SPEAKER CHANGE POINT DETECTION USING DEEP NEURAL NETS

Vishwa Gupta

Centre de recherche informatique de Montréal (CRIM)

{Vishwa.Gupta}@crim.ca

ABSTRACT

We investigate the use of deep neural nets (DNN) to provide initial speaker change points in a speaker diarization system. The DNN trains states that correspond to the location of the speaker change point (SCP) in the speech segment input to the DNN. We model these different speaker change point locations in the DNN input by 10 to 20 states. The confidence in the SCP is measured by the number of frame synchronous states that correspond to the hypothesized speaker change point. We only keep the speaker change points with the highest confidence. We show that this DNN-based change point detector reduces the number of missed change points for both an English test set and a French dev set. We also show that the DNN-based change points reduce the diarization error rate for both an English and a French diarization system. These results show the feasibility of DNNs to provide initial speaker change points.

Index Terms— Deep Neural Networks, DNN, change point detection, speaker diarization.

1. INTRODUCTION

Audio or speaker change point detection is the process of locating time points (or frames) in an audio stream that correspond to a transition from one speaker to another, or from music to speech or vice-versa. These change points have many uses including speaker diarization, as clues to possible scene changes in scene analysis, for tracking speakers in a conversation, etc. When the speaker change point detector is part of a speaker diarization system, the goal in general is to use a fast change point detector that may have many false alarms but works well enough that down stream processes (like BIC or GMM based cluster merging [1][2][3][4][5][6]) can merge homogeneous audio segments and eventually lead to a good speaker diarization system.

The basic principle of a fast audio change point detector is to use a short duration window of a few seconds and use a similarity measure to decide whether the midpoint of this window is a potential speaker change point. We then slide this window frame by frame over the entire audio to mark all the potential speaker change points. Some examples of the metrics used to classify the window as a potential change point is symmetric KL2 distance metric [3], Generalized likelihood ratio [4], a local Gaussian divergence measure [1] [2]. In [7], the authors train an autoassociative neural net on the left half of the window (of 1 sec duration) to model the distribution of features in this segment. They then score the right side of the window and use an error threshold to classify whether the mid point of the window is a change point or not. All these methods segment audio into 2 to 3 second segments on an average by controlling a threshold value. The idea is to minimize miss rate for the correct change points. The false alarms can be reduced later by the BIC and GMM-based clustering processes downstream.

In [8], the authors use a Gaussian mixture model (GMM) obtained by adapting the speech segment to a universal background model, and a cross likelihood ratio to segment audio (instead of a KL2 distance metric or a generalized likelihood ratio or a local divergence measure). Their goal was to produce as accurate a change point detector as possible. In most multi-stage speaker diarization systems [1] [2] [4] [5], the GMM clustering stage occurs later on when the clusters are larger and the GMM clustering is much more effective.

Like the systems in [1] [2] [4], we also use a fast change point detector (CPD). This CPD is based on a symmetric KL2 distance metric [5] that quickly generates potential speaker change points. Consecutive change points belonging to the same speaker are then merged using BIC clustering, followed by further refined clustering stages that include GMM-based clustering. This initial change point detector was also the input to the single Gaussian based modified BIC algorithm that gave competitive speaker diarization results [6] in the ETAPE evaluation of French broadcast audio [9].

The idea in this paper is to replace this fast change point detector using a deep neural net (DNN). We train the DNN with short speech segments around the speaker change points from both English and French broadcast audio. We show that the speaker change points generated by the DNN miss fewer speaker change points and lead to a reduction in the overall diarization error rate.

2. ACOUSTIC TRAINING AND TEST DATA

The training audio for DNNs comes from both French and English broadcast audio. The French broadcast audio consists of 44 different broadcasts from ETAPE training data [9] for a total of 26.4 hours of audio. The English data comes from 114 files from 1997 Hub4 English broadcast news training data (97 hours in total). These audio files were chosen because they have been well segmented into speaker turns. The validation set for DNN training was created by excluding 2 English training audio files and one French training audio file from the training set.

For the French test set, we used the ETAPE development set that consists of 15 files of 10 minutes to 1 hour in duration for a total duration of 8.6 hours. The audio files were recorded from French radio and TV programs. These programs contained both broadcast news and talk shows. Roughly 6.7 hours of these audio files were well segmented into speaker turns so that we could measure both the diarization error rate, and the false alarm and miss rate of the speaker change point detector.

For the English test set, We used the 6 audio files from RT 2002 English evaluation data (files bn02en_1 through bn02en_6) since they had good transcription of speaker turns (4.5 hours in total).

This work has been made possible by investments from CANARIE's Research Software Program (www.canarie.ca) and from the Ministère de l'économie, innovation et exportation (MEIE) of Gouvernement du Québec.

978-1-4673-6997-8/15/$31.00 ©2015 IEEE

All the training and test audio was downsampled to 8KHz in order to use the diarization system in an internet-based transcription system where the audio can come from mobile or land-line telephones or from skype or other similar applications connected to the internet using arbitrary microphones. The downsampling increased the diarization error rate (DER) for the French ETAPE Dev set from 13% to 22%, while the DER for the English test set went down from 15% to 13%.

3. SPEAKER CHANGE POINT DETECTION USING DNN

In our current change point detector (CPD) [5], we take a 1.5 second window on either side of the change point and estimate a KL2 distance metric. We estimate this metric for every frame of the audio. The CPD algorithm then looks for a maximum of this KL2 distance metric in these overlapping 1.5 sec windows, and classifies this maximum as a potential change point if it exceeds a distance threshold.

In the case of the deep neural net (DNN) used for CPD, we actually train the DNN so that the output states of the DNN correspond to the presence or absence of a speaker change point in the speech segment input to the DNN. So during DNN training, each frame needs to be labeled in a way that we can later decode the speaker change point with a reasonable precision. If we just label the frames that contain the actual change point as state 0 and the rest of the frames as state 1, then this strategy leads to a very sparse state 0. During scoring with the DNN trained in this fashion, every frame gets labeled with state 1. In order to get reasonable frequency of occurrence of states corresponding to speaker change points, we modified the state labeling as follows.

Let us assume that we input 15 frames to the DNN as shown in Fig. 1. We divide this input into 5 segments of 3 frames each. If the actual speaker change point (CP) falls in the first segment, we label the output state corresponding to the current DNN input as state 0, if the CP falls in the second segment, we label the output state corresponding to the current input as state 1,..., and if the CP falls in the last segment then we label the output state as state 4. If there is no change point in the 15-frame input to the DNN, then the corresponding output state is 5. In this example, we train a DNN with 15 input frames and 6 output states. In this scenario, each state containing the CP occurs three times as frequently, and we can increase the confidence in the change point by looking for frame sequential occurrence of state 4 three times, followed by state 3 three times,..., followed by state 0 three times. (Note that the states will occur from 4 to 0 as we move the DNN over the audio from left to right).

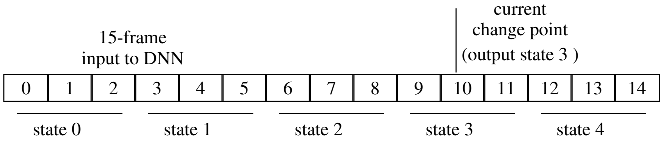

Fig. 1. *15-frame input to DNN illustrating how output state is assigned to this DNN input during DNN training. In this example, the speaker change point corresponds to frame 10, so the output state is 3. If there was no speaker change point inside these 15 frames, then the output state assigned would be state 5.*

During search for speaker change points, we compute the forward scores through the DNN frame by frame. For each frame, we find the output state with the maximum a posteriori likelihood. We then label the frame with the label of this output state. In an ideal scenario, the output state sequence around a change point will look like:

“...5 5 5 4 4 4 3 3 3 2 2 2 1 1 1 0 0 0 5 5 5 ...”

Note that in this example, state 2 corresponds to the change point in the center of the input to the DNN. The change point location accuracy in our case is +/- 3 frames or 30 msec. (Note that frame advance is 10 msec). However, in the ideal case (as in the example above), we assume that the change point is at the center frame of the sequence of frames labeled with state 2. In this case, the CP location accuracy is +/- 1 frame advance (or 10msec). In order to associate a confidence measure with the change point, we add up all the frame synchronous states around a sequence of frames labeled with state 2. For example, in the example above, the confidence count is 15 (we only count states 0 through 4). So in the following sequence of frames labeled with the closest states, the confidence count is 8. The frames that contribute to the confidence count are labeled with a 1 in the 0-1 sequence shown below the state sequence.

“...5 4 5 3 4 5 3 0 3 2 2 2 0 3 1 0 3 5 5 5 5 ...”

“...0 0 0 0 1 0 1 0 1 1 1 1 0 0 1 1 0 0 0 0 0 ...”

In the real scenario, we tried two different strategies. In the first one, we input 101 frames to the DNN (center frame +/- 50 frames) and divide 101 frames into 9 states (16, 10, 10, 10,10,10, 10, 10, 15). In other words, if the change point falls in the first 16 frames input to the DNN, then the output state is 0, if CP falls in the next 10 frames then the output state is 1, ..., and if the CP falls in the last 15 frames (of the 101 frame window) then the output state is 8. Output state 9 corresponds to input frames with no change point. So the maximum confidence count can be as high as 101.

In the second strategy, we input 201 frames to the DNN (center frame +/- 100 frames), and these 201 frames are divided into 19 states depending on where the change point falls in these 201 frames: first 16 frames, next 10 frames, ..., last 15 frames. The output state 20 corresponds to input frames without any change point inside it.

For both these strategies, during decoding, we first look for a sequence of at least 3 frames labeled with the center state (state 4 in the first strategy and state 9 in the second strategy). We assume that the CP is at the center frame of this sequence of frames, and add all the frame synchronous output states from this center frame, and assign this sum as the confidence value of this change point. This confidence value is later used as a threshold to accept or reject the change point.

4. TRAINING THE DNN FOR CHANGE POINT DETECTION

We train the DNN from all the training data (26 hours of French audio and 97 hours of English audio). During training, for every sequence of frames input to the DNN, we need the corresponding DNN output state. These output states are input to the DNN through alignment files in Kaldi [10]. The training audio is already marked with speaker change points through corresponding transcription files. We use these known change points to generate alignment files (in Kaldi format) that contain sequence of states (one output state per frame) as explained in Sec. 3. In most cases, there were only 0 or 1 change points in any given sequence of frames input to the DNN during training. However, in a few cases, there was more than one change point in the sequence of frames input to the DNN. In this case, the identity of the DNN output state label was based on the change point closest to the center frame.

We initially trained the DNN with the entire 123 hours of training audio (26 hours of French and 97 hours of English). This data contains a total of 27,500 change points, or approximately one change point every 16 seconds. In this training scenario, for the first strategy with 10 output states, state 10 (no change point in the window) occurs 140 times more frequently than the other states. In the second strategy with 20 output states, state 20 (no change point in the window) is 120 times more frequent than the other states. The result of training DNN with such a biased training data is that during decoding, only state 10 (or 20 for the second strategy) shows up as the state with the highest likelihood for virtually all the test frames.

In order to balance the occurrence of different states, we trained the DNN only with the frames within 1.1 secs of each change point for the first strategy (with 10 output states). In this scenario, state 10 is only about 10 times more frequent than the other states. However, the total training audio is reduced from 123 hours to 16.8 hours. In the second strategy (with 20 output states), we train the DNN with the frames within 2.1 secs of each change point. In this case, state 20 is about 20 times more frequent than the other states, while the training audio is reduced to 32 hours. In these scenarios, we were able to train the DNNs and get reasonable CPD results. The validation data was also similarly modified. So the validation data contains only 1.1 secs of audio on each side of a change point (1st strategy) or 2.1 secs of audio on each side of a change point (2nd strategy).

To get good CPD accuracy, we tried different architectures for the DNNs, varying from 2 to 5 hidden layers and from 50 to 1000 neurons per hidden layer. It was only possible to train DNNs with 2 hidden layers starting from random initialization of weights. The DNNs with more than 2 hidden layers were trained by first training a DNN with 2 hidden layers, then iteratively adding one more hidden layer with random initialization and then retraining the DNN. The DNNs between 3 and 5 hidden layers with between 400 and 1000 neurons gave good CPD accuracy. With the fully trained DNN, the frame accuracy for the validation data varied between 55% and 62%.

We tried two different feature parameters: cepstrum and delta cepstrum as one feature set, and TRAP features [11] as another feature set. The cepstrum + delta cepstrum based features gave significantly worse results than the TRAP features. To compute the TRAP features, we first normalize the 23-dimensional filterbank features to zero mean per audio file. For the first strategy, 101 frames of these 23-dimensional filterbank features (50 frames on each side of current frame) are spliced together to form a 2323-dimensional feature vector. This 2323-dimensional feature vector is transformed using a hamming window (to emphasize the center), passed through a discrete cosine transform and the dimensionality is reduced to either 23*20 or 23*40 or 23*60. The number 20, 40 or 60 is the number of DCT values we keep when we take the cosine transform of each feature in the 101 frame window. It turns out that 40 works the best in the first strategy (with 101 frame window), and a value of between 60 and 100 works better in the second strategy with a 201 frame window. So in the first strategy, the input feature vector is 920-dimensional (23*40) and in the second strategy, the feature vector is either 60*23 dimensional or 100*23 dimensional. This feature vector is globally normalized to have zero mean and unit variance. This normalized feature vector is then input to the DNN. The feature vector is advanced by one frame every time. Even though we train each DNN with both the English and French training data together, the same DNN did not give the best CPD results for both English and French.

5. CHANGE POINT DETECTION RESULTS WITH DNN

The speaker change points are found by scoring each frame with the DNN to find the output state posterior likelihoods. The input to the DNN is the TRAP features computed from this frame together with +/- 50 frames (in the first strategy) or with +/- 100 frames (in the second strategy) as outlined in the previous section. This frame is then given the label of the output state with the highest posterior likelihood. This frame labeling process converts the frame sequence into an output state sequence. We then look for a contiguous sequence of the output state corresponding to the change point in the center of the window. In the first strategy (with 10 states), we look for a contiguous sequence of state 4 with a minimum run of 3. The center of every such run is marked as a potential change point. The confidence corresponding to each potential change point is also computed as outlined in Sec. 3.

We then compute the miss rate and the false alarm rate for the change points at a specified confidence threshold. The miss rate is the percentage of actual change points missed. An actual change point is missed if there is no change point within 0.25 seconds of this actual change point. Similarly, we compute false alarms as change points that are not within 0.25 secs of any actual change point.

The DNN that gave the best results for the English test set (from RT 2002) corresponds to DNN trained with strategy 1 (with a total of 10 output states) has 3 hidden layers with 1000 neurons per hidden layer and sigmoid non-linearity in each hidden layer, one input layer, and one softmax output layer. The TRAP features input to the DNN have 23x40 values (keep 40 DCT values out of 101) computed from a window of 101 frames (center frame +/- 50 frames). The English test set has a total duration of 3445 secs (or 57.4 mins) with 183 change points or roughly one change point per 19 secs.

Table 1 gives the miss rate and the false alarm rate per min at different confidence thresholds. We measure the false alarm rate by average number of false alarms per minute. We compare these performance figures with the change point detector using KL2 metric in our multi-stage diarization system [5] used during ETAPE evaluation. The false alarms and the the miss rate for this KL2-metric-based change point detector are shown in Table 2. As we can see from the two tables, the miss rate at the same false alarm rate is significantly lower for the DNN-based change point detector.

Table 1. *Percentage missed and false alarm rate/min for DNN-based change point detector for English test set.*

conf thresh	0	10	20	30	40	50
% missed	11	12	16	24	31	43
false alarms/min	29.3	26.9	22.7	18.5	13.7	9.6

Table 2. *Percentage missed and false alarm rate/min for KL2-metric-based change point detector for English test set.*

% missed	60
false alarms/min	33.0

The DNN that gave the best results for the French Dev set (the French Dev set for speaker diarization during ETAPE evaluation in 2011) corresponds to DNN trained with strategy 1 (with a total of 10 output states), has 5 hidden layers with 400 neurons per hidden layer and sigmoid non-linearity in each hidden layer, one linear input layer, and one softmax output layer. The TRAP features input to the DNN have 23x40 values (keep 40 DCT values out of 101) computed from a window of 101 frames (center frame +/- 50 frames). The French dev set has a total duration of 24354 secs (or 6.765 hours) with 2214 change points or roughly one change point per 11 secs.

The French dev set is more difficult than the English test set as it contains both broadcast news and talk shows. The talk shows have short speaker turns, speaker overlaps and speech with music segments and background audience noise.

Table 3. *Percentage missed and false alarm rate/min for DNN-based change point detector for French dev set.*

conf thresh	0	10	20	30	40	50
% missed	26	29	36	43	52	62
false alarms/min	40.0	35.8	30.0	24.3	19.1	13.6

Table 4. *Percentage missed and false alarm rate/min for KL2-metric-based change point detector for French dev set.*

% missed	75
false alarms/min	32

Table 3 gives the miss rate and the false alarm rate per min at different confidence thresholds. We compare the performance figures in Table 3 with the change point detector using KL2 metric embedded in our multi-stage speaker diarization system. The false alarms and the miss rate for this KL2-based change point detector are shown in Table 4. As we can see from the two tables, the miss rate at the same false alarm rate is significantly lower for the DNN-based change point detector.

6. SPEAKER DIARIZATION USING CHANGE POINTS FROM DNN

In order to evaluate the speaker diarization performance of the change points obtained from the DNNs, we substituted the change points from the KL2 metric in our speaker diarization system by the change points from the DNNs. The two diarization systems are shown side by side in Fig. 2. The left side of the figure uses the KL2 metric for change point detection while the right side uses the change points from the DNNs. The rest of the flowchart is kept exactly the same. We do adjust the various thresholds in order to get minimum DER.

For the change points from the DNNs, we adjusted the confidence threshold per audio file to get one false alarm per n secs of audio for each file. Table 5 shows the diarization error rate (DER) for the English test set with varying n, while Table 6 shows the DER for the French dev set with varying n. We see that at $n = 2.4$, the DER for English test set goes down from 12.91% (for KL2-based CPD) to 12.51% (for DNN-based CPD), and for French dev set the DER goes down from 21.95% (for KL2-based CPD) to 21.12% for the DNN-based CPD. The reduction in DER is small probably due to the limitation of the GMM-based agglomerative clustering algorithm.

Table 5. *Variation in diarization error rate (DER) with n for the English test set. The DER with the KL-2 based change point detector is 12.91%*

n	2.2	2.3	2.4	2.5	2.6	2.8	3.1
DER	12.8	12.65	12.51	12.52	12.58	12.76	12.85

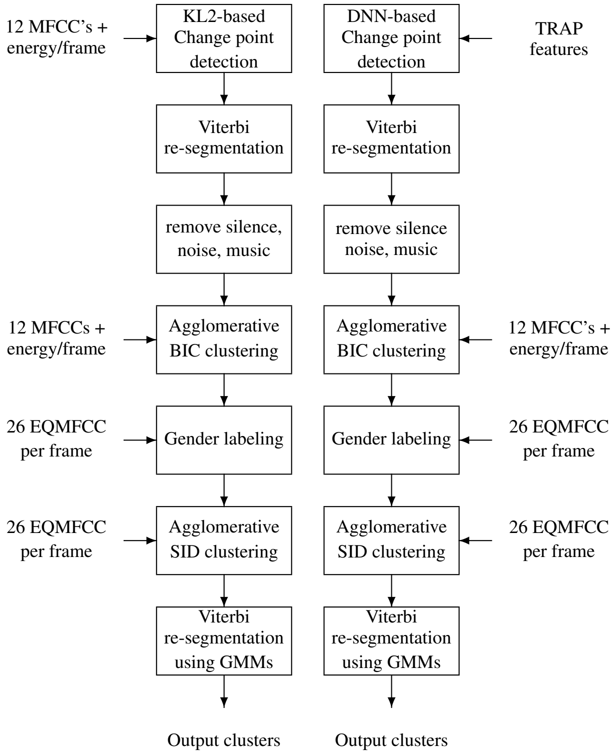
12 MFCC's + energy/frame
KL2-based Change point detection
DNN-based Change point detection
TRAP features
Viterbi re-segmentation
Viterbi re-segmentation
remove silence, noise, music
remove silence noise, music
12 MFCCs + energy/frame
Agglomerative BIC clustering
Agglomerative BIC clustering
12 MFCC's + energy/frame
26 EQMFCC per frame
Gender labeling
Gender labeling
26 EQMFCC per frame
26 EQMFCC per frame
Agglomerative SID clustering
Agglomerative SID clustering
26 EQMFCC per frame
Viterbi re-segmentation using GMMs
Viterbi re-segmentation using GMMs
Output clusters
Output clusters

Fig. 2. *The multi-stage speaker diarization system using KL2 metric as the change point detector is shown on the left, while the speaker diarization system using the DNN-based change point detector is shown on the right.*

Table 6. *Variation in diarization error rate with n for the French test set. The DER with the KL-2 based change point detector is 21.95%*

n	2.3	2.4	2.5	2.6
DER	21.63	21.12	21.15	21.31

7. CONCLUSIONS

We have shown that we can train DNNs to locate speaker change points that can be used as a starting point in a speaker diarization system. Compared to a change point detector using a KL2-based metric, these change points have a lower miss rate at the same false alarm rate. The key to training reasonable DNNs for change point detection is to associate output states with the location of the change points within the frames input to the DNN, and to train the DNN with only the audio around the speaker change points. We also show that substituting the change points from KL2 metric by the change points from DNNs for speaker diarization results in a lower DER for both an English and a French diarization system.

Currently, the DNNs for change point detection are trained with only a small amount of acoustic data (16.8 hours). The reason is that we can only train from audio around the speaker change points. We plan to increase this data significantly in order to see its effect on change point detection. Any significant results from these experiments will be reported at the conference.

8. REFERENCES

[1] Barras, C., Zhu, X., Meignier, S., Gauvain, J., “Multistage Speaker Diarization of Broadcast News”, IEEE Trans. ASLP, vol. 14, no. 5, 1505–1512, 2006.

[2] Sinha, R., Tranter, S. E., Gales, M. J. F., Woodland, P. C., “The Cambridge University March 2005 Speaker Diarisation System”, Interspeech 2005, pp. 2437–2440.

[3] Siegler, M., Jain, B., Stern, R., “Automatic segmentation and clustering of broadcast news audio”, Proc. DARPA Speech Recognition Workshop, Feb. 1997, pp. 97–99.

[4] S. Meignier and T. Merlin, “LIUM SPKDIARIZATION: An open source toolkit for diarization”,in CMU SPUD workshop, Dallas, Tx, 2010.

[5] V. Gupta, G. Boulianne, P. Kenny, P. Ouellet, and P. Dumouchel, “Speaker Diarization of French Broadcast News”, Proc. ICASSP-2008, pp. 4365–4368.

[6] T. Stafylakis, P. Kenny, V. Gupta, and P. Dumouchel, “Compensation for inter-frame correlations in speaker diarization and recognition”, Proc. ICASSP-2013, pp. 7731–7735.

[7] S. Jothilaksmi, S. Palanivel, V. Ramalingam, “ Unsupervised speaker segmentation using autoassociative neural network”, International Jour. Comp. Appl., vol. 1, no. 7, 2010.

[8] A. Malegaonkar, A. Ariyaeeinia, and P. Sivakumaran, “Efficient speaker change detection using adapted Gaussian mixture models”, IEEE Trans. ASLP, vol. 15, No. 6, Aug 2007, pp. 1859–1869.

[9] G. Gravier, G. Adda, N. Paulsson, M. Carré, A. Giraudel, O. Galibert,“The ETAPE corpus for the evaluation of speech-based TV content processing in the French language”, Proc. LREC 2012, Istanbul, Turkey.

[10] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hanneman, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer and K. Vesely, “The Kaldi Speech Recognition Toolkit”, IEEE 2011 Workshop on Automatic Speech Recognition and Understanding, 2011.

[11] F. Grézl, “TRAP-based Probabilistic Features for Automatic Speech Recognition”, Doctoral Thesis, dept. Computer Graphics & Multimedia, Brno Univ of Technology, Brno 2007.