Avadhuta Lit Review 2020v3

Project Title

Machine Learning & a New Shorthand for Faster Text Input

Date

11/23/2020

Society demands efficiency. This is evident with several recent inventions, such as 5G and online medical treatment. However, while text messages can be sent in less than 5 seconds, society still types on the same QWERTY keyboard used in 1874. Society is not limited by the speed technology works but rather the speed at which humans type. As part of the Information Age, it is time for change.

Developing a New Shorthand

Beating Gregg Shorthand

Hence, my problem last year was that even Gregg Shorthand (the world's current leading option for the fastest writing convention) was not enough to satisfy the needs of today's world. The world needs a simpler, more efficient solution. Thus, it is hypothesized that it should be possible to create a new shorthand that utilizes less strokes and shorter strokes than Gregg Shorthand, the current leader, to minimize writing time.

Since my new shorthand was able to represent the same phonetic sounds with fewer characters, the hypothesis was supported. Indeed, it was possible to create a new shorthand utilizing less strokes and pixels, E3. From here, it is possible to look towards digital storage of this information as well. By representing the same characters on paper with digital values, it may be possible to store data more efficiently, thus saving more space and processing time. Furthermore, with faster processing speeds, it could help with a large variety of tasks that include everything from self-driving cars to processing online purchases faster.

The Shorthand in Context

This year, the shorthand will be put to use when answering the following question: How can a faster method for data input be found that does not need specialized equipment, such as a steno, and is accessible to everyone? To better understand this year's project, some background knowledge on shorthands and machine learning would be beneficial along with a short description of my project last year and prior research.

Making Writing More Efficient

Language

Foremost, it is important to note that written language in general has some areas for improvement. For example, one article focuses on drawing conclusions between 17 languages across 9 language families to demonstrate the relationships between speech, information density, number of syllables etc (*Different Languages, Similar Encoding Efficiency: Comparable Information Rates across the Human Communicative Niche | Science Advances*, n.d.). The project had seemingly taken into consideration several variables such as sex and language family that could have contributed to different rates in information and speech. After several mathematical tests and syntagmatic analysis, Figure 1 showed that languages varied greatly in speech while they were remarkably similar in Information rate. Information rate was defined in the experiment as the product of the speech rate and information density.

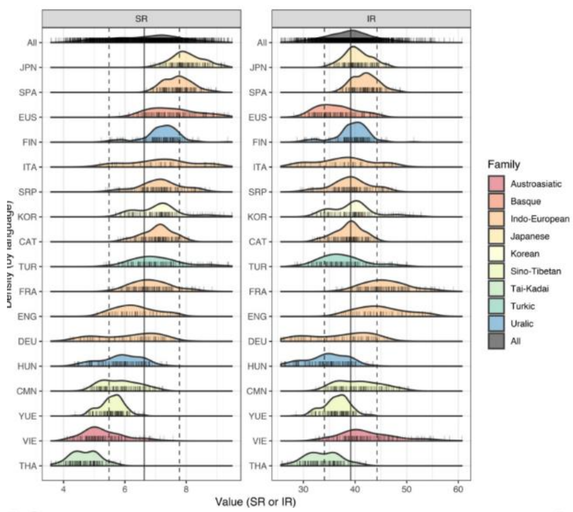

Figure 1. Speech Rate (SR) and Information Rate (IR) across Languages

The Digital World

This same concept can be seen in a Unicode article as in a simple article on zip files, but this uses slightly different methods (*Fast Compression Algorithm for UNICODE Text*, n.d.). The first method it touches upon is Lempel-Ziv compression that takes previous segments, which may be beneficial depending on how repetitive the sample is. Then they also use a hash table which is a method almost identical to zip files.

However, it also goes into detail about the storage space of the characters. Compression and decompression are nothing but a matter of assigning values to different segments of the text. There are also some same algorithms and times (in microseconds) for how long this compression and the process should take.

	Unicode compression		zlib, level 1		zlib, level 9	
	compressed size [B]	time [μs]	compressed size [B]	time [μs]	compressed size [B]	time [μs]
English (1014 B)	560	0.18	405	2.8	377	3.2
Russian (982 B)	618	0.19	464	2.9	443	3.3
Chinese (1018 B)	841	0.23	726	3.8	719	3.8

Figure 2. Unicode Built-In Compression Rate Across 3 Languages

But Natural Language Processing (NLP) itself is a bit different itself. It deals more with the syntactical and semantic sides of language and is thus a more complicated process that is case dependent and largely prone to error compared to Unicode compression. This field includes many types of models and techniques ranging from Hidden Markov Models to simple Backwards Propagation (*Natural Language Processing: An Introduction | Journal of the American Medical Informatics Association | Oxford Academic*, n.d.). NLP has come a long way and is even embodied in modern inventions such as IBM's Watson. This leads us to a discussion of its application in this year's project.

Machine Learning for Text

The output of the first Machine Learning program will output information written in the DSK, however this is not English. In order to get it back into English, lexers and parsers will have to be used, or another Machine Learning algorithm that strictly deals with text to put the information into context (*US7027974B1 - Ontology-Based Parser for Natural Language Processing - Google Patents*, n.d.). Furthermore, although the information about English words in IPA is phonetic like the shorthand, the IPA will have to go through a dichotomy in order to resemble the DSK.

IPA in conjunction with Unicode

The groups solution was a phonetic-based text input method for a patent that also focuses on the phonetic aspect of language to translate ideas between different mediums (*US8200475B2 - Phonetic-Based Text Input Method - Google Patents*, n.d.). The patent details a system where a program accurately maps the sounds in a certain language so that they can be expressed in other scripts and languages, via a phonetic based string layout. The system works by first taking a word, character, or phrase from a target language and giving it a specific IPA designation along with certain Unicode IDs. From here, one can find the corresponding layout in the target language. The patent also protects the layout of the keyboard used for this system.

Essentially, the process can be simplified as a piece of information being transferred from the source language to Unicode then the designated Phonetic String. This string is fed into a phonetic mapping engine which has a fixed scheme that it uses to produce results concisely and reliably. The system also has a series of three large processors that first work together to double check the work that works with different information for example Computer 110 works primarily on the backend. Ultimately, this is intended as a reusable program on many different platforms including stationary and mobile for many different environments. It is also applicable to word processing programs. Nevertheless, this is only for text; images are a different matter.

Introducing Semantics

Looking beyond the pronunciation of words, their semantical sides also have weight. The logic naturally embedded into language can be taken advantage of to organize such a large inventory of words. In this project, the natural language processing system aims to loom for specific keywords or types of keywords that may give hints to the literary subject and predicates of sentences. By identifying these attributes, the program can turn these into a format that is more easily readable by search engine and more robotic operators. Ultimately a lexer and parser will need to be used to deal with this new fundamental type of data. A simplified model is shown below in Figure 7.

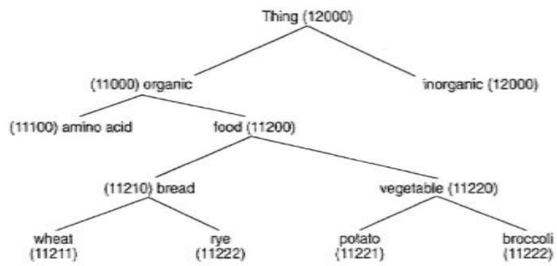

Fig. 7

Figure 3. Semantic Dichotomy of Words

Semantics in Technology

In a way, there is an ever-growing shift away from the rough idea and towards one-to-one translation. Part of this movement includes the new deep learning algorithm developed by Quoc Le and his team at Google (*Google's New Translation Software Is Powered by Brainlike Artificial Intelligence | Science | AAAS*, n.d.). This part of deep learning is referred to as neural machine translation. This is a series of several layers of processors working together in tandem. The fact that translating language requires deeper cognitive abilities is a debated topic. This was proven due to the old mindset of splicing the sentence which simply did not work sometimes. By using vectors to connect related words and ideas however, they were able to consistently remove most translation errors. This hints at further growth in this field in the nearby future.

Machine Learning for Images

Image recognition is an important part of this project because it is the first step of getting the shorthand from an image into a format where functions can be performed to get it into English. This can utilize one of two methods (or both): blocking and weighted matrices. Blocking is a method in image recognition that uses distinguishing characteristics to split the image into parts that the algorithm can tackle one by one (Mohamed & Rohm-Ensing, n.d.). From this point, these sections of the image can be split up into rows and columns in which certain areas have more weight than others and can be represented with vectors (*US8200475B2 - Phonetic-Based Text Input Method - Google Patents*, n.d.)

Weighted Matrices

These two main steps of blocking and the weight matrix can be found in many image recognition studies, however, one specific study focused on using these techniques to read Gregg Shorthand. A different method that focused more on thickness was meanwhile used to read Pittman Shorthand (*Automatic Recognition and Transcription of Pitman's Handwritten Shorthand—An Approach to Shortforms - ScienceDirect*, n.d.). However, to measure the success of the Gregg Shorthand project, the engineers behind this project decided to use two artificial networks both offline and online and compare them in a variety of elements such as computing time, computing power needed, and accuracy (Rajasekaran, 2014). While the online and offline processes differed in the location of the computations (on-server or off-server), their computations were similar.

However, the core weighted matrix system categorized every pixel in a 32 x 32 area into either black or white (Rajasekaran, 2014). 32 x 32 was chosen for its reliability yet is small enough to efficiently work with. A model for this matrix would have to be prepped but after its filtering, softening, sharpening, or embossing, the output would resemble the following.

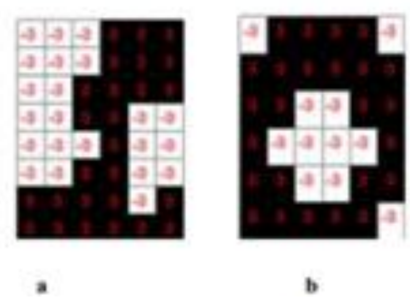

Figure 4. Weighted Matrices for Gregg Shorthand

Expanding on Weighted Matrices

Another study titled, "Automatic recognition and transcription of Pitman's handwritten shorthand—An approach to shortforms," uses a similar matrix to store information about the image but used more values than just 1s and 0s (*Automatic Recognition and Transcription of Pitman's Handwritten Shorthand—An Approach to Shortforms - ScienceDirect*, n.d.). The study starts off by listing a series of unofficial yet extremely helpful and efficient changes to Pitman shorthand including new abbreviations and the changes of certain vowel clusters. To track these changes and to measure whether these changes are significant, they trained an algorithm to read the shorthand and evaluate its efficiency to write and legibility. Examples of their matrix diagrams or processing is shown below.

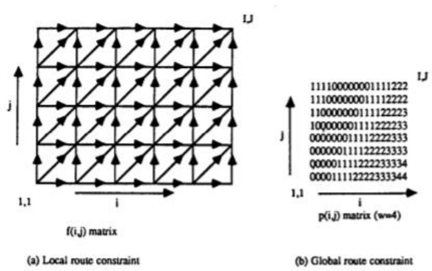

Figure 5. Weighted and Scaled Matrices

To compare this to the Gregg Shorthand example, they took a similar approach to many matrix-based vector analysis procedures as other Handwritten Shorthand Recognition systems. However, the caveat is that in this system it uses a graded scale to gauge it more in detail than just zeros and ones. However, this will lead to dealing with heuristics and thus the workload is heavier on the Machine Learning side.

Monitoring the Direction of Writing with Polar Functions

Another article started by addressing the fact that currently shorthands are important and relevant due to their usage in digital data entry displays. Pitman's wpm rate makes it the ideal tool to use for the job (*Segmentation and Recognition of Phonetic Features in Handwritten Pitman Shorthand - ScienceDirect*, n.d.). First, the article works on analyzing the patterns within the strokes of Pitman Shorthand. It marks the change of direction and the order to check how many times the direction of writing changes. From there, the samples must be segmented so that the flow of different characters combined with each other can be realized and to prep for the Box Model. This Box Model is a helpful tool for AI to break up sensing objects or features in visual data. From here the computer represents the strokes as a series of polar functions and applies the summative vectors and weighted matrices approach standard in Machine Learning. Here is a diagram of the preliminary portion however.

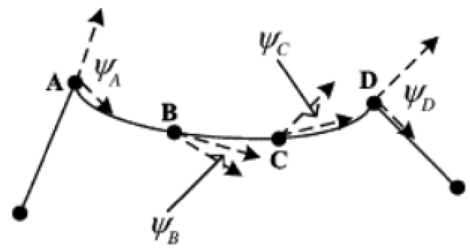

Figure 6. Directional Movements in Pitman

Works Cited

Automatic recognition and transcription of Pitman's handwritten shorthand—An approach to shortforms—ScienceDirect. (n.d.). Retrieved November 24, 2020, from https://www.sciencedirect.com/science/article/abs/pii/0031320387900082

Different languages, similar encoding efficiency: Comparable information rates across the human communicative niche | Science Advances. (n.d.). Retrieved November 24, 2020, from https://advances.sciencemag.org/content/5/9/eaaw2594

Mohamed, A., & Rohm-Ensing, E. (n.d.). *English-Arabic Handwritten Character Recognition using Convolutional Neural Networks*. 7.

Fast Compression Algorithm for UNICODE Text. (n.d.). Retrieved November 24, 2020, from http://unicode.org/notes/tn31/

Google's new translation software is powered by brainlike artificial intelligence | Science | AAAS. (n.d.). Retrieved November 24, 2020, from https://www.sciencemag.org/news/2016/09/google-s-new-translation-software-powered-brainlike-artificial-intelligence

Natural language processing: An introduction | Journal of the American Medical Informatics Association | Oxford Academic. (n.d.). Retrieved November 24, 2020, from https://academic.oup.com/jamia/article/18/5/544/829676

Segmentation and recognition of phonetic features in handwritten Pitman shorthand—ScienceDirect. (n.d.). Retrieved November 24, 2020, from https://www.sciencedirect.com/science/article/abs/pii/S0031320307004426

Rajasekaran, R. (2014). Statistical review of Online/Offline Gregg Shorthand Recognition using CANN and BP - A Comparative analysis. *International Journal of Advanced Information Science and Technology*, 12.

US7027974B1—Ontology-based parser for natural language processing—Google Patents. (n.d.). Retrieved November 24, 2020, from https://patents.google.com/patent/US7027974B1/en

US8200475B2—Phonetic-based text input method—Google Patents. (n.d.). Retrieved November 24, 2020, from https://patents.google.com/patent/US8200475B2/en