IberSPEECH 2018
21-23 November 2018, Barcelona, Spain

Differentiable Supervector Extraction for Encoding Speaker and Phrase Information in Text Dependent Speaker Verification

Victoria Mingote, Antonio Miguel, Alfonso Ortega, Eduardo Lleida

ViVoLab, Aragón Institute for Engineering Research (I3A), University of Zaragoza, Spain
{vmingote,amiguel,ortega,lleida}@unizar.es

Abstract

In this paper, we propose a new differentiable neural network alignment mechanism for text-dependent speaker verification which uses alignment models to produce a supervector representation of an utterance. Unlike previous works with similar approaches, we do not extract the embedding of an utterance from the mean reduction of the temporal dimension. Our system replaces the mean by a phrase alignment model to keep the temporal structure of each phrase which is relevant in this application since the phonetic information is part of the identity in the verification task. Moreover, we can apply a convolutional neural network as front-end, and thanks to the alignment process being differentiable, we can train the whole network to produce a supervector for each utterance which will be discriminative with respect to the speaker and the phrase simultaneously. As we show, this choice has the advantage that the supervector encodes the phrase and speaker information providing good performance in text-dependent speaker verification tasks. In this work, the process of verification is performed using a basic similarity metric, due to simplicity, compared to other more elaborate models that are commonly used. The new model using alignment to produce supervectors was tested on the RSR2015-Part I database for text-dependent speaker verification, providing competitive results compared to similar size networks using the mean to extract embeddings.

Index Terms: Text Dependent Speaker verification, HMM Alignment, Deep Neural Networks, Supervectors

1. Introduction

Recently, techniques based on discriminative deep neural networks (DNN) have achieved a substantial success in many speaker verification tasks. These techniques follow the philosophy of the state-of-the-art face verification systems [1][2] where embeddings are usually extracted by reduction mechanisms and the decision process is based on a similarity metric [3]. Unfortunately, in text-dependent tasks this approach does not work efficiently since the pronounced phrase is part of the identity information [4][5]. A possible cause of the imprecision in text-dependent tasks could be derived from using the mean as a representation of the utterance as we show in the experimental section. To solve this problem, this paper shows a new architecture which combines a deep neural network with a phrase alignment method used as a new internal layer to maintain the temporal structure of the utterance. As we will show, it is a more natural solution for the text-dependent speaker verification, since the speaker and phrase information can be encoded in the supervector thanks to the neural network and the specific states of the supervector.

In the context of text-independent speaker verification tasks, the baseline system based on i-vector extraction and Probabilistic Linear Discriminant Analysis (PLDA) [6][7] are still among the best results of the state-of-the-art. The i-vector extractor represents each utterance in a low-dimensional subspace called the total variability subspace as a fixed-length feature vector and the PLDA model produces the verification scores. However, as we previously mentioned, many improvements on this baseline system have been achieved in recent years by progressively substituting components of the systems by DNNs, thanks to their larger expressiveness and the availability of bigger databases. Examples of this are the use of DNN bottleneck representations as features replacing or combined with spectral parametrization [8], training DNN acoustic models to use their outputs as posteriors for alignment instead of GMMs in i-vector extractors [9], or replacing PLDA by a DNN [10]. Other proposals similar to face verification architectures have been more ambitious and have trained a discriminative DNN for multiclass classifying and then extract embeddings by reduction mechanisms [11] [12], for example taking the mean of an intermediate layer named usually bottleneck layer. After that embedding extraction, the verification score is obtained by a similarity metric such as cosine similarity [11].

The application of DNNs and the same techniques as in text-independent models for text-dependent speaker verification tasks has produced mixed results. On the one hand, specific modifications of the traditional techniques have been shown successful for text-dependent tasks such as i-vector+PLDA [13], DNNs bottleneck as features for i-vector extractors [14] or posterior probabilities for i-vector extractors [14][15]. On the other hand, speaker embeddings obtained directly from a DNN have provided good results in tasks with large amounts of data and a single phrase [16] but they have not been as effective in tasks with more than one pass phrase and smaller database sizes [4][5]. The lack of data in this last scenario may lead to problems with deep architectures due to overfitting of models.

Another reason that we explore in the paper for the lack of effectiveness of these techniques in general text-dependent tasks is that the phonetic content of the uttered phrase is relevant for the identification. State-of-art text-independent approaches to obtain speaker embeddings from an utterance usually reduce temporal information by pooling and by calculating the mean across frames of the internal representations of the network. This approach may neglect the order of the phonetic information because in the same phrase the beginning of the sentence may be totally different from what is said at the end. An example of this is the case when the system asks the speaker to utter digits in some random order. In that case a mean vector would fail to capture the combination of phrase and speaker. Therefore one of the objectives of the paper is to show that it is important to keep this phrase information for the identification process, not just the information of who is speaking.

In previous works we have developed systems that need to store a model per user which were adapted from a universal background model and the evaluation of the trial was based on

10.21437/IberSPEECH.2018-1

a likelihood ratio [17][18]. One of the drawbacks of this approach is the need to store a large amount of data per user and the speed of evaluation of trials, since likelihood expressions were dependent on the frame length. In this paper, we focus on systems using a vector representation of a trial or a speaker model. We propose a new approach that includes alignment as a key component of the mechanism to obtain the vector representation from a deep neural network. Unlike previous works, we substitute the mean of the internal representations across time which is used in other neural network architectures [4][5] by a frame to state alignment to keep the temporal structure of each utterance. We show how the alignment can be applied in combination with a DNN acting as a front-end to create a supervector for each utterance. As we will show, the application of both sources of information in the process of defining the supervector provides better results in the experiments performed on RSR2015 compared to previous approaches.

This paper is organized as follows. In Section 2 we present our system and especially the alignment strategy developed. Section 3 presents the experimental data. Section 4 explains the results achieved. Conclusions are presented in Section5.

2. Deep neural network based on alignment

In view of the aforementioned imprecisions in the results achieved in previous works for this task with only DNNs and a basic similarity metric, we decided to apply an alignment mechanism due to the importance of the phrases and their temporal structure in this kind of tasks. Since same person does not always pronounce one phrase at the same speed or in the same way due to differences in the phonetic information, it is usual that there exists an articulation and pronunciation mismatch between two compared speech utterances even from the same person.

In Fig. 1 we show the overall architecture of our system, where the mean reduction to obtain the vector embedding before the backend is substituted by the alignment process to finally create a supervector by audio file. This supervector can be seen as a mapping between an utterance and the state components of the alignment, which allows to encode the phrase information. For the verification process, once our system is trained, one supervector is extracted for each enroll and test file, and then a cosine metric is applied over them to achieve the verification scores.

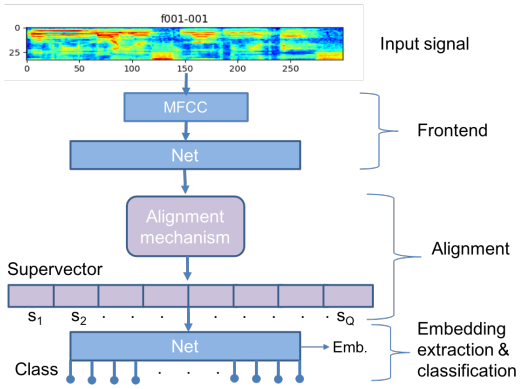


Figura 1: *Differentiable neural network alignment mechanism based on alignment models. The supervector is composed of Q vectors s_Q for each state.*

2.1. Alignment mechanism

In this work, we select a Hidden Markov Model (HMM) as the alignment technique in all the experiments, but other possibilities could be to select Gaussian Mixture Model (GMM) or DNN posteriors. In text-dependent tasks we know the phrase transcription which allows us to construct a specific left-to-right HMM model for each phrase of the data and obtain a Viterbi alignment per utterance.

One reason to employ a phrase HMM alignment was due to its simplicity for training independent HMM models for different phrases used to develop our experiments without the need of phonetic information for training. Another reason was that using the decoded sequence provided by the Viterbi algorithm in a left-to-right architecture it is ensured that each state of the HMM corresponds to at least one frame of the utterance, so no state is empty.

The process followed to add this alignment to our system is detailed below. Once models for alignment are trained, a sequence of decoded states γ=$(q_1, ..., q_t)$ where q_t indicates the decoded state at time t with $q_t \in \{1, ..., Q\}$ is obtained. Before adding these vectors to the neural network they are preprocessed and converted into a matrix with ones and zeros in function of its correspondences with the states which makes possible to use them directly inside of the neural network. In this way, we put ones at each state according to the frames that belong to this state as a result of this process, we have the alignment matrix $A \in \mathbb{R}^{T \times Q}$ with its components a_{tq_t}=*1* and $\sum_q a_{tq}$=*1* which means that only one state is active at the same time.

For example, if we train an HMM model with 4 states and we obtain a vector γ and apply the previous transformation, the resultant matrix A would be:

$$\gamma = [1,1,1,2,2,3,3,4] \rightarrow A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1)$$

After this process, as we show in Fig. 2, we added this matrix to the network as a matrix multiplication like one layer more, thanks to the expression as a matrix product it is easy to differentiate and this enables to backpropagate gradients to train neural network as usual. This matrix multiplication allows assigning the corresponding frames to each state resulting in a supervector. Then, the speaker verification is performed with this supervector. The alignment as a matrix multiplication can be expressed as a function of the input signal to this layer x_{ct} with dimensions *(c × t)* and matrix of alignment of each utterance A with dimensions *(t × q)*:

$$s_{cq} = \frac{\sum_t x_{ct} \cdot a_{tq}}{\sum_t a_{tq}} \quad (2)$$

where s_{cq} is the supervectors with dimensions *(c × q)*, where there are q state vectors of dimension c and we normalize with the number of times state q is activated.

2.2. Deep neural network architecture

As a first approximation to check that the previous alignment layer works better than extracting the embedding from the mean reduction, we apply this mentioned layer directly on the

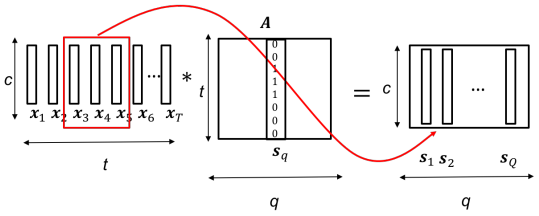


Figura 2: *Process of alignment, the input signal x is multiplied by an alignment matrix A to produce a matrix with vectors s_Q which are then concatenated to obtain the supervector.*

input signal over the acoustic features thus we obtain the traditional supervector. However, we expect to improve this baseline result, so we propose to add some layers as front-end previous to the alignment layer and train them in combination with the alignment mechanism.

For deep speaker verification some simple architectures with only dense layers [4] have been proposed. However, lately it has been tried to employ deep neural networks as Residual CNN Networks [5] but in text-dependent task it has not achieved the same good results as previous simple approaches.

In our network we propose a straightforward architecture with only a few layers which include the use of 1-dimension convolution (1D convolution) layers instead of dense layers or 2D convolution layers as in other works. Our proposal is to operate in the temporal dimension to add context information to the process and at the same time the channels are combined at each layer. The context information which is added depends on the size of the kernel used in convolution layer.

To use this type of layer, it is convenient that the input signals have the same size to concatenate them and pass to the network. For this reason, we apply a transformation to interpolate or fill with zeros the input signals to have all of them with the same dimensions.

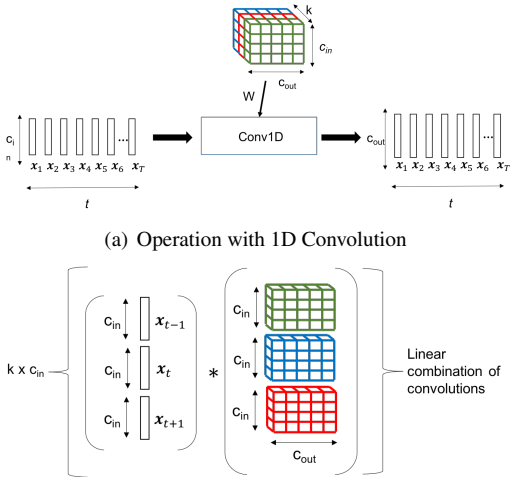
The operation of the 1D convolution layers is depicted in Fig. 3, the signal used as layer input and its context, the previous frames and the subsequent frames, are multiplied frame by frame with the corresponding weights. The result of this operation for each frame is linearly combined to create the output signal.

3. Experimental Data

In all the experiments in this paper, we used the RSR2015 text-dependent speaker verification dataset [19]. This dataset consists of recordings from 157 male and 143 female. There are 9 sessions for each speaker pronouncing 30 different phrases. Furthermore, this data is divided into three speaker subset: background (bkg), development (dev) and evaluation (eval). We develop our experiments in Part I of this data set and we employ the bkg and dev data (194 speakers, 94 female/100 male) for training. The evaluation part is used for enrollment and trial evaluation.

4. Results

In our experiments, we do not need the phrase transcription to obtain the corresponding alignment, because one phrase dependent HMM model has been trained with the background partition using a left-to-right model of 40 states for each phrase. With these models we can extract statistics from each utterance of the database and use this alignment information inside our DNN architecture. As input to the DNN, we employ 20 dimensional Mel-Frequency Cepstral Coefficients (MFCC) with their first and second derivatives as features for obtaining a final input dimension of 60. On these input features we apply a data augmentation method called Random Erasing [20], which helps us to avoid overfitting in our models due to lack of data in this database.



(a) Operation with 1D Convolution

(b) Example of the convolution operation

Figura 3: *Operation with 1D Convolution layers, 3(a) general pipeline of this operation. 3(b) example of how k context frames from input are multiplied by the weight matrix W and the output is equivalent to a linear combination of convolutions.*

On the other hand, the DNN architecture consists of the front-end part in which several different configurations of layers have been tested as we will detail in the experiments, and the second part of the architecture which is an alignment based on HMM models. Finally, we have extracted supervectors as a combination of front-end and alignment with a flatten layer and with them we have obtained speaker verification scores by using a cosine similarity metric without any normalization technique.

A set of experiments was performed using Pytorch [21] to evaluate our system. We compare a front-end with mean reduction with similar philosophy as [4][5] to the feature input directly or a front-end both followed by the HMM alignment. In the part of the front-end, we implemented 3 different layer configurations: one convolutional layer with a kernel of dimension 1 equivalent to a dense layer but keeping the temporal structure and without adding context information, one convolutional layer with a kernel of dimension 3, and three convolutional layers with a kernel of dimension 3.

In Table 1 we show equal error rate (EER) results with the different architectures trained on the background subset for female, male and both partitions together. We have found that, as we expected, the first approach with mean reduction mechanism for extracting embeddings does not perform well for this text-dependent speaker verification task. It seems that this type of embeddings do not represent correctly the information to achieve discrimination between the correct speaker and phrase both simultaneously. Furthermore, we show how changing the typical mean reduction for a new alignment layer inside the DNN achieves a relative improvement of 91.62 % in terms of the EER %.

Nevertheless, these EER results were still quite high, so we decided that the results can be improved training with background and develop subsets together. In Table 2, we can see that if we use more data for training our systems, we achieve better performance especially in deep architectures with more than

Cuadro 1: *Experimental results on RSR2015 part I [19] eval subset, where EER % is shown. These results were obtained by training only with bkg subset.*

Architecture		**Fem**	**Male**	**Fem+Male**
Layers	Kernel			
$FE:3C+mean$	3	11,20 %	12,13 %	11,70 %
$Signal+alig.$	–	1,43 %	1,37 %	1,54 %
$FE:1C+alig.$	1	1,16 %	0,98 %	1,56 %
$FE:1C+alig.$	3	1,04 %	0,77 %	1,20 %
$FE:3C+alig.$	3	0,86 %	1,00 %	0,98 %

one layer, this improvement is observed for both architectures. This fact remarks the importance of having a large amount of data to be able to train deep architectures. In addition, we performed an experiment to illustrate this effect in Fig.4 where we show how if we increase little by little the amount of data used to train, the results progressively improve although we can see that the alignment mechanism makes the system more robust to training data size.

Cuadro 2: *Experimental results on RSR2015 part I [19] eval subset, showing EER %. These results were obtained by training with bkg+dev subsets.*

Architecture		**Fem**	**Male**	**Fem+Male**
Layers	Kernel			
$FE:3C+mean$	3	9,11 %	8,66 %	8,87 %
$Signal+alig.$	–	1,43 %	1,37 %	1,54 %
$FE:1C+alig.$	1	1,17 %	0,98 %	1,55 %
$FE:1C+alig.$	3	1,07 %	0,78 %	1,24 %
$FE:3C+alig.$	3	0,58 %	0,70 %	0,72 %

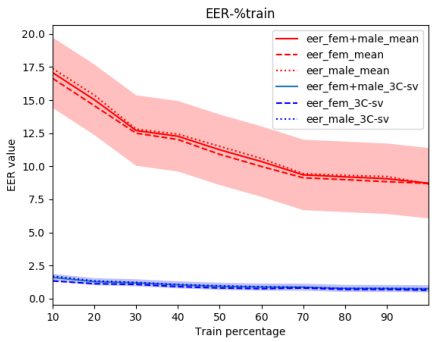
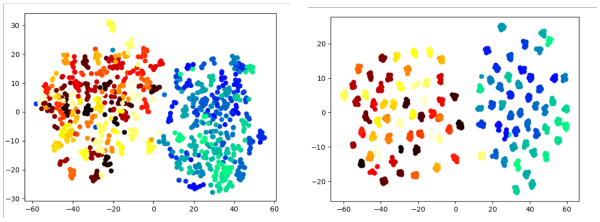


Figura 4: *Results of EER % varying train percentage where standard deviation is shown only for both gender independent results.*

For illustrative purposes, we also represent our high-dimensional supervectors in a two-dimensional space using t-SNE [22] which preserves distances in a small dimension space. In Fig.5(a), we show this representation for the architecture which uses the mean to extract the embeddings, while in Fig.5(b) we represent the supervectors of our best system. As we can see in the second system the representation is able to cluster the examples from the same person, whereas in the first method is not able to cluster together examples from the same person. On the other hand, in both representations data are auto-organized to show on one side examples from female identities

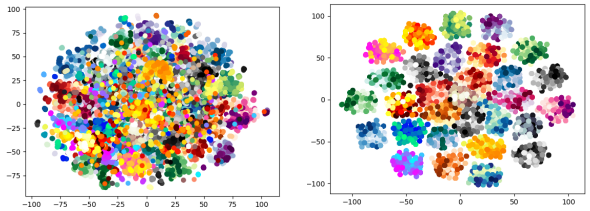


(a) Mean embeddings (b) Supervectors

Figura 5: *Visualizing Mean embeddings vs Supervectors for 1 phrase from male+female using t-SNE, where female is marked by cold color scale and male is marked by hot color scale.*

and on the other side examples from male identities.

Furthermore, we illustrate in Fig.6 the same representation in the previous figure, however in this case we represent the embeddings and the supervectors of the thirty phrases from female identities. With this depiction we checked something that we had already observed in the previous verification experiments since the embeddings from mean architecture are not able to separate between same identity with different phrase and same identity with the same phrase which is the base of text-dependent speaker verification task.



(a) Mean embeddings (b) Supervectors

Figura 6: *Visualizing Mean embeddings vs Supervectors for 30 phrases from female using t-SNE. Each phrase is marked by one different color scale.*

5. Conclusions

In this paper we present a new method to add a new layer as an alignment inside of the DNN architectures for encoding meaningful information from each utterance in a supervector, which allows us to conserve the relevant information that we use to verify the speaker identity and the correspondence with the correct phrase. We have evaluated the models in the text-dependent speaker verification database RSR2015 part I. Results confirm that the alignment as a layer within the architecture of DNN is an interesting line since we have obtained competitive results with a straightforward and simple alignment technique which has a low computational cost, so we can achieve better results with other more powerful techniques.

6. Acknowledgements

This work has been supported by the Spanish Ministry of Economy and Competitiveness and the European Social Fund through the project TIN2017-85854-C4-1-R, by Gobierno de Aragón/FEDER (research group T36_17R) and by Nuance Communications, Inc. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

7. References

[1] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1701–1708, 2014.

[2] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 815–823.

[3] H. V. Nguyen and L. Bai, "Cosine similarity metric learning for face verification," in *Asian conference on computer vision*. Springer, 2010, pp. 709–720.

[4] Y. Liu, Y. Qian, N. Chen, T. Fu, Y. Zhang, and K. Yu, "Deep feature for text-dependent speaker verification," *Speech Communication*, vol. 73, pp. 1–13, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.specom.2015.07.003

[5] E. Malykh, S. Novoselov, and O. Kudashev, "On residual cnn in text-dependent speaker verification task," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10458 LNAI, pp. 593–601, 2017.

[6] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of interspeaker variability in speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 5, pp. 980–988, 2008.

[7] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[8] A. Lozano-Diez, A. Silnova, P. Matejka, O. Glembek, O. Plchot, J. Pešán, L. Burget, and J. Gonzalez-Rodriguez, "Analysis and optimization of bottleneck features for speaker recognition," in *Proceedings of Odyssey*, vol. 2016, 2016, pp. 352–357.

[9] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1695–1699.

[10] O. Ghahabi and J. Hernando, "Deep belief networks for i-vector based speaker recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1700–1704.

[11] G. Bhattacharya, J. Alam, and P. Kenny, "Deep speaker embeddings for short-duration speaker verification," in *Proc. Interspeech*, 2017, pp. 1517–1521.

[12] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Proc. Interspeech*, 2017, pp. 999–1003.

[13] H. Zeinali, H. Sameti, and L. Burget, "Hmm-based phrase-independent i-vector extractor for text-dependent speaker verification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 7, pp. 1421–1435, 2017.

[14] H. Zeinali, L. Burget, H. Sameti, O. Glembek, and O. Plchot, "Deep neural networks and hidden markov models in i-vector-based text-dependent speaker verification," in *Odyssey-The Speaker and Language Recognition Workshop*, 2016, pp. 24–30.

[15] S. Dey, S. Madikeri, M. Ferras, and P. Motlicek, "Deep neural network based posteriors for text-dependent speaker verification," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5050–5054.

[16] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2016-May, no. Section 3, pp. 5115–5119, 2016.

[17] A. Miguel, J. Villalba, A. Ortega, E. Lleida, and C. Vaquero, "Factor Analysis with Sampling Methods for Text Dependent Speaker Recognition," *Proceedings of the 15th Annual Conference of the International Speech Communication Association, Interspeech 2014*, no. September, pp. 1342–1346, 2014.

[18] A. Miguel, J. Llombart, A. Ortega, and E. Lleida, "Tied Hidden Factors in Neural Networks for End-to-End Speaker Recognition," pp. 2819–2823, 2017.

[19] A. Larcher, K. A. Lee, B. Ma, and H. Li, "Text-dependent speaker verification: Classifiers, databases and RSR2015," *Speech Communication*, vol. 60, pp. 56–77, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.specom.2014.03.001

[20] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," *arXiv preprint arXiv:1708.04896*, 2017.

[21] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.

[22] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.