Automatic Speech Recognition: Using Vosk on accented speech data

Cordelia Horch

I. Introduction

It is known that current speech recognition programs have difficulty recognizing speech patterns of people with accents and from many underrepresented groups [1]. This is due to the lack of diversity of speech patterns in the data used to train such models. Using such a narrow corpora excluded linguistic features that are found in accented English or specific dialects. As computers and specifically speech recognition technology increasingly becomes part of our daily lives, it is problematic and unacceptable to force some to adapt their speech patterns to be understood. Additionally, many rely on these technologies for communication such as those with speech or hearing disabilities [2]. It is imperative that members of these communities can be understood by anyone and can understand anyone through the use of these programs.

There are many automatic speech recognition (ASR) models currently available. A literature review from 2018 broadly categorized speech recognition techniques into seven main categories: neural network based speech recognition, Fuzzy logic based, Wavelet based, Optimization algorithm based, Dynamic Time Warping (DTW) algorithm based, sub-band based speech recognition and other approaches [3]. The majority of research I have read utilize artificial neural networks. After experimenting with multiple different ASR models, I settled on Vosk which is a neural network model that uses Kaldi style data processing which is a widely used speech recognition toolkit. Vosk is available in 18 natural languages and multiple programming languages. I chose to write in Python.

Another avenue of research related to accented speech recognition, is to use a model to adapt highly accented speech and reduce the linguistic variation found in these speech samples. This technique was applied successfully with English language data produced by Japanese speakers [4]. For my project, I am interested in performance between groups of languages without additional adaptation models applied. When analyzing the performance of this model, I grouped language by family according to the common ancestry of each language. These groupings show that Indo-European languages are over represented in the dataset. Any future work with this data should account for the over-representation of certain languages when analyzing model performance.

II. Data

The dataset I worked with is the Speech Accent Archive published on Kaggle (The Speech Accent Archive). This dataset contains 2140 speech samples of speakers from 177 countries with 214 different native languages. There are three files included in the dataset: reading-passage.txt, the text all the speakers read, speakers_all.csv, demographic information on every speaker, recording, a zipped folder containing .mp3 files with speech. The following is the passage that every speaker reads,

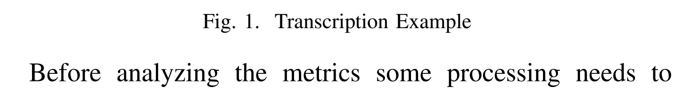
> Please call Stella. Ask her to bring these things with her from the store: Six spoons of fresh snow peas, five thick slabs of blue cheese, and maybe a snack for her brother Bob. We also need a small plastic snake and a big toy frog for the kids. She can scoop these things into three red bags, and we will go meet her Wednesday at the train station.

Since the speakers all read the same short paragraph in English, this is a good dataset to use to analyze the differences in how a speech recognition model performs in response to various accents.

III. Methodology

To install Vosk I followed the installation instructions on this website https://alphacephei.com/vosk/install. I used the pip environment to install Vosk with the command ¡pip3 install vosk¿. Both Python and pip need to be up to date to run this. Next, go to the vosk-api git repository, https://github.com/alphacep/vosk-api. I wrote in Python, so in the Python folder find examples, then test_ffmpeg.py. ffmpeg is a method to convert video and audio files, and it is necessary since the dataset consists of mp3 files. Put test_ffmpeg into a folder in your local environment then download the model and unpack it as 'model' in that same folder(download from https://alphacephei.com/vosk/models). To run use the code ¡ python3 test_ffmpeg.py file.mp3¿

In this project, I tweaked the test_ffmpeg.py code to take in a directory and loop through each recording in the directory. Additionally, the original code printed out partial transcriptions as the model went, and I changed it to only give the final text. The code I ran is ¡ python3 my_test.py recordings ¿ filename.txt ¿ which then creates a text file that is formatted like a dictionary with keys being the name of the audio file and values being the transcription. Once done, each entry should look like this,



```
{"kikongo1.mp3": "these still ask her to bring these things we hear from the store six
bowl of fresh snow peas five the slaves of blue cheese and maybe a snake far broader ball
for our one we also need a small plastic snake and a big toy frog four kids she can scoop
these things into three red base and will and we will go later wednesday at the train
station",
```

Fig. 1. Transcription Example

Before analyzing the metrics some processing needs to be done. The model includes possessives and common contractions, so I took out the apostrophes and replaced the contractions with the fully written out form (Ex. replaced don’t with do not). Additionally, I actual passage each participant read and removed capital letters, punctuation and trimmed white space so it could be compared with the transcriptions.

I used the python package JiWER to calculate word error rate (WER), match error rate (MER) and word information lost (WIL). WER is the proportion of word errors to words processed. Essentially,

$$\frac{\text{Substitutions + Insertions + Deletions}}{\text{Number of Words Spoken}}$$

MER is the probability of a given match being incorrect and WIL is a simple approximation to the proportion of word information lost. All three values are between zero and one with zero indicating perfect performance (no errors from the ASR model). The word error rate is most commonly used in ASR assessment, however as argued by Meyer and Green , ”WER measure is ideally suited only to Connected Speech Recognition (CSR) applications where output errors can be corrected by typing. For almost any other type of speech recognition system a measure based on the proportion of information communicated would be more useful.” [5]. Hence the introduction of the other performance calculations.

IV. Results and Analysis

Since I was interested in instances where the model failed to perform well, I chose to analyze the worst 100 data points. Beginning with WER, the worst 100 transcriptions are summarized in the graph below

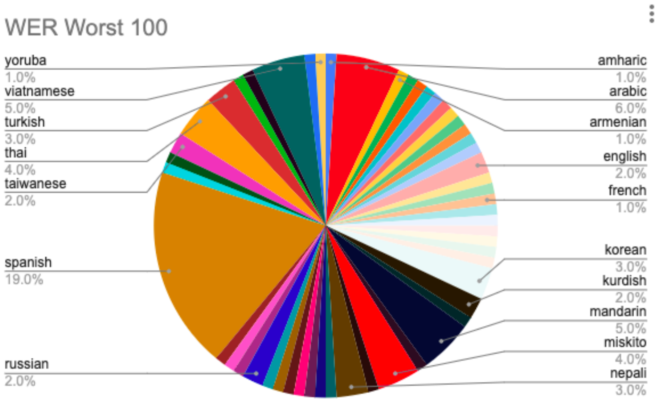


Fig. 2. Word Error Rate: Worst 100

There are 17 different languages represented in this pie chart, so clearly there was not a strong consensus of poor performance around any particular language. However, Spanish has a higher percentage than expected. 19% of data points in the top 100 worst in terms of word error rate are transcriptions from Spanish speakers. This is surprising since Spanish and English are relatively similar languages; they share many similar words, similar sounds, and have the same alphabet.

When looking at MER, the same patterns appear. Again 17 languages are represented (although the languages differ), and Spanish has the highest percentage.

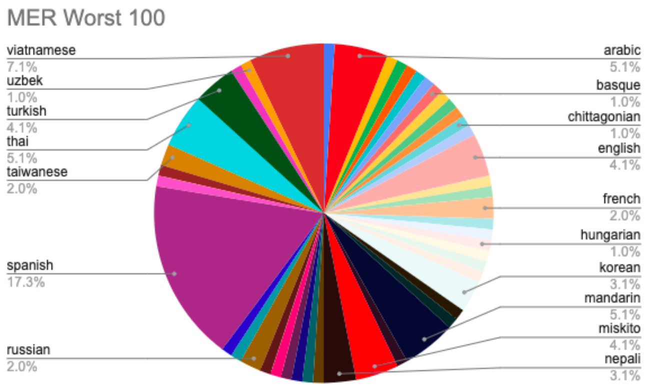


Fig. 3. Match Error Rate: Worst 100

However, this can be accounted for if we look into the dataset itself. Although there are over 200 languages represented, they are not represented equally in the data. The language with the most speakers in the set is by far English with 579 instances. As predicted, the model does very well with English, the top 100 for all three metrics have majority English speakers including many scores of 0 indicating a perfect match between transcription and the passage being read. After English, Spanish is the second most represented language with 162 instances. From there most languages appear less than 80 times and many even less than 10. This over representation of Spanish as compared to the other languages in the dataset most likely accounts for the large percentage we see in the worst 100 WER and MER performances.

Since so many of the languages in the charts above appear only once, I proposed another way of visualizing the data. Instead, we could group languages by language family which is a concept from linguistics which captures the ancestry of languages. Languages that come from the same root are in the same language family. If we graph the Word Error Rate Worst 100 in this way, we get a new graph.

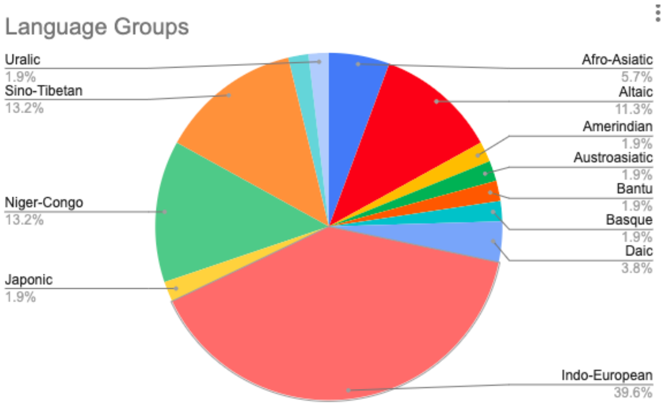


Fig. 4. Word Error Rate: Worst 100 by Language Family

Here we can see that Indo-European languages account for a little over a third of the languages in the worst 100. However again, this is probably due to the fact that Indo-European languages are over represented in the data set. Indo-European is the largest language family with 425 and has native speakers in all six inhabited continents. Additionally, Indo-European is a very broad category that includes the Italic (romance) languages such as Italian, Spanish and French, Germanic languages, Slavic, Indo-Iranian, Celtic, Baltic and more [6]. In future work, we could sort language families down into smaller groups to research similar languages more closely.

To help visualize the breadth of the Indo-European languages, we can look at the following Language Family map where Indo-European is represented in red.

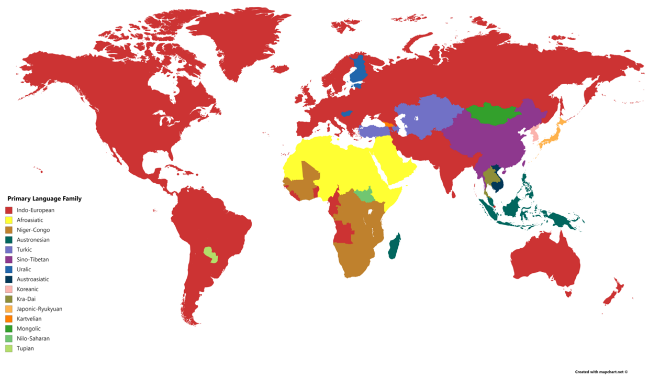


Fig. 5. Map of Language Families

V. Conclusion

This project has many avenues for future work:

1) Explore the data more: calculate different metrics, look at best 100, make a graphs from full data set.
2) Analyze Language Families more deeply: break language family down into smaller sub groups, research connections between languages that could account for model performance.
3) Analyze which words and sounds are most commonly mis-transcribed by the ASR model to propose improvements for model.

In summary, I used the automatic speech recognition software Vosk on a large dataset of accented speech samples. Then I analyzed the performance using word error rate, match error rate and word information lost. The results were somewhat surprising but inconclusive. More analysis would need to be done into which languages cause the worst performance for this model and why in terms of linguistic features such as specific phoneme pronunciation before improvements to the model can be proposed

References

[1] H.-p. Shen and et al., "Model generation of accented speech using model transformation and verification for bilingual speech recognition," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 14, no. 2, pp. 1–24, 2004.

[2] A. Glasser, "Automatic speech recognition services," *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019.

[3] e. a. Haridas, Arul Valiyavalappil, "Ta critical review and analysis on techniques of speech recognition: The road ahead," *International Journal of Knowledge-Based and Intelligent Engineering Systems*, vol. 22, no. 1, pp. 39–57, 2018.

[4] e. a. Radzikowsk, Kacper, "Support software for automatic speech recognition systems targeted for non-native speech." *Proceedings of the 22nd International Conference on Information Integration and Web-Based Applications amp; Services*, 2020.

[5] A. C. Morris, V. Maier, and P. D. Green, "From WER and RIL to MER and WIL: improved evaluation measures for connected speech recognition," in *INTERSPEECH 2004 - ICSLP, 8th International Conference on Spoken Language Processing, Jeju Island, Korea, October 4-8, 2004*. ISCA, 2004. [Online]. Available: http://www.isca-speech.org/archive/interspeech_2004/i04_2765.html

[6] "Language families, language family groups, subgroups of languages." [Online]. Available: http://www.italiantechnicaltranslations.com/language-family-groups.htm