ROBUST SPEAKER RECOGNITION USING UNSUPERVISED ADVERSARIAL INVARIANCE

Raghveer Peri, Monisanka Pal, Arindam Jati, Krishna Somandepalli, Shrikanth Narayanan

Signal Analysis and Interpretation Laboratory, University of Southern California, Los Angeles, CA, USA

ABSTRACT

In this paper, we address the problem of speaker recognition in challenging acoustic conditions using a novel method to extract robust speaker-discriminative speech representations. We adopt a recently proposed unsupervised adversarial invariance architecture to train a network that maps speaker embeddings extracted using a pre-trained model onto two lower dimensional embedding spaces. The embedding spaces are learnt to disentangle speaker-discriminative information from all other information present in the audio recordings, without supervision about the acoustic conditions. We analyze the robustness of the proposed embeddings to various sources of variability present in the signal for speaker verification and unsupervised clustering tasks on a large-scale speaker recognition corpus. Our analyses show that the proposed system substantially outperforms the baseline in a variety of challenging acoustic scenarios. Furthermore, for the task of speaker diarization on a real-world meeting corpus, our system shows a relative improvement of 36% in iос. Furthermore, for the task of speaker diarization on a real-world

1. INTRODUCTION

Obtaining robust speaker embeddings i.e., low-dimensional representations from speech signals that capture speaker characteristics, is a particularly challenging problem given the diverse nature of possible variability in the signal. The signal variability could arise from various nuisance factors such as background acoustic noise, room reverberation, microphone placement etc. The presence of such variability in the signal makes tasks which rely on speaker-discriminative features such as speaker verification and speaker diarization even more challenging. This serves as a motivation to extract speaker embeddings that are invariant to nuisance factors.

Until recently, much of the speaker verification research was based on generative modeling based embeddings such as i-vectors [2]. Since i-vectors contain both speaker and channel information, they fail to provide speaker embeddings robust to the nuisance factors [4], requiring additional supervised compensation steps [5]. With the advances in deep learning technologies, robust speaker modeling approaches based on neural networks have been proposed [6-8]. These techniques can be broadly categorized into two classes: data augmentation and adversarial invariance. In [6], a time-delay deep neural network (TDNN) based model was proposed, that was trained on variable length utterances to generate fixed length speaker representations using a cross entropy loss. In [8], a large corpus of audio recordings from various sources was combined, which was further augmented by artificially adding background noise and music at varying signal-to-noise levels. In order to simulate the effect of reverberation, audio signal was convolved with various room impulse responses. Speaker embeddings, called x-vectors, extracted using this technique have provided state-of-the-art performance in applications such as speaker verification [8] and diarization [9]. One inherent drawback with such data augmentation approaches is that they learn specific variations of the acoustic signal and tend to degrade in performance when tested on unseen acoustic variations, as shown in [10]. Further, data augmentation techniques do not explicitly ensure that irrelevant information is removed from the speaker representations, as shown through various probing tasks in [11].

A promising research direction in this context is domain adversarial training to make speaker representations robust to recording conditions [12-15]. However, a majority of these techniques are supervised, i.e., they require labelled nuisance factors, which might not be readily available in many real-world scenarios. This necessitates unsupervised adversarial training, that can learn speaker representations robust to channel and other acoustic variability without knowledge of any particular nuisance factor. Such work in the speech domain has been largely unexplored. Recently, an unsupervised approach to induce invariance for automatic speech recognition was introduced in [16]. However, the goal of that work was to remove speaker-specific information from the speech representations.

In this work, we explore a method of inducing robustness in speaker embeddings to cope with challenging acoustic environments, where no prior information about the recording conditions is readily available. We adopt a recently proposed unsupervised adversarial invariance (UAI) architecture [17] to extract two disentangled representations from x-vectors. One representation is trained to contain only speaker-discriminative information, while all other information is captured in the second embedding. We empirically show that embedding learnt using our method is able to capture speaker-related information better than the decoderinput x-vectors, while forcing other information pertaining to the nuisance factors to be captured in a second embedding. Our proposed method is different from previously proposed adversarial invariance techniques for speaker embeddings in that, our model does not rely on any supervised information about the nuisance factors.

2. METHODS

2.1. Feature extraction

As described before, x-vectors have shown to be robust for speaker recognition tasks and achieve state-of-the-art performance. Therefore, we use these as input features for our model. The input features were extracted using a pre-trained, publicly available TDNN based embedding system [1]. It takes frame-level MFCC features as input and produces segment-level x-vectors.

[1] https://kaldi-asr.org/models/m7
2.2. Adversarial nuisance invariance

Although x-vectors have produced state-of-the-art performance for speaker recognition tasks, they have also been shown to capture information related to nuisance factors \[^{[11]}\]. Our objective, in using unsupervised invariance technique, is to further remove the effects of the nuisance signals from the x-vectors.

Fig \[^{[1]}\] shows the full UAI architecture, which consists of an encoder (Enc), decoder (Dec), predictor (Pred) and two disentanglers (Dis\(_1\)) and (Dis\(_2\)). Enc maps the input utterance-level x-vector \(x\) into two latent representations \(h_1\) and \(h_2\), each used for different downstream tasks. Pred classifies \(h_1\) as belonging to one of the known speakers producing a one-hot encoded representation at its output, \(\hat{y}\). Meanwhile, a dropout module, Dropout, randomly removes some dimensions from \(h_1\) to create a noisy version denoted by \(h'_1\). Then the decoder Dec takes a concatenation of \(h'_1\) and \(h_2\) and reconstructs the input x-vector, denoted by \(\hat{x}\). In addition, the latent embeddings \(h_1\) and \(h_2\) are passed through two different modules Dis\(_1\) and Dis\(_2\). The goal of these modules is to predict \(h_2\) from \(h_1\) and vice-versa. The parameters of encoder, decoder and predictor are denoted by \(\Theta_e\), \(\Theta_d\) and \(\Theta_p\), respectively, while those of the disentanglers are denoted by \(\Phi_{dis1}\), \(\Phi_{dis2}\). We use categorical cross entropy loss for the predictor \((L_{pred})\) and mean square error loss for both the disentanglers as well as for the decoder \((L_{recon}, L_{Dis1}, L_{Dis2})\).

The central idea in the UAI method is to setup a minimax game between the main model comprising the modules Enc, Dec and Pred and the adversarial model comprising Dis\(_1\) and Dis\(_2\). The goal here is to maximize the predictive power of \(h_1\) for speaker classification and reconstruct the input features with \(h_2\), simultaneously minimizing the predictive power between \(h_1\) and \(h_2\), thereby disentangling the two representations. Equations \[^{[1]}\) and \[^{[2]}\) describe the loss functions of the main and adversarial models, respectively.

\[
\begin{align*}
L_{main} &= \alpha L_{pred}(y, \hat{y}) + \beta L_{recon}(x, \hat{x}) \quad (1) \\
L_{adv} &= L_{Dis1}(h_2, \hat{h}_2) + L_{Dis2}(h_1, \hat{h}_1) \quad (2)
\end{align*}
\]

The task of the decoder is to reconstruct the input x-vectors by minimizing \(L_{recon}\). Since the decoder receives a noisy version of \(h_1\), as the training progresses, Dec learns to treat \(h_1\) as an unreliable source of information for the reconstruction task and thus is forced to squeeze all information into \(h_2\). However, this is not sufficient to ensure that \(h_1\) and \(h_2\) contain complementary information. Hence, explicit “disentanglement” between the two latent representations is encouraged by training the two tasks involving \(L_{main}\) and \(L_{adv}\) in an adversarial fashion, consistent with related previous work \[^{[17]}\) \[^{[18]}\). The adversarial training can be setup as shown in Equation \[^{[3]}\), where \(\alpha\), \(\beta\) and \(\gamma\) are tunable parameters.

\[
\min_{\Theta_e, \Theta_d, \Theta_p, \Phi_{dis1}, \Phi_{dis2}} \max_L \left( L_{main} + \gamma L_{adv} \right) \quad (3)
\]

In our experiments, the modules, Enc, Dec, Pred, Dis\(_1\) and Dis\(_2\) comprised of 2 hidden layers each. Enc and Dec had 512 units in each layer, while the disentangler modules had 128 units in each layer. For Pred, 256 and 512 were used as number of hidden units. The dropout probability of the Dropout module was set to 0.75. We set the weights for the losses as \(\alpha = 100\), \(\beta = 5\) and \(\gamma = 50\). Parameters were tuned by observing the convergence of the losses on a pre-determined subset of the training data. The model was trained for 350 epochs with a batch size of 128. In each epoch, the model was trained with 10 batches of adversarial model update for every 1 batch of main model update. Both objectives were optimized using the Adam optimizer with 1e\(-3\) and 1e\(-4\) learning rates respectively and a weight decay factor of 1e\(-4\) for both. The dimensions for the embeddings \(h_1\) and \(h_2\) were chosen as 128 and 32 respectively.

3. DATASETS

In this work, we designed experiments to analyze general speaker verification performance, while also performing controlled experiments for two main sources of variability that can occur in real-world audio recordings. We also perform probing tasks to understand better the information contained in the speaker embeddings that we extract. In this section, we provide details of the publicly available datasets that we use for the experiments.

AMI: To evaluate the performance of the proposed embeddings on the speaker diarization task, we use a subset of the AMI meeting corpus \[^{[19]}\) that is frequently used for evaluating diarization performance \[^{[20]}\) \[^{[21]}\). It consists of audio recordings from 26 meetings.

V19-eval and V19-dev: We use the VOiCES data corpus \[^{[22]}\) to evaluate the performance of our system with respect to the baselines on a speaker verification task and perform probing tasks to examine the systems. It consists of recordings collected from 4 different rooms with microphones placed at various fixed locations, while a loudspeaker played clean speech samples from the Librispeech \[^{[23]}\) dataset. Along with speech, noise was played back from loudspeakers present in the room, to simulate real-life recording conditions. Fig \[^{[2]}\) shows one such room configuration and data collection setup where “Distractor” represents noise source and the green circles represent the 12 available microphones.

We use two subsets of this data corpus, the development portion of the VOiCES challenge data \[^{[22]}\) referred to as V19-dev and the evaluation portion referred to as V19-eval. V19-dev is used for probing experiments as discussed in Section 4.2, as it contains annotations for 200 speaker labels, 12 microphone locations and 4 noise types (none, babble, television, music). V19-eval is used for experiments to study the robustness of the systems to various nuisance factors. We use the evaluation portion of the dataset for robustness analysis as it contained more challenging recording conditions than the development portion.

Vox: Our training data consists of a combination of the development and test splits of VoxCeleb2 \[^{[24]}\) and the development split...
We evaluate the baseline and the proposed methods for verification task on the V19-eval dataset. Following standard practice [8], we perform dimensionality reduction using linear discriminant analysis (LDA) and score the verification trials using a probabilistic linear discriminant analysis (PLDA) backend for both the proposed embeddings and the baseline. The LDA and PLDA models were learnt on the training data for our proposed system, while for the baseline system we used the pre-trained models. For the embeddings extracted using our method, we use a dimension of 96 after LDA, while for x-vectors we use 150 as the reduced dimension. Consistent with general practice [8], equal error rate (EER) was used as the metric for evaluation.

Following [22] and [27], we used knowledge of the nuisance factors annotations available in V19-eval dataset to study the various factors affecting the performance of speaker verification. For these experiments we consider two distinct nuisance factors, noise conditions: none, babble and television noise and microphone location: far-mic, near-mic and obstructed-mic (microphone hidden in the ceiling).

We further distinguish between the recordings collected at 2 different microphone locations (far-mic vs. near-mic) while examining the performance in noisy conditions.

The experimental setup for the controlled conditions is shown in Fig [2]. As mentioned in Section 3, the green circles, numbered 1-12, represent microphones located at various distances from the main loudspeaker. In all experiments, the enrolment utterances were collected from source data used to playback from the loudspeaker, consistent with [24]. We choose a different set of test utterances depending on the experiment being performed. For example, to evaluate performance in the noisier (near-mic) scenario, we use the utterances that were recorded from mics 1 and 2 as shown in Fig. 2 as the test utterances. Similarly for the noisy (far-mic) scenario, test utterances are pooled from mics 5 and 6.

## 4. EXPERIMENTS

We setup the following experiments to study the different aspects of our system:

1. Robustness analysis of speaker verification (V19-eval dataset)
2. Unsupervised clustering (V19-dev dataset)
3. Speaker diarization with oracle speech segment boundaries (AMI dataset)

**Baseline:** We used x-vectors extracted from the pre-trained model [2] as baseline, to test if the proposed model is able to improve robustness of speaker embeddings by removing the nuisance factors from x-vectors. In the results in Sections 4.1 and 4.2, we denote the baseline method by x-vector and the method using the proposed embeddings by $h_1$. In Section 4.3, the baselines are denoted by Baseline 1 and Baseline 2, which are defined in the section.

### 4.1. Speaker Verification

#### 4.1.1. Setup

We evaluate the baseline and the proposed methods for verification task on the V19-eval dataset. Following standard practice [8], we

---

```plaintext
Table 1: Statistics of datasets (utt refers to utterances, spk refers to speakers)

| Name       | Purpose               | No.utt | No.spk | Nuisance annotations available |
|------------|-----------------------|--------|--------|-------------------------------|
| AMI        | diarization           | 26     | 29     | no                            |
| V19-eval   | verification          | 11,392 | 47     | yes                           |
| V19-dev    | clustering            | 15,904 | 200    | yes                           |
| Vox        | train                 | 1.2M   | 7323   | no                            |

† Refers to number of sessions
```

of VoxCeleb1 [26] datasets. This is consistent with the split that was used to train the pre-trained x-vector model (mentioned in Section 2.1), but with no data augmentation. It consists of speaker annotated in-the-wild recordings from celebrity speakers. As such the dataset is sourced from unconstrained recording conditions. For brevity, henceforth, we refer to this subset of the VoxCeleb dataset as Vox.

Table 1 shows the statistics for the different datasets used in our work. We ensured that the speakers contained in one dataset had no overlap with the speakers from any other dataset.

---

```plaintext
Table 2: Speaker verification (% EER) vs. nuisance factors (V19-eval)

| Noise              | Placement     | x-vector | $h_1$ |
|--------------------|---------------|----------|-------|
| (near-mic)         | none          | 3.34     | 3.99  |
| babble             | 5.41          | 4.86     |
| television         | 3.28          | 4.15     |
| (far-mic)          | none          | 7.43     | 6.26  |
| babble             | 21.93         | 19.79    |
| television         | 10.80         | 9.05     |
| mic placement      | near-mic      | 4.17     | 4.41  |
| far-mic            | 14.97         | 12.79    |
| obstructed-mic     | 6.34          | 5.67     |
| overall            | 10.30         | 9.07     |
```

---

1. We ensured that the speakers contained in one dataset had no overlap with the speakers from any other dataset.
2. For brevity, henceforth, we refer to this subset of the VoxCeleb dataset as Vox.

---

Fig. 2: Example room configuration in VOiCES dataset [22]. Distractor represents noise source and green circles represents microphones

---

[https://kaldi-asr.org/models/m7]
In experiments on the effect of microphone placement, the results show that our method performs comparable to the baseline in near-mic scenario and outperforms the baseline in the more challenging far-mic and obstructed-mic scenarios.

The last row in Table 2 shows the overall speaker verification performance using test utterances from all the microphones under all noise conditions. In this experiment we see a relative 10\% EER improvement by our system over the baseline.

### 4.2. Clustering analysis of embeddings

#### 4.2.1. Setup

In order to further probe the information contained in the latent representations, we analyze clustering performance of the embeddings. We expect \( h_1 \) to perform best when clustering speakers and \( h_2 \) to cluster well with respect to the nuisance factors. We use normalized mutual information (NMI) between the embeddings and the ground truth clusters as a proxy for the speaker/nuisance related information contained in each of the embeddings. The ground truth clusters here are obtained from the annotations available in the V19-dev dataset. Clustering is performed using k-means (mini batch k-means implementation in [28]) with the known number of clusters.

#### 4.2.2. Results

Table 3 reports results comparing the performance of both our embeddings and the baseline in clustering speakers and nuisance factors (noise type and microphone location). We conducted permutation tests [29] between the clustering results of the different experiments to test for statistical significance.

Clustering by speaker, we see that \( h_1 \) performs significantly better than x-vectors (absolute 4.3\% as shown in Table 3). This suggests that our method is able to extract more speaker-discriminative information from x-vectors. Furthermore, as expected, \( h_2 \) showed relatively poor performance in clustering speakers.

Clustering by nuisance factors, as expected, \( h_2 \) is the most predictive. Also, \( h_1 \) doesn’t cluster well according to the nuisance factors. Consistent with our findings in Section 4.1, x-vectors have a significantly higher NMI scores than \( h_1 \) (row 3, Table 3). This suggests that the proposed embedding is able to capture speaker information, invariant to microphone placement better than x-vectors. We found significant differences in the reported NMI scores, suggesting that our method is able to disentangle the 2 different streams of information, speaker-related and nuisance-related.

#### 4.3. Speaker diarization using oracle speech segment boundaries

We further extend the analysis by examining the effectiveness of our proposed speaker embedding in speaker diarization task [30]. Since the goal of this work is to investigate the speaker-discriminative nature of embeddings, we consider only speaker clustering in the diarization task and assume prior knowledge of speaker homogeneous segments and the number of speakers, as was done in past studies [20, 30]. The proposed speaker diarization system (denoted by \( h_1 \)) is based on embeddings extracted from speaker-homogeneous segments followed by k-means clustering as the backend. We compare our system with two competitive baselines that use x-vectors from pre-trained model as input features.

| System | Baseline 1 | Baseline 2 | \( h_1 \) |
|--------|------------|------------|----------|
| Avg. DER (%) | 11.91 | 11.51 | 7.28 |

Table 4: Diarization with oracle speech segment boundaries and known number of speakers (AMI)

One baseline (denoted by Baseline 1) uses k-means clustering on the extracted x-vectors. The other baseline is the state-of-the-art diarization system [30] (denoted by Baseline 2) which uses PLDA scoring and agglomerative hierarchical clustering (AHC).

Diarization error rate (DER) [31] averaged across all sessions in the AMI dataset are shown in Table 4. We see that our proposed system outperforms both Baseline 1 and Baseline 2 systems by a relative 38\% and 36\% in DER respectively. This suggests that the proposed speaker embeddings contain more speaker discriminative information than x-vector embeddings and hence are better suited for speaker clustering across datasets.

### 5. CONCLUSIONS

We present an adversarial invariance approach to obtain speaker embeddings robust to various sources of acoustic variability present in speech signals. The embeddings are learnt by disentangling speaker-related information from all other factors without supervised information about the acoustic conditions. We evaluate these embeddings for various tasks such as speaker verification, clustering and diarization. Experimental results suggest that our method is able to produce robust speaker embeddings in a variety of challenging acoustic scenarios. In the future, we will focus on obtaining speaker representations using low-level audio features such as spectrograms, while further improving their robustness in other challenging acoustic conditions.

### 6. REFERENCES

[1] Gregory Sell, David Snyder, Alan McCree, Daniel Garcia-Romero, Jesús Villalba, Matthew Maciejewski, Vimal Manohar, Najim Dehak, Daniel Povey, Shinji Watanabe, et al., “Diarization is hard: Some experiences and lessons learned for the jhu team in the inaugural dihard challenge.,” in Interspeech, 2018, pp. 2808–2812.

[2] 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, May 2011.

[3] D. Garcia-Romero, X. Zhou, and C. Y. Espy-Wilson, “Multi-condition training of gaussian plda models in i-vector space for noise and reverberation robust speaker recognition,” in 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), March 2012, pp. 4257–4260.

[4] Alex Solomonoff, William M Campbell, and Carl Quillen, “Nuisance attribute projection,” Speech Communication, pp. 1–73, 2007.
