Improving the Speech Activity Detection for the DARPA RATS Phase-3 Evaluation

Jeff Ma

Raytheon BBN Technologies, Cambridge, MA 02138, USA
jma@bbn.com

Abstract

This paper presents the work that we conducted for building the speech activity detection (SAD) systems for the phase 3 evaluation of the RATS program. The work focused on improving the SAD performance with the neural network (NN) approach. The major efforts include reducing the false rejections errors by extensions of speech regions in the training references and use of post-processing NNs, and removing channel variations by design of channel bottleneck features with the deep NN learning approach. With these efforts more 25% relative improvements were achieved over the phase 2 evaluation system. The bigger contribution of the design of the bottleneck features was the enhancement of the SAD system performance on new channels. Our results revealed that the bottleneck features were able to improve SAD performance on new channels significantly.

Index Terms: speech activity detection, neural network, bottleneck features

1. Introduction

The DARPA RATS (Robust Automatic Transcription of Speech) program focused on developing techniques for performing tasks on audios transmitted through communication channels that are extremely noisy or highly distorted. It included 4 tasks: SAD, language identification (LID), speaker identification (SID) and key word spotting (KWS). This paper presents work we conducted on the SAD task. SAD detects when human speech occurs in audio signals and is an important component of many speech processing applications. The Patrol team, including BBN, University of Maryland (UM), Brno University of Technology (BUT), etc., worked on building SAD systems. For the first two phase evaluations of the RATS program, the team built frame-level speech/non-speech classifiers through two approaches, Gaussian mixture model (GMM) and multi-layer perceptrons (MLP) [1]. BBN mainly worked on building the GMM system. It was found that the noise-robust cortical feature [2], developed at UM, greatly improved the SAD performance of the GMM system. BUT worked on building the MLP system. The GMM and MLP systems were combined to produce the final outputs. During the 2nd phase evaluation BBN also took effort in building SAD systems with neural network approaches. However, due to the time limit, the NN systems were not trained well to produce good results as expected. For the 3rd phase evaluation BBN continued working on the NN approaches. This paper describes our work for improving the NN systems.

This paper is organized as follows: Section 2 addresses the data used for developing our SAD systems. Section 3 elaborates on a design of bottleneck features with the deep NN approach to capture channel variation information and integration of the features into the SAD NN training. Section 4 describes the improvements we made on the NN systems, and Section 5 presents the benefits from the bottleneck features on new channels.

2. Data

The Linguistic Data Consortium (LDC) created data collections for the RATS participants to develop systems. For the SAD task LDC selected audio recordings from existing speech corpora that have annotations, such as the Fisher English and Arabic Levantine conversational telephone speech collections, and also created new collections specific to the RATS project. The new collections include telephone conversations in Arabic Levantine, Pashto, and Urdu. All these recordings were retransmitted through 8 different noisy communication channels, labeled with letters A through H [3]. Three data releases were issued for the SAD task, which include 2,077 hours of audio in total. For simplicity of notation, we denote this training data corpus as “Train2Khrs”. LDC also provided reference annotations that mark speech (S), non-speech (NS), and non-transmitted (NT) regions for all the data. We used all this data to train our SAD systems.

LDC also issued test data sets for the development purpose. One of the test sets was named “dev2”, which includes 389 audio files and about 100 hours of data in total. In this paper we report SAD performance measured on this data set.

To measure the SAD performance, two types of errors are defined as: 1) false rejection (FR) rate, \( P(\text{FR}) = \frac{D(\text{FR})}{D(S)} \) where \( D(\text{FR}) \) is the total durations of falsely rejected speech and \( D(\text{FA}) \) the total duration of falsely accepted non-speech; \( D(S) \) and \( D(\text{NS}) \) are the total amounts of scored speech and non-speech, respectively. While these errors are computed, forgiveness collars of 500ms and 200ms are applied to the non-speech and speech sides, respectively, of each annotated non-speech/speech boundary in the references. In this paper we report SAD performance measured in terms of two metrics, the equal error rate (EER) – the operating point at which the \( P(\text{FR}) \) is equal to \( P(\text{FA}) \) and “PER@1%” – the \( P(\text{FR}) \) when \( P(\text{FA}) \) is equal to 1%, which is one of the RATS SAD operating points for the phase 3 evaluation.

3. Features

With the GMM approach each of the speech and non-speech classes were modeled with one GMM. 512-component GMMs were trained with the Maximum Likelihood (ML) criterion for the phase 1 evaluation system. For the phase 2 evaluation the number of components was increased to 2,048 and the GMMs were trained with the maximum mutual information (MMI) criterion, which improved the SAD performance significantly. For the phase 3 evaluation we focused on the neural network approach, NNs were trained to estimate posteriors of frames being speech or non-speech.
3.1. Long-span features
As shown in [1], used together with the PLP features, the cortical features (Cort) improved performance of the GMM system significantly. The cortical features are high-dimensional, multi-scale spectrotemporal modulation features that are extracted from a window of 0.5 seconds and have been shown to be very robust to noise on a speech detection task [3]. The dimensionality was reduced to 140 by using tensor principal component analysis (PCA) when used in the SAD training. The best configuration for combining the types of features was found [1] to append the 140-dimensional cortical features to the concatenation of 31 consecutive PLP frames. Each PLP frame consists of energy and the first 14 coefficients, thus the dimension of the combined features was 605 (≈31×15+140). The heteroscedastic LDA (HLDA) was then applied to reduce the dimensionality to 45 for training the GMMs. We trained NNs with the same feature configuration, but removing the HLDA projection. The 605-dimensional features were directly used as inputs to the NN training. The HLDA projection can be treated as one hidden layer of NN and jointly trained with other layers to minimize the S/NS classification errors.

3.2. Channel neural network bottleneck features
We trained channel dependent (CD) GMMs for the previous evaluation systems. Since each channel had sufficient data (> 200 hours), the CD GMMs were able to better model the different characteristics of the 8 channels than the channel independent (CI) GMMs did. Rather than modeling different channel characteristics, one more popularly used strategy is to remove channel variations from model training by various normalization techniques, such as listed in [4]. We tried to use the constrained maximum likelihood linear regression (CMLLR) technique [5, 6, 7] to remove the channel variations from the SAD system training. Since the clean audios are available, we estimated a CMLLR transform for each channel separately. The two SAD NNs were set to have the same hidden structures (two hidden layers, 800×60). The hidden layer for the NNs were configured as “600x30x300” and “600x60x300”, and thus the CBN features were directly used as inputs to the NN training. The best configuration for combining the types of features was found [1] to append the 140-dimensional cortical features (Cort) improved performance of the GMM system significantly. The cortical features are high-dimensional, multi-scale spectrotemporal modulation features that are extracted from a window of 0.5 seconds and have been shown to be very robust to noise on a speech detection task [3]. The dimensionality was reduced to 140 by using tensor principal component analysis (PCA) when used in the SAD training. The best configuration for combining the types of features was found [1] to append the 140-dimensional cortical features to the concatenation of 31 consecutive PLP frames. Each PLP frame consists of energy and the first 14 coefficients, thus the dimension of the combined features was 605 (≈31×15+140). The heteroscedastic LDA (HLDA) was then applied to reduce the dimensionality to 45 for training the GMMs. We trained NNs with the same feature configuration, but removing the HLDA projection. The 605-dimensional features were directly used as inputs to the NN training. The HLDA projection can be treated as one hidden layer of NN and jointly trained with other layers to minimize the S/NS classification errors.

We trained NNs to classify the 9 channels (8 noisy ones plus the clean one) with the SAD training data, which had channel information provided. The NNs were set to have 3 hidden layers and the middle hidden layers (HL) were chosen as the bottleneck layers. We called the linear outputs from nodes at the bottleneck layers as channel bottleneck (CBN) features. These CBN features were assumed to carry channel variation information. Then, we integrated these CBN features into the SAD NN training, hoping the channel variations were removed from the original features during the NN training. We found that it was important to normalize the CBN features before using them. Our comparisons showed the Gaussian normalization (GN) conducted on each audio separately produced the best results. Figure 1 demonstrates the generation and normalization of the CBN features and the integration of the features into the SAD NN training.

The NN training with more than 2,000 hours of data is time-consuming. So we first explored the CBN features with a small training corpus that was created by randomly selecting ¼ of the data from each of 9 channels in the “Train2Khrs” training corpus. The total amount of data was about 500 hours.

We set the bottleneck layer to have much fewer nodes than other hidden layers do, thus the CBN features are low dimensional and can be integrated at ease into the SAD NN training. We then trained the SAD NNs with each of the “CBN30” and “CBN60” features appended to the input features, separately. The two SAD NNs were set to have the same hidden structures (two hidden layers, 800x60). The performance of the two SAD NNs is shown in Table 1.

Table 1. Performance of NN SAD systems trained with 30, 60 and 100 dimensional CBN features

<table>
<thead>
<tr>
<th>Features (dimension)</th>
<th>EER</th>
<th>PFR@1%</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLP+Cort (605)</td>
<td>2.02</td>
<td>4.09</td>
</tr>
<tr>
<td>PLP+Cort+CBN30d (635)</td>
<td>1.83</td>
<td>3.30</td>
</tr>
<tr>
<td>PLP+Cort+CBN60d (665)</td>
<td>1.89</td>
<td>3.48</td>
</tr>
</tbody>
</table>

First, compared to the baseline (no CBN features used), both of the CBN features were able to improve the SAD performance (5-15% relative). Second, the larger dimensional CBN features (“CBN60”) did not produce better performance (slightly worse, eventually). So, we limited the dimension of the CBN features to be under 60 in our experiments after.
4. Training NN SAD systems

We used the “Train2Khrs” training corpus in all training experiments reported here. We first trained two CI NN systems with two (1500x500) and three (800x150x400) hidden layers, which are denoted as “CI-NN1” and “CI-NN2”, respectively. As shown in Table 2 (the column “Chn-Dep”) indicates if the system is channel dependent or not), the two CI NNs produced similar performance, so uses of 2 or 3 hidden layers was not critical to the NN training. A simple combination (by taking average) of the two CI NNs (denoted as “CI-NN1, 2”) produced minor gains. Though the gain was not large, we adopted this two-NN combination strategy in our final system training to reduce the risk that training of a single NN may not converge well. The “MMI-GMM” system, listed in Table 2 as well, is the 2,048-component GMM system trained with the MMI criterion – our best single system for the phase 2 evaluation. As can be seen, the CI NNs significantly outperformed the CD “MMI-GMM” system (over 20% relative in terms of EER).

As mentioned earlier, the CD GMM systems produced better performance. The CD NN systems reported in [10] also produced superior performance. Therefore, we worked on building channel dependent NN systems as well. The outputs were the weighted sums of the CD NNs and the weights were the corresponding channel posteriors. We trained a 5-layer channel classification NN (structured as “605x800x50x500x9”) to estimate posteriors for the 8 channels. The channel classification accuracy measured on the “dev2” test set was 100%.

Due to uses of less training data we trained CD NNs with reduced sizes. Two CD NN systems, trained with 2 and 1 hidden layers (denoted as “CD-NN2” and “CD-NN3”), respectively, are show in Table 2. As can be seen, the two CD NN systems produced similar performance. However, they fell behind the CI NNs. This indicates that the data from the different channels benefited each other in the CI NN training. Due to the time limit, we did not further investigated on the CD NNs and just concentrated on working with the CI NNs.

As mentioned earlier, forgiveness collars of 0.5s and 0.2s are applied to the non-speech and speech sides of each annotated SNS reference boundaries, respectively, while SAD outputs are scored. A SAD hypothesis boundary is scored correct if it is located inside the region, [RSB-0.5s, RSB+0.2s], where RSB means the reference speech boundary. Thus, extending the speech regions in the training data by a time period shorter than 0.5s would not hurt the SAD performance. Also, as a consequence of the forgiveness collars, NS regions shorter than 0.7s are treated as speech during the scoring. Considering these two facts, we adjusted the training reference annotations by extending speech regions at both sides with a time length $E (<0.5s)$ and converting NS regions that were shorter than a threshold $T (<0.7s)$ to speech. These adjustments tended to make the model trained with more speech frames at the boundaries and then help reducing the FA errors. We re-trained the “CI-NN1+2” NN system with the training references adjusted by setting $E=0.5s$ and $T=0.5s$. As shown in Table 2, this new system, named as “+ATR”, improved the performance slightly.

The concatenation of 31 PLP frames covers a time span of 0.31s with the frame step at 0.01s. To cover longer context spans we trained NNs to post-process the SAD NN output posteriors. The inputs to the post-processing NNs were the concatenation of 69 frames of the SAD NN outputs. Due to the longer context span, we increased $T$ to 0.6s when converting short NS regions to speech in the training references. Besides, we trained 5 NNs with 5 different $E$ values, -0.02s (negative mean the opposite, shrinking), 0.02s, 0.08, 0.14s, and 0.2s. The average of 5 NN outputs was computed as the final SNS posteriors. In this way, the posteriors near the boundaries were better smoothed. We carried out this post-processing on the outputs of the “+ATR” system and trained the NNs with only one hidden layer that had 100 nodes. The SAD performance after this post-processing, denoted as “+ATR+PP”, is shown in Table 2. As can be seen, the post-processing produced further gains (compared to the “+ATR” system).

4.2. Use of the CBN features

We generated 50 dimensional CBN features with a 5-layer channel classification NN (“605x800x50x500x9”), trained with the “Train2Khrs” corpus. We re-trained the “+ATR+PP” system with these CBN features (denoted as “CBN50”) integrated. The performance of this system, named as “+ATR+CBN50+PP”, is shown in Table 2. Comparison of the two systems (“+ATR+PP” vs. “+ATR+CBN50+PP”) shows there were minor gains from these CBN features (6% relative on “PFR@1%”, no gains on EER), Recall that (shown in Table 1) about 15% relative improvements were observed when we investigated the CBN features on the small training corpus. The smaller gains were obtained when the NNs were trained with the large training corpus. The reason was likely that the NN learnt the channel characteristics better when the channels had sufficient data.

5. Improving SAD on new channels

The final goal of the RATS program was to run the systems developed for the 4 tasks on field data, which came from channels different from the 8 noisy channels. Therefore, robustness to new channels is one important factor to be considered during the development of systems. We worked on increasing robustness of the SAD system to new channels. The
design of the CBN features mainly targeted at improving the SAD performance on new channels. The situation we were assumed to face was that a large amount of data from a new channel is available, but was not labeled. The new data hence cannot be used in the SAD model training. However, it can be added, as an extra channel, to the channel classification NN training for learning characteristics of the new channel. The CBN features generated would capture some of the new channel characteristics and thus they would help the SAD NN learn information of the new channel when integrated in the training.

To simulate this situation we treated channel B as a new channel. Excluding the channel B data from the NN training, we re-trained the “+ATR” NN system. This new NN system denoted as “+ATR-noAB”. To measure the SAD performance on the new channel we computed the “EER” and “PFR@1%” scores only on the channel B data in the “dev2” test set. The performance of the two systems is shown in Table 3. As can be seen, there were big degradations (EER: from 1.54 to 7.67; PFR@1%: from 2.36 to 30.29) when the channel B was not observed in the training.

<table>
<thead>
<tr>
<th>Systems</th>
<th>On B in dev2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EER</td>
</tr>
<tr>
<td>+ATR</td>
<td>1.54</td>
</tr>
<tr>
<td>+ATR+CBN50</td>
<td>1.59</td>
</tr>
<tr>
<td>+ATR-noB</td>
<td>7.67</td>
</tr>
<tr>
<td>+ATR+CBN50-noB</td>
<td>5.64</td>
</tr>
</tbody>
</table>

As described, the “CBN50” features, generated in Section 4.2, could still be used in the SAD NN training. Thus, we re-trained both the “+ATR” and “+ATR-noAB” systems with the “CBN50” bottleneck features integrated. These two new systems are denoted as “+ATR+CBN50” and “+ATR+CBN50-noAB”, respectively, and their performance measured on channel B is also shown in Table 3. As shown, when channel B was observed in training, the “CBN50” features did not help on channel B (“+ATR” vs. “+ATR+CBN50”), although they improved the overall performance on “dev2” (shown in Table 2). However, when channel B was not observed in the training, the CBN features helped improving the performance significantly (“+ATR-noB” vs. “+ATR+CBN50-noB”).

The degradation on the new channel was large even with the CNB features used. Our first investigation on this was to exclude the cortical features from the training. We re-trained “+ATR” and “+ATR-noB” systems with only PLP features. The performance of the two new systems, name as “PLP_NN” and “PLP_NN-noB”, is shown in Table 4. Comparing these 4 systems, we see that the use of the cortical feature improved the performance significantly when the channels were observed in training, but caused big degradations on unobserved channels. Hence, the cortical features seemed to be more channel-specific. We then re-trained the 5-layer channel classification NN with the PLP only and with the structure configured as “465x600x30x300x9”. With this NN we generated 30-dimensional CBN features and integrated them to the training of the “PLP_NN” and “PLP_NN-noB” systems. The two new systems are denoted as “PLP_NN+CBN30p” and “PLP_NN+CBN30p-noB”, respectively. As shown in Table 4, these CBN features helped improving the performance greatly on both situations where channel B was observed (“PLP_NN” vs. “PLP_NN+CBN30p”) and unobserved (“PLP_NN-noB” vs. “PLP_NN+CBN30p-noB”) in the training.

<table>
<thead>
<tr>
<th>Systems</th>
<th>On B in dev2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EER</td>
</tr>
<tr>
<td>PLP_NN</td>
<td>2.52</td>
</tr>
<tr>
<td>PLP_NN-noB</td>
<td>4.08</td>
</tr>
<tr>
<td>PLP_NN+CBN30p</td>
<td>2.04</td>
</tr>
<tr>
<td>PLP_NN+CBN30p-noB</td>
<td>3.58</td>
</tr>
<tr>
<td>PLP_GMM</td>
<td>3.20</td>
</tr>
<tr>
<td>PLP_GMM-noB+MLLR</td>
<td>4.14</td>
</tr>
</tbody>
</table>

To compare with the commonly used MLLR adaptation technique, we trained two channel independent MMI-GMM (2,048-components) systems with PLP features only, one with and the other without channel B data included. The MLLR was used to adapt the GMM (only means) when the channel B was not excluded from the training. The performance of these two systems, “PLP_GMM” and “PLP_GMM-noB+MLLR”, is also listed in Table 4. First, we see that the NN system, as expected, outperformed the GMM system (“PLP_GMM” vs. “PLP_GMM-noB+MLLR”). Second, on the new channel the NN trained with the CBN features produced superior performance, compared to the MLLR-adapted GMM system (“PLP_NN+CBN30p-noB” vs. “PLP_GMM-noB+MLLR”).

6. Conclusions

We have described the work conducted mainly with the NN approach for building our SAD systems for the RATS phase 3 evaluations. First, our results showed the CI NN system was able to produce 20% relative improvements over the channel dependent GMM system – our phase 2 evaluation system. Then, the efforts to reduce the FR errors by the extensions of speech regions in the training references and the use of the post-processing NNs further improved the performance.

The design of the CBN features and the integration of them into the SAD NN training targeted at removing channel variations from the NN training. Our results revealed that the CBN features helped improve the SAD performance. The bigger benefits of the CBN features were the enhancement of the SAD performance on new channels. The results showed that the SAD degradation was greatly reduced with the use of CBN features.

The cortical features improved the SAD performance greatly when all the channels were observed in the NN training. However, big degradations occurred on new channels. This suggested the cortical features are more channel-specific.

---
1 We observed channel A was strongly corrected with B, so we excluded the channel A data as well from the training to avoid potential impacts from channel A on B.
7. References


