Interspeech 2018
2-6 September 2018, Hyderabad

Deeply Fused Speaker Embeddings for Text-Independent Speaker Verification

Gautam Bhattacharya[1,2], *Jahangir Alam*[2], *Vishwa Gupta*[2], *Patrick Kenny*[2]

[1]McGill University
[2]Computer Research Institute of Montreal (CRIM)

gautam.bhattacharya@mail.mcgill.ca

Abstract

Recently there has been a surge of interest is learning speaker embeddings using deep neural networks. These models ingest time-frequency representations of speech and can be trained to discriminate between a known set speakers. While embeddings learned in this way perform well, they typically require a large number of training data points for learning. In this work we propose deeply fused speaker embeddings - speaker representations that combine neural speaker embeddings with i-vectors. We show that by combining the two speaker representations we are able to learn robust speaker embeddings in a computationally efficient manner.

We compare several different fusion strategies and find that the resulting speaker embeddings show significantly different verification performance. To this end we propose a novel fusion approach that uses an attention model to combine i-vectors with neural speaker embeddings. Our best performing embedding achieves an error rate of 3.17% using a simple cosine distance classifier. Combining our embeddings with a powerful Joint Bayesian classifier, we are able to further improve the performance of our speaker embeddings to 2.22%, which gave a 7.8% relative improvement over the baseline i-vector system.

Index Terms: speaker embeddings, recurrent neural networks, i-vectors, attention model, fusion

1. Introduction

Speaker verification tackles the problem of authenticating a person's identity based on their voice. The problem is setup as follows. Given two recordings, we need to determine if they belong to the same person or not. In this work we consider text-independent speaker verification, wherein there are no constraints placed on the phonetic content of the test recordings [1]. Over the past decade, the i-vector speaker representation has emerged as the dominant approach in text-independent speaker verification [2]. When i-vectors are combined with a probabilistic linear discriminant analysis (PLDA) classifier, state-of-the-art performance is achieved on a number of standard datasets, including the one used in this work. A great strength of the i-vector representation is that it can be learned in an unsupervised way. In doing so, the approach works best when the recordings that the i-vectors are derived from are relatively long (greater than 30 seconds).

A recent trend in the speaker verification community involves using deep neural networks to learn speaker representations [3, 4, 5, 6, 7]. The task is two fold. First a recording of arbitrary length needs to be transformed into a representation of fixed size, and ideally, of low dimensionality. More importantly, this representation needs to be discriminative, so as to allow us to easily discriminate between speakers. We hence forth refer to such representations as speaker embeddings.

In this work we make use of Recurrent Neural Networks (RNN) for learning neural speaker embeddings. RNN's are a natural choice for modeling time-series data such as speech, and have been used extensively in speech and speaker recognition [8, 9]. We draw inspiration from multi-modal machine learning to explore different strategies for learning joint speaker representations by fusing i-vectors and neural speaker embeddings. Joint representations are formed by projecting unimodal representations into multi-modal space [10]. Recently, model-based fusion techniques using neural networks have become popular in several domains including visual question answering, gesture recognition and video description generation [11, 12, 13].

From our experiments we find that by fusing i-vectors with recurrent speaker embeddings during model learning leads to robust embeddings that perform well even with a simple cosine distance classifier. We also note that our model is able to learn speaker embeddings with significantly fewer training datapoints compared to approaches that only time-frequency representations of speech for learning embeddings [3]. These models require many millions of data points for training, whereas our fused model is able to learn speaker embeddings using ≈750000 data-points.

The remainder of the paper is organized as follows. We begin with an overview of our deeply fused speaker embeddings. We highlight different ways in which embeddings can be learned by combining recurrent networks and i-vectors. Moreover, we show from our experiments that the way in which fusion is performed is crucial to the verification performance of the embeddings. Our best model makes use of a content-based attention model, that uses embedded i-vectors to attend over RNN hidden states. This model achieves an equal error rate (EER) of 3.17 % using a simple cosine distance classifier, and further improves to 2.22% when combined with a Joint Bayesian classifier. We also perform an experiment to determine which components of our fused embeddings contribute the most to speaker verification performance. We conclude with some final remarks and propose directions for future work.

2. Deep Fusion Speaker Embeddings

The idea of combining i-vectors with the input to a neural network is not a new one, and is widely used for speaker adaptation in speech recognition systems [14]. In the context of this work, the i-vectors provide speaker and channel information at a global scale, as they are extracted using full recordings. On the other hand, neural speaker embeddings are trained on short snippets of speech and thus capture this information on a shorter time-scale. Unlike the approach in [14], which is an early fusion approach, we are primarily interested in late fusion strategies. Our main contribution is to propose a novel approach for late fusion based on neural attention models. Fusion techniques have also been popular in the speaker verification, albeit at the score

10.21437/Interspeech.2018-1688

level [15]. Recently it was shown that a neural speaker embeddings fuse well with i-vectors at the score level [16]. Drawing inspiration from this result, our goal is to fuse i-vectors into our neural speaker embeddings during model learning.

Figure 1. illustrates the general framework we use to learn speaker embeddings in this work. The raw speech, represented by 40 dimensional log filterbank features is processed by a recurrent neural network. We used 1 second chunks of audio for RNN training. The i-vector corresponding to the same recording is processed by a multi-layer perceptron (MLP). The key component of our model is the embedding module, which decides how to combine the RNN hidden states into a single vector. The output of the embedding module is then concatenated with the output of the MLP before being fed to the output layer of the network.

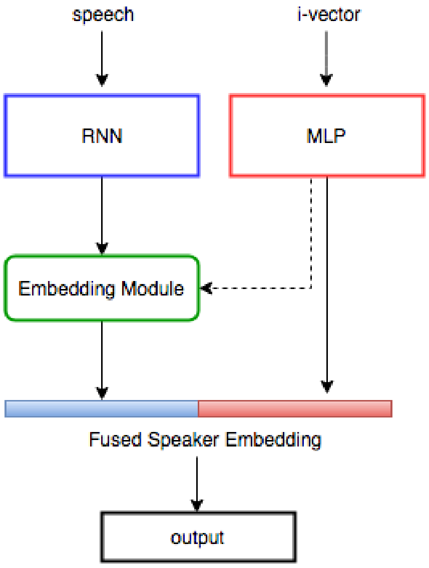

Figure 1: *Deeply Fused Speaker Embedding Framework*

2.1. Recurrent Neural Networks

Recurrent Neural Networks (RNN) extend feed-forward networks in order process sequential data of arbitrary length. This is achieved through a recurrent connection between its hidden states, and by sharing weights across time-steps of the input sequence [17]. These features allow RNNs to capture temporal dependencies in the data. Consequently, RNNs have been used extensively in a variety of machine learning problems ranging from translation and dialogue systems to speech and speaker recognition.

$$h_t = f(W_{ih}x + W_{hh}h_{t-1} + b) \tag{1}$$

Equation 1 represents the output computed by a RNN for a given time-step t. x_t is the input at this time and h_{t-1} is the RNN hidden state at the previous time-step. While RNNs are 'deep in time' by design, performance can often be improved by stacking multiple RNN layer atop one another. In this work we make use of a gated RNN variant known as Long Short Term Memory (LSTM) networks [18], which has been popular due to its robustness to the vanishing gradient problem that plagues vanilla RNNs. Equation 1. also implies that RNNs output a hidden state corresponding to every time-step in a sequence. In this work we show that these hidden states can be combined in a variety of way, and certain combinations lead to significantly better speaker embeddings than others.

2.2. Multi Layer Perceptron

As mentioned in the introduction, i-vectors constrain the speaker and channel variability is a recording to lie in a low dimensional subspace. Consequently, we could use the i-vector directly in our model by either concatenating it to the RNN input or to the output of the embedding module. However, we found it beneficial to non-linearly transform the i-vectors using a single layer feed forward network. More importantly, our model benefited from using separate processing streams for the raw speech and the i-vectors. Non-linearly projecting the i-vector also gives us more flexibility in terms of combining it with the output of the RNN. We provide details of these combination strategies in the next section.

2.3. Embedding Module

The embedding module (green box in figure. 1) represents the key component of our proposed method. This module decides how to combine the sequence of hidden states output by the RNN into a single vector. A simple approach is to take the RNN hidden state corresponding to the last time step of the sequence and treat it as the sequence 'summary'. This summary is then concatenated to the non-linearly transformed i-vector before being fed to the output layer of the network. We refer to this model as 'last-step' in our experiments.

We now describe two other ways of deriving a single vector from a set of hidden states based on the idea of neural attention models. Attention models provide a mechanism that allows neural networks to focus on a specific portion of its input. The approach was made popular by using one RNN to attend over another RNN [19]. Crucially this mechanism is differentiable, which implies that the attention parameters can be learned along with the rest of the parameters of the network.

2.3.1. Self-Attention

Attention models have been most successful in sequence to sequence mapping problems [19]. Neural networks can also use attention to highlight the most relevant parts of the input in classification problems [20]. This type of attention is known as monotonic or self attention. In this work we use the simple monotonic attention model proposed in [21, 22].

$$e_i = a(h_i) \tag{2}$$

$$\alpha_t = \frac{exp(e_t)}{\sum_{k=1}^{T} exp(e_k)} \tag{3}$$

$$c = \sum_{t=1}^{T} \alpha_t h_t \tag{4}$$

The self attention mechanism is driven by a small neural network **a** that assigns a score to each hidden state. These scores are then used to calculate a weighted sum of the hidden states. This weighted average is then concatenated to the MLP output before being fed to the output layer of the network.

2.3.2. Content-driven Attention

Neural attention models are typically driven by query vectors. A query vector is used to generate attention scores for all the time-steps encoded by a RNN. These scores can be generated by using dot product, cosine distance or by a small neural network. The main difference between this model and the self-attention model is how scores are calculated for each hidden state. Equation 2. gets augmented with an additional query vector **q**. Equations 3. and 4. remain the same.

$$e_i = a(h_i, q) \tag{5}$$

In a sequence-to-sequence model, a new query vector is produced at every time-step of the decoder. In our proposed model, the query vector remains fixed, and is represented by the non-linearly transformed i-vector. Instead of a neural network, we use a cosine kernel to compute attention scores. Consequently, we need to insure that the transformed i-vector has the same size the hidden state of the RNN.

We note that this model is the only one that combines the RNN hidden states and the i-vector within the embedding module (dashed line in figure 1). Intuitively, the model is encouraging the hidden states of the RNN to be similar to the corresponding i-vector.

2.4. Loss Function

Neural networks can be used to learn discriminative speaker embeddings by adopting one of two training strategies. The simplest approach is to minimize the cross-entropy loss over speakers in the training set. A criticism of this method is that the network cross-entropy does not match the task we are finally interested in, i.e., verification. A second training strategy involves minimizing a contrastive loss function [23]. These models have been shown to improve performance over cross-entropy models for verification tasks [24], however this difference is quite small for speaker verification. Contrastive losses are also harder to optimize, require more data and often require careful mining of negative examples [24].

In this work we choose to train cross-entropy networks for the purpose of extracting speaker embeddings. One of the reasons that we are motivated to take this approach is due to the specific nature of the dataset used in this work. The NIST-SRE training data set contains a fairly large number of speakers, however, the number of recordings per speaker is fairly small. On the other hand, most of the recordings are quite long. The current state of the art deep speaker embedding model on this dataset originally used a contrastive loss [25]. However, that initial work used a proprietary dataset, and for NIST-SRE they trained cross-entropy models [16].

3. Experiments and Results

All of our experiments were conducted on the NIST-SRE 2010 evaluation set. We used the NIST-SRE 2004-2008 data for training the neural speaker embeddings and the Joint Bayesian classifier. We compare our proposed deeply fused embeddings with i-vectors in terms of speaker verification.

3.1. Feature Extraction

We extracted 40-dimensional log filterbank features from the raw speech using a sliding window of 25ms and a hop size of 10ms. These features were used as input to the RNN. We also extracted 600-dimensional i-vectors using a 2048 component, full covariance Universal Background Model (UBM).

3.2. Network Training

All RNN models are trained using 1 second segments of speech corresponding to 100 log filterbank frames. We filtered the training data and only retained speakers with 4 or more recordings. This leads to a training dataset consisting of 4032 speakers. We use 10% of the training data as a validation set for early stopping. Gradient computation is performed using backpropagation through time (BPTT) and weight updates are done using the Adam optimizer [26].

3.3. Extracting Speaker Embeddings

Once the network is trained, it can be used as a feature extractor to obtain speaker embeddings. We discard the output layer, and use the concatenated outputs of the embeddings module and MLP as our speaker embeddings. We extract embeddings from non-overlapping one second chucks of a recording and average them to obtain an utterance level speaker embedding.

3.4. Comparison of Fusion Techniques

In this experiment we compare the speaker verification performance of the different embeddings described in section 3. We also trained a model by simply concatenating the i-vector to each filterbank frame and fed it to the RNN. This approach is similar to how speaker adaptation is done in speech recognition [14] and is a early fusion method.

For this experiment we considered only the female part of the NIST-SRE 2010 evaluation data. Scoring for speaker verification was performed using cosine distance. We kept the number of RNN hidden units the same for all the models, in order to keep the comparison as fair as possible. Speaker embeddings were extracted from all the model using the procedure detailed in section 3.3.

Table 1: *Comparison of different fusion approaches.*

Model	EER (%)
Early Fusion	15.69
Last Step	13.21
Self-Attention	5.61
IV-Attention	**3.77**

From table 1. we see that the early fusion approach (simple feature concatenation) produces the worst result. The results also suggests that fusing i-vectors with neural speaker after layers of non-linear processing is beneficial. Among the late fusion models, we see that the models employing attention perform significantly better than last-step model. This result is somewhat expected, given that the attention based models make use of all the RNN hidden states as opposed to only the last one. The best performance was shown by the content-driven attention model, with an EER of 3.77%. Notably, this is the only model wherein the i-vector has a direct influence on the RNN through the attention model.

3.5. Tuning Network Depth and Width

In the previous experiment we used a bidirectional RNN with 200 hidden units, and a MLP with 400 hidden units. The fused speaker embeddings are 800-dimensional, comprising of the concatenated forward and backward RNN hidden states and the MLP output. We maintain this symmetry between the outputs of the RNN and MLP in all models. Having established our best performing model, we experimented with the deeper and wider networks using this configuration.

Table 2: *Comparison of different network architectures*

Model	Units	Layers	EER (%)
BiLSTM	200	1	3.77
BiLSTM	200	2	3.83
BiLSTM	**512**	**1**	**3.34**
BiLSTM	512	2	3.40

From Table 2. we see that making the network deeper leads to a slight degradation in verification performance. On the other hand, making the network wider (increasing from 200 to 512 hidden units) does lead to a significant improvement from 3.77% to 3.34% EER. We did not see any further improvement by making the network even wider, or by stacking RNN layers. We make used of speaker embeddings derived from this model for the remainder of our experiments.

3.6. Joint Bayesian Classifier

In our previous experiments we scored speaker verification trials using a simple cosine distance classifier. In this section we experiment with a powerful Joint Bayesian (JB) classifier [27], which has recently been shown to perform slightly better than a probabilistic linear discriminant analysis (PLDA) [28]. An advantage of the JB model compared to PLDA is that there is no need to determine the subspace dimension, and the algorithm was shown to converge faster than PLDA in [27]
Table 3. compares PLDA and the JB model using i-vectors. We see that the verification performance of the two classifiers is comparable.

Table 3: *Speaker Verification using Joint Bayesian model*

Embedding	Classifier	EER(%)
i-vector	PLDA	2.68
i-vector	**JB**	**2.64**

The deeply fused embeddings used in this experiment are derived using a 512-dimensional bidirectional LSTM and a 1024-dimensional MLP for processing i-vectors. The resulting fused embedding is 2048-dimensional. In order to use these embeddings with the joint Bayesian classifier we use principal component analysis to reduce the dimensionality of the embeddings to 600. For these experiments we report verification performance on both male and female parts of the NIST-SRE 2010 evaluation data.

Table 4: *This is an example of a table*

Embedding	Classifier	Female	Male	Pooled
i-vector	JB	2.64	2.23	2.41
Fused	cosine	3.34	3.00	3.17
Fused	**JB**	**2.45**	**2.11**	**2.22**

Table. 4 compares the speaker verification performance of i-vectors and our proposed deeply fused embeddings using the Joint Bayesian model. We see that the fused embeddings improve over the performance of i-vectors yielding a EER of 2.22 compared to 2.41. The fused embeddings also perform quite well using simple cosine distance, showing performance that it competitive with i-vectors/JB.

4. Analysis

The deeply fused speaker embeddings proposed in this work have several useful characteristics. We believe that the results using cosine distance point to the robustness of the embeddings, as it implies that much of the channel effects have been removed. In order gain a deeper understanding of the fused embeddings we perform an experiment wherein we score the individual parts of the fused embedding separately. This is possible as our fused embeddings concatenate the outputs of the embedding module and the MLP.

Table 5: *Comparison of components of Fused Embeddings. Results are reported on the Male part of SRE 2010 using cosine distance for scoring.*

Model	EER (%)
Fused Embedding	3.00
RNN Embedding	3.00
MLP Embedding	45.17

Table 5. compares the performance of the different components of our fused embeddings. Interestingly, we see that the speaker embedding derived from the RNN yields the same performance as the fused embedding. On the other hand, the MLP part performs badly with a EER of 45.17%. While the i-vector plays an important role in shaping the RNN embeddings through the attention model, it appears that in order to do so it becomes less discriminative towards speakers.

5. Conclusions

In this work we proposed deeply fused speaker embeddings that fuse i-vectors with RNN based neural speaker embeddings. We compared different strategies for fusion and found that different methods lead to significantly different verification performance. Our best embeddings use non-linearly transformed i-vectors to drive a content-based attention model, which in turn shapes the RNN embedding. This model achieves a EER of 3.17% with cosine distance, compared to 2.41% achieved by combining i-vectors with a JB classifier. When we use the JB model with the fused embeddings we achieve a 7.8% relative improvement over i-vectors, achieving an EER of 2.22%. From our analysis of the fused embeddings, we make a surprising finding in that most of the discriminative power is retained in the RNN component of the embedding, while the MLP part performs poorly. While this is the case, the MLP part of the embedding still plays a crucial role in shaping the RNN embeddings and remains a crucial component of our model.

Given the good performance of this model with cosine distance, in the future we will explore training our fused embeddings using a contrastive loss function that better matches this scoring criteria. We are also interested in exploring domain adaptation strategies so as to adapt our embeddings to new languages.

6. References

[1] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: From features to supervectors," *Speech communication*, vol. 52, no. 1, pp. 12–40, 2010.

[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, 2011.

[3] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," *ICASSP, Calgary*, 2018.

[4] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *arXiv preprint arXiv:1705.02304*, 2017.

[5] L. Li, Y. Chen, Y. Shi, Z. Tang, and D. Wang, "Deep speaker feature learning for text-independent speaker verification," *arXiv preprint arXiv:1705.03670*, 2017.

[6] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," *arXiv preprint arXiv:1710.10467*, 2017.

[7] C. Zhang and K. Koishida, "End-to-end text-independent speaker verification with flexibility in utterance duration," in *Automatic Speech Recognition and Understanding Workshop (ASRU), 2017 IEEE*. IEEE, 2017, pp. 584–590.

[8] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4580–4584.

[9] 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.

[10] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[11] H. Gao, J. Mao, J. Zhou, Z. Huang, L. Wang, and W. Xu, "Are you talking to a machine? dataset and methods for multilingual image question," in *Advances in neural information processing systems*, 2015, pp. 2296–2304.

[12] N. Neverova, C. Wolf, G. Taylor, and F. Nebout, "Moddrop: adaptive multi-modal gesture recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 8, pp. 1692–1706, 2016.

[13] Q. Jin and J. Liang, "Video description generation using audio and visual cues," in *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*. ACM, 2016, pp. 239–242.

[14] A. Senior and I. Lopez-Moreno, "Improving dnn speaker independence with i-vector inputs," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 225–229.

[15] N. Brummer, J. Cernocky, M. Karafiát, D. A. van Leeuwen, P. Matejka, P. Schwarz, A. Strasheim *et al.*, "Fusion of heterogeneous speaker recognition systems in the stbu submission for the nist speaker recognition evaluation 2006," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2072–2084, 2007.

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

[17] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

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

[19] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[20] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," *arXiv preprint arXiv:1703.03130*, 2017.

[21] C. Raffel and D. P. Ellis, "Feed-forward networks with attention can solve some long-term memory problems," *arXiv preprint arXiv:1512.08756*, 2015.

[22] G. Bhattacharya, J. Alam, T. Stafylakis, and P. Kenny, "Deep neural network based text-dependent speaker recognition: Preliminary results," *Odyssey 2016*, pp. 9–15, 2016.

[23] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, vol. 2. IEEE, 2006, pp. 1735–1742.

[24] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[25] 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 *Spoken Language Technology Workshop (SLT), 2016 IEEE*. IEEE, 2016, pp. 165–170.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[27] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun, "Bayesian face revisited: A joint formulation," in *European Conference on Computer Vision*. Springer, 2012, pp. 566–579.

[28] Y. Wang, H. Xu, and Z. Ou, "Joint bayesian gaussian discriminant analysis for speaker verification," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5390–5394.