# TEXT ADAPTATION FOR SPEAKER VERIFICATION WITH SPEAKER-TEXT FACTORIZED EMBEDDINGS

*Yexin Yang†, Shuai Wang†, Xun Gong, Yanmin Qian, Kai Yu*

MoE Key Lab of Artificial Intelligence
SpeechLab, Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China
{yangyexin, feixiang121976, gongxun, yanminqian, kai.yu}@sjtu.edu.cn

## ABSTRACT

Text mismatch between pre-collected data, either training data or enrollment data, and the actual test data can significantly hurt text-dependent speaker verification (SV) system performance. Although this problem can be solved by carefully collecting data with the target speech content, such data collection could be costly and inflexible. In this paper, we propose a novel text adaptation framework to address the text mismatch issue. Here, a speaker-text factorization network is proposed to factorize the input speech into speaker embeddings and text embeddings and then integrate them into a single representation in the later stage. Given a small amount of speaker-independent adaptation utterances, text embeddings of target speech content can be extracted and used to adapt the text-independent speaker embeddings to text-customized speaker embeddings. Experiments on RSR2015 show that text adaptation can significantly improve the performance of text mismatch conditions.

***Index Terms—*** speaker verification, text-dependent, text mismatch, adaptation

## 1. INTRODUCTION

Speaker verification aims to verify the client's claimed identity based on his/her speech. Considering the constraint on the speech content, speaker verification can be classified into two categories: text-dependent and text-independent. The former task requires the same speech content for the enrollment and test utterances, while the latter doesn't pose such a requirement, giving users more flexibility.

For the text-independent speaker verification task, the speaker embedding extractor is usually trained on a large amount of unconstrained speech data, the text information is implicitly normalized, which is beneficial since the final speaker embeddings should get rid of the phonetic variability. Despite the good performance on the text-independent task [1, 2, 3, 4, 5], directly applying the same model to the text-dependent task is problematic, for which text information is important. The common method to address such performance degradation is to collect training data which has the same speech content with the evaluation data, and this approach is usually adopted by companies for the wake-up word based speaker verification [6, 7, 8, 9, 10]. However, recollecting application specific training data can be very expensive and inflexible.

In real applications, the challenge not only comes from the text mismatch from the training data and evaluation data but also from the text mismatch between the enrollment and test data in the evaluation. For example, it's common in real applications where users would like to use multiple keywords to wake up smart devices. For example, Google devices allow "OK Google" and "Hey Google" [10, 11]. Some applications even involve more different keywords.

In this paper, to avoid recollecting a large amount of application specific training data, we consider these two kinds of text mismatch in the "text-adaptation framework", in which text-independent speaker embeddings are adapted to customized text-dependent speaker embeddings according to a specific input. We proposed a speaker-text factorization network which contains four parts: one generic feature learner, a speaker sub-net for text-independent "spk" embedding extraction, a text sub-net for "text" embedding extraction, and a "combination" sub-net to learn a text-adapted representation based on the information provided by "text" embedding.

Different evaluation sets considering different types of text mismatch are derived from the RSR2015 [12] dataset, for the traditional text-dependent task where the text mismatch only exists between the training and evaluation data, the "text "embedding is computed from the same utterance as the "spk" embedding. For the case where text mismatch also appears between the enrollment and test data, we collected a very small amount of utterances from arbitrary speakers (different from the evaluation set) to compute the text embedding and adapt the enrollment speaker embeddings. Experimental results show a remarkable performance improvement for both conditions. Furthermore, the "spk" embedding extracted from the speaker sub-net with no text adaptation also outperforms the original x-vector baseline on the standard Voxceleb evaluation set.

## 2. RELATED WORK

### 2.1. x-vector

x-vector [1, 13] is a time-delay neural network (TDNN) based speaker embedding learning framework. The model contains several frame-level time-delay layers, followed by a statistics pooling layer which aggregates the frame-level representation into a single segment-level representation. One or more embedding layers after the pooling layer can be incorporated in the segment-level layers to extract speaker embeddings.

†: These authors have contributed equally to this work

Yanmin Qian and Kai Yu are the corresponding authors

Authors would like to thank Johan Rohdin for providing phoneme labels

This work has been supported by the China NSFC project No. U1736202. Experiments have been carried out on the PI supercomputer at Shanghai Jiao Tong University

### 2.2. Segment-level phonetic label definition

Researchers have investigated integrating the phonetic information into the speaker modeling process, most of which follow the frame-level multi-task learning paradigm [14, 15]. In our previous work, we proposed a framework to consider phonetic information at the segment level, which is more compatible with the segment-level trained x-vector. The key point is how to define the phonetic labels for one speech segment, with multiple phonemes involved. We adopted a naive way as follows: for a given segment $\mathbf{x}$ with $N$ frames, the corresponding segment-level phoneme label $\mathbf{y}^t$ is represented as

$$\mathbf{y}^t = \{y_1, y_2, \ldots, y_C\}, \quad y_c = \frac{N_c}{N}$$

where $C$ is the size of the chosen phoneme set. $N_c$ denotes the number of occurrences of the $c$-th phoneme in $\mathbf{x}$.

## 3. TEXT-ADAPTATION FRAMEWORK

A typical deep speaker verification task involves three phases:

- Training: the speaker embedding extractor is trained with a large amount of pre-collected data.
- Evaluation
  - Enrollment: new speakers are enrolled by generating speaker embeddings via the well-trained extractor.
  - Test: each test utterance is evaluated using the enrolled model of the claimed identity to make the verification decision.

For the text-independent task, we don't pose any requirement on the text match either between the training and evaluation data or between the enrollment and test data.

For the traditional text-dependent task, directly applying the systems trained for the text-independent speaker verification task usually achieves very poor performance due to the text mismatch between the training and evaluation data. The current state-of-the-art text-dependent speaker verification systems share the same methods with the text-independent ones, while the training data are collected for the customized application. For example, the training speech segments in [6, 7] and [8] are collected from a large amount of speakers sharing the same content "OK Google" or "Hey Cortana". Despite the good results achieved using this approach, it's expensive and inflexible to recollect the training data for each different phrases.

Moreover, real-world applications usually don't follow the standard text-dependent regime. There are also scenarios where text mismatch also exists between enrollment and test utterances. For instance, it's common in some conditions users would like to use multiple keywords to wake up smart devices. For example, Google devices support both "OK Google" and "Hey Google" simultaneously [10, 11]. Some applications may require even more different keywords. Is it possible to allow the user only to enroll one of them and test on other different keywords?

### 3.1. Text-adaptation for speaker embeddings

To summarize the problems mentioned above, we would like to address the following two text-mismatch conditions:

- text mismatch exists between the training and evaluation data, which is the traditional text-dependent task.
- text mismatch exists not only between the training and evaluation data but also between the enrollment and test data.

In the case that the pre-collected training data share the same text with the evaluation data, the text information is implicitly modeled in the speaker extractor. However, to address the two types of text mismatch mentioned above without the recollection of a huge amount of training data, the text information modeling should be explicitly considered. In this paper, we proposed a framework in which text-independent speaker embeddings could be adapted to text-dependent speaker embeddings, while the text information could be customized according to the input.

### 3.2. Speaker-text factorization network

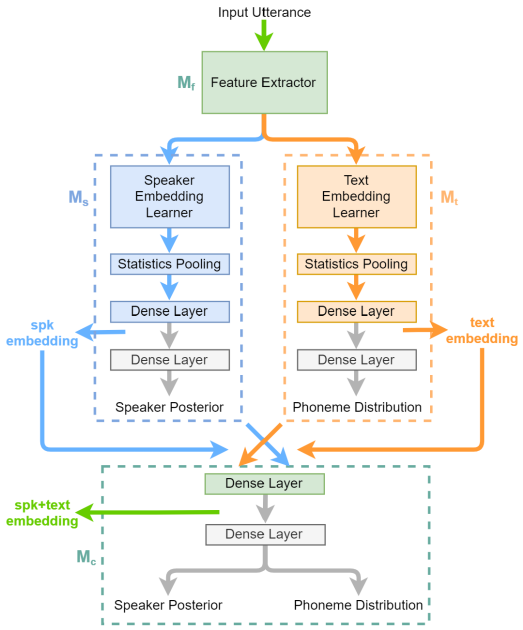


**Fig. 1**. The proposed speaker-text factorization network.

As depicted in Figure 1, The proposed model contains four parts: generic feature extractor $M_f$, two parallel sub-nets $M_s$ and $M_t$ for speaker discrimination and phoneme distribution learning respectively, and the "combination" sub-net $M_c$ for integrating both speaker and phonetic information. Following [2], phoneme classifier $M_t$ predicts the normalized categorical occurrences of phonemes in one input segment and speaker classifier $M_s$ is a standard one predicting speaker classes. The speaker embedding $\text{ebd}_s$ and phoneme based text embedding $\text{ebd}_t$ extracted from $M_s$ and $M_t$ are then concatenated as the input into the combination network $M_c$, aiming to recover both the speaker identity and phonetic information. The model is trained jointly. Given the features of one training segment pair $[\mathbf{x}_s, \mathbf{x}_t]$ and the corresponding speaker label $\mathbf{y}^s$ and phoneme label $\mathbf{y}^t$, the loss is defined as $\mathcal{L}_{total} = \mathcal{L}_{s1} + \mathcal{L}_{t1} + \mathcal{L}_{s2} + \mathcal{L}_{t2}$, where

$$\begin{aligned}
\mathcal{L}_{s1} &= \text{CE}(M_s(M_f(\mathbf{x}_s)), \mathbf{y}^s) \\
\mathcal{L}_{t1} &= \text{KLD}(M_t(M_f(\mathbf{x}_t)), \mathbf{y}^t) \\
\mathcal{L}_{s2} &= \text{CE}(M_c([\text{ebd}_s, \text{ebd}_t]), \mathbf{y}^s) \\
\mathcal{L}_{t2} &= \text{KLD}(M_c([\text{ebd}_s, \text{ebd}_t]), \mathbf{y}^t)
\end{aligned}$$

"spk" embedding $\text{ebd}_s$ is computed from $\mathbf{x}_s$ using the speaker sub-net, while the "text" embedding $\text{ebd}_t$ is from $\mathbf{x}_t$ using the text sub-

net. To better decouple speaker and phonetic information, the training pair $[\mathbf{x}_s, \mathbf{x}_t]$ is randomly sampled from the training data, which can be identical or two utterances from two different speakers.

The "spk" embeddings extracted from the speaker sub-net are directly used for the text-independent task since no text adaptation is needed. For the traditional text-dependent speaker verification task, where the text embedding could be accurately computed for the enrollment and test data, the "text" embedding comes from the same utterance as the "spk" embedding. For the other scenario where text-mismatch exists from the enrollment and test utterance in the target trials, we use a very small amount of pre-collected data with the target text (e.g., 10 utterances from other speakers) to compute the "text" embedding and then use it to adapt the "spk" embeddings computed from the enrollment utterances, while the genuine "text" embeddings are used for the test utterances.

## 4. EXPERIMENTAL SETUPS

### 4.1. Data

The Voxceleb and RSR2015 datasets are used in our experiments. All the phoneme labels are generated by a phoneme recognizer. More details could be referred to [2]. Details about training and evaluation data preparation for the text-adaptation and text-independent evaluations will be given below, and the trial definitions are available online to reproduce our results on the customized RSR2015 evaluation set[1].

#### *4.1.1. Training set*

For our experiments, Voxceleb2 development set is used for training the neural network and the Probabilistic Linear Discriminant Analysis (PLDA) back-end. This set contains 5994 speakers with 1092009 utterances. To train the neural network, we follow the data preparation process in Kaldi Voxceleb recipe, which cut the utterances to segments with length ranging from 2s to 4s. It should be noted that, unlike the recipe, we didn't use any data augmentation.

#### *4.1.2. Text-adaptation evaluation set*

Two individual evaluation sets are created corresponding to different text-mismatch cases:

**Mismatch between training and evaluation data**: When the mismatch is only between training and evaluation data, it becomes the traditional text-dependent task. The evaluation set is derived from the evaluation portion of RSR2015 [12] Part I. It contains 30 fixed phrase utterances of 3-4s duration from 106 speakers (57 males, 49 females). Every phrase is spoken 9 times by each speaker, 3 of which are taken for registering, and the rest are used for testing. As shown in Table 1, for a standard text-dependent task, there are four possible types of trials, among which TC represents the target trials and TW, IC, IW denote three non-target conditions.

**Table 1**. Types of trials for the text-dependent task

| | Correct Content | Wrong Content |
|---|---|---|
| Target | *TAR-correct* (TC) | *TAR-wrong* (TW) |
| Impostor | *IMP-correct* (IC) | *IMP-wrong* (IW) |

Since in the original trial definition, about $90\%$ of the original trials belong to the very easy IW case, we generate our own trial list following the ratio: TC:TW:IC:IW=1:3:3:3.

[1] https://github.com/Xflick/RSR2015_trials

**Mismatch between enrollment and test data**: As mentioned before, there are 30 different fixed phrases in the RSR2015 part1 evaluation set. We randomly select ten of them and generate ten evaluation subsets. Two enrollment conditions are considered, the text-independent and text-dependent. For the former condition, the text of three enrollment utterances is randomly selected, while for the latter one, the text is shared among the enrollment utterances and has no overlap between the text of test utterances. The text embedding for adaptation is computed using 10 random utterances from the development set with the same text as target text (speakers are other from the evaluation set). To better exhibit the text awareness of our speaker-text factorization network, we increase the number of TC trials. And the trial list for this specific task follows the ratio: TC:TW:IC:IW=1:1:1:1.

#### *4.1.3. Text-independent evaluation set*

**Voxceleb1 evaluation set**: To validate the performance of our baseline and proposed system, we first report the results on the official Voxceleb1 evaluation set. The cleaned trial lists are used: VoxCeleb1 (denoted as Voxceleb1-O, O for "original"), Voxceleb1-E (extended), and cleaned Voxceleb1-H (hard).

**RSR2015 evaluation set**: To better exhibit the effectiveness of our proposed framework, we designed one more text-independent evaluation set based on RSR2015, all the trial pairs are the same as the ones defined for traditional text-dependent case in Section 4.1.2, while the trials from TW condition are now treated as target.

### 4.2. System configurations

Standard x-vector [1] system with five time delay layers and two dense layers is used as our baseline system. The proposed Factorization Net system is modified from the baseline system, with three time delay layers extracting generic features. $M_s$ and the $M_t$ have identical structures, both having two time delay layers, one statistics pooling layer and two dense layers, except that one is for speaker classification and the other is for phonetic information prediction. $M_c$ has two dense layers in common and two output layers for two tasks. It is notable that when extracting only *speaker embedding*, the Factorization Net has exactly the same structure as the baseline TDNN model.

40-dimensional Fbank features are used for model training. The neural networks are trained on 4 GPUs with a batch size of 256. Stochastic gradient descent with learning rate 0.01, momentum 0.9 and weight decay 1e-4 is used to optimize the model. Batch normalization is applied after ReLU activation function.

PLDA is applied on Voxceleb1 evaluation set to validate the correctness of our system, for other evaluation sets, to get rid of the impact of PLDA compensation and focus on the properties of learned embeddings, and simple cosine scoring is utilized. All architectures are implemented in PyTorch [16]. We report the performance of our models in terms of Equal error rate (EER) and Minimum detection cost (minDCF) with $P_{tar}$ set as 0.01.

## 5. RESULTS AND ANALYSIS

### 5.1. Validation results on the text-independent task

x-vector was proposed for the text-independent task, to show the correctness of the baseline model and the proposed model, and we first report the results on the standard Voxceleb1 evaluation set in Table 2. Since we only used the clean training data, the baseline is quite

**Table 2**. Validation experiments on Voxceleb 1 evaluation set

| System Configuration | | Voxceleb1_O | | Voxceleb1_E | | Voxceleb1_H | |
|---|---|---|---|---|---|---|---|
| Architecture | Embedding Type | EER | minDCF | EER | minDCF | EER | minDCF |
| TDNN | spk | 2.888 | 0.3281 | 3.055 | 0.3272 | 5.026 | 0.4646 |
| Factorization Net | spk | **2.595** | **0.2940** | **2.784** | **0.2990** | **4.703** | **0.4292** |

strong compared to the ones in the literature [17, 18, 19]. As shown in Table 2, the embedding extracted from the speaker sub-net of the proposed model reduces the EERs on Voxceleb_O, Voxceleb_E and Voxceleb_H from 2.888%, 3.055% and 5.026% to 2.595%, 2.784% and 4.703%, respectively. A similar improvement can also be observed in terms of minDCF. For the experiments on the RSR2015 text-independent evaluation set, similar performance improvement for the speaker embedding from the speaker sub-net is observed.

**Table 3**. Experiments on RSR2015 text-independent evaluation set

| System Configuration | | EER (%) | minDCF |
|---|---|---|---|
| Architecture | Embedding Type | | |
| TDNN | spk | 7.220 | 0.7068 |
| Factorization Net | spk | **6.239** | **0.6721** |

### 5.2. Text adaptation to address the text-mismatch problems

#### *5.2.1. Mismatch between training and evaluation data*

As shown in Table 4, when the mismatch happens between training and evaluation data, integrating text information into embedding significantly improves the performance. The EER of the system is reduced from $6.671\%$ to $1.542\%$, while the minDCF is reduced from $0.5234$ to $0.1246$.

**Table 4**. Experiments on RSR2015 text-adaptation evaluation set (mismatch between training and evaluation data)

| System Configuration | | EER (%) | minDCF |
|---|---|---|---|
| Architecture | Embedding Type | | |
| TDNN | spk | 6.671 | 0.5234 |
| Factorization Net | spk | 6.010 | 0.5144 |
| | spk+text | **1.542** | **0.1246** |

Table 5 shows the results when different error types are individually analyzed. The errors TW and IW, which are resulted from the wrong text, are greatly reduced as expected. The speaker error (IC) is also decreased from $1.919\%$ to $1.101\%$.

**Table 5**. EERs with regard to different error types on RSR2015 text-adaptation evaluation set (mismatch between training and evaluation data)

| System Configuration | | EER (%) | | |
|---|---|---|---|---|
| Architecture | Embedding Type | TW | IC | IW |
| TDNN | spk | 10.60 | 1.919 | 1.007 |
| Factorization Net | spk | 10.32 | 1.385 | 0.7867 |
| | spk+text | **2.454** | **1.101** | **0.1573** |

#### *5.2.2. Mismatch between enrollment and test data*

As shown in Table 6, when the mismatch happens between not only training and test data, but also enrollment and test data, most systems fail on the task. However, by using the target text embedding instead of the genuine text embedding to adapt enrollment "spk" embedding, the system performance is substantially improved, which shows the effectiveness of our factorization net to generate text-customized speaker embeddings.

**Table 6**. Text-independent/Text-dependent enrollment (introduced in Sec 4.1.2) EERs(%) on RSR2015 text-adaptation evaluation set (mismatch between enrollment and test data)

| Subset | TDNN | Factorization Net | | |
|---|---|---|---|---|
| | spk | spk | spk +text | spk +adapt_text |
| 1 | 27.34/29.21 | 27.05/28.31 | 26.42/26.29 | **12.26/15.07** |
| 2 | 24.61/22.73 | 26.03/24.09 | 24.87/23.18 | **13.71/13.99** |
| 3 | 22.18/24.33 | 23.41/25.37 | 20.75/21.49 | **10.84/14.49** |
| 4 | 22.56/20.19 | 22.88/19.97 | 24.87/17.14 | **6.912/7.227** |
| 5 | 28.16/32.45 | 26.74/30.99 | 27.93/34.84 | **10.51/16.98** |
| 6 | 22.46/21.32 | 22.76/21.74 | 21.38/23.40 | **9.985/10.81** |
| 7 | 22.20/25.09 | 22.02/25.23 | 22.02/27.67 | **7.898/12.52** |
| 8 | 29.97/29.09 | 29.64/28.88 | 30.24/30.44 | **14.21/15.32** |
| 9 | 22.86/24.73 | 22.43/25.04 | 22.91/24.66 | **9.450/14.56** |
| 10 | 22.91/25.50 | 22.51/25.34 | 23.15/25.76 | **9.083/12.35** |
| avg | 24.53/25.46 | 24.55/25.50 | 24.45/25.49 | **10.49/13.33** |

## 6. CONCLUSIONS AND FUTURE WORKS

Text mismatch between the training data and evaluation data can lead to huge performance degradation for the text-dependent speaker verification. One common solution is to collect application-specific training data that share the same text information as the evaluation data. To get rid of the expensive and inflexible data collection process and take advantage of the large amount of unconstrained speech data, we proposed a "text-adaptation speaker verification" framework, in which the text-independent speaker embeddings could be adapted to text-customized ones according to the specific adaptation input. A speaker-text factorization network is proposed, which first factorizes a speech segment into a text-independent speaker embedding and a speaker-independent text embedding and then recombines them as one single embedding containing both information. We first verify the proposed method without text adaptation on standard text-independent Voxceleb evaluation set and observe consistent performance improvement on all the three trial lists. Results on three customized evaluation sets derived from the RSR2015 dataset show that the proposed method using text adaptation can greatly reduce the errors caused by the text-mismatch between the training and evaluation data and between the enrollment and test data.

In the future work, we will make more efforts to allow the model to utilize simple plain text instead of the text embedding computed from specific audios for the text adaptation on speaker embeddings.

## 7. REFERENCES


[1] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018*. IEEE, 2018.

[2] Shuai Wang, Johan Rohdin, Lukáš Burget, Oldřich Plchot, Yanmin Qian, Kai Yu, and Jan Černockỳ, "On the usage of phonetic information for text-independent speaker embedding extraction," *Proc. Interspeech 2019*, pp. 1148–1152, 2019.

[3] Zili Huang, Shuai Wang, and Kai Yu, "Angular softmax for short-duration text-independent speaker verification.," in *Interspeech*, 2018, pp. 3623–3627.

[4] Zili Huang, Shuai Wang, and Yanmin Qian, "Joint i-vector with end-to-end system for short duration text-independent speaker verification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4869–4873.

[5] Chunlei Zhang and Kazuhito Koishida, "End-to-end text-independent speaker verification with triplet loss on short utterances.," in *Interspeech*, 2017, pp. 1487–1491.

[6] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 4052–4056.

[7] Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer, "End-to-end text-dependent speaker verification," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5115–5119.

[8] Shi-Xiong Zhang, Zhuo Chen, Yong Zhao, Jinyu Li, and Yifan Gong, "End-to-end attention based text-dependent speaker verification," in *2016 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 171–178.

[9] Yichi Zhang, Meng Yu, Na Li, Chengzhu Yu, Jia Cui, and Dong Yu, "Seq2seq attentional siamese neural networks for text-dependent speaker verification," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6131–6135.

[10] FA Rezaur rahman Chowdhury, Quan Wang, Ignacio Lopez Moreno, and Li Wan, "Attention-based models for text-dependent speaker verification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5359–5363.

[11] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno, "Generalized end-to-end loss for speaker verification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4879–4883.

[12] Anthony Larcher, Kong Aik Lee, Bin Ma, and Haizhou Li, "Text-dependent speaker verification: Classifiers, databases and rsr2015," *Speech Communication*, vol. 60, pp. 56–77, 2014.

[13] David Snyder, Daniel Garcia-Romero, Daniel Povey, and Sanjeev Khudanpur, "Deep neural network embeddings for text-independent speaker verification," *Proc. Interspeech 2017*, pp. 999–1003, 2017.

[14] Yuan Liu, Yanmin Qian, Nanxin Chen, Tianfan Fu, Ya Zhang, and Kai Yu, "Deep feature for text-dependent speaker verification," *Speech Communication*, vol. 73, pp. 1–13, 2015.

[15] Yi Liu, Liang He, Jia Liu, and Michael T. Johnson, "Speaker embedding extraction with phonetic information," *CoRR*, vol. abs/1804.04862, 2018.

[16] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer, "Automatic differentiation in pytorch," 2017.

[17] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman, "Voxceleb2: Deep speaker recognition," *arXiv preprint arXiv:1806.05622*, 2018.

[18] Xu Xiang, Shuai Wang, Houjun Huang, Yanmin Qian, and Kai Yu, "Margin matters: Towards more discriminative deep neural network embeddings for speaker recognition," *arXiv preprint arXiv:1906.07317*, 2019.

[19] Weidi Xie, Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "Utterance-level aggregation for speaker recognition in the wild," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5791–5795.