Discussion

Since there is evidence to support the notion the Machine Learning algorithm is largely successful and the shorthand itself is shown to a significant advantage when measuring by Speed Score, it can be asserted the Digital Shorthand Key serves as an optimal method for faster transcription. The p-values significantly less than 0.05 also reinforce the ideas that the Machine Learning model is successful almost 90% of the time and that the chance of Gregg Shorthand outdoing the Digital Shorthand Key in terms of Speed Score is less than 1%. Hence, the Digital Shorthand Key is a reliable and efficient medium for quickly transcribing text that embodies an image recognition and English word mapping elements.

Nevertheless, specialized equipment, such as Steno Machines, also serve as a popular alternative for those looking to write at high speeds at the price of extra equipment (MacMillan, 2016). However, the Digital Shorthand Key aims to provide an equivalent free for all and available at a moment's notice.

The Digital Shorthand Key also proves itself to be an improvement upon a Pitman Shorthand recognition system. The study focused on interpreting Pitman shorthand achieved a loss of about 10%, proving to be a formidable competitor, but the authors also conceded there were a reasonable number of common errors among their shorthand experts (Ma et al., 2008). From analyzing the mistakes, the researchers also suggested a variety of amendments to the shorthand to make it more user friendly. While Machine Learning algorithms can be applied to any shorthand, it takes a new more efficient shorthand to surpass its competitors.

This project itself is an application of the observation made in a linguistics study. The study published on ScienceMag conducted an experiment in which they compared languages in terms of information density and determined that there was indeed sizeable variance across languages (Coupe et al., 2019).

Likewise, studies and articles on Unicode also detail how the repetition in language can be leveraged in text file compression (Studený, n.d.). In terms of Unicode, it creates a HashMap-like dictionary of shorter terms to represent strings, longer terms.

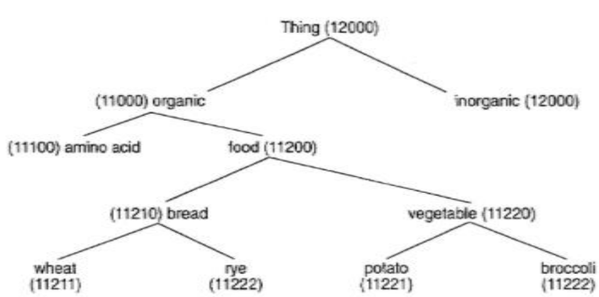
The model was trained exclusively using common English words in famous linguistic-used stories and the most common words in the English lexicon. When the model is confronted with new words or slang it may not recognize, it may be comfortable recognizing characters that it has gotten used to and assume incorrect meanings. In this scenario, the accuracy of the model would be an overestimate.

Likewise, the loss of the model would fluctuate greatly even among a smaller range of ~5,000 steps. This allows from interpretations of the total loss to be anywhere from 0.5 to 0.13. In this regard, the loss chosen to be reported, 0.113, would be an underestimation of the model's abilities and success.

Similarly, errors within the application will always persist too. Ambiguity in language found by computers is one of the biggest problems present whenever computers meet language and at the core of Natural Language Processing (Nadkarni et al., 2011). A future avenue of study would be to reduce the little amount of ambiguity this system is bound to face. This can be done via variety of methods, name some low-level Natural Language Processing tasks such as Part-of-Speech assignation.

The ontology-based parser from Busch serves an excellent example of an existing patent using this step in Natural Language Processing methodology (Busch et al., 2001). The invention takes input, translates it phonetically, and tracks the part of speech to create a more fluid parser for typing applications. This project builds upon this work by using this element of the methodology to solve another text input problem, albeit not via text but rather shorthand.

To understand how this solution may impact this product, look at the word identification process once again. This method of going back from the Digital Shorthand Key via an organized dichotomy is like how researchers at Google used a new Artificial Intelligence to dramatically increase the translation success of words and phrases (Mataic, 2016). In Google program, they split all the words up into trees with different subbranches. For example, you would find the word "dog" on the "pets" branch on the "animals" branch on the "living" branch. In the case of the Digital Shorthand Key, however, the twelve characters in the Digital Shorthand Key would see branches less like the example on the left, Google's semantic tree, and more like the example on the right, representative of this project and shows the possible interpretations for the letters t and o in the word.



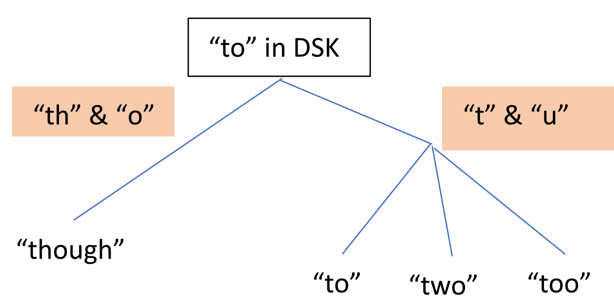


Figure 4. and 5. Semantic Trees for Words

The following two trees show how a team from Google classifies words (left) and how the Digital Shorthand Key identifies words.

Since “to,” “two,” “too,” and “though” are going to typically found in different parts of sentences, some of these context clues embedded naturally into language may also help solve this current error. These are the same context clues people apply when distinguishing homophones verbally. As of right now however, this is a minor problem as the algorithm is designed to allow the user to choose from multiple possible interpretations.

In all, the system provides a quick and easy way to communicate information through this model majority of the time. It provides a quantitative metric of the effectiveness of the trained model and a new medium for faster digital transcription for everyone in all environments. The digital shorthand system can help people save time writing on touchscreen devices such as iPads, write without the traditional keyboard (easier for the visually impaired), and standardize or facilitate shorthands in the medical industry.