Self-Attentive Speaker Embeddings for Text-Independent Speaker Verification

Yingke Zhu[1]*, Tom Ko*[2]*, David Snyder*[3]*, Brian Mak*[1]*, Daniel Povey*[3]

[1]Department of Computer Science & Engineering
The Hong Kong University of Science & Technology
[2]Huawei Noahs Ark Research Lab, Hong Kong, China
[3]Center for Language and Speech Processing & Human Language Technology Center of Excellence
The Johns Hopkins University, USA

{yzhuav,mak}@cse.ust.hk, {tomkocse, david.ryan.snyder, dpovey}@gmail.com

Abstract

This paper introduces a new method to extract speaker embeddings from a deep neural network (DNN) for text-independent speaker verification. Usually, speaker embeddings are extracted from a speaker-classification DNN that averages the hidden vectors over the frames of a speaker; the hidden vectors produced from all the frames are assumed to be equally important. We relax this assumption and compute the speaker embedding as a weighted average of a speaker's frame-level hidden vectors, and their weights are automatically determined by a self-attention mechanism. The effect of multiple attention heads are also investigated to capture different aspects of a speaker's input speech. Finally, a PLDA classifier is used to compare pairs of embeddings. The proposed self-attentive speaker embedding system is compared with a strong DNN embedding baseline on NIST SRE 2016. We find that the self-attentive embeddings achieve superior performance. Moreover, the improvement produced by the self-attentive speaker embeddings is consistent with both short and long testing utterances.

Index Terms: speaker recognition, deep neural networks, self-attention, x-vectors

1. Introduction

Speaker verification (SV) is the task of accepting or rejecting the identity claim of a speaker based on some given speech. There are two broad categories of SV systems: text-dependent and text-independent SV systems. Text-dependent SV systems require the content of input speech to be fixed, while text-independent SV systems do not.

Over the years, the combination of i-vectors [1] and probabilistic linear discriminant analysis (PLDA) [2] has been the dominant approach for text-independent SV tasks [3, 4, 5]. Also, hybrid approaches that incorporate deep neural networks (DNNs) trained for automatic speech recognition (ASR) into the i-vector system have proved to be beneficial in some conditions [6, 7, 8, 9, 10]. However, the ASR DNN adds significant computational complexity to the i-vector system and also requires transcribed data for training. Moreover, the success of this approach has been primarily isolated to English-language datasets [11]. On the other hand, recent work demonstrates that more powerful SV systems can be built from directly training a speaker discriminative DNN [12, 13, 14, 15, 16, 17]. Heigold *et al*. introduced an end-to-end system for a text-dependent SV task, that was jointly trained to map frame-level features to speaker embeddings and to learn a similarity metric to compare embedding pairs [13]. The system was then adapted to the more general task of text-independent SV in [15]. The work in [16] divided the end-to-end system into two components: a DNN to produce speaker embeddings and a separately trained PLDA classifier to compare embedding pairs. Compared to the end-to-end approach, this method requires less data to be effective and has the added benefit of facilitating reuse of the methods developed over the years for processing and comparing i-vectors. We continue to use this two-stage approach in this work.

Most DNN-based SV systems use a pooling mechanism to map variable-length utterances to fixed-dimensional embeddings. In a feed-forward architecture, this is usually enabled by a pooling layer that averages some frame-level DNN features over the whole input utterance. In early systems, such as the d-vector in [12], the DNN was trained at the frame-level, and pooling is performed by averaging activation vectors of the the last hidden layer over all frames of an input utterance. The work in [15, 16, 17] proposed adding a statistics pooling layer that aggregates DNN hidden vectors over the whole utterance of a speaker, and computes its mean and standard deviation. The statistics vectors were then concatenated together to form a fixed-length representation of the input utterance at the segment level. Speaker embeddings are derived from further processing of these segment-level representations. However, in most prior work, this pooling mechanism assigns equal weight to each frame-level feature. Zhang *et al*, proposed using an attention model to combine the frame-level features for a text-dependent SV application [14]. The attention model takes phonetic posterior features and phonetic bottleneck features as extra sources, and learn the combination weights for frame-level features.

This paper proposes an extension of the x-vector architecture described in [17]. In order to better utilize the speaker information in the input speech, we propose using frame-level weights that are learned by a structured self-attention mechanism and incorporated into a weighted statistics pooling layer. In contrast to the work in [14], our task is text-independent and there's a language mismatch between the training and testing data, so the phonetic information may not be helpful or even available. The self-attention mechanism was originally proposed for extracting sentence embeddings for natural language processing tasks [18]. We adapt the self-attention mechanism in [18] to text-independent SV based on the system in [17].

2. Speaker verification systems

We compare the proposed methods with two x-vector-based SV baseline systems. All systems are built using the Kaldi speech recognition toolkit [19].

This work is done during Yingke Zhu's internship at Noah's Ark Research Lab.

2.1. The x-vector baseline system

The x-vector baselines are based on the systems described in [17]. A speaker discriminative DNN is trained to produce speaker embeddings called x-vectors, and a PLDA backend is used to compare pairs of speaker embeddings.

The input acoustic features are 23-dimensional MFCCs with a frame-length of 25ms that are mean-normalized over a sliding window of up to 3 seconds. An energy-based VAD is employed to filter out non-speech frames from the utterances.

The DNN used in the x-vector baseline system is depicted in Figure 1. The first five layers l_1 to l_5 are constructed with a time-delay architecture and they work at the frame level. Suppose t is the current time step. Frames from $(t-2)$ to $(t+2)$ are spliced together at the input layer. The next two layers splice the output of the previous layer at time steps $\{t-2, t, t+2\}$ and $\{t-3, t, t+3\}$, respectively. No temporal contexts are added to the fourth and fifth layers. Thus, the total temporal context after the third layer is 15 frames.

The statistics pooling layer aggregates over frame-level output vectors of the DNN, and computes their mean and standard deviation. This pooling mechanism enables the DNN to produce fixed-length representation from variable-length speech segments. The mean and standard deviation are concatenated together and forwarded to two additional hidden layers l_6 and l_7, and finally a softmax output layer. The DNN is trained to classify speakers in the training set. After training, the softmax output layer and the last hidden layer are discarded, and speaker embeddings are extracted from the affine component of l_6. The system uses PLDA backend for scoring, which is described in section 2.3. All neural units are rectified linear units (ReLUs).

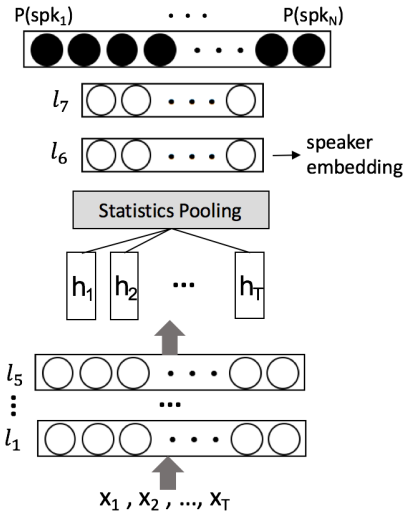

Figure 1: *Structure of the DNN in the x-vector baseline system.*

2.2. Self-attentive speaker embeddings

Self-attention mechanism can be effectively used to encode a variable-length sequence into some fixed-length embeddings. Inspired by the structured self-attention mechanism proposed in [18] for sentence embedding, we adapt it to improve speaker embeddings in the x-vector baseline system shown in Fig. 1.

In the current x-vector system, the statistics pooling layer treats all the frame-level outputs from its previous hidden layer equally. However, not all frames provide 'equal' speaker-discriminative information to the upper layers. For instance, non-speech frames that unfortunately pass the VAD and short pauses are not useful, and some phonetic contents can be more speaker-discriminative. In this paper, the statistics pooling layer is replaced by a self-attention layer as shown in Figure 2 to derive a weighted mean and a standard deviation vector from the outputs of the previous hidden layer over each speech segment. The weights are learned with the self-attention mechanism to maximize speaker classification performance for the whole system.

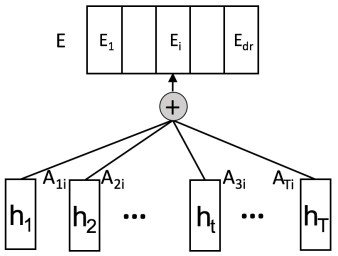

Figure 2: *Structure of the self-attention layer.*

Suppose a speech segment of duration T produces a sequence of T output vectors $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_T\}$, where $\mathbf{h}_t$ is the hidden representation of input frame $\mathbf{x}_t$ captured by the hidden layer below the self-attention layer. Let the dimension of $\mathbf{h}_t$ be d_h. Thus, the size of $\mathbf{H}$ is $d_h \times T$. The self-attention mechanism takes the whole hidden representation $\mathbf{H}$ as input, and outputs an annotation matrix $\mathbf{A}$ as follows:

$$\mathbf{A} = softmax(g(\mathbf{H}^T\mathbf{W}_1)\mathbf{W}_2) \tag{1}$$

where $\mathbf{W}_1$ is a matrix of size $d_h \times d_a$; $\mathbf{W}_2$ is a matrix of size $d_a \times d_r$, and d_r is a hyperparameter that represents the number of attention heads; $g(\cdot)$ is some activation function and ReLU is chosen here. The $softmax(\cdot)$ is performed column-wise.

Each column vector of $\mathbf{A}$ is an annotation vector that represents the weights for different $\mathbf{h}_t$. Finally the weighted means $\mathbf{E}$ is obtained by

$$\mathbf{E} = \mathbf{H}\mathbf{A}. \tag{2}$$

When the number of attention heads $d_r = 1$, $\mathbf{E}$ is simply a weighted mean vector computed from $\mathbf{H}$, and it is expected to reflect an aspect of discriminative speaker characteristics in the given speech segment. Apparently, speakers can be discriminated along multiple aspects, especially when a speech segment is long. By increasing d_r, we can easily have multiple attention heads to learn different aspects from a speaker's speech. To encourage diversity in the annotation vectors so that each attention head can extract dissimilar information from the same speech segment, a penalty term P is introduced when $d_r > 1$:

$$P = \|(\mathbf{A}^T\mathbf{A} - \mathbf{I})\|_F^2 \tag{3}$$

where $\mathbf{I}$ is the identity matrix and $\|\cdot\|_F$ represents the Frobenius norm of a matrix. P is similar to L2 regularization and is minimized together with the original cost of the whole system.

2.3. PLDA backend

We use the same type of PLDA backend as [16, 17] for comparing pairs of embeddings. The embeddings are centered, and projected using LDA, which reduces the dimension from 512

Table 1: *Results on SRE16 with various systems.*

		Cantonese		Tagalog		pool	
		EER(%)	DCF16	EER(%)	DCF16	EER(%)	DCF16
mean only	baseline	7.33	0.516	19.42	0.813	14.06	0.666
	attn-1	6.13	0.500	17.05	**0.783**	12.18	0.642
	attn-2	6.15	0.472	**16.43**	0.791	12.05	0.633
	attn-5	**5.81**	**0.451**	16.44	0.790	**11.88**	**0.623**
mean+stddev	baseline	5.39	0.425	15.20	0.766	11.02	0.596
	attn-1	5.16	0.386	14.49	**0.728**	10.74	0.558
	attn-2	4.84	0.385	14.41	0.736	10.30	0.561
	attn-5	**4.61**	**0.380**	**14.15**	0.730	**10.21**	**0.556**
ivector [16]		8.3	0.549	17.6	0.842	13.6	0.711

to 150. After dimensionality reduction, the representations are length-normalized and modeled by PLDA. The scores are normalized using adaptive s-norm [20].

3. Experimental setup

3.1. Model configuration

In the x-vector baseline system, the inputs size is 115 including context, and there are 512 nodes in each of the first four frame-level hidden layers l_1 to l_4, while the last frame-level layer l_5 has $d_h = 1500$ hidden nodes. Each of the two segment-level layers l_6 and l_7 has 512 nodes. For the self-attention layer, d_a is set to 500.

3.2. Training data

The training data consists primarily of English telephone speech (with a smaller amount of non-English and microphone speech), taken from Switchboard datasets, past NIST speaker recognition evaluations (SRE) and Mixer 6. The Switchboard portion consists of Switchboard 2 Phase 1, 2, 3 and Switchboard Cellular, and it contains about 28k recordings from 2.6k speakers. The SRE portion consists of NIST SREs data from 2004 to 2010 along with Mixer 6 for a total of about 63k recordings from 4.4k speakers. The four data augmentation techniques described in [17], namely, babble, music, noise, and reverb[21] are applied to increase the amount of training data and to improve the robustness of the system. The clean data, together with the augmented data are used to train the speaker embedding DNN system, and only the clean and augmented SRE subset is used to train the PLDA classifier.

3.3. Evaluation

System performance is assessed on NIST 2016 speaker recognition evaluations (SRE16) [22]. SRE16 consists of Cantonese and Tagalog telephone speech. The length of enrollment segments is about 60 seconds, and the length of test segments varies from 10 to 60 seconds. The performance is reported in terms of equal error rate (EER) as well as the official evaluation metric DCF16 for SRE16 [22], which is computed from a normalized detection cost function (DCF) averaged from two operation points with $P_{Target} = 0.01$ and $P_{Target} = 0.005$, respectively.

4. Results

In the following results, '*baseline*' refers to the x-vector baseline described in Section 2.1. The label '*attn-k*' denotes the self-attentive embedding systems described in Section 2.2 with *k* attention heads.

4.1. Overall results

The SRE16 results are summarized in Table 1. In producing the *mean-only* results, the various systems only utilize 1st-order statistics to generate speaker embeddings. That is, 'baseline' computes simple unweighted mean from all the frames of an input utterance, whereas 'attn-k' computes the weighted mean using a self-attention layer. On the other hand, both the 1st- and 2nd-order statistics are used in the *mean+stddev* results.

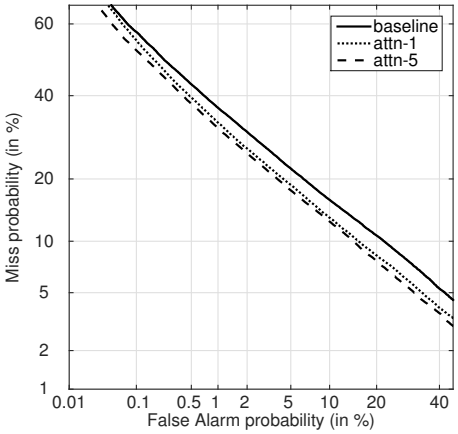

Figure 3: *DET curve for mean-only systems when SRE16 results are pooled across Cantonese and Tagalog.*

In general, self-attention systems outperform the baseline systems that derive speaker embeddings from simple averaging pooling layer, and more attention heads achieve greater improvement. For example, when only mean vectors are used, the single-head attention system is 16% better in EER and 3% better in DCF16 on Cantonese, and 12% better in EER and 4% better in DCF16 on Tagalog. On the other hand, the 5-head system outperform the baseline by 21% in EER and 13% in DCF16 on Cantonese, and 15% better in EER and 3% better in DCF16 on Tagalog. When the performances are pooled across the two languages, the best self-attentive system outperforms the baseline by 16% in EER and 6% in DCF16. Figure 3 shows the detec-

tion error tradeoff (DET) curves for mean only systems when the performance is pooled across Cantonese and Tagalog.

With the incorporation of standard deviation information, the performance of the self-attentive embedding systems is more stable and they continue to outperform the respective baselines. We see that the single-head attention system achieves 5% improvement in EER on both Cantonese and Tagalog. The best multi-head system with 5 heads is 14% better in EER and 10% better in DCF16 on Cantonese, and 7% better in EER and 5% better in DCF16 on Tagalog. Figure 4 reports the DET curve for systems using both mean and standard deviation.

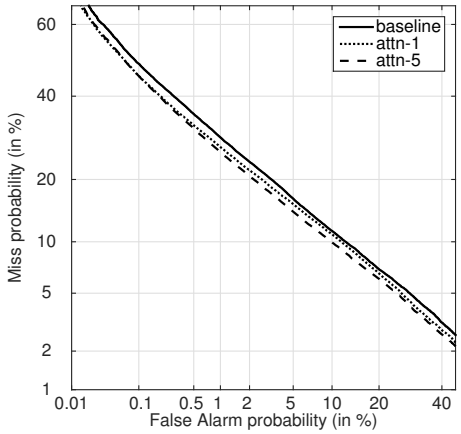

Figure 4: *DET curve for mean+stddev systems when SRE16 results are pooled across Cantonese and Tagalog.*

We also compare the performance of the self-attentive speaker embedding systems with a traditional i-vector system reported in Snyder *et al.* [16]. Pooled across languages, the best 5-head self-attentive embedding system performs better than the i-vector system 25% in EER and 22% in DCF16.

4.2. Results on test utterances of different durations

We also investigate the interplay between performance and utterance duration. Test utterances are divided into 3 groups according to their speech durations. Table 2 and Table 3 report the mean+stddev performance of various systems on the three different duration groups. We can see that with few exceptions, (a) self-attentive embeddings bring improvement across all the different duration groups; (b) as expected the SV performance is better with longer utterances; (c) in general, the self-attentive systems perform better with more heads. For instance, on Cantonese, the single-head system achieves 2% improvement in EER for utterances in the first two groups and 9% in the last group; the improvement in DCF16 is fairly consistent among all duration groups and it is about 10% for the single-head system. Larger gains are made by multi-head systems. The 5-head self-attentive system achieves 13-16% improvement in EER and about 11% in DCF16 for utterances in all the duration groups. On Tagalog, the largest improvement is obtained by the single-head system: it is around 5% better in EER and 2-6% better in DCF16 for all duration groups.

Notice that in our current experiments, in order to provide enough training examples per speaker and to increase diversity in the training examples, we have chunked the training utterances into segments of 200–400 frames. After DNN training, the speaker embeddings are extracted from the entire recoding. Therefore, there may be a mismatch between training and testing duration. If more data are available so that we do not need to chunk the data, even better performance may be achieved.

Table 2: *EER(%) on SRE16*

	baseline	attn-1	attn-2	attn-5
Cantonese				
10s-20s	6.95	6.84	6.22	5.91
20s-40s	5.37	5.29	4.73	4.52
40s-60s	4.39	3.98	3.91	3.83
Tagalog				
10s-20s	18.21	17.51	17.55	17.10
20s-40s	14.84	13.89	13.98	13.86
40s-60s	13.50	12.83	12.37	12.24

Table 3: *DCF16 on SRE16*

	baseline	attn-1	attn-2	attn-5
Cantonese				
10s-20s	0.530	0.485	0.492	0.474
20s-40s	0.432	0.388	0.383	0.383
40s-60s	0.360	0.323	0.324	0.319
Tagalog				
10s-20s	0.837	0.820	0.829	0.828
20s-40s	0.775	0.739	0.738	0.735
40s-60s	0.700	0.656	0.675	0.666

5. Conclusion

We propose a new method to extract speaker embeddings for text-independent speaker verification by introducing a self-attention mechanism into DNN embeddings. The new speaker embeddings are evaluated on SRE16, which is a challenging task since there is a language mismatch between the predominately English training data and the Cantonese or Tagalog evaluation data. We find that the proposed self-attentive speaker embedding outperforms a traditional i-vector system and a strong DNN embedding baseline when tested on utterances of different lengths. By increasing the number of attention heads, consistent improvement is further obtained. We believe the training strategy with chunks of speech segments may not be optimal for self-attention mechanism. In the future work, we will modify the training strategy and try on larger training corpus. We will also investigate different penalty terms for multi-head attention.

6. Acknowledgements

The work described in this paper was partially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. HKUST16215816).

7. References

[1] 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.

[2] S. Ioffe, "Probabilistic linear discriminant analysis," in *European Conference on Computer Vision*. Springer, 2006, pp. 531–542.

[3] P. Kenny, "Bayesian speaker verification with heavy-tailed priors," in *Proceedings of the Odyssey*, 2010, p. 14.

[4] N. Brümmer and E. De Villiers, "The speaker partitioning problem," in *Proceedings of the Odyssey*, 2010, p. 34.

[5] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Proceedings of Interspeech*, 2011.

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

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

[8] D. Garcia-Romero, X. Zhang, A. McCree, and D. Povey, "Improving speaker recognition performance in the domain adaptation challenge using deep neural networks," in *Proceedings of the IEEE Workshop on Spoken Language Technology*, 2014, pp. 378–383.

[9] D. Snyder, D. Garcia-Romero, and D. Povey, "Time delay deep neural network-based universal background models for speaker recognition," in *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop*, 2015, pp. 92–97.

[10] F. Richardson, D. Reynolds, and N. Dehak, "Deep neural network approaches to speaker and language recognition," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1671–1675, 2015.

[11] O. Novotný, P. Matějka, O. Glembeck, O. Plchot, F. Grézl, L. Burget, and J. Černocký, "Analysis of the dnn-based sre systems in multi-language conditions," in *Spoken Language Technology Workshop (SLT)*. IEEE, 2016.

[12] E. Variani, X. Lei, E. McDermott, I. Moreno, and J. 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.

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

[14] S. Zhang, Z. Chen, Y. Zhao, J. Li, and Y. Gong, "End-to-end attention based text-dependent speaker verification," in *Spoken Language Technology Workshop (SLT), 2016 IEEE*. IEEE, 2016, pp. 171–178.

[15] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *Proceedings of the IEEE Workshop on Spoken Language Technology*, 2016, pp. 165–170.

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

[17] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2018.

[18] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," *Proceedings of the International Conference on Learning Representations*, 2017.

[19] 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 *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop*, 2011.

[20] D. E. Sturim and D. A. Reynolds, "Speaker adaptive cohort selection for Tnorm in text-independent speaker verification," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, 2005, pp. I–741.

[21] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 2017, pp. 5220–5224.

[22] "NIST speaker recognition evaluation 2016," https://www.nist.gov/itl/iad/mig/speaker-recognition-evaluation-2016, 2006.