Submission from CMU towards 1st MultiTarget Speaker Detection and Identification Challenge

SaiKrishna Rallabandi and Alan W Black

Language Technologies Institute, Carnegie Mellon University, PA, USA
{srallaba,awb}@cs.cmu.edu

Abstract

In this document, we present the entry from CMU towards MCE 2018 [1]. This begins with a brief description of the tasks and evaluation framework. We show via experiments that inverted Euclidean distance can be used instead of cosine distance. Based on this, we present a submission that aims to learn a similarity metric using Siamese nets optimized using euclidean distance between the representations.

Index Terms: speech processing, convolutional neural networks, strength of excitation, classification, emotion

1. Introduction

Applications related to speech technology have grown rapidly over the last decade and span both human-human as well as human-machine interactions. The pace at which these applications(such as smart speakers) are being deployed has been on a steady rise and has gone hyperbolic in the last couple of years. On the one hand, these applications are now making a transition from benign and low profile scenarios (such as digital assistant) to high risk scenarios (such as medical transcription analysis, forensics, etc). On the other hand, new application domains are surfacing (such as Google Duplex) since disrupting innovations such as Wavenet[2] and LAS[3] have been pushing core speech technology further and further.

Given this impact, there has been a rise in interest in the research community towards building 'failsafes' against possible abuse of such advances [4, 5]. Multitarget speaker detection and identification challenge [1] is another evaluation in this vein. The aim of this challenge is two fold: (1) Given a test utterance, detect if the utterance was possibly spoken by a speaker from 'blacklist cohort'. (2) If indeed the utterance is from the cohort, identify the speaker.

1.1. Significance of the Challenge

The challenge aims to evaluate how well current machine learning algorithms are able to detect a restricted set of speakers. This task has semblance to a widely deployed module in any sophisticated email program: spam detection. Having said that, spam detection is based on the 'content' of the message while the objective of current challenge is to detect the 'transmitter' of message: speaker. The data used in this version of challenge is from telephone conversations. This means there is a domain match with deployment conditions and hence, provides us relatively more reliable way to interpret (final)model behavior.

2. Statistics and Task Description

In this section we highlight only some important aspects related to data and tasks. For a comprehensive description we refer the readers to [1]. We are given 600 dimensional iVectors for each utterance. Provided data has a train and validation split. Every speaker has one utterance in the validation set. In other words, length of speakers and length of validation set are the same. There are 3631 black listed speakers in total and 5000 non black listed speakers (also referred to as background speakers). Thus, length of validation set is 8631. Each black listed speaker has 3 utterances in the train set. There are more than 4 utterances per each back ground speaker in the train set. The total number of utterances in train set is approx. 41k(10893 + 30952). The test set will have approx. 16k utterances - approximately double the length of validation set.

3. Experiments

In this section, some experimental results are presented. We first describe the experiments performed only on Top S stack detection. Since this can be formulated as a binary classification task, we believe this would serve as a case study to investigate if non linear models are suitable for applying in the context of present task.

3.1. Top S Stack Detector

3.1.1. Baseline

We have implemented a simplistic baseline for this task using Feed Forward architecture. The Network configuration was '512R128R64R2S' where R represents ReLu activation and S denotes sigmoid activation. Baseline was implemented using both Keras [6] and PyTorch[7]. The Equal Error rates from baseline are mentioned in the table 1.

Table 1: *Equal Error Rates from Baselines of Top S Stack Detection*

EER[%]	Keras	PyTorch
Average	1.08	0.6

3.1.2. ***Linearity of Model: Decision Trees and Random Forests vs DNN***

iVectors are derived using a factorization based approach. Therefore we have investigated the performance of our baseline with a linear model: Decision Trees. We have further combined three trees based on feature bagging to form Random Forests, similar to the approach mentioned in [8]. We have conducted 10 trials with each model: Decision Tree and Random Forest for validating this. Table 2 presents the results of this experiment.

An interesting observation surfaced when we looked at the recall scores in addition to EER alone. We consider this important since task 02 depends on the output from task 01. Al-

Table 2: *Equal Error Rates from experiments based on Model Linearity*

EER[%]	Decision Tree	Random Forest
Best	14.32	0.26
Average	15.10	**0.26**

though Random Forest based models showed good EER scores, the blacklist Recall score was poor (0.58).

3.1.3. System Clusterdiff

The idea in this system is that there are some 'types' of speakers. We hypothesize that explicitly factoring these types might help the models better discriminate the blacklist speakers. Since neural networks are known to be powerful feature extractors [9] and combiners, we posit that this might help identify the blacklist cohort. To implement this system, use cluster the 600 dim. iVectors into 64 classes using K Means clustering. We then append an embedding of the cluster identity to the features while training the model. Other approach in the same vein would be to use a bottleneck representation. However, we have not observed much advantage to using bottleneck features in the context of the current task.

3.2. Task 02: Top 1 Stack Detector

3.2.1. Extended Baseline

In this subsection, we describe a minimal modification to the official baseline system. Instead of taking plain cosine distance, we investigate if weighting the cosine distance helps improve the error rate.

For accomplishing this, we follow the approach proposed in [10]: We first calculate weighted distance between each of the iVectors. We then combine inverse of this weighted distance with the cosine distance to formulate a new measure of similarity. Since iVectors are continuous as opposed to discrete document vectors in the context of text classification, we also consider using Euclidean distance instead of hamming distance.

3.2.2. Dimensionality Reduction

Since we have access to a limited number of utterances per speaker, we have investigated if appending the original dimensionality using representation learnt by approaches such as PCA helps.

3.2.3. Siamese Networks

Experiments from the previous subsection prove that we can use Euclidean distance (or) a weighted version instead of pure cosine distance. Inspired by this, we investigate if we can learn the similarity using architectures such as Siamese networks [11]. We have employed a convolutions architecture with 5 layers, each reducing the dimensionality in half except the first layer. The number of neurons in the first layer was 512. We have used a dropout of 0.2 and the network was optimized using SGD with learning rate of 0.01. Learning rate was scheduled using validation loss. Along the same lines, we have also investigated training using a triplet objective [12].

Table 3: *Equal Error Rates from Extended Baseline for Top 1 Stack Detector*

System	Top S	Top 1	Confusion
Baseline	2	13.41	492
Inverse Euclidean * Cosine	1.72	12.78	469
Inverse Euclidean * Cosine No Norm	1.54	13.99	514
Inverse Euclidean	1.74	12.50	459
Inverse Euclidean No Norm	1.54	13.99	514
Inverse Euclidean + Cosine	1.96	13.03	477
Inverse Euclidean + Cosine No Norm	1.54	13.99	514
Baseline + PCA	2.69	12.94	468
Euclidean Baseline + PCA	2.35	12.26	444
Only PCA	2.94	12.62	455
Only PCA + Euclidean	2.48	11.76	424
Bottleneck + Euclidean	11.38	35.8	952
Bottleneck + Cosine	18.84	31.76	924
Siamese Net	**1.69**	**11.24**	**410**
LosslessTriplet Loss	2.35	12.26	444

4. Conclusion

In this document, we present the entry from CMU towards MCE 2018 [1]. We experimentally show that inverted Euclidean distance can be employed as similarity metric. Based on this, we attempt to learn a similarity metric using Siamese architecture.

5. References

[1] S. Shon, N. Dehak, D. Reynolds, and J. Glass, "Mce 2018: The 1st multi-target speaker detection and."

[2] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio." in *SSW*, 2016, p. 125.

[3] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4960–4964.

[4] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, C. Hanilçi, M. Sahidullah, and A. Sizov, "Asvspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[5] Z. Wu, J. Yamagishi, T. Kinnunen, C. Hanilçi, M. Sahidullah, A. Sizov, N. Evans, and M. Todisco, "Asvspoof: the automatic speaker verification spoofing and countermeasures challenge,"

IEEE Journal of Selected Topics in Signal Processing, vol. 11, no. 4, pp. 588–604, 2017.

[6] F. Chollet *et al.*, “Keras,” 2015.

[7] A. Paszke, S. Gross, S. Chintala, and G. Chanan, “Pytorch,” 2017.

[8] A. W. Black and P. K. Muthukumar, “Random forests for statistical speech synthesis,” in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[9] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[10] B. Li and L. Han, “Distance weighted cosine similarity measure for text classification,” in *International Conference on Intelligent Data Engineering and Automated Learning*. Springer, 2013, pp. 611–618.

[11] R. Hadsell, S. Chopra, and Y. LeCun, “Dimensionality reduction by learning an invariant mapping.” IEEE, 2006, pp. 1735–1742.

[12] M.-O. Arsenault, “Lossless triplet loss https://towardsdatascience.com/lossless-triplet-loss-7e932f990b24?gi=43d5e9be22a.”