Author Identification With Varying Clip Lengths

Lucas Croslyn and Tyler Gosbee
St. Francis Xavier University
Department of Computer Science
Antigonish, Nova Scotia, Canada
x2019dvh@stfx.ca and x2019fep@stfx.ca

Abstract

The motivation for this paper was to evaluate the effectiveness of various models on identifying authors based on sampled audio data. The audio data consisted of 50 different authors, generally taken from public domain audio book recordings. Each author had approximately 1 hour worth of audio, split into 1 minute clips. The audio data was tested with an LSTM based model on the direct audio data, as well as 2 models were tested on the text extracted from the audio samples. The goal of this analysis was to demonstrate the effectiveness of the various models as the length of the data analyzed varies.

1 Introduction

For this project, we decided to tackle the problem of author identification. Author identification is the process of taking either audio data or text data and trying to predict who said/wrote it. This is a big problem recently with it being easier and easier to fake information online. With the rise of AI voice generation even spoken words are not fully enough to know if a person actually said something which can pose not just misinformation but potentially security issues as well.

1.1 Previous Work

One paper that was done on the topic of speaker identification was by Reynolds (1995). In this paper, the datasets used were different to ours. The datasets used in the paper were the TIMIT, NTIMIT, Switchboard and YOHO. The TIMIT was the ideal dataset where there was no noise in the data and the quality was controlled. The NTIMIT had the same sentences and speakers but the quality of the data was worse, there was noise and microphone variability. The Switchboard had different data but was of similar quality to the NTIMIT data while the YOHO data was another set of speakers and it seems like the words were predefined and had mostly overlap between the speakers but was not specified in too much detail, it also had near ideal audio quality.

The model that they utilized in the paper was something called a Gaussian Mixture Model. However, before they used it, they transformed the audio data into a different scale. The audio was put into what is known as a mel-scale and the mel-scale cepstral coefficients are utilized for the models since it allows them to do feature reduction. They then made a feature vector to use in the model. The Gaussian Mixture Model itself will do predictions based on the maximum likelihood that the test sample is any given author based on the training data.

The paper's analysis involved an investigation into how well their model worked against each of the datasets described before. Each dataset was tested on an increasing number of speakers. Verification was also done on a new set of samples and an analysis was done to see if males or females were identified better. The TIMIT had the best accuracy overall and the YOHO dataset also had very high accuracy. This was not too surprising as the audio quality being better should lead to higher accuracy. The NTIMIT and Switchboard datasets performed a lot worse overall than the other models, which, again is not surprising. One of the interesting results was that as the number of speakers increased, the accuracy of the NTIMIT and Switchboard data actually decreased by a fair amount while the TIMIT accuracy stayed the same (The YOHO dataset was not tested for this). The NTIMIT dataset especially had a hit in quality going from 10 test speakers to 100 as it started $>90\%$

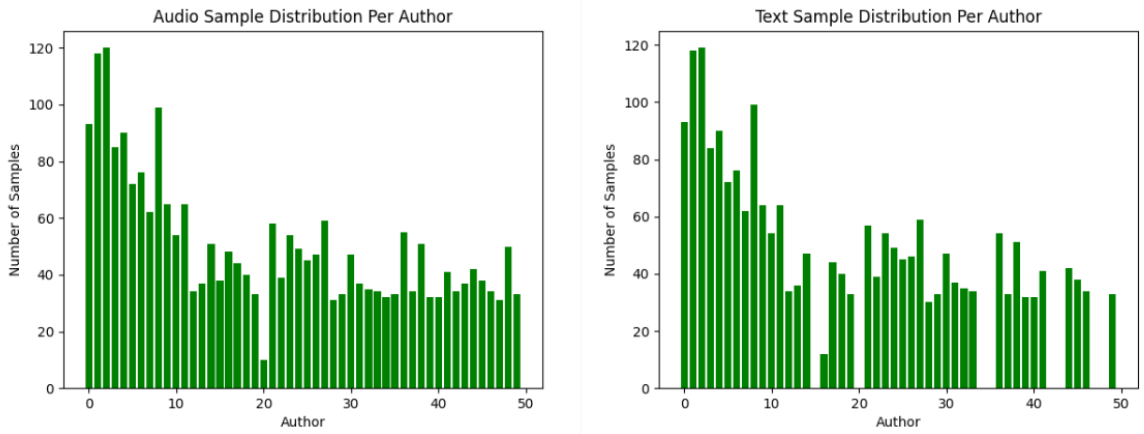

Figure 1: Author Distributions

accuracy down to <80% while the TIMIT dataset stayed at near perfect (>99%) always. This shows that the quality of the audio data being used can impact greatly how well a model will perform, even if that model does well overall.

The other main results discussed in the paper is that their results during verification showed that the model was worse at predicting the female speakers over the male speakers for each dataset. While it wasn't discussed in detail why, it could be possibly because there were more males than females in most of the datasets causing a data imbalance.

2 Dataset

The dataset we chose came from Jain (2019). It is composed of 50 different speakers who have approximately an hour of audio each. Each sample is about a 1 minute long section of the hour. When reading the audio data into python, some corruption issues arose, which meant we could not use the full dataset. The data we were able to use was composed of 2511 total samples with a varying amount from each speaker. The audio for the dataset was clear as the audio seemed to come from audiobooks and educational videos. There was a mix of masculine and feminine voices and the speakers had a variety of accents but all spoke English.

2.1 Preprocessing

The audio dataset was split into 80% training and 20% validation. The data was also transformed into text from the audio utilizing Google's speech recognition API which caused any punctuation to not be included. Some of the audio samples were not transformed due to issues with the API and were thrown out but only for the text data, the audio samples were kept. The text data only had 42 of the speakers, with now only 2196 total samples. The final distributions can be seen in Figure 1. Generally, as can be seen in the distributions, in the case where any text samples were lost, all the samples for the author were lost.

The audio data was then compressed from 44kHz into 11kHz so that training/testing with the audio data would not cause memory issues. The text data was also tokenized using the NLTK tokenizer so that it could be used in certain learning models easier.

3 Models

3.1 Model 1

The first model that we utilized was on the converted text inputs. This model is a lexicon type model where there is a list of tokens associated with each speaker. The training consists of going through each speaker's tokenized documents and creating a set of each unique token seen overall for that author. The sets between authors can have duplications for words, especially stop words which were not excluded. Given some test document, the prediction will be the author which had the most number of tokens that were in the test document.

3.2 Model 2

The second model that was used was a similarity based model. We generated what can be known as a binarized author-term matrix initially for the training. Each row of the matrix was for each dif-

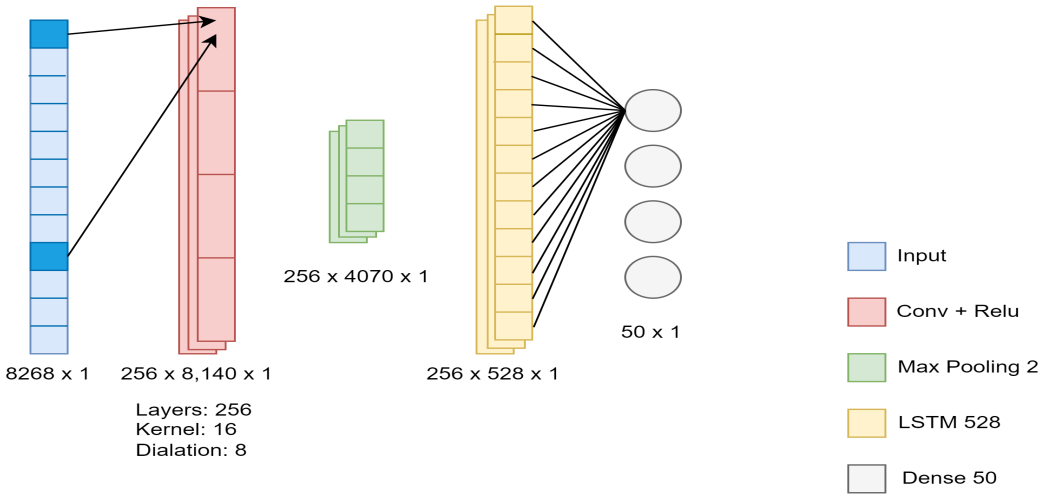

Figure 2: LSTM Model Architecture

ferent author we had. The columns were for the unique tokens seen. The cell was the number of training documents from that author that contained that token. For example, if author 1 said the word "normal" only one time in ten different training documents, cell (Author 1, normal) would be 10 while if author 2 said normal ten times in only one training document, cell (Author 2, normal) would be 1.

For prediction, given some test document with an unknown author, that document will be made into a binarized vector of which tokens appear in it. This is so that the cosine similarity can be calculated between this vector and each row of the author-term matrix described above. The predicted author will be the one that had the highest cosine similarity value to that test document vector.

3.3 Model 3

The third model we utilized was an LSTM based model that was trained on the audio data inputs. This model consisted of 5 layers. The first layer was an input layer for the ∼6s of audio data or 8268 data points in this case. This was followed by a 1d-CNN that iterated over the data with 256 layers, a kernel size of 16 and a dilation of 8. This was then fed through a max pooling layer with a size of 2, dropping half of the data. The reduced data was then fed into the LSTMs with a memory size of 528. Finally the output from the LSTMs were fed into a dense layer with a neuron for each output, in this case 50. Figure 2 shows a visual outline of the LSTM model architecture.

The training data was split with an 80/20 distribution for training and testing respectively, in order to ensure that overfitting did not occur as we were training the model. The testing data was then cut into the appropriate size clips of ∼6s in order to train the model with manageable memory usage. The training was repeated for 15 epochs over the ∼5800 small clips, before the model began overfitting and the training was stopped.

Testing was then able to be performed by splitting a validation clip into the appropriate size clip, or further the longer sample could be fed in to be predicted by splitting the sample and averaging the predictions.

4 Results

For our results, we decided to see how our accuracy for each model would change if each testing sample was split into smaller samples. In order to accomplish this, we first combined all of the audio/text data for an author's verification data into one file. Then, it was split into an appropriate number of samples to represent the ∼6s and ∼30s samples (2 times previous number of samples per author for ∼30s samples and 10 times previous number of samples per author for ∼6s samples). This means that the text data was only an approximation of what was said in each of those split

Table 1: Model Results

Model	Recall	Precision	F1-Score	Overall Accuracy
Lexicon ∼6s	0.366	0.495	0.387	0.423
Lexicon ∼30s	0.717	0.781	0.729	0.768
Lexicon ∼60s	0.825	0.862	0.829	0.864
Similarity ∼6s	0.342	0.417	0.351	0.379
Similarity ∼30s	0.739	0.780	0.743	0.778
Similarity ∼60s	0.861	0.880	0.864	0.895
LSTM ∼6s	0.905	0.909	0.905	0.905
LSTM ∼30s	0.957	0.957	0.955	0.958
LSTM ∼60s	0.973	0.977	0.974	0.967

samples, time constraints meant that we could not transform each of the smaller samples again over again. Each of the accuracy scores in Table 1 were the macro averaged versions of that score for.

As can be seen in the Table 1, we see generally the LSTM is the most accurate, followed by the similarity model, followed by the lexicon for most lengths of the validation samples used. In all cases providing more data to a model improved its accuracy, but in the case of the LSTM it was much better at handling the large decrease in data length achieving a better accuracy than both the similarity and lexicon model with full access to the sample.

The ability for the LSTM to handle the difference in length makes sense, as even in just 6 seconds of data, there were still ∼8000 data points to take information from. In the case of the word based models they had to deal with very little data, as only a handful of words would be said within those 6 seconds, drastically affecting the ability to predict the author. The LSTM is also generally a more powerful model than the text based models used as well, leading to the improved performance.

5 Conclusion

In conclusion, the results of our project indicate that the more complex audio-based LSTM model outperformed the other text-based models. When we reduced the size of each of the samples so there is less data per sample, the LSTM was still able to accurately predict who said it while the other models struggled due to the limited text per sample. This means that when given some audio where the speaker is unknown, using a pre-trained LSTM model can be a good option because the testing does not take too much time per sample, although training the LSTM can take longer than the other models.

Looking ahead, it would be interesting to test the performance of the models when dealing with a newly added author who does not have as much training data as the others. This scenario can help assess the ability of the models to generalize to new and unknown speakers. Additionally, testing the models with less training data can provide insights into how well they can learn with limited data, which is a crucial consideration in many real-world applications where obtaining large amounts of training data may not be feasible.

References

Vibhor Jain. 2019. Speaker recognition audio dataset. https://www.kaggle.com/datasets/vjcalling/speaker-recognition-audio-dataset.

Douglas A. Reynolds. 1995. Automatic speaker recognition using gaussian mixture speaker models.