JOINT SPEAKER DIARIZATION AND RECOGNITION USING CONVOLUTIONAL AND RECURRENT NEURAL NETWORKS

Zhihan Zhou, Yichi Zhang, Student Member, IEEE, and Zhiyao Duan, Member, IEEE

Dept. of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627, USA

ABSTRACT

Speaker diarization (detecting who-spoke-when using relative identity labels) and speaker recognition (detecting absolute identity labels without timing) are different but related tasks that often need to be completed simultaneously in many scenarios. Traditional methods, however, address them independently. In this paper, we propose a method to jointly diarize and recognize speakers from a collection of conversations. This method benefits from the sparsity and temporal smoothness of speakers within a conversation and the large-scale timbre modeling across recordings and speakers. Specifically, we employ one convolutional neural network (CNN) to perform segment-level speaker classification and another CNN to detect the probability of speaker change within a conversation. We then concatenate the output of both CNNs and feed it into a recurrent neural network (RNN) for joint speaker diarization and recognition. Experiments on different datasets show promising performance of our proposed approach.

***Index Terms*—** Speaker diarization, speaker recognition, convolutional neural network, recurrent neural network, speak change detection

1. INTRODUCTION

Speaker recognition aims to recognize the identity of a speaker from his/her utterances, yet the time boundaries of such utterances do not need to be detected. Speaker diarization, on the other hand, aims to detect "who spoke when" during a conversation, yet speaker identities can be relative within the conversation (e.g., Speaker No. 1 vs. John Smith). In many scenarios, however, speaker recognition and speaker diarization are both needed. Take the call center as an example, it may want to recognize a caller's identity and other paralinguistic parameters (e.g., emotion) from the caller's speech so that it can quickly direct the caller to a specialized agent to improve the caller's satisfaction. In this case, the call center would need a system that is able to diarize the conversation between the caller and the initial agent and recognize the caller's identity against a pretrained model. Therefore, jointly diarizing a conversation and recognizing the identity of conversational partners is an interesting and useful problem to investigate.

One naive way to achieve joint speaker diarization and recognition in a conversation is to segment the conversation into short segments and recognize the speaker(s) (if any) in each segment independently. This, however, does not fully exploit the many useful properties of the problem, which allow the two tasks to benefit each other. On one hand, speaker diarization helps speaker recognition. First, within a conversation, the identity of an active speaker is likely to be stationary within a short period of time; this is a property that speaker diarization techniques often exploit (e.g., speaker change detection [1]), and can help smooth speaker recognition results. Second, within a conversation, there are usually only a few speakers, i.e., the identity of an active speaker at a moment can only come from a small set of people of a large identity database; this can help reduce the search space of speaker recognition significantly. On the other hand, speaker recognition techniques explicitly or implicitly learn speaker models from many recordings of many different speakers. This cross-speaker, cross-context learning helps the speaker models to capture highly discriminative features of speech. When they are applied to speaker diarization, the clustering of the same speaker within a conversation can also be benefited.

In this paper, we develop a method to jointly diarize and recognize speakers from a set of conversations. It not only estimates the time boundaries of utterances of each speaker, but also recognizes the absolute identity of a set of speakers of interest, provided that training speech of these speakers are available. Our method exploits the unique properties of the problem and allows the two tasks to benefit each other.

Specifically, we first use one Convolutional Neural Network (CNN), called CNN1, to classify the absolute speaker identity of the set of speakers of interest on equally spaced segments of each conversation. CNN was first introduced in [2] and has been successfully used in image classification and audio recognition [3, 4, 5]. We incorporate a sparsity term in the loss function to account for the fact that only a few speakers are present in each conversation. We then use another CNN, called CNN2, to perform Speaker Change Detection (SCD) on each conversation to model the temporal continuity of speaker identities, where we design a loss function to

This work was partially supported by the National Science Foundation Grant No. 1617107. We acknowledge NVIDIA's GPU donation as well as Mingqing Yun's effort on data annotation and some experiments.

bias towards false alarms. Finally we concatenate the outputs of both CNNs and feed it into a Recurrent Neural Network (RNN) for joint speaker recognition and diarization. Through the RNN, the CNN1 discriminative features and CNN2 temporal continuity information can be integrated together.

2. RELATED WORKS

Many recent advances adopt i-vector extraction [6, 7, 8] for speaker diarization followed by a probabilistic linear discriminant analysis (PLDA) based scoring function [9] to cluster speakers. However, due to the clustering performance relying on the size of segments, such systems could not work well for short segment processing. Also, feature embedding was proposed to embed the speech utterance into a pre-defined anchor space [10]. Deep neural networks can also be used to create speaker embeddings [11]. However, most speaker diarization systems work for relative label identification. In this paper, we propose to not only estimate the time boundaries of the utterances of each speaker, but also identify the speaker's absolute identity. We further improve our predicted result with Speaker Change Detection (SCD), which determines the specific time of speaker change. A common way [8] is calculating the distance between two sliding windows' contents, using Kullback-Leibler divergence [12] and Generalized Likelihood Ratio as distance metrics. Deep Neural Network (DNN) was also applied in [13], where pre-computed features that contain information about each segment were fed as input to the DNN. Using CNN to detect speaker change has been introduced by [14], in which a conversation is divided into consecutive windows with overlaps and a regression task is performed to predict the speaker change probability between 0 and 1 in each window. In this paper, we further exploited this model for our work.

3. PROPOSED APPROACH

The overall structure of our proposed method is shown in Figure 1. It has two CNNs for segment-level speaker identity classification and Speaker Change Detection (SCD), respectively. Then it is followed by an RNN to integrate the information of classification and SCD together, to generate a more robust speaker identity prediction for each segment.

3.1. CNN1 for Segment-Level Speaker Classification

CNN1 is used to classify recording a spectral segment into a certain speaker identity. The input to CNN1 is a log-mel spectrogram of 0.2 second long (26 frames) with no overlap, with 39 frequency bands covering 0 to 4000 Hz. The STFT frame and hop size are 16 ms and 8 ms.

As shown in Figure 2, each segment corresponds to a label from 0 to N (positive integer), where 0 denotes silence and 1 to N denotes the N possible speakers.

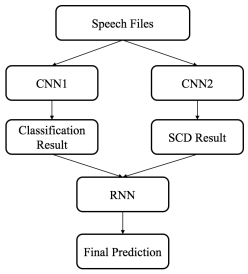

Fig. 1. The overall structure of our proposed method.

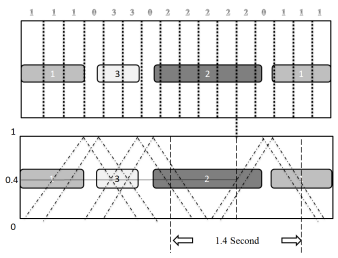

Fig. 2. Recording track segmentation and data preparation. It shows how we separately use different information of the same conversation segment to train CNN1 and CNN2.

In Figure 3, CNN1 consists of 4 convolutional layers and every two convolutional layers are followed by a max pooling layer. For each convolutional layer, zero padding and Batch Normalization (BN) [15] are adopted with Rectified Linear Unit (ReLU) activation. Every fully connected layer has a dropout rate of 0.5 to avoid over fitting [16]. Softmax activation is used in the output layer to generate $N+1$ dimensional probabilistic output.

Theoretically, CNN1 output could have various speaker identity combinations. However, it is reasonable to assume only limited amount of speakers per recording (e.g., 2 speakers) and CNN1 should present sparse output pattern. So we design the CNN1 loss function with sparsity constraint as:

$$loss = y_{true} \times \log(y_{pred}) + \sqrt{y_{pred}}, \tag{1}$$

where the first term is cross-entropy [17] and the second term is an L-0.5 norm regularizer to force the output layer prediction to be sparse. Stochastic Gradient Descent (SGD) is used as the optimizer and the learning rate is 0.01.

3.2. CNN2 for Speaker Change Detection

CNN2 estimates the speaker change probabilities as a regression task. Following [14] with some modifications. The input to CNN2 is a log-mel spectrogram of 1.4 second long (141 frames) and a hop of 200 ms, with 128 frequency bands covering 0 to 4000 Hz. The STFT frame and hop size are 64 ms and 10 ms. Predictions are made on spectrograms of this

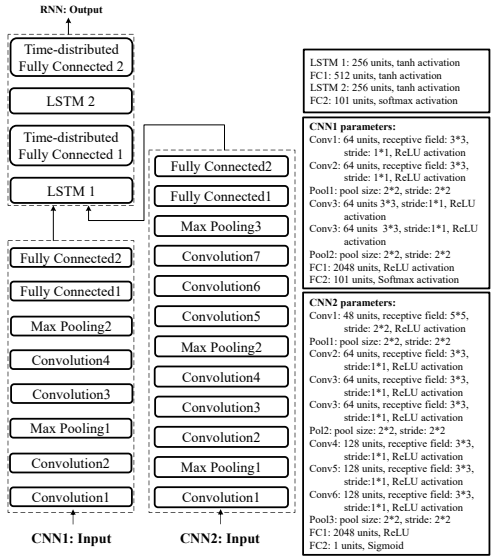

Fig. 3. Our proposed model: CNN1 + CNN2 + RNN. Parameters are for the CALLHOME experiment. Layer sizes are reduced to 1/8 for the prisoner dataset.

size, for every 200 ms throughout a recording. Such setting guarantees synchronized time steps for CNN1 and CNN2.

For annotations, we adopt the triangle-shaped soft boundaries, which has values between 0 and 1, indicating speaker change probabilities described in [14]. Speaker change point is defined as the moment when a certain speaker starts or stops speaking activity. As shown in Figure 2, for each speaker change point, there is a tolerance of 0.6 seconds. For each 1.4-second long window, we denote the center of the windows as t_{mid} and the nearest speaker change time as t_{SCD}, and the label of each window can be represented as:

$$label = \max\left\{0, \frac{5}{3} \times (0.6 - |(t_{mid} - t_{SCD})|)\right\}. \quad (2)$$

In Figure 3, the output layer of CNN2 has only one node with sigmoid activation to generate an output value in (0, 1). We define a new loss function as:

$$loss = (0.1 + y_{true}) \times (y_{true} - y_{pred})^2, \quad (3)$$

which allows windows with larger ground-truth label values to have larger weights. This makes CNN2 detect more speaker change points but prone to false alarms. SGD is used as the optimizer and the learning rate is 0.01.

3.3. RNN for Combining Results Together

In Figure 3, after obtaining the predicted classification and SCD results, they are concatenated and fed into the RNN, which further refines the speaker classification results. We employ two LSTM layers [18] with 128 units and tanh activation, each followed by a time-distributed fully connected layer. Softmax activation is adopted for the output layer. We choose categorical cross-entropy as the loss function, RMSprop as the optimizer, and the learning rate is 0.001.

4. EXPERIMENTAL RESULTS

4.1. Datasets

We first adopt the CALLHOME American English Speech dataset, where 50 conversation recording tracks are used [19]. Each recording has two distinct speakers, so there are 100 different speakers in total. For all recordings, the first 30%, the following 20%, and the rest 50% length of the whole recording, contribute to three subsets denoted as D_{train_c}, D_{train_r}, and D_{test}, respectively. D_{train_c} is used to train both CNN1 and CNN2. D_{train_r} is used to train the RNN, by feeding the CNN1 and CNN2 predictions on it to the RNN. D_{test} is used for testing these trained models.

Then, we use another prison dataset that contains two-side telephone conversation recordings between a prisoner and an external partner. In total there are 10 prisoners and each prisoner has 10 recordings. The same external partner may appear in several recordings of each prisoner, yet only the prisoners' identities are included in the ground-truth, not the external partners. The dataset also contains the prisoner-side only recordings of these conversations, where the external partner's voice is greatly attenuated. We use an energy-based method from pyAudioAnalysis [20] to generate speaker diarization annotations. Specifically, we first denote speaker activities (starting time and ending time) from two-side recordings by energy detection, then follow the same step for the prisoner-side recordings. By subtracting the annotations of the two recordings, the external partner's activities can also be annotated. Finally, we manually checked the annotations and corrected minor errors.

As we do not have the ground-truth identity of the external speakers, we simply treat all of them across all recordings as a single Universal Background Class (UBC). This is in fact a more practical setup in real scenarios. For each prisoner, the first 3 out of the 10 recordings are used to train CNN1. We do not use them to train CNN2, but directly use the pre-trained CNN2 from the CALLHOME dataset. This is to verify our assumption that SCD is less related to the actual identify of speakers. The next 2 recordings of each prisoner are used to train the RNN, by feeding the cNN1 and CNN2 predictions to the RNN; the rest 5 recordings are used for testing.

4.2. CNN1 Restricted to Ground Truth Identities

As CNN-based models are widely used in speaker diarization in recent years [11, 13, 14], to benchmark the performance of our proposed approach, we further constructed another CNN-based segment-level speaker classification method for diarization on the CALLHOME dataset. Differently, however, we provide oracle side information to this CNN. Specifically, we

Table 1. Predicted accuracy (mean ± std) comparisons.

Method	Acc.
(1) CNN1 w/ cross-entropy loss	0.711 ± 0.019
(2) CNN1 w/ sparsity constraint loss	0.741 ± 0.009
(3) CNN1 in (2) + all zeros SCD	0.743 ± 0.008
(4) CNN1 in (2) + predicted SCD	0.829 ± 0.004
(5) CNN1 in (2) + GT SCD	0.867 ± 0.003
(6) CNN1 restricted to GT identities	0.847 ± 0.007

use CNN1 with the sparsity term in the loss function to predict speaker identity within each segment of a conversation, but instead of making predictions across all of the 101 classes, the predictions are restricted to the 3 classes of the ground-truth speakers in the conversation plus the silence. Note that this speaker information is not available in many application scenarios and is not made available to our proposed methods.

4.3. CALLHOME Dataset Result

We train the model on the CALLHOME dataset for 10 times with different initializations. In Table 1, we compare the averaged predicting accuracies by using (1) cross-entropy as the loss function, (2) our proposed loss function with sparsity constraint, (3) integrating sparsity constraint CNN1 with all zeros SCD using RNN, (4) integrating sparsity constraint CNN1 with predicted SCD using RNN, (5) integrating sparsity constraint CNN1 with ground truth SCD using RNN, and (6) CNN1 restricted to ground truth identities. In (3), (4), and (5), RNN serves as the purpose of integrating identity classification and SCD information together.

First, the newly proposed loss function with sparsity constraint not only improves the prediction accuracy but also makes the results more stable with smaller std value, while we notice that the model is occasionally trapped to a local minimum when using cross-entropy as the loss function. Second, integrating CNN1 and CNN2 results significantly improves the classification performance. Compared with (2) that adopts sparsity loss function only , we achieved relatively 11.9% improvement of accuracy in (4), indicating that the temporal continuity information of speakers provided by the SCD result is very helpful for speaker classification. Third, precise SCD is the key for final classification performance improvement. By integrating CNN1 prediction with artificial all-zeros, CNN2 predicted SCD, and ground truth SCD, we observe the trend of increasing RNN classification accuracy. Fourth, the result of our proposed method in (4) is quite close to (6), a method which is only useful when there is specific information to significantly decrease the possible range of every test sample. On the contrary, our proposed method is much more practical since it does not need any additional information but can achieve almost the same accuracy as (6).

Table 2. Precision and recall for 10 prisoners.

ID	Pre.	Rec.	ID	Pre.	Rec.
1	0.921	0.776	6	0.933	0.832
2	0.767	0.836	7	0.235	0.006
3	0.796	0.837	8	0.941	0.753
4	0.786	0.838	9	0.743	0.777
5	0.899	0.830	10	0.370	0.607

4.4. Prison Dataset Result

We use the same method as CALLHOME dataset except that CNN2 model is trained on the CALLHOME data for SCD. As speaker change detection model learns the speaking behaviors, patterns, styles, etc. across different speakers, we assume that this model can generalize well to the new prison dataset. RNN works for a 12 classes classification (10 prisoners + 1 universal other speaker + silence), so the input of RNN is a 12-dimensional vector including the SCD result. We reduce the number of neurons in every layer to be 1/8 of the given structure in Figure 3 to avoid overfitting. Since we only care about the diarization and recognition of the prisoner but not the external partner, this is essentially an information retrieval task. Hence precision and recall are used.

Experimental results for each prisoner are listed in Table 2. First, Performances in most prisoner recordings are consistent. The average precision (0.739) and recall (0.709) suggests that our proposed system works well on the prison dataset. Second, it also can be inferred that SCD information does apply from one dataset to another. It supports our assumption that SCD is more related to the natural conversation patterns, other than the identity of specific speakers. Third, performance from No. 7 and No. 10 prisoners are low, especially for the recall of No. 7 prisoner. By listening to these corresponding recordings, we find that some external partners share very similar timber with the prisoner. In some recordings more than two speakers appear in the conversation. Background music and babble noise are often present in the recordings. All of these factors form the likely reasons of the poor performance on these recordings.

5. CONCLUSION AND FUTURE WORK

In this paper, we proposed a joint speaker diarization and recognition system using two CNNs and one RNN. We used CNN1 to classify the absolute speaker identity, and CNN2 to perform speaker change detection. Outputs from both CNNs are fed into an RNN for joint speaker diarization and recognition. Experiments show that our approach achieves satisfying speaker diarization and recognition results, which is comparable with the extremely powerful but unpractical method: CNN1 restricted to ground truth identities. It also shows that SCD plays an important role in the final RNN prediction.

6. REFERENCES

[1] Zhenhao Ge, Ananth N. Iyer, Srinath Cheluvaraja, and Aravind Ganapathiraju, "Speaker change detection using features through a neural network speaker classifier," *arXiv preprint arXiv:1702.02285*, 2017.

[2] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[3] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 6645–6649.

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2012, pp. 1097–1105.

[5] Yoon Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[6] Mohammed Senoussaoui, Patrick Kenny, Themos Stafylakis, and Pierre Dumouchel, "A study of the cosine distance-based mean shift for telephone speech diarization," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 1, pp. 217–227, 2014.

[7] Yan Xu, Ian McLoughlin, Yan Song, and Kui Wu, "Improved i-vector representation for speaker diarization," *Circuits, Systems, and Signal Processing*, vol. 35, no. 9, pp. 3393–3404, 2016.

[8] Mickael Rouvier, Grégor Dupuy, Paul Gay, Elie Khoury, Teva Merlin, and Sylvain Meignier, "An open-source state-of-the-art toolbox for broadcast news diarization," in *Proc. Interspeech*, 2013, pp. 1477–1481.

[9] Gregory Sell and Daniel Garcia-Romero, "Speaker diarization with plda i-vector scoring and unsupervised calibration," in *Proc. IEEE Spoken Language Technology Workshop*, 2014.

[10] Mickael Rouvier, Pierre-Michel Bousquet, and Benoit Favre, "Speaker diarization through speaker embeddings," in *Proc. 23rd IEEE European Signal Processing Conference (EUSIPCO)*, 2015, pp. 2082–2086.

[11] Pawel Cyrta, Tomasz Trzciski, and Wojciech Stokowiec, "Speaker diarization using deep recurrent convolutional neural networks for speaker embeddings," in *Proc. International Conference on Information Systems Architecture and Technology*, 2017, pp. 107–117.

[12] James M. Joyce, "Kullback-Leibler divergence," in *International Encyclopedia of Statistical Science*, pp. 720–722. Springer, 2011.

[13] Vishwa Gupta, "Speaker change point detection using deep neural nets," in *Proc. Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, 2015, pp. 4420–4424.

[14] Marek Hrúz and Zbyněk Zajíc, "Convolutional neural network for speaker change detection in telephone speaker diarization system," in *Proc. Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, 2017, pp. 4945–4949.

[15] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. International Conference on Machine Learning (ICML)*, 2015, pp. 448–456.

[16] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting.," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[17] Pieter-Tjerk De Boer, Dirk P. Kroese, Shie Mannor, and Reuven Y. Rubinstein, "A tutorial on the cross-entropy method," *Annals of Operations Research*, vol. 134, no. 1, pp. 19–67, 2005.

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

[19] Alexandra Canavan, David Graff, and George Zipperlen, "CALLHOME American English Speech LDC97S42," *Linguistic Data Consortium*, 1997.

[20] Theodoros Giannakopoulos, "pyAudioAnalysis: An open-source python library for audio signal analysis," *PLOS One*, vol. 10, no. 12, 2015.