INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

What Does the Speaker Embedding Encode?

Shuai Wang, Yanmin Qian, Kai Yu

Key Lab. of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering
SpeechLab, Department of Computer Science and Engineering
Brain Science and Technology Research Center
Shanghai Jiao Tong University, Shanghai, China

wsstriving@gmail.com, yanminqian@sjtu.edu.cn, kai.yu@sjtu.edu.cn

Abstract

Developing a good speaker embedding has received tremendous interest in the speech community. Speaker representations such as *i*-vector, *d*-vector have shown their superiority in speaker recognition, speaker adaptation and other related tasks. However, not much is known about which properties are exactly encoded in these speaker embeddings. In this work, we make an in-depth investigation on three kinds of speaker embeddings, i.e. *i*-vector, *d*-vector and RNN/LSTM based sequence-vector (*s*-vector). Classification tasks are carefully designed to facilitate better understanding of these encoded speaker representations. Their abilities of encoding different properties are revealed and compared, such as speaker identity, gender, speaking rate, text content and channel information. Moreover, a new architecture is proposed to integrate different speaker embeddings, so that the advantages can be combined. The new advanced speaker embedding (*i*-*s*-vector) outperforms the others, and shows a more than 50% EER reduction compared to the *i*-vector baseline on the RSR2015 content mismatch trials.

Index Terms: speaker identification, *i*-vector, *d*-vector, *s*-vector, speaker embedding

1. Introduction

Speaker recognition is the task of recognizing the identity of a speaker through his voice. Considering the restriction on the spoken text, speaker recognition can be classified into two categories, text-dependent and text-independent. There has been lots of work on both tasks, and the system performance has been improved dramatically during the last few years.

Since proposed in [1], Gaussian Mixture Model-Universal Background Model with maximum a posteriori (GMM-UBM-MAP) has dominated the field of speaker recognition for a long time. In this framework, the speaker's model is derived from a well-trained UBM with MAP adaptation. However, GMM-UBM-MAP suffers from the *issue of data sparsity in the enrollment phase*: only a part of the UBM parameters are well adapted [1, 2]. Researchers tended to find a method to correlate the different Gaussian components of the UBM and perform some "Global Transform". Super-vector was proposed and achieved competitive results when combined with a SVM classifier [3, 4]. Since then, exploring a fixed-length vector representation receives more and more interest. P. Kenny proposed a Joint Factor Analysis (JFA) [5] framework as a compensation method in the super-vector space and aimed to model speaker and channel factors in separate subspaces. The following *i*-vector [6] approach simplifies the JFA framework by modelling a single total variability subspace. The success of deep learning in speech recognition [7] motivated researchers to apply Deep Neural Networks (DNNs) to speaker recognition. DNN based acoustic models were used instead of the GMM in the *i*-vector framework in several works [8, 9]. An alternative approach is to use DNN to extract bottleneck features [10, 11, 12, 13] or speaker representations directly [14, 15, 16, 12], whereas *d*-vector [14] is the most typical one. Researchers also tried to learn an utterance-level representation [17, 18], using long-short term memory (LSTM) [19] networks.

Although some work has been carried out on finding feasible speaker embeddings[6, 14, 16, 17, 18], few researchers have drawn a systematic investigation into which properties are exactly encoded in these speaker embeddings. In [20], the authors proposed a methodology to perform fine-grained analysis on sentence vectors which are used in natural language processing (NLP) tasks. Inspired by this, we designed the methodology to facilitate better understanding of these encoded speaker representations. The following insights regarding different speaker embedding methods are exposed:

- Though *i*-vector is usually used in text-independent speaker recognition, it can encode spoken terms (which words are presented) to a good extent, however, the sequence information such as word order can not be encoded.
- *i*-vector has the best speaker discrimination ability.
- LSTM/RNN based sequence vector (*s*-vector) is inferior at modelling speaker identity, but can encode the spoken terms and word orders to a notably extent.
- Appropriate combination of *i*-vector and *s*-vector can take both advantages and result in a new better speaker embedding.

The rest of this paper is organized as follows. Three kinds of speaker embeddings (*i*-vector *d*-vector, and *s*-vector) will be introduced in Section 2. Section 3 describes the adopted methodology. Experiments and analysis are given in Section 4. Section 5 introduces an approach on combining *s*-vector and *i*-vector, namely *i*-*s*-vector), moreover, results on RSR2015 part 1 dataset are presented. The conclusions are given in Section 6.

2. Vector-based Speaker embedding

2.1. *i*-vector

In the *i*-vector framework [6], the speaker- and session-dependent super-vector $\mathbf{M}$ (derived from UBM) is modeled as

$$\mathbf{M} = \mathbf{m} + \mathbf{Tw} \tag{1}$$

This work was supported by the Shanghai Sailing Program No. 16YF1405300, the China NSFC projects (No. 61573241 and No. 61603252) and the Interdisciplinary Program (14JCZ03) of Shanghai Jiao Tong University in China. Experiments have been carried out on the PI supercomputer at Shanghai Jiao Tong University.

http://dx.doi.org/10.21437/Interspeech.2017-1125

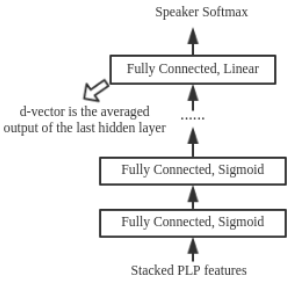

Figure 1: *d-vector extraction*

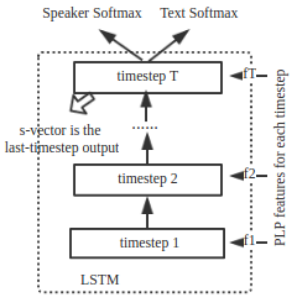

Figure 2: *s-vector extraction*

where $\mathbf{m}$ is a speaker and session-independent super-vector, $\mathbf{T}$ is a low rank matrix which captures speaker and session variability, and *i*-vector is the posterior mean of $\mathbf{w}$.

2.2. *d*-vector

Within the *d*-vector framework, a speaker-discriminative DNN is firstly trained. Then, it is used to extract frame level vectors from the last hidden layer. These vectors are averaged over the whole sentence to obtain the utterance-level *d*-vector [14]. The model architecture for *d*-vector extraction is shown in Figure 1.

2.3. *s*-vector

s-vector refers to a sequence-vector or summary-vector. speaker recognition can be treated as a sequence labelling problem. Thus recurrent neural networks or the more advanced long-short term memory (LSTM) version can be applied. As mentioned in [18], the sequence-level representation can be a last-timestep speaker embedding, attention-based speaker embedding or averaged speaker embedding. In this paper, the last-timestep speaker embedding is utilized, i.e. the hidden output at the last timestep is used as the sequence representation (*s*-vector). Further investigation can be referred to in [18]. It should be noticed that sequence training is non-trivial, therefore we adopt a multi-task approach, similar to [16], the model architecture is shown in Figure 2.

3. Approach

All experiments are based on an assumption: **if one property is encoded in the speaker representation, a classifier to predict such property can be trained, and the classification accuracy depends on how well the property has been encoded in these speaker embeddings**. Given the speaker embeddings, we create training data and train a classifier to predict a specific property based on these representations. In this work, we mainly focus on the following properties:

- Speaker related, such as identity, gender and speaking rate.
- Text related, such as spoken terms, word order and utterance length.
- Channel related, such as the handset id and noise type.

The experiments are performed on three kinds of different speaker embeddings: *i*-vector, *d*-vector and *s*-vector. After extracting the speaker embeddings, the corresponding prediction tasks will be conducted. To better verify the embedding's properties, the classifier should be simple, thus a Multi-Layer Perceptrons (MLP) with only a single hidden layer and ReLU activation is used in all the experiments.

3.1. The Prediction Tasks

Based on the rules above, 7 prediction tasks are designed:

- **Speaker Identity Task**: This task measures to what extent the speaker embedding encodes the speaker's identity, which is crucial for the speaker recognition. The evaluation set contains 106 different speakers.
- **Speech Text Task**: This task measures to what extent the spoken text is encoded in the speaker embedding. In this experiment, 30 different sentences are included, leading to a 30-class classification task.
- **Spoken Term Task**: This task measures to what extent the speaker embedding encodes the identities of words within it. The dataset we choose only contains 147 different words, so each embedding is tested using 147 binary classifiers (logistic regression) to verify whether a certain word is present. An embedding is classified as true when all words in the corresponding utterance are correctly verified.
- **Word Order Task**: This task measures to what extent the speaker embedding encodes word order. We simplify this task to "chunk order" task. Given two WAV format utterances $\mathbf{u}_1$, $\mathbf{u}_2$ and their concatenation $\mathbf{s}$, the goal is to predict whether $\mathbf{u}_1$ appears before or after $\mathbf{u}_2$ in $\mathbf{s}$ given their speaker embeddings $\mathbf{se}_{u1}, \mathbf{se}_{u2}$ and $\mathbf{se}_s$. This is modeled as a binary classification task, where the input is the concatenation of $\mathbf{se}_{u1}, \mathbf{se}_{u2}$ and $\mathbf{se}_s$. Through flipping the order of $\mathbf{u}_1$ and $\mathbf{u}_2$ in $\mathbf{s}$, positive samples and negative samples are generated.
- **Utterance Length Task**: This task measures to what extent the spoken utterance length is encoded in the speaker embedding. According to the utterance length (after VAD), we split the samples into 4 categories: 1-3s, 4-6s, 7-9s, 10-12s (A gap is preserved for better discrimination). Long utterances are obtained via concatenating short ones. A four-class classifier is trained in this prediction task.
- **Channel Task**: This task measures to what extent the channel information (6 channels corresponding to 6 handsets in this paper) is encoded in the speaker embedding. A six-class classifier is trained in this prediction task.
- **Speaker Gender Task**: This task measures to what extent the speaker embedding encodes the speaker's gender. A binary classifier is used in this prediction task.
- **Speaking Rate Task**: This task measures to what extent the speaker embedding encodes the speaking rate. In the original data set, all speakers speak at normal speed. We generate wav files at $0.5\times$ speed and $1.5\times$ speed. A three-class classifier is trained in this prediction task.

4. Experiments and Analysis

4.1. Exprimental setup

All experiments are based on the RSR2015 part 1 dataset[21], consisting of overall 300 speakers, whereas 143 are females and 157 are males. Each speaker pronounces 30 fixed sentences (each for 9 sessions) selected from the TIMIT [22] database to cover all English phonemes. The average recording duration is 3.20 s. The whole set is divided into background (*bkg*), development (*dev*) and evaluation (*eval*) subsets.

The *bkg* data is used for the training of GMM-UBM, *i*-vector extractor, *d*-vector DNN extractor, *s*-vector LSTM extractor. 39-dimensional PLP features (13-dim static feature with delta and delta-delta features) are used in all experiments. The *eval* data is used for the prediction tasks.

Table 1: *Subset definition of RSR2015 part 1*

Subset	# Female speaker	# Male speaker	# Total
bkg	47	50	97
dev	47	50	97
eval	49	57	106

- ***i*-vector**: The *i*-vector system comprises of a 1024-gaussian GMM-UBM, and *i*-vector extractors with different dimensions. The whole process was implemented using Kaldi [23].
- ***d*-vector**: The DNN comprises of 1 input layer with 429 (39-dim features along with left and right context frames) neurons, 5 hidden layers with 1024 neurons, an extract layer with $|vector|$ neurons, and an output softmax layer with 97 neurons (one for each speaker). Sigmoid is used as the activation function. Succeeding RBM initialization, the DNN is fine-tuned using SGD based BP algorithm. After the speaker discriminative DNN training, the *d*-vector is obtained by averaging the frame level representations extracted from the last hidden layer.
- ***s*-vector**: The initial LSTM comprises of a 39-dim input layer and a classic LSTM hidden layer, followed by a softmax layer with 97 neurons. By varying the LSTM neuron number in the hidden layer, we extract *s*-vector with different dimensions. However, since the LSTM is trained utterance-wise, the number of samples is limited, leading to poor training accuracy. Instead, we train the LSTM in a multi-task learning mode with both speaker and text output layer (97 neurons representing the speakers + 30 neurons representing the text), similar to the idea proposed in [16]. This LSTM model is optimized using RMSprop algorithm [24].

In all the following figures, the green line with the notation *i*-*s*-vector (unidirectional LSTM based) is the newly proposed speaker embedding which will be described in Section 5.

4.2. Speaker identity task

For speaker recognition, the speaker discrimination ability is important. As shown in Figure 3, *i*-vector performs much better than the other two embeddings, whereas *s*-vector performs worst. LSTM/RNN operates at utterance level, leading to limited training samples, which has a negative impact on its generalization capability.

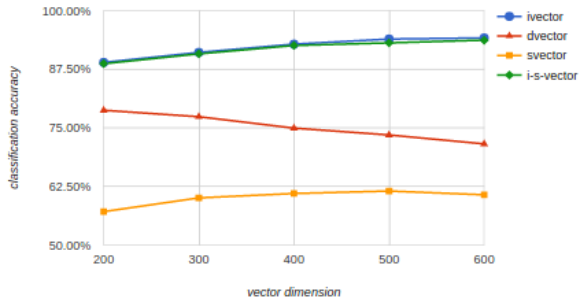

Figure 3: *Speaker identity task*

4.3. Speech text task

As shown in Figure 4, both *i*-vector and *s*-vector perform very well in the speech text classification, both approaching 100% accuracy. *d*-vector also achieves nearly 95% accuracy, although it averages out frame-level representations. To clarify this abnormal observation, *spoken term task, word order task* and *utterance length task* are performed.

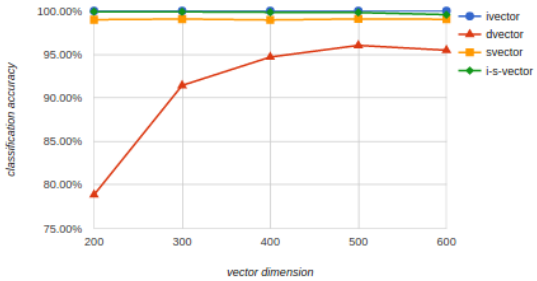

Figure 4: *Speech text task*

4.3.1. Spoken term task

As described in Section 3, this task measures whether it can be inferred which words are included in an utterance given its speaker embedding. The result is shown in Figure 5. *s*-vector outperforms *i*-vector when the vector dimension is larger than 300. *d*-vector doesn't encode any word information and cannot predict spoken terms, which is reasonable since the average pooling removes most related information.

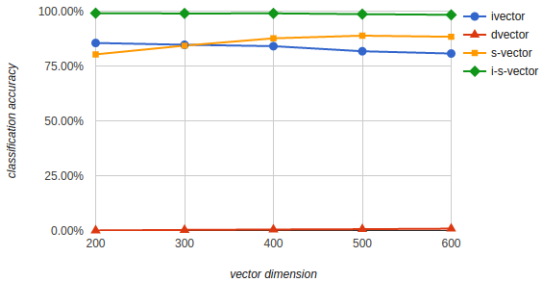

Figure 5: *Speech content task*

4.3.2. Word order task

Figure 6 shows the speaker embedding comparison on the word order task. Since it's a binary task, the baseline is 50%. *d*-vector doesn't preserve any order information, and *i*-vector also performs poorly. In contrast, *s*-vector is more robust and superior to the others on the order prediction, and achieves almost 100% accuracy on this task. The results coincide with the fact that *i*-vector and *d*-vector don't distinguish sample orders and recurrent network takes the sample orders explicitly into modelling.

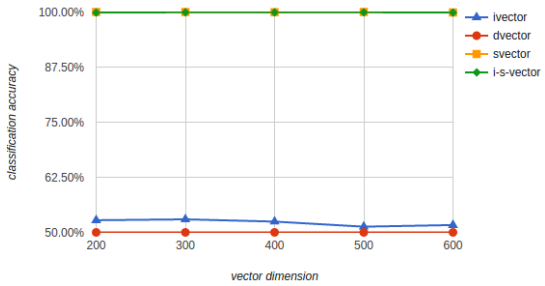

Figure 6: *Word order task*

4.3.3. Utterance Length task

Based on our design described in Section 3, there are four categories in the length task, leading to a naive baseline accuracy of 25% (1/4). As shown in figure 7, *s*-vector and *i*-vector outperforms *d*-vector as expected.

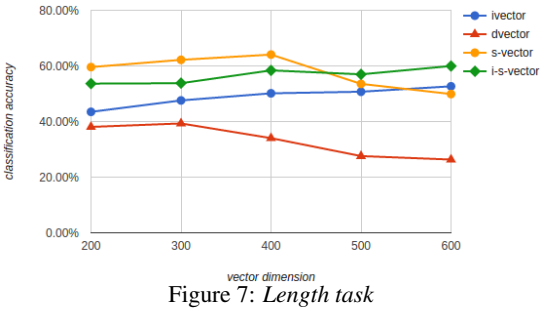

Figure 7: *Length task*

4.4. Channel task

6 different handsets were used to record the utterances of RSR2015 dataset, which are marked as 6 channels. As shown in Figure 8, the prediction accuracies of all three speaker embedding types are much higher than the baseline (1/6=16.7%). It means that these embeddings also encode the channel-related information, which will make them sensitive to the channel mismatch or distortion. Accordingly, the channel compensation may be important when using these speaker embeddings.

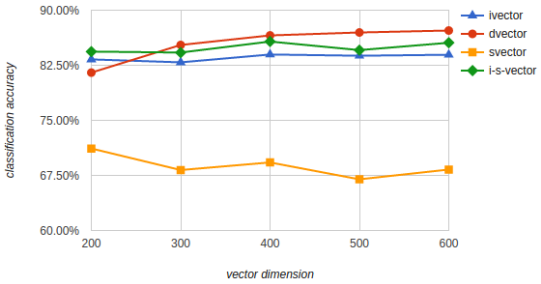

Figure 8: *Channel task*

4.5. Speaker gender and Speaking rate task

Regarding the speaker gender task, all three speaker embeddings achieve above 97% accuracy, *d*-vector even approaches 100%. Concerning speaking rate prediction, the individual accuracies of *i*-vector, *d*-vector and *s*-vector systems are 90%, 95% and 77% respectively. *d*-vector still performs the best.

5. Combination on *i*-vector and *s*-vector

Based on the experiments and analysis above, it's observed that different speaker embeddings have their own properties. For instance, *i*-vector shows better speaker discrimination ability, whereas *s*-vector shows its superiority on encoding text information. Considering both text and speaker discriminative information are important for the text-dependent speaker verification (TDSV), we propose a multi-task framework to train LSTM as an auxiliary model to add more text information (sequence information) to *i*-vector. The only difference from the original *s*-vector framework is that we concatenate the last-step hidden output (*s*-vector) and corresponding *i*-vector before feeding it to the speaker softmax, and the entire architecture is optimized with the multi-task learning training mode.

As shown in Figures 3 to 7, it is observed that this new speaker embedding achieves remarkable positions on almost all tasks, which demonstrates the superiority of *i*-*s*-vector.

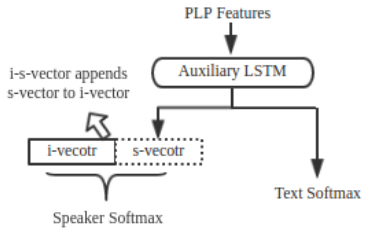

Figure 9: *Proposed i-s-vector framework for TDSV*

5.1. Evaluation on RSR2015 part 1

In text-dependent speaker verification, three test conditions are defined according to three impostor types: (I) the content doesn't match (II) the speaker doesn't match (III) neither the speaker or the content match. Results on RSR2015 part1 Evaluation Set can be found in Table 2, and **cosine similarity** is used as the scoring metric.

Table 2: *EER(%) comparison of TDSV on RSR2015 part1*

vectors/conditions	**I**	**II**	**III**
i-vector	0.35	**1.13**	0.06
i-vector + *s*-vector	0.28	**1.13**	0.03
i-*s*-vector (LSTM)	0.17	1.98	0.03
i-*s*-vector (BLSTM)	**0.11**	1.72	**0.02**

The first line is the *i*-vector baseline, which is comparable to the results in the literature [21, 25], and the second line is the system concatenating *i*-vector and *s*-vector directly. The bottom part are the systems using proposed *i*-*s*-vector described above. It is observed that there is a more than 50.0% EER reduction compared to the *i*-vector baseline in the content-mismatch conditions, i.e. I and III, and using Bidirectional LSTM can achieve a further improvement. However, there is also an obvious degradation in condition II. Accordingly the proposed *i*-*s*-vector can perform pretty well in "speaker-dependent content matching problems" [25]. Moreover, the direct concatenation of *i*-vector and *s*-vector (*i*-vector + *s*-vector) can obtain gains on conditions I and III, but the improvements are relatively smaller than those from the proposed *i*-*s*-vector framework.

6. Conclusion

A comprehensive analysis on *i*-vector, *d*-vector and LSTM/RNN based sequence vector *s*-vector are compared and analyzed in this paper, and the results reveal:

- As state-of-the-art technology in the speaker recognition field, *i*-vector can encode the speaker identity to a large extent. Moreover, the GMM-UBM modelling on phonemes enables *i*-vector to encode the speech content, but the order of words can not be encoded.
- LSTM based *s*-vector performs pretty well on embedding the speech content and word order, which may be particularly useful in tasks such as "random digits" speaker recognition.
- All kinds of speaker embeddings encode channel information to some extent, which means more efforts should be taken on channel compensation.
- Sequence information encoded by *s*-vector can be combined to *i*-vector using an appropriate architecture. The proposed multi-task learning framework to combine *i*-vector and *s*-vector can achieve more than 50% EER reduction on content mismatch trials of RSR2015 part1 evaluation.

7. References

[1] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[2] S. Shum, "Unsupervised methods for speaker diarization," Ph.D. dissertation, Massachusetts Institute of Technology, 2011.

[3] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using gmm supervectors for speaker verification," *IEEE signal processing letters*, vol. 13, no. 5, pp. 308–311, 2006.

[4] W. M. Campbell, D. E. Sturim, D. A. Reynolds, and A. Solomonoff, "Svm based speaker verification using a gmm supervector kernel and nap variability compensation," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 1. IEEE, 2006, pp. I–I.

[5] P. Kenny, "Joint factor analysis of speaker and session variability: Theory and algorithms," *CRIM, Montreal,(Report) CRIM-06/08-13*, 2005.

[6] 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.

[7] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[8] 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.

[9] P. Kenny, V. Gupta, T. Stafylakis, P. Ouellet, and J. Alam, "Deep neural networks for extracting baum-welch statistics for speaker recognition," in *Proc. Odyssey*, 2014, pp. 293–298.

[10] T. Fu, Y. Qian, Y. Liu, and K. Yu, "Tandem deep features for text-dependent speaker verification." in *INTERSPEECH*, 2014, pp. 1327–1331.

[11] 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.

[12] Y. Tian, M. Cai, L. He, and J. Liu, "Investigation of bottleneck features and multilingual deep neural networks for speaker verification." in *Interspeech*, 2015, pp. 1151–1155.

[13] F. Richardson, D. Reynolds, and N. Dehak, "A unified deep neural network for speaker and language recognition," *arXiv preprint arXiv:1504.00923*, 2015.

[14] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4052–4056.

[15] L. Li, Y. Lin, Z. Zhang, and D. Wang, "Improved deep speaker feature learning for text-dependent speaker recognition," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2015 Asia-Pacific*. IEEE, 2015, pp. 426–429.

[16] N. Chen, Y. Qian, and K. Yu, "Multi-task learning for text-dependent speaker verification," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[17] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5115–5119.

[18] G. Bhattacharya, J. Alam, T. Stafylakis, and P. Kenny, "Deep neural networks based text-dependent speaker verification."

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[20] Y. Adi, E. Kermany, Y. Belinkov, O. Lavi, and Y. Goldberg, "Fine-grained analysis of sentence embeddings using auxiliary prediction tasks," *arXiv preprint arXiv:1608.04207*, 2016.

[21] 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.

[22] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1," *NASA STI/Recon technical report n*, vol. 93, 1993.

[23] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[24] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, 2012.

[25] 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.