Predicting a Speaker's Accent and Origin

Mingyang Cai, Eric Liobis, Brad Pender, Garrett Wight

November 2019

Abstract

The goal of this project is to use recordings of people speaking to train a model that can predict the speaker's accent origin. We will be using an existing language processing network and transfer learning in order to train the model on the new data set. We will be using a Kaggle data set that contains a speech accent archive data as the primary information for training the model.

1 Introduction

Our inspiration for this project came from a data set on Kaggle called the Speech Accent Archive. We initially wanted to predict what type of accent a speaker had but decided to focus on where the speaker was from. We focused our model on trying to predict the latitude and longitude of the speakers origin using spectrogram images of each speakers audio file.

2 Custom Loss Function

For our problem we initially thought that a typical loss function would not be optimal. We are not measuring distance on a plane but distance on a sphere. Mean Squared Error can be helpful in setting a good target but a more accurate metric would be to define a loss function based on the distance on a sphere. We constructed a custom loss function in tensorflow that calculated the distance on a sphere and optimized over that.

We later determined that the custom loss function was not actually as successful as a standard MSE loss function however. We initially thought that was due to an error in our custom function, but after further consideration, it did make some sense that MSE would lead to a better model. Our reasoning for this is that our custom loss function was, by design, not uniform. As such, it did not give the best metric for determining closeness of two coordinates. There may still be a small error with our custom loss function, but after some extensive testing, it seemed to be behaving appropriately.

3 Transfer Learning Model Approach

Our first approach involved using a VGG16 model and utilizing transfer learning. We removed the top layers and replaced them with a regression layer to predict latitude and longitude coordinates. We used Adam as the optimizer with a learning rate of 0.00001. The top layer is replaced with a 16-output Dense layer that uses ReLU as an activation function and a 2-output Dense layer into that becomes our latitude/longitude prediction. We trained with 100 epochs and a 0.2 validation split.

3.1 Data Manipulation

The data set includes English speech samples from 177 different countries. All of the speakers read the same text which provides continuity between the different accents. The data includes mp3 files for each of the speakers and a .csv file with information about their age, birthplace, native language, country, and sex.

We downloaded the mp3 data and converted the mp3's to wav files for easy use in python. We then created a script to convert the wav files to spectrogram images. This was done using a mel scale for the spectrogram which is a perceptual scale of pitches that is equal in distance from one another when heard by humans. The spectrograms were generated using the librosa library functions for melspectrograms. All of the images generated were the same height (256 pixels) and the same distribution of frequencies. The length of the image depended upon the length of the wav files. To achieve continuity in our set of images, we resized all of the images to a 256 by 256 pixel image to compare the speech patterns. We recognize that the length of the file can be a feature for recognizing where the speaker is from but we decided not to include it initially.

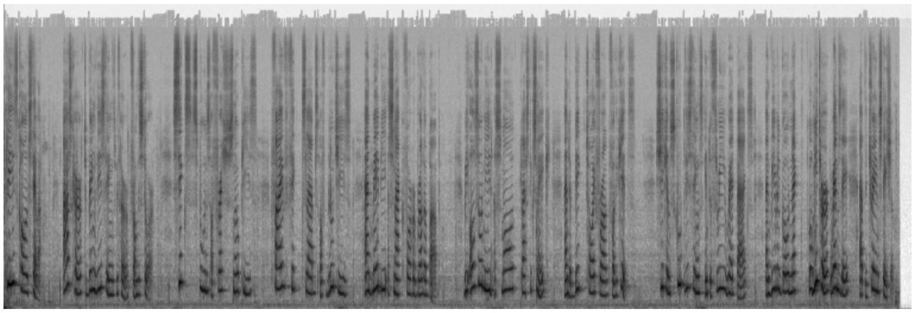

Figure 1: Spectrogram representation of a .wav file

For our output, we first looked at a classification problem where we tried to predict the country of origin of the speaker. Our dataset however was skewed with only 1 or 2 points for some countries and hundreds for others. Also, some countries (such as the US) have a large range of accents that such a varied dataset would lead to higher error. Instead we turned it into a regression problem. With the place of origin we ran a script to get the latitude and longitude for each data point. The project goal was to then predict the parameters and minimize the distance on a sphere between the actual and predicted values in the test set.

3.2 Results

The Transfer Learning Model was unfortunately not as effective as we had hoped and resulted in about a 5000 kilometer validation error. The RMSE was about 52.37 degrees. Figure 2 below shows the evolution of the loss function over the epochs. As you can see, the training loss keep going down because of our massive imbalance between parameters and the size of the training set while the validation level out. We guess that this large of an error is primarily due to the small quantity of data we had. We only had 2,138 spectrogram images but many more trainable parameters to work with. Data augmentation would also likely have been unsuccessful due to the nature of spectrogram images.

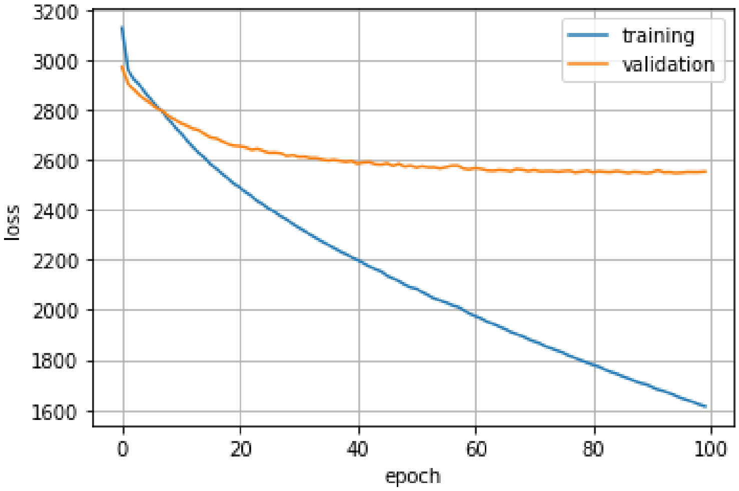

Figure 2: Loss trends over 100 epochs

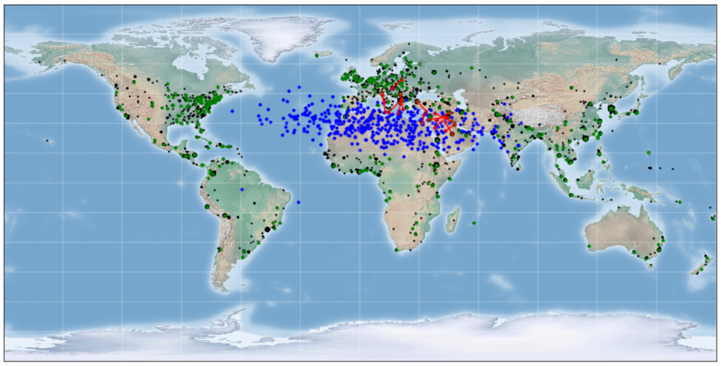

Figure 3: Predictions within 1000 km (Black are training data, green are test data, blue is predictions)

For our baseline we used the mean of all the training data as a simple bias regressor. With this we received an RMSE of 54.24. So while our model performed better than a simple baseline, it only did so slightly.

4 Recurrent Neural Network Model Approach

Recurrent Neural Networks (RNN) are known to work well with speech recognition tasks [1][2]. Previous researches have also shown it's ability on accent recognition tasks [3]. Therefore, we decided to try reproducing the RNN part in [3].

4.1 Limitations

The first thing we weren't able to perform is voice activity detection (VAD). Pre-processing speech signals can help removing unintended pauses during the speeches, so that the speeches can better align with each other. However, there isn't any off-the-shelf solution of VAD, and constructing VAD by hand is beyond our knowledge and time. We've also considered segmenting the speeches into words and perform word-to-word comparisons. However, speech segmentation turned out to be a complicated problem that is beyond our knowledge, and the current solutions do not provide perfect results [4].

4.2 Pre-Processing

Each signal of about 20 seconds was first re-sampled to 131,072 samples. Then, the signal was split into 128-sample windows. Short-term features were extracted from each 128-sample frame. Every signal was then normalized to -1.0 to 1.0 range.

4.3 Recurrent Neural Network

We trained the RNN on the short-term features from those 128-sample frames. The predictions of all frames of each signal are averaged to get the prediction of the signal. Similar to the model in [3], our RNN has a structure as follows: Input data get sequentially fed into the RNN frame-by-frame. Two hidden layers with 512 long short term memory (LSTM) nodes were used. In each LSTM node, there is a cell state regulated by a forget gate, an input gate and an output gate. The activation function for the gates was a 'logistic sigmoid' and for updating the cell state we used a 'tanh'. The accent label was assigned to every 128-sample frame. LSTM nodes take the outputs of the previous frame as an input for the current frame, which allows the network to learn long-term features. As a result, the network should be able to learn both differences in articulation and differences in how articulation changes over time for different accents. For training, we set the dropout probability to 0.5 [5]. The optimizer we used was the RMSProp algorithm with a learning rate of 0.001 and a batch size of 128 signals.

4.4 Results

Despite the success in [3], we weren't able to train a network that can effectively predict origins of speakers. The model stopped improving after about 100 epoch with about 5000 km as the loss. Further examination of the predictions reveals that the model essentially makes the same prediction regardless of the inputs. We believe that the biggest reason of such disappointing result is our lack of training signals. We only have about 2000 speech signals in total, which means that the samples for training is only about 1500. Also, failing to perform VAD in pre-processing may also made our input data more chaotic, which raises the difficulty of learning for the network.

5 Conclusion

Overall, our network only performed slightly better than the baseline. There are many factors that could have led to this. Our model was prone to overfitting and adding regulatization would have certainly helped capture this. In addition, we could have cut the data into smaller sample so we could drastically increased the size of the data set and the resolution on each individual sample. Using MSE as a loss function also worked sufficiently fine, however, it did not take into account the spherical nature of the earth and the prediction algorithm was certainly limited by that.

References

[1] H. Zen and H. Sak, "Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis," in

2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2015, pp. 4470–4474.

[2] Y. Jiao, M. Tu, V. Berisha, and J. Liss, "Online speaking rate estimation using recurrent neural networks," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2016, pp. 5245–5249.

[3] Y. Jiao, M. Tu, V. Berisha, and J. M. Liss, "Accent identification by combining deep neural networks and recurrent neural networks trained on long and short term features." in *Interspeech*, 2016, pp. 2388–2392.

[4] A. H. H. N. Torbati, J. Picone, and M. Sobel, "Speech acoustic unit segmentation using hierarchical dirichlet processes." in *INTERSPEECH*, 2013, pp. 637–641.

[5] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *arXiv preprint arXiv:1409.2329*, 2014.