

Faster Transcription Using Machine Learning & a New Shorthand



Ronit Avadhuta

Phrase 1

How can we reduce the time it takes digitally transcribe information?

Phrase 2

The final product should be able to reliably transfer written shorthand to digital text.

Introduction | Contribution to Field

Write Digitally Faster

A new, custom, and efficient shorthand (loops, curls, and squiggles)

- help people save time writing on touchscreen devices
- write without the traditional keyboard (easier for the visually impaired)
- standardize or facilitate shorthands in the medical industry.

Main Results

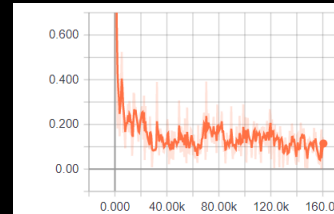


Figure 1. Total Loss from 0 Steps to 160,000 Steps
This is a graph of the progression of Loss (inaccuracy) on the y-axis over time measured in training steps on the x-axis.

Analysis | One Proportion Z-Test

$$z = \frac{\hat{p} - p_0}{\sqrt{\frac{p_0(1 - p_0)}{n}}}$$

\hat{p} = Observed Population
 p_0 = Null Hypothesized Value
 n = Sample Size
 z = Z Score

Observed = 154/174 (Calculated via loss)

z-score of ~39.08 standard deviations from the mean

$p \leq 0.00001$ ($p \leq 0.05$).

The algorithm is effective with loss of ~0.11.

Abstract

Faster Transcription with the DSK
From the User Interface to Countless Applications

Write Anywhere
Any User on Any Touch-Screen Device can draw to write.

User Interface

Components

- WST will analyze what the user is trying to say
- TST will analyze what the shorthand is trying to say
- More about this in the Methodology Graphical Abstract

Methods

WST & TST

Writing vs Text

Data Collection

Samples of the Shorthand are drawn
These images then need to be "binned" to identify individual characters within a segment.
These images also need the same dimensions and resolution.

English IPA
A key to refer to the IPA of many words of the English will be generalized to fit the 12 characters of the Digital Shorthand Key

Programming

80% Train 20%
As an industry standard, the data will be organized into two groups.
Layers from the Keras Library will sort the data into nodes of different layers and draw connections.

Matching
The goal of this portion is to effectively draw information from the 12 shorthand characters and context to try to get English words

Fine-Tuning

Overtuning & Sample Size
Different tools such as Data Augmentation, Dropout, etc will be used to accommodate training sets of different sizes, proportions, and near-world applicability.

Duplicate Words
Many words will have different keys encoding for the same value. To get away from these mistakes, look the parts of speech each word is referring to.

Table 1. Comparison of Models Across Shorthands

This Table compares the Digital Shorthand Key, Gregg, and Pitman in Loss, Training Size, and Information Density. Higher Information Density indicates a more efficient Shorthand and more efficient Algorithm as well.

Shorthands	Loss	Training Size	Information Density
DSK	11.27%	~500	1.53
Gregg	13.50%	Database	~1.34
Pitman	10.24%	4,000	-

Table 2. Comparison of Average RGB and Stroke Count (Speed Score)

Dividing these two qualities for each experimental groups its own speed score, representative of Information Density

Name	Gregg	E1	E2	E3
Average RGB	251.25	250.48	251.44	253.57
Strokes	187	219	247	166

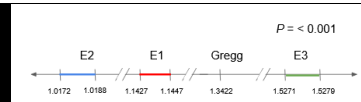


Figure 2. Ranges of Information Density Between Groups
This number line shows where the Information Density of each group (± 3 standard deviations) lies.

Conclusion

Info Density: 1.52

Loss = 11.37%

Steps = 160,000

Images = 500

- 1 - /x/
- 2 -> /r/
- 3 -> /k/
- 4 -> /b/ /w/ /p/
- 5 -> /n/
- 6 -> /s/
- 7 -> /t/
- 8 -> /m/
- 9 -> /e/
- 10 -> /d/
- 11 -> /g/
- 12 -> /f/

Superior speed score
Information Density

Novel ML algorithm to
digitally transcribe
information with
Novel Shorthand.

Supporting Documentation

- QR Code to
- Literature Review
- Project Proposal
- Project Notes
- & More!

