spoken"	KLA analysis of MASC spoken transcripts (phone and face-to-face: 25,783 words)
"COCA blogs"	KLA analysis of Corpus of Contemporary American English blog samples
"Google website"	KLA analysis of the Google home page (accessed 10/20/2020)
"Software languages"	KLA analysis of the "Tower of Hanoi" (programming languages A-Z from Rosetta Code)

Table 4. Text data.

3. Results

In the results below, we computed the total score for a keyboard layout as the average of the scores for every pair of keys. For optimization of the arrangement of the letters in the Engram layout, we computed over 23 million scores for different arrangements of the 24 letters in the home columns. For comparisons against other keyboard layouts, none of which reserve all of the keys in the middle columns for non-letter characters, we instead computed scores for 32 keys, including the 24 letters in the home columns, the middle columns, and keys 112 and 113.

3.1. Optimized layouts

Of the 1,360,800 layouts generated from the 30 fixed initializations (2.5), the top-scoring layout contains a different sequence of home-key consonants than in any of the initializations:

--0- -IEA ---U ---- TSNR ----

After permitting the further exchange of letters (2.6) and scoring 21,772,800 more layouts, 9 layouts had unique home-key letter assignments, with the top-scoring layout below:

Y	P	O	X	F	C	M	W
H	I	E	A	T	S	N	R
G	K	J	U	D	V	B	L

3.2. Stability test results

3.2.1. Test 1 results: After repeatedly selecting random letters in the top-scoring layout, creating over 80 million layouts from every permutation of these letters, and computing their scores, we could not find a higher-scoring layout.

3.2.2. Test 2 results: Replacing the flow matrix with the inter-key speed matrix changes the ranking of the layouts, but the top-scored layout stays in top place, attesting to its efficiency with respect to typing speed. We included an additional test, where we multiplied the flow matrix by the finger strength matrix when scoring. The top-scored layout remained in first place once again.

3.2.3. Test 3 results: Removing each scoring parameter did not affect the top-scoring layout’s rank, attesting to its robustness to parameter perturbations.

3.3. The winning Engram layout

These test results all corroborate the choice of the top-scoring layout. For the final letters Q and Z, our scoring model gave a higher score for Q above Z:

Y	P	O	X	F	C	M	W	Q
H	I	E	A	T	S	N	R	Z
G	K	J	U	D	V	B	L	

The complete Engram layout consists of all letter and non-letter key assignments (available by default or via the Shift key, shown as left and right pairs below):

[{ 1 | 2= 3~ 4+ 5< 6> 7^ 8& 9% 0*] } / \
yY pP oO xX ' (") fF cC mM wW qQ # \$ % & ' `
hH iI eE aA , ; . : tT sS nN rR zZ
gG kK jJ uU _ - ? ! dD vV bB lL

3.4. Comparisons of the Engram layout against other layouts

Table 5 contains scores computed for different 32-key keyboard layouts using Engram's scoring model and (i) Google's bigram frequency data (first data column), and (ii) bigram frequency data derived from each of the text sources in Table 4 (remaining columns). The Engram layout scored higher than any other layout for all text sources.

Layout	Google bigrams	Alice	Romeo	Gita	Memento	100K tweets	20K tweets
Engram	0.02514	0.01953	0.02111	0.01948	0.02612	0.03319	0.02995
Halmak	0.02474	0.01931	0.02086	0.01922	0.02588	0.03294	0.02962
Norman	0.02422	0.01903	0.02049	0.01897	0.02543	0.03231	0.02898
Workman	0.02451	0.01910	0.02063	0.01901	0.02564	0.03260	0.02931
QGMLWB	0.02330	0.01835	0.01974	0.01817	0.02450	0.03123	0.02817
Colemak Mod-DH	0.02426	0.01880	0.02037	0.01885	0.02524	0.03220	0.02882
Colemak	0.02437	0.01886	0.02050	0.01889	0.02537	0.03238	0.02896
ASSET	0.02375	0.01834	0.02000	0.01832	0.02488	0.03176	0.02842
Capewell	0.02390	0.01866	0.02019	0.01864	0.02497	0.03188	0.02852
Dvorak	0.02370	0.01852	0.02008	0.01839	0.02480	0.03157	0.02840
QWERTY	0.02135	0.01685	0.01814	0.01670	0.02243	0.02882	0.02583

Table 5. Engram scoring model scores for layouts using publicly available text data.

Table 6 contains scores for the same layouts according to the online Keyboard Layout Analyzer. The Engram layout scored in first place for 6 of the 11 text sources: Alice in Wonderland (Ch. 1), Bhagavad Gita, 100,000 tweets, MASC tweets, COCA blogs, and Google website; it scored in second place to Dvorak for Romeo and Juliet and to QGMLWB for the 20,000 tweet dataset. The Dvorak layout scored higher than the Engram layout for three text sources, more than any other layout, according to the Keyboard Layout Analyzer’s scoring algorithm.

Layout	Alice in Wonderland	Romeo Juliet	Bhagavad Gita	Memento screenplay	100K tweets	20K tweets	MASC tweets	MASC spoken	COCA blogs	Google website	Software languages
Engram	66.65	58.16	58.75	54.49	61.77	56.01	58.23	62.14	67.93	33.81	48.09
Halmak	66.25	57.02	57.45	55.03	60.86	55.53	57.13	62.32	67.29	30.41	47.60
Norman	62.76	53.21	53.44	52.33	57.43	53.24	53.90	59.97	62.80	28.29	46.01
Workman	64.78	56.67	56.97	54.29	59.98	55.81	56.25	61.34	65.27	29.28	47.76
QGMLWB	65.45	55.67	55.57	54.07	60.51	56.05	56.90	62.23	66.26	33.05	45.72
Colemak Mod-DH	65.74	56.05	57.52	54.91	60.75	54.94	57.15	61.29	67.12	31.85	48.50
Colemak	65.83	56.12	57.63	54.94	60.67	54.97	57.04	61.36	67.14	31.48	48.65
ASSET	64.60	54.63	56.09	53.84	58.66	54.72	55.35	60.81	64.71	33.05	47.52
Capewell	63.40	55.67	56.56	53.45	59.27	52.84	55.28	59.80	64.29	28.27	45.62
Dvorak	65.86	58.18	57.29	55.09	60.93	55.56	56.59	62.75	66.64	28.85	45.55
QWERTY	53.06	43.74	44.92	44.25	48.28	44.99	44.59	51.79	52.31	24.28	39.89

Table 6. Keyboard Layout Analyzer scores for layouts using publicly available text data.

Discussion

In this study, we presented a set of ergonomics principles relevant to touch typing on standard (diagonal) and specialized (orthonormal) computer keyboards. We encoded these principles in a scoring model as part of a systematic approach for developing optimized keyboard layouts. As an example application, we created the Engram keyboard layout optimized for touch typing in the English language using letter-pair frequencies from Google’s N-gram data. We conducted a variety of tests to evaluate the robustness of the winning layout to perturbations in letter arrangements and scoring parameter settings. The winning layout is the outcome of careful initializations and scoring of over 100 million layouts. Finally, we used our Engram scoring model and the online Keyboard Layout Analyzer’s simple scoring method to compare the Engram layout with 10 prominent keyboard layouts. For this comparison, we used a variety of publicly available text sources, including hundreds of thousands of tweets, spoken transcripts, etc. The Engram layout scored higher on all of these text sources than any other keyboard layout using the Engram scoring algorithm, and scored in first place for more text sources than any other layout using the Keyboard Layout Analyzer’s scoring algorithm.

The Engram keyboard layout can be installed on Linux, macOS, and Windows (<https://keyman.com/keyboards/engram>) and the open source software can be used to create other optimized keyboard layouts in different languages or tailored to different character-pair frequency data (<https://github.com/binarybottle/engram>).

Acknowledgments

I would like to thank my family for supporting this endeavor, the Keyman tools and community for helping to make distribution of the Engram layout possible, and to that damned DEC workstation at the MIT Media Lab that introduced me to repetitive strain injury 25 years ago, which has prompted me over the years to experiment with voice dictation, one-handed and keyless keyboards, foot pedals, foot mouse, and wonderful keyboards like the Kinesis Advantage and Ergodox. I would also like to express my gratitude to all of my predecessors who have made valiant efforts to improve our relationship with computers by advancing the ergonomics of keyboard designs and keyboard layouts. This article was comfortably typed using the Engram layout on macos and linux laptops.

Funding details

This work was independently supported by the author.

Disclosure statement

The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Data availability and deposition

All of the bigram frequency and timing data used to optimize the Engram layout accompanies the open source software in the engram public GitHub repository (<https://github.com/binarybottle/engram>). All of the text data used to evaluate the Engram layout is in the text_data public GitHub repository (https://github.com/binarybottle/text_data).

References

- Amell, T. K., and Kumar, S. (2000). "Cumulative trauma disorders and keyboarding work". *International journal of industrial ergonomics*, 25(1), 69-78.
- Buften, M.J., Marklin, R.W., Nagurka, M.L., Simoneau, G.G. (2006). "Effect of keyswitch design of desktop and notebook keyboards related to key stiffness and typing force", *Ergonomics*, 49:10, 996-1012, DOI: 10.1080/00140130600577437
- Çakir, A. (1995). "Acceptance of the adjustable keyboard", *Ergonomics*, 38:9, 1728-1744, DOI: 10.1080/00140139508925223
- Cook, V.J. (2013). "Standard punctuation and the punctuation of the street" in M. Pawlak and L. Aronin (eds.), *Essential Topics in Applied Linguistics and Multilingualism*, Springer International Publishing Switzerland (2013), 267-290.
- Deshwal, P.S. and Deb, K. (2006). "Ergonomic Design of an Optimal Hindi Keyboard for Convenient Use". *2006 IEEE International Conference on Evolutionary Computation*, 2187-2194.
- Dvorak, A. and Dealey, W. L. (1936). Typewriter keyboard. US Patent 2,040,248.
- Dvorak, A., Merrick, N.L., Dealey, W.L., and Ford, G.C. (1936). *Typewriting Behavior: Psychology applied to teaching and learning Typewriting*. American Book Company, New York, USA.

Eggers, J., Feillet, D., Kehl, S., Wagner, M. O., & Yannou, B. (2003). "Optimization of the keyboard arrangement problem using an ant colony algorithm." *European Journal of Operational Research*, 148(3), 672-686.

Fagarasanu, M., and Kumar, S. (2003). "Carpal tunnel syndrome due to keyboarding and mouse tasks: a review". *International Journal of Industrial Ergonomics*, 31(2), 119-136.

Ferguson, D., and Duncan, J. (1974). "Keyboard design and operating posture". *Ergonomics*, 17(6), 731-744.

Garza, J.L.B., Eijkelhof, B.H.W., Johnson, P.W., Raina, S.M., Rynell, P.W., Huysmans, M.A., van Dieën, J.H., van der Beek, A.J., Blatter, B.M., Dennerlein, J.T. (2012). "Observed differences in upper extremity forces, muscle efforts, postures, velocities and accelerations across computer activities in a field study of office workers", *Ergonomics*, 55:6, 670-681, DOI: 10.1080/00140139.2012.657692

Gerard, M.J., Jones, S.K., Smith, L.A., Thomas, R.E., Wang, T. (1994). "An ergonomic evaluation of the Kinesis Ergonomic Computer Keyboard", *Ergonomics*, 37:10, 1661-1668, DOI: 10.1080/00140139408964943

Goettl, J.S., Brugh, A.W., Julstrom, B.A. (2005). "Call me e-mail: arranging the keyboard with a permutation-coded genetic algorithm". In *Proceedings of the 2005 ACM symposium on Applied computing* (pp. 947-951).

İşeri, A., Ekşioğlu, M. (2015). "Estimation of digraph costs for keyboard layout optimization." *International Journal of Industrial Ergonomics*, Volume 48, 2015, Pages 127-138, doi:10.1016/j.ergon.2015.04.006.

Khorshid, E., Alfadli, A. and Majeed, M. (2010). "A new optimal Arabic keyboard layout using genetic algorithm", *Int. J. Design Engineering*, Vol. 3, No. 1, pp.25-40.

Krzywinski, M. (2015). "Typing Effort Model" in "Carpalx".
http://mkweb.bcgsc.ca/carpalx/?typing_effort (Accessed December 1, 2020)

Ladany, S. P. (1975). A model for optimal design of keyboards. *Computers & Operations Research*, 2(1), 55-59.

Lee, D.L., Kuo, P.L., Jindrich, D.L., Dennerlein, J.T. (2009). "Computer keyswitch force-displacement characteristics affect muscle activity patterns during index finger tapping". *Journal of Electromyography and Kinesiology*, 19(5), 810-820.

Lee, X. (2013). "Computer Languages Character Frequency." Date: 2013-05-23. Last updated: 2020-06-29. http://xahlee.info/comp/computer_language_char_distribution.html (Accessed Dec. 1, 2020)

Light, L.W. and Anderson, P.G. (1993). Typewriter Keyboards via Simulated Annealing. *AI Expert*.

Malas, T.M., Taifour, S.S., Abandah, G.A. (2008). "Toward optimal Arabic keyboard layout using genetic algorithm". *Proc. 9th Int'l Middle Eastern Multiconf. on Simulation and Modeling (MESM 2008)*, 50-54.

Malik, S., and Findlater, L. (2013). "Punctuation input on touchscreen keyboards: analyzing frequency of use and mode-switching costs." University of Maryland, College Park. HCIL-2013-17.

Martin BJ, Armstrong TJ, Foulke JA, et al. (1996). "Keyboard Reaction Force and Finger Flexor Electromyograms during Computer Keyboard Work." *Human Factors*. 1996;38(4):654-664. doi:10.1518/001872096778827288

Martin, J. (1981). "A study of keying skills and various alphanumeric keyboards." PhD Thesis. Loughborough University. https://repository.lboro.ac.uk/articles/thesis/A_study_of_keying_skills_and_various_alphanumeric_keyboards/9360608 (Accessed Dec. 1, 2020)

McLoone, H.E., Jacobson, M., Clark, P., Opina, R., Hegg, C., Johnson, P. (2009). "Design and evaluation of a curved computer keyboard", *Ergonomics*, 52:12, 1529-1539, DOI: 10.1080/00140130903215321

Nemshilov, N. (2016). "Halmak Keyboard Layout". GitHub. <https://github.com/MadRabbit/halmak> (Accessed December 1, 2020)

Norvig, P. (2012). "English Letter Frequency Counts: Mayzner Revisited or ETAOIN SRHLDCU" <http://www.norvig.com/mayzner.html> (Accessed Dec. 1, 2020)

Noyes, J. (1983). "The QWERTY keyboard: a review." *International Journal of Man-Machine Studies*, 18(3), 265–281. doi:10.1016/s0020-7373(83)80010-8

Onsorodi AHH and Korhan O. (2020). Application of a genetic algorithm to the keyboard layout problem. *PLoS ONE* 15(1): e0226611. doi:10.1371/journal.pone.0226611

Pacheco, G., Palmeira, E, Yamanaka, K. (2020). Using Genetic Algorithms to Design an Optimized Keyboard Layout for Brazilian Portuguese. In: *Encontro Nacional de Inteligencia Artificial e Computacional (ENIAC)*, 17, 2020, Evento Online. Porto Alegre: Sociedade Brasileira de Computação, 2020. p. 437-448. doi:10.5753/eniac.2020.12149

Radwin, R.G., Ruffalo, B.A. (1999). "Computer key switch force-displacement characteristics and short-term effects on localized fatigue". *Ergonomics*, 42(1), 160-170.

Rempel, D., Serina, E., Klineberg, E., Martin, B.J., Armstrong, T.J., Foulke, J.A., Natarajan, S. (1997). "The effect of keyboard keyswitch make force on applied force and finger flexor muscle activity". *Ergonomics*, 40(8), 800–808. <https://doi.org/10.1080/001401397187793>

Rempel, D., Barr, A., Brafman, D., & Young, E. (2007). "The effect of six keyboard designs on wrist and forearm postures". *Applied ergonomics*, 38(3), 293-298.

Rempel, D., Nathan-Roberts, D., Chen, B.Y., Odell, D. (2009). "The effects of split keyboard geometry on upper body postures", *Ergonomics*, 52:1, 104-111, DOI: 10.1080/00140130802481040

Rose, M. J. (1991). "Keyboard operating posture and actuation force: Implications for muscle over-use". *Applied Ergonomics*, 22(3), 198-203.

Ruhlen, H., and Pressey, S. (1924). A Statistical Study of Current Usage in Punctuation. *The English Journal*, 13(5), 325-331. doi:10.2307/802253

Salvo J.M.R., Raagas C.J.B., Medina M.T.C.M., Portus A.J.A. (2016). "Ergonomic Keyboard Layout Designed for the Filipino Language." In: Goonetilleke R., Karwowski W. (eds) *Advances in Physical Ergonomics and Human Factors. Advances in Intelligent Systems and Computing*, vol 489. Springer, Cham. doi:10.1007/978-3-319-41694-6_41

Serina, E.R., Tal, R., Rempel, D. (1999). "Wrist and forearm postures and motions during typing", *Ergonomics*, 42:7, 938-951, DOI: 10.1080/001401399185225

Sun, Kun & Wang, Rong. (2018). "Frequency Distributions of Punctuation Marks in English: Evidence from Large-scale Corpora." *English Today*. doi:10.1017/S0266078418000512

Wagner, M. O., Yannou, B., Kehl, S., Feillet, D., & Eggers, J. (2003). "Ergonomic modelling and optimization of the keyboard arrangement with an ant colony algorithm." *Journal of Engineering Design*, 14(2), 187-208. doi:10.1080/0954482031000091509

Walker, C.P. (2003). "Evolving a more optimal keyboard". *Course project: Introduction to evolutionary computation, Missouri University of Science & Technology*.

Yin, P.Y. and Su, E.P. (2011). Cyber Swarm optimization for general keyboard arrangement problem. *International Journal of Industrial Ergonomics*, 41(1), 43-52.

Zecevic, A., Miller, D.I., Harburn, K. (2000). "An evaluation of the ergonomics of three computer keyboards", *Ergonomics*, 43:1, 55-72, DOI: 10.1080/001401300184666

Zipp, P., Haider, E., Halpern, N., & Rohmert, W. (1983). "Keyboard design through physiological strain measurements". *Applied Ergonomics*, 14(2), 117-122.