VoiceID: End-to-End Text-Independent Speaker Verification

Piyush Vyas[1], *Darshan Shinde*[2]

[1]Department of Intelligent Systems Engineering, Indiana University, USA
[2]Department of Data Science, Indiana University, USA

piyush@iu.edu, dshinde@indiana.edu

Abstract

In this paper, we introduce an end-to-end speaker verification system that works directly on raw waveforms. We compare our system with a baseline LSTM system proposed in [1] which needs MFCC features as input and maps an utterance to a speaker embedding. We show that the embeddings learned by our end-to-end system are strong representation of speaker specific discriminant features and are competitive to the embeddings learned by other systems which require hand-crafted feature enginerring. We also introduce a limited vocabulary spoken commands dataset for speaker verification, which we name "PS60k". Experiments performed on our PS60k dataset show that the proposed end-to-end model is equally competent to the LSTM baseline model for the text-independent speaker verification task.

1. Introduction

Voice Assistants are ubiquitous, from smartphones to space shuttles and expanding. We all use an intelligent voice assistant in some form. It can be either Google Assistant or Alexa or Siri. Voice assistants have made our life easier than before, but it has also exposed us to several types of malicious attacks. It is easy to fool these assistants by any non-authorized user and instruct them to perform unintended operations like "Text my wife that you're dominating!". Hence, there is a need to develop system which are capable of differentiating authorized users and non-authorized users.

Speaker verification is the process of verifying speaker identity by matching the representation of the incoming test phrase to the small set of speaker-dependent enrollment phrases. When the lexicon of the spoken utterance is constrained to a single word of a phrase across all speakers, then this process is referred to as global password text-dependent speaker identification. Text-dependent speaker verification systems needs keyword-specific data to train the model. Text-dependent speaker verification system needs the speaker to say exactly the enrolled or given password. Whereas Text-independent Speaker Verification is a process of verifying the identity without constraint on the speech content. Compared to the text-dependent speech verification system, it is more convenient because the user can speak freely to the system.

In this project, we are focusing on building a text-independent end-to-end speaker verification system. A system which will take raw waveform as input and predict whether the spoken utterance/phrase belongs to the true speaker or not. The benefit of making the system end-to-end is that it will remove an extra overhead of extracting speech specific features like Short-Term Fourier Transform (STFT), Log-Filter banks, Mel-Frequency Cepstral Coefficients (MFCC) etc. for training the model. The feature extraction process needs more domain-specific knowledge and few model assumptions. Building an end-to-end system, will result in exclusion of the extra overhead of feature extraction. Such approach will result in simple and efficient systems, requiring little domain specific knowledge and making comparatively less model assumptions.

The remainder of the paper is organised as follows. Section 2. discuss about the prior work published by other authors in the literature. Section 3. gives the brief overview of the baseline LSTM architecture proposed in [1] and introduces our proposed end-to-end approach to speaker verification. Section 4 focuses on the PS60k dataset. How we collected the data and processed it. An information about experimental settings and evaluation metrics can be found in section 5. Section 6 talks about the experiment results. We conclude the paper with some final remarks in section 7.

2. Previous Work

Erik Marchi et al.[1] proposed a Long-Short Term Memory based system for speaker verification which takes as input 20 MFCC frames and learns discriminant utterance-level representation of the speech signal while Georg Heigold et al. [2] proposed a similar system that takes as input 40-dim log-filter banks instead of MFFCs. Since both [1] and [2] used their internal datasets for the purpose of training and evaluating their systems, a direct comparision with their systems is not possible. In a different approch, Jee-weon Jung et al.[3] used an hybrid ResNet + LSTM system which requires pre-emphasis of raw wavforms and a pre-training scheme [4] to avoid speaker overfitting.

3. Methods

3.1. Baseline System

Model architecture: The baseline model shown in Figure 1. and proposed by [1] has a single recurrent layer containing 512 hidden LSTM units. While vanilla LSTM layer have multiple outputs at each timestamp, we only connect the last LSTM output to the next fully-connected linear layer of 128 units in order to obtain a single, utterance-level speaker representation. To classify the speaker representations into different speaker classes, another fully-connected linear layer is used which applies a log softmax activation function to it's outputs.

Features: The input to the LSTM is simply the sequence of MFCC frames (20 MFCCs per frame are extracted by a data window of 25 ms, 100 frames per second). Since the utterances in the dataset are of variable lengths, we replicate them till the length of the utterance is not equal to the longest utterance in the entire dataset. Utterances are not replicated at test time.

3.2. Proposed End-to-End System

In the end-to-end model, we are targeting the speaker verification directly from the waveform signal, without any pre-

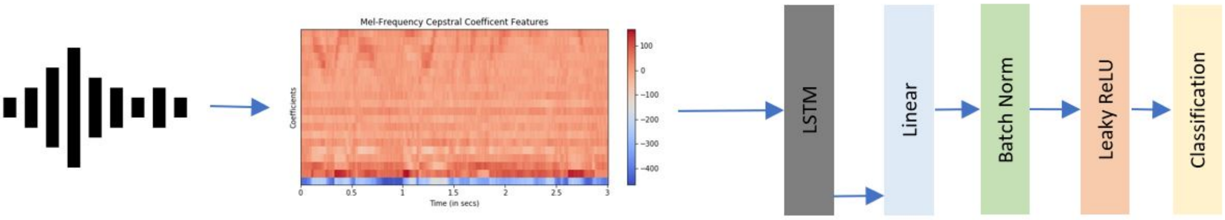

Figure 1: *LSTM baseline model which takes in 20 MFCCs as input features extracted from raw waveforms.*

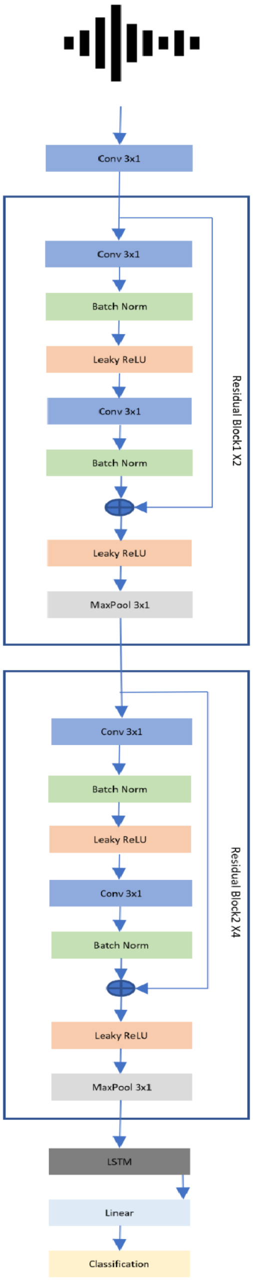

Figure 2: *End-to-End model with residual connections.*

processing of the input data. We use the similar replication scheme here as we did in the LSTM baseline.

Model Architecture: We have designed a DNN model, which comprises a convolution block at the start to extract features. This convolution block contains a 1-d convolution layer (kernel size 3, stride 3, and 128 out channels), a batch normalization layer, and the LeakyReLU activation layer.

There are six residual blocks followed by the first convolution block, which are similar to ResNet. Each residual block is a sequential block of a convolution layer (kernel size 3, output channel 256), batch normalization layer, LeakyReLU activation (negative slope 0.3), a convolution layer, and a batch normalization layer. The convolution layers of the first two residual blocks have 128 output channels. There is an identity connection inside each residual block after the second batch normalization layer.

An LSTM layer (512 hidden units) is then attached to aggregate the frame-level features into a single utterance level representation. We consider only the last output of the LSTM layer which is fed to a fully-connected layer of 128 units. Then, we use a fully-connected layer to transform the 128 dim features to respective classes, to which we apply softmax activation.

We have made several changes to existing RawNet architecture [3]. Instead of using the GRU recurrent layer, we used an LSTM recurrent layer. We also reduced the number of hidden units in the recurrent layer from 1024 to 512 to prevent drastic reduction in features.

4. Dataset

4.1. Data collection

Since we couldn't find any publicly available dataset that met our project requirements we created our own dataset, PS60k. The dataset contains a total of 60k utterances spoken by 60 different speakers from China, India, and the US. In order for the dataset to be balanced and gender unbiased, we recorded equal number of male and female speakers from each nationality. For every speaker in the dataset we have 20 different variable length utterances, 10 containing "Hey Siri" commands and 10 containing "Hey Portal" commands. To make the dataset even more diverse and text-independent, we gave speakers the liberty to speak the commands however they felt comfortable in. All the utterances were recorded using iPhone in a fairly quiet environment at a sampling rate of 48kHz. Additionally, we also recorded 10 different background noises like traffic, subway, cafe (babble), airport announcements, music, bonfire, birds' chirping, flush, shower, and whooshing sound from AC vent using iPhone to add background noise.

4.2. Background Noise Insertion

For each recording, 10 different background noises are inserted and for each noisy utterance the SNR levels are varied at -5, 0, 10, 15 and 25 decibels. A one-hot vector label is associated with each data sample.

5. Experiments and Evaluation Metrics

5.1. Experimental Settings

We train our system to minimize the negative log-likelihood of the softmax distribution:

$$loss((x,y);\theta) = -log\left[\frac{e^{z_s}}{\sum_{k=1}^{K} e^{z_k}}\right]$$

where θ are the network parameters, K is the num of speakers in the dataset, x is the utterance/phrase, y is the target speaker's index, s is the target speaker, z are the unnormalized log probabilities predicted by the linear transformation comprised in the softmax layer: $z_s = {w_s}^T h + b_s$, where w_s and b_s are the weights and the bias, and h is the vector of activations of the last fully-connected layer. Adam optimization algorithm with a learning rate of $1e-3$ and L2 weight decay of $3e-3$ was used to minimize the loss function.

At training time we use a batch size of 128 and split it across two Tesla V100 GPUs for parallel training. We only feed one utterance/phrase at test time to check the performance of the model with variable utterance/phrase lengths.

5.2. Evaluation Metrics

We measure the performance of our speaker verification system as a combination of an Imposter Accept (IA) rate and a False Reject (FR) rate. A False Reject (or Miss) is observed when the test phrase belongs to the speaker but the system rejects it. This sort of error tends to occur more often in acoustically noisy environments, such as in a cafe or during a concert. We report FR's as a fraction of the total number of rejects even when the test phrase was of the speaker. Imposter Accept (IA) occurs when the test phrase is confused to be from the speaker even when it isn't.

In our case, we are comparing all 6k test phrases against the 60 speakers we have in our dataset. The cases in which the system correctly verifies the speaker are considered to be accurate. While the failure cases belong to one of the two following scenarios. 1. When we are testing against a speaker S_i and the system fails to verify the speaker by the test utterance x_i actually spoken by the speaker S_i, then we consider this as a False Reject (FR). 2. When the utterance x_i is wrongly predicted to be of speaker S_i. We consider it as an Imposter Accept (IA). Figure 7 shows more detail about this.

6. Results

The results clearly show that our proposed end-to-end system outperforms the LSTM baseline on our dataset. Figure 3 shows comparison of per batch loss trends where our approach has a lower loss value than the baseline. Even though the baseline model starts with the higher training accuracy, our model catches up with the baseline quickly. This phenomena can be seen in Figure 4. Figure 5 and 6 show that our model again performs better in term of imposter accept rate and false reject rate. For IA and FR, the lower is the better.

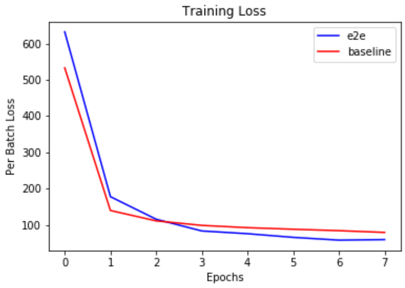

Figure 3: *Per Batch Loss Comparison*

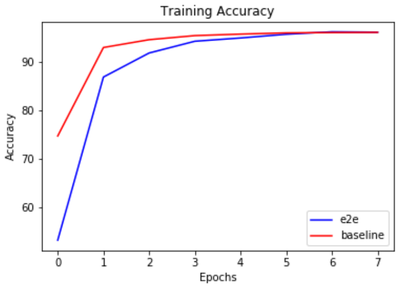

Figure 4: *Train Accuracy Comparison*

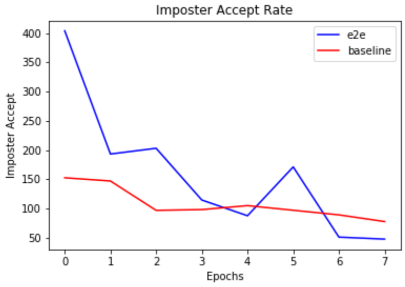

Figure 5: *Imposter Accept Rate Comparison*

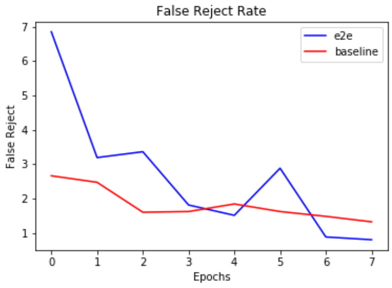

Figure 6: *False Reject Rate Comparison*

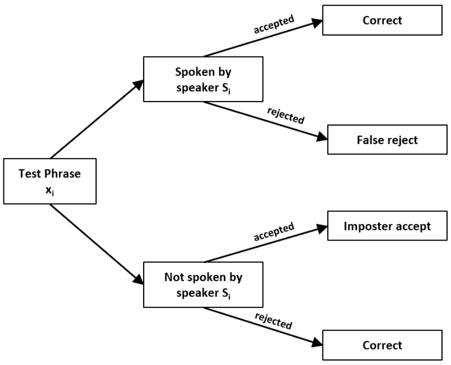

Figure 7: *Imposter Accept (IA) and False Reject (FR) cases*

7. Conclusion

The end-to-end model we proposed in this paper doesn't require any hand-crafted feature engineering, pre-processing such as standardization or normalization and works directly on raw waveforms. The proposed model is able to learn speaker-specific discriminant features that are comparable to the features extracted by the LSTM baseline system. Future work may focus on incorporating more speakers from nationalities not already present in the dataset to make it more generalizable to a diverse group of speakers. The Curriculum Learning [1] based training of the system is something that we look forward to apply during training of our end-to-end model. [3]

8. References

[1] E. Marchi, S. Shum, K. Hwang, S. Kajarekar, S. Sigtia, H. Richards, R. Haynes, Y. Kim, and J. Bridle, "Generalised discriminative transform via curriculum learning for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 5324–5328.

[2] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," 2015.

[3] J.-W. Jung, H.-S. Heo, J.-h. Kim, H.-J. Shim, and H.-J. Yu, "Rawnet: Advanced end-to-end deep neural network using raw waveforms for text-independent speaker verification," 09 2019, pp. 1268–1272.

[4] J. weon Jung, H. soo Heo, I. ho Yang, H. jin Shim, and H. jin Yu, "Avoiding speaker overfitting in end-to-end dnns using raw waveform for text-independent speaker verification," in *Proc. Interspeech 2018*, 2018, pp. 3583–3587. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1608