Optimizing Keyboard Layouts for English Text

David Sommerfield – Computer Science Department

WINONA STATE UNIVERSITY

Introduction

Layout optimization is difficult and historically has relied on theoretical decisions or manual observations of data instead of statistically robust modeling. QWERTY, a product of these observations, was popularized by its usage on typewriters in the 1870s and, in spite of its age, remains as the most common keyboard layout today. The search space for layouts grows factorially, making it impossible to search exhaustively, necessitating smart ways of searching, such as metaheuristic optimization techniques like simulated annealing and genetic algorithms. The focus of this project is implementing such a technique on a predictive model of typing speed (R^2:0.78, MAE:12.7ms) to generate a layout optimized with real-world data, remedying the shortcomings of traditional layout optimization methods.

Data Processing

Two data sets are used to construct a predictive model of typing time: The 136M Keystrokes Dataset for data on keystroking patterns and their correlation with typing speed and the iWeb corpus for ngram frequency information.

The 136M Keystrokes Dataset

This dataset comprises 8,228 hours of typing data from 168,000 volunteers across four keyboard layouts: azerty, dvorak, qwerty, and qwertz. Data is limited to participants who used 9 to 10 fingers for typing to approximate touch typing and ensure consistency in the input methodology. Approximate string matching is used to identify typos; then, a sliding window is used to decompose the data into bistrokes and tristrokes. Transition times for each typo-free nstroke are grouped by key-coordinate ngram pairs and then averaged with outliers being removed by IQR.

M	R	T	C	W	,	K	A	E	’
L	N	D	S	V	Y	U	O	I	G
H	X	Z	B	F	.	-	Q	J	P
SPACE									

$(-5,3)$	$(-4,3)$	$(-3,3)$	$(-2,3)$	$(-1,3)$	$(1,3)$	$(2,3)$	$(3,3)$	$(4,3)$	$(5,3)$
$(-5,2)$	$(-4,2)$	$(-3,2)$	$(-2,2)$	$(-1,2)$	$(1,2)$	$(2,2)$	$(3,2)$	$(4,2)$	$(5,2)$
$(-5,1)$	$(-4,1)$	$(-3,1)$	$(-2,1)$	$(-1,1)$	$(1,1)$	$(2,1)$	$(3,1)$	$(4,1)$	$(5,1)$
$(0,0)$									

Figure 1. Key-coordinate Character Mapping and Frequency Heatmap

Typing Error	Example String
Insertion	But thankk you for the offer
Deletion	But tha k you for the offer
Substitution	But thabk you for the offer

Figure 2. Types of Typing Errors

iWeb Corpus

The iWeb corpus is a comprehensive collection of high-quality text and one of the largest available English corpora. A sliding window decomposes it into bigrams and trigrams, capturing their frequency of occurrence.

Bigram	Occurences	Trigram	Occurences
th	9709171	the	6076523
he	8552661	ing	3227179
in	7913861	and	2998065
an	6389345	ion	1716878
er	6348583	ent	1519196

Figure 3. Top 5 Bigrams and Trigrams Extracted from the Corpus

Defining The Cost Function

Data Analysis

Ngram frequency significantly influences typing speed, with a high frequency aiding muscle memory regardless of placement. After accounting for the effect of frequency, we can assess the influence of key placement and identify three distinct categories of bistroke:

- **ALT:** Alternating bistroke. A bigram typed on both hands by alternating sides of the keyboard. Fastest category.
- **SHB:** same-hand bistroke. Opposite of an ALT.
- **SFB:** same-finger bistroke. A bigram typed using the same finger twice. Slowest category, especially for high WPMs.

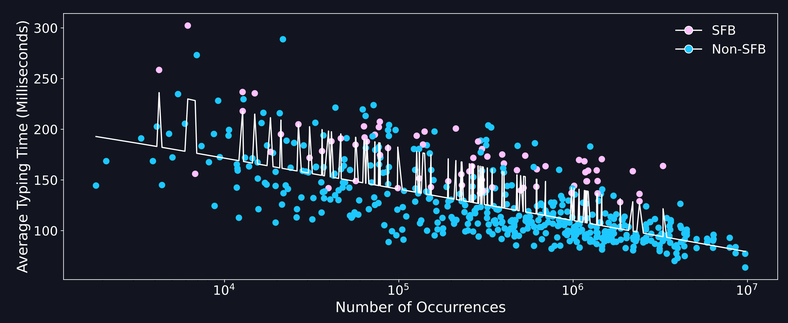


Figure 4. The Effect of SFBs and Frequency on Typing Speed (>80WPM)

Cost Function Construction

To prevent overfitting on our sparse data, we devise a cost function from our findings. Using the Levenberg-Marquardt algorithm the parameters are fine-tuned to fit the typing data. For a bistroke b, of category $i = \{1, 2, 3\}$, the cost is:

$$C(b) = (p_0 \log(f(b) + p_1) + p_2) \times \left(1 + P_x(b)P_y(b) + P_x^{(i)}(b)P_y^{(i)}(b)\Delta\right)$$

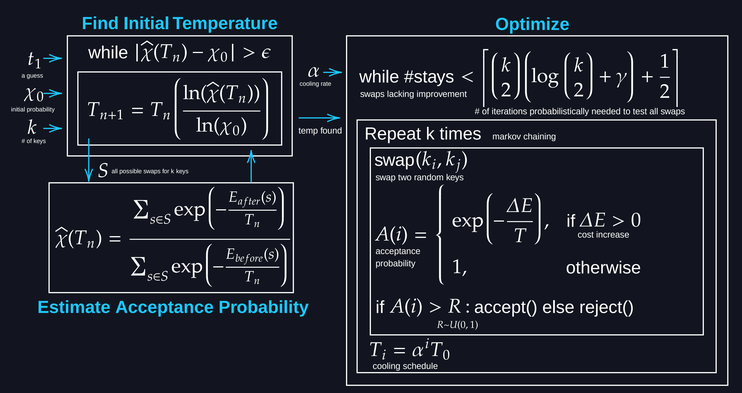
Δ is the row-stagger adjusted distance between keys if it's an SFB and 1 otherwise. P_x and P_x are row and column penalties.
For a tristroke t, the cost is:

$$C(t) = C(b_1(t)) + C(b_2(t)) + P_x(s(t))P_y(s(t))\Delta$$

$b_1(t), b_2(t)$ are the constituent bistrokes of t, $S(t)$ is its associated skipstroke, and Δ represents the distance if a skipstroke is present, and 0 otherwise. For fast typists, same-finger skipstrokes (strokes separated by one stroke sharing a finger) behave more like delayed same-finger bistrokes. The final cost of a layout l is the sum of each tristroke's cost and frequency, effectively estimating the typing time for a given layout $C(l) = \Sigma_{t \in T}(C(t) \times f(t))$

Simulated Annealing

Simulated annealing optimizes by gradually reducing the probability of accepting suboptimal moves over time. The acceptance probability is guided by the temperature parameter and a cooling schedule that diminishes it. For our problem space, a move is simply swapping two keys. The full algorithm is outlined below:



Results and Future Work

For the final result (fig. 1), a layout of the most common 30 characters was optimized. According to the time prediction model, the converged solution is 8% quicker than QWERTY. To create better predictive models in the future, we developed an open-source tool called Kiakl to collect more diverse keystroke data from volunteers on a greater variety of alternative layouts.