A PHONETICALLY AWARE SYSTEM FOR SPEECH ACTIVITY DETECTION

Luciana Ferrer[2], Martin Graciarena[1], Vikramjit Mitra[1]

[1] Speech Technology and Research Laboratory, SRI International, California, USA
[2] Departamento de Computación, FCEN, Universidad de Buenos Aires and CONICET, Argentina

ABSTRACT

Speech activity detection (SAD) is an essential component of most speech processing tasks and greatly influences the performance of the systems. Noise and channel distortions remain a challenge for SAD systems. In this paper, we focus on a dataset of highly degraded signals, developed under the DARPA Robust Automatic Transcription of Speech (RATS) program. On this challenging data, the best-performing systems are those based on deep neural networks (DNN) trained to predict speech/non-speech posteriors for each frame. We propose a novel two-stage approach to SAD that attempts to model phonetic information in the signal more explicitly than in current systems. In the first stage, a bottleneck DNN is trained to predict posteriors for senones. The activations at the bottleneck layer are then used as input to a second DNN, trained to predict the speech/non-speech posteriors. We test performance on two datasets, with matched and mismatched channels compared to those in the training data. On the matched channels, the proposed approach leads to gains of approximately 35% relative to our best single-stage DNN SAD system. On mismatched channels, the proposed system obtains comparable performance to our baseline, indicating more work needs to be done to improve robustness to mismatched data.

Index Terms— Speech activity detection, deep neural networks, bottleneck features, degraded channels

1. INTRODUCTION

Speech-activity detection (SAD) is an essential component in many speech processing tasks, such as speech recognition, speaker verification, and language identification. SAD can also be used in a scenario where vast amounts of audio data are searched for the rare presence of speech. Errorfull SAD can greatly degrade the performance of these systems [1].

Though most SAD systems work well on relatively clean signals, their performance greatly degrades with the presence of noise. In recent years, the DARPA RATS program has provided a great opportunity for researchers to work on very challenging noisy and distorted data, with a focus on speech activity detection, keyword spotting, language identification, and speaker verification. Under this program, performance of SAD systems in highly degraded acoustic conditions has been greatly improved by using different techniques, such as long-span and robust features, feature-level fusion and better modeling methods [2, 3, 4, 5, 6]. The best of these systems rely on deep neural networks (DNNs) to predict speech and non-speech posterior probabilities using many different input features, including a wide range of features specially designed for noise robustness [5, 6].

In this work, we propose to extend the work based on DNNs to take into account phonetic information. Humans are able to detect speech under extreme conditions, even when they do not speak the language or when speech is unintelligible. They expect speech to be formed by units that sound like phones concatenated into familiar sequences. A region containing a sound which resembles a phone but lasts several seconds without change would probably not be labelled as speech by a human. While current systems can, in theory, if given enough contextual information and enough training data, model all this information within a single DNN trained to predict speech and non-speech posteriors, we hypothesize that adding structure to the problem by creating features that are phonetically rich might facilitate the modeling of what speech should sound like. To this end, we propose a novel two-stage system. The first stage consists of a bottleneck DNN trained to predict senone posteriors. The activations at the bottleneck layer, which should mostly contain information about the phonetic content [7], are then used as input into another model to classify each frame as speech or non-speech.

The use of bottleneck activations from a DNN trained to predict senone posteriors has been previously proposed for the speaker verification [8, 9] and language recognition tasks [10, 11]. For language recognition, the systems based on these features have become the state of the art, giving significantly better performance than previous approaches [11, 12]. For speaker verification, bottleneck features give a significant gain when fused with standard acoustic features [8]. We hypothesize that these features should also be useful for the SAD task, particularly because they are trained as a low-dimensional representation of the phonetic content in each frame, which should make them useful for discriminating speech versus non-speech. In this paper we explore the use of bottleneck features for SAD and show promising results, specially when the conditions in the training data for the senone DNN are matched to those in the test data.

2. GMM- AND DNN-BASED SAD

GMM-based SAD systems are composed of two GMMs, one trained on speech frames, and another one trained on non-speech frames [4]. Given a test sample, the likelihood of the feature vector extracted from each frame is computed with respect to each of the two models. The logarithm of the ratio of the speech and non-speech likelihoods is then computed for each frame. In a final step, these LLRs (log-likelihood ratios) are smoothed by averaging their values over a rolling window typically 31 to 71 frames long. The final SAD decisions are made by thresholding these LLRs, with a threshold chosen based on the desired operating point. For some applications, the resulting speech regions (any contiguous frames for which the LLR value was above the threshold) are padded with a certain number of

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-15-C-0037. The views, opinions, and/or findings contained in this article are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. Distribution Statement A: Approved for Public Release, Distribution Unlimited.

frames (30 in our experiments) on each side. This padding reduces the amount of missed speech near the detected speech regions while potentially increasing the false alarm rate.

DNN-based SAD systems compute the LLRs by using a DNN trained to predict the posterior of the speech and non-speech classes at the output layer. The posteriors are converted into LLRs by using Bayes rule, assuming equal priors for both classes. These LLRs are processed the same way as for the GMM-based systems to obtain the SAD decisions.

For both of these families of systems, the input features are generally mel frequency cepstral coefficients (MFCCs), perceptual linear prediction (PLPs), or log mel filter bank energies (LMFBs), concatenated over several frames (generally, 31) to include contextual information. Dimensionality reduction techniques can be used on the concatenated vector [2, 3]. For GMM modeling, contextual information is sometimes represented through deltas and double deltas or a discrete cosine transform [4]. Other options, like cortical features, which are already long-term, have also been used [5].

3. PHONETICALLY AWARE SAD SYSTEM

We propose a two-stage approach to SAD in which the first stage operates as a feature extractor for the second stage, which is a standard DNN-based SAD system as described above. Figure 1 shows a scheme of the system. The features generated by the first stage are given by the activations in a bottleneck layer of a DNN trained to predict senone posteriors. Senones are defined as tied-states within context-dependent phones and are the unit for which observation probabilities are computed during automatic speech recognition (ASR). Given that the posteriors at the output layer of the DNN have to be computed as a function of only the activations in the bottleneck layer, we can see these bottleneck (BN) features as a low-dimensional representation of the phonetic content in each frame. This should make these features useful for discriminating speech versus non-speech. The input features to the senone DNN are given by standard frame-level acoustic features concatenated over a relatively short context.

The second stage in the proposed approach models these features using the same procedure as for standard acoustic features, training a DNN to predict SAD labels. As for the baseline system, contextual information is provided to the SAD DNN by concatenating features from several adjacent frames. Nevertheless, because the dimension of the BN features is relatively large, and to avoid creating very large input vectors for the DNN, we first smooth the features across time over a window of size S frames and then take one every S frames for concatenation.

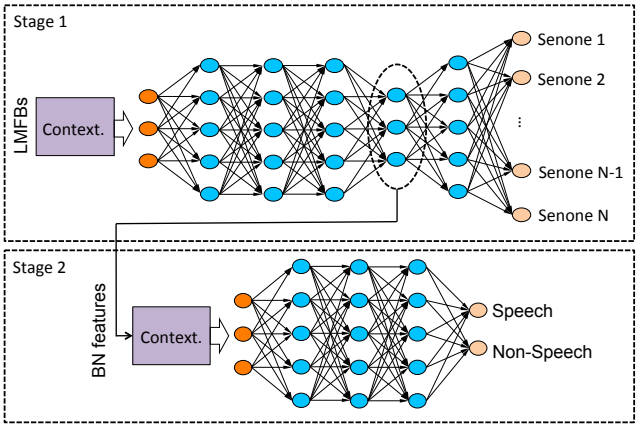

Fig. 1. Flow diagram of the proposed phonetically-aware SAD system. The number of nodes in each layer are kept small for this schematic figure but the number of layers in both DNNs are the ones used for the experiments. A bottleneck DNN is trained to predict senone targets using contextualized LMFB features. The activations in the bottleneck layer are then used as input to another DNN, after another stage of contextualization. This second DNN is trained to predict speech versus non-speech.

4. EXPERIMENTS

In this section we describe the setup for our experiments and present the results.

4.1. Train and test datasets

Both the training and test data used in the experiments came from to the Linguistic Data Consortium (LDC) collections for the DARPA RATS program [13]. Conversational telephone recordings were retransmitted using a multilink transmission system at LDC. Several combinations of transmitters and receivers were used to retransmit, resulting in extremely noisy and distorted signals.

For training the senone DNNs, we used the RATS keyword-spotting (KWS) training sets for Farsi and Levantine Arabic. This data contains eight transmission channels plus the source signals and includes word-level transcriptions. The phonesets for the two languages were merged, mapping the symbols that correspond to a similar phone in the two languages to the same symbol. The Farsi phoneset contains 29 phones, the Levantine phoneset contains 38 phones, and the merged phoneset contains 46 phones. The transcriptions, mapped to use this new phoneset, were used to train an HMM-GMM ASR system with 3353 senones. The senones were obtained automatically using a decision tree approach. The resulting HMM-GMM system was then used to force-align the same training data to obtain senone-level alignments, which were used to train the DNNs. More details on the HMM-GMM model used to create the alignments can be found in Section 5.1 of [14]. The same alignments produced for that paper were used here.

Note that, for senone DNN training, only segments containing mostly speech and short pauses are used, including some padding on each side of a segment to ensure that they start and end during non-speech. The KWS training data contains a total of 151 hours, approximately 15% of it being non-speech.

For training the SAD DNNs, we used the RATS SAD training data. Channel D was not included in the SAD data due to annotation problems. The clean source was included among the channels. This data contains 830 hours of speech and 677 hours of non-speech. Given the large size of this data set, during GMM and DNN training we select 1 every 5 frames.

For testing, we used data from the SAD Dev-1 collection released between 2011 and 2012, which was retransmitted using the same channels as the training data. We call this the "seen channel" data. We also tested on a separate dataset extracted from the novel channel collection, released by LDC in 2014. This data was created using different transmitter/receiver pairs, new transmitter/receiver locations and longer distances than the original set. We used 8 of the released channels (A, D, G, H, K, M, Q, and R), discarding the ones with a clear problem in the annotations and leaving out some chan-

nels for future evaluation on a held-out set (these channels are not used for the results in this paper). The low speech-density data and the composite data released in this collection were not used for these experiments due to the bad quality of the annotations. We call this the "unseen channel" data. Both datasets were divided into adaptation and test sets. We do not present adaptation results in this paper. Results are shown on the test set to keep results consistent with future work. The seen channel data contains 4.4 hours of audio, while the unseen channel data contains 5.6 hours. In both cases, approximately half of the time is speech.

Note that, although the channels are the same across the KWS and the SAD training datasets and the seen channel test data, the languages in the SAD data are a superset of those in the KWS data, containing speech in English, Pashto and Urdu, as well as Levantine Arabic and Farsi.

4.2. Performance metric

Two types of error can be computed for SAD: (1) the miss rate (the proportion of speech frames labeled as non-speech) and (2) the false alarm rate (the proportion of non-speech frames labeled as speech). In Phase 4 of the RATS program, a "forgiveness" collar was used around all annotated speech regions. False alarm errors over those regions were disregarded. The official value for the collar was 2.0 seconds. Yet, because many non-speech regions in the RATS data are quite short, this long collar reduced the amount of scored non-speech by a large proportion, reducing the statistical significance of the results. For this reason, for the results in this paper, we used a smaller collar of 0.5 seconds.

The RATS metrics for SAD were the equal error rate (EER), the miss rate when the miss and false alarm rates are equal, and the miss rate when the false alarm rate is equal to some fixed value (e.g., 1%). To obtain these values, the LLR threshold chosen to make the final decisions is swiped across a range of values. These two metrics ignore the problem of threshold selection. In practice, the threshold must be selected during development. The actual error will then depend both on the goodness of the LLRs and the choice of threshold. Hence, for this work we present results in terms of a metric commonly used in speaker recognition: the detection cost function (DCF), sometimes also called Cdet [15]. The DCF is computed as a weighted sum of the miss and false alarm rates. In this work, we use equal weights. The minimum value of this DCF (when the threshold is selected to optimize DCF on the test data) is, by definition, no more than two times the EER. The actual DCF is the DCF value when the threshold is chosen a priori, without knowledge of the test set. In our case, we set the threshold for actual DCF to 0. This is the optimal value for this equal-weight DCF if we assume LLRs are well calibrated. The DCF has also been chosen as the primary metric for the NIST OpenSAD evaluation [16], though with uneven weights for false alarms and misses.

To obtain the DCFs we post processed the LLRs from each of the systems as described in Section 2 using an average filter of 41 frames, thresholded them with the threshold for minimum or actual DCF, padded each resulting speech region with 0.3 seconds on each side, and finally summed false alarm and miss rates to calculate the DCF. We do the calculation by channel. The reported results are averages across channels in each of the two data sets.

4.3. Systems

We compare three different systems, two baseline systems and our proposed system.

MFCC-GMM (baseline 1): GMM-based SAD system as described in Section 2. Both speech and non-speech GMMs have 512 Gaussians with diagonal covariance. The input features are given by 20-dimensional MFCCs, normalized over each waveform by subtracting the mean and dividing by the standard deviation of each coefficient, with the exception that, for C0, the maximum is subtracted rather than the mean. This normalization for C0 gives a small but consistent gain over using the mean. Contextualization for this system is done by taking the first four DCT coefficients for each original feature across a window of 31 frames and appending those coefficients to the feature vector corresponding to the center frame. This works better for modeling with GMMs than the simple concatenation used for DNNs. The final feature vector for this system is of size 100. For this system we do not try different context lengths since, as we will see, its performance for the 31-frame context is significantly worse than that of the DNN baseline below.

MFCC-DNN (baseline 2): DNN-based SAD system as described in Section 2. Three hidden layers of size 500 are used for these DNNs. This structure was optimal in our experiments for both test sets, though performance was quite robust to variations in the structure of the DNN, given the large amount of training data available. The same normalized MFCC features described above are used for this system, except that contextualization is done by concatenation. This was found to be better than using the DCTs when DNNs are used for modeling. The final dimension of the feature vector input to the DNN is given by 20xW, where W is the size of the window (an odd integer). For the longer contexts, we smooth the features with a window of S frames, using one out of S frames for concatenation, as also done for the BN-DNN system below. The final feature dimension is given by 20x(1+(W-1)/S). We explore different values for W and S.

BN-DNN (proposed): As for the MFCC-DNN system, this system consists of a DNN with three hidden layers of size 500 trained to predict SAD labels. The input features are the activations from the bottleneck layer of another DNN trained to predict 3353 senones (Section 4.1). The structure of this DNN is similar to that commonly used in ASR [17] and contains 5 hidden layers, with the bottleneck layer at position 4. The size of the bottleneck is set to 50, which was better for this task than the more standard value of 80. The input features to this DNN are given by 40-dimensional log mel filterbank (LMFB) energies (the same ones used to compute the 20 cepstral coefficients for the MFCC-GMM and MFCC-DNN systems above), normalized by subtracting the median from each dimension within a window of size 201 centered at each frame. This sliding-window normalization worked better than a normalization performed over the whole waveform. The features are concatenated over a window of 15 frames for input to the senone DNN, as usually done for ASR [18]. Finally, the BN features obtained from the first DNN are concatenated over a window of size W. As for the MFCC-DNN system, we smooth the features over S frames and concatenate one out of S frames to create a final feature vector of dimension 50x(1+(W-1)/S), that is then used as input to the SAD DNN. As for the MFCC-DNN baseline, we explore different values for W and S.

For all systems, we band-pass filtered the signals with lowest energy given by 200 Hz and highest energy given by 3300 Hz before feature extraction. For the DNN-based baseline system, LMFB features gave worse performance than the MFCCs, while for the proposed system, LMFBs worked better than MFCCs as input to the first stage. This is the reason for using different input features for each system.

4.4. Results

Table 1 shows the results for all three systems, with different contextualization windows and the skipping step set to keep the dimension of the input to the model similar across windows. We can see that the DNN baseline system significantly outperforms the GMM baseline system on both datasets for a context window of 31 frames. We also find that, for the MFCC-DNN baseline, long contexts lead to significant gains over the usual 31 frame window used in most of the recent SAD papers [4, 6]. The optimal context length seems to be around 71 frames, with the longer context of 111 frames giving a small gain on the seen channels and a small loss on unseen channels.

Finally, we can see that the proposed system leads to large gains on the seen channel data of 35% on minimum DCF and 40% on actual DCF. On the other hand, the performance on unseen channels is not significantly different from that of the baseline DNN system. This indicates that more work is required to ensure a better generalization of the BN features on unseen conditions. The optimal context length for this system is between 31 and 71 frames, with no consistent gain obtained from a longer context. Note, though, that the effective context for this system is, in fact, 15 frames longer than the window size W, since the BN features already consider a context of 15 frames in their computation.

Table 1. Min DCF (mDCF) and actual DCF (aDCF) for seen and unseen channels for different baseline and proposed systems. The second column ("W/S") indicates the size of the contextualization window W and, for the DNN approaches, the number of frames S used for smoothing and skipping before concatenation. The third column shows the final dimension of the concatenated feature vector input to the SAD DNN.

System	W/S	Dim	Seen		Unseen	
			mDCF	aDCF	mDCF	aDCF
MFCC-GMM	31/-	100	4.38	6.74	12.92	18.91
MFCC-DNN	31/-	620	2.48	4.10	11.99	16.70
	71/2	720	2.41	3.30	11.28	14.34
	109/3	740	2.30	3.18	12.23	14.77
BN-DNN	31/3	550	1.80	2.66	11.21	14.92
	71/7	550	1.48	1.88	11.83	14.72
	111/11	550	1.55	1.93	11.72	14.22

5. CONCLUSIONS

We proposed a new approach to SAD composed of two stages. In the first stage, a bottleneck DNN is trained to predict the senone posteriors. The activations of the bottleneck layer are then used to create features for a second DNN, trained to predict speech/non-speech posteriors. We show that this approach leads to 35-40% relative gains on channels seen during training with respect to a single-stage DNN baseline with MFCC input features. For unseen channels, the system performs comparable to the baseline, indicating that the system might be safe to use on conditions different from those present in the training data.

In the near future, we plan to explore different structures for the bottleneck DNN, including convolutional neural networks. We will also work on the issue of robustness to unseen channels, exploring different normalization approaches for both the features input to the DNN and the bottleneck features themselves. Finally, we will explore using fewer classes (for example, monophones) at the output of the phonetic DNN. This might result in more robust models, less sensitive to variations in the channel characteristics.

6. REFERENCES

[1] J. Ramirez, J. M. Górriz, and J. C. Segura, *Voice activity detection. fundamentals and speech recognition system robustness*, INTECH Open Access Publisher, 2007.

[2] T. Ng, B. Zhang, L. Nguyen, S. Matsoukas, X. Zhou, N. Mesgarani, K. Veselỳ, and P. Matejka, "Developing a speech activity detection system for the DARPA RATS program," in *Proc. Interspeech*, Portland, USA, Sept. 2012.

[3] G. Saon, S. Thomas, H. Soltau, S. Ganapathy, and B. Kingsbury, "The IBM speech activity detection system for the DARPA RATS program," in *Proc. Interspeech*, Lyon, France, Aug. 2013.

[4] Martin Graciarena, Abeer Alwan, Dan Ellis, Horacio Franco, Luciana Ferrer, John HL Hansen, Adam Janin, Byung Suk Lee, Yun Lei, Vikramjit Mitra, et al., "All for one: Feature combination for highly channel-degraded speech activity detection," in *Proc. Interspeech*, Lyon, France, Aug. 2013.

[5] Jeff Ma, "Improving the speech activity detection for the DARPA RATS phase-3 evaluation," in *Proc. Interspeech*, Singapore, Sept. 2014.

[6] S. Thomas, G. Saon, M. Van Segbroeck, and S. S. Narayanan, "Improvements to the IBM speech activity detection system for the DARPA RATS program," in *Proc. ICASSP*, Brisbane, Australia, May 2015.

[7] M. McLaren, L Ferrer, and A. Lawson, "Exploring the role of phonetic bottleneck features for speaker and language recognition," in *submitted to ICASSP 2016*, 2016.

[8] M. McLaren, Y. Lei, and L. Ferrer, "Advances in deep neural network approaches to speaker recognition," in *Proc. ICASSP*, Brisbane, Australia, May 2015.

[9] F. Richardson, D. A. Reynolds, and N. Dehak, "A unified deep neural network for speaker and language recognition," in *Proc. Interspeech*, Dresden, Sept. 2015.

[10] Yan Song, Bing Jiang, YeBo Bao, Si Wei, and Li-Rong Dai, "i-vector representation based on bottleneck features for language identification," *Electronics Letters*, vol. 49, no. 24, pp. 1569–1570, 2013.

[11] P. Matejka, L. Zhang, T. Ng, S. H. Mallidi, O. Glembek, J. Ma, and B. Zhang, "Neural network bottleneck features for language identification," in *Proc. Odyssey-14*, Joensuu, Finland, June 2014.

[12] L. Ferrer, Y. Lei, and M. McLaren, "Study of senone-based deep neural network approaches for spoken language recognition," *submitted to IEEE Trans. Audio Speech and Language Processing*, 2015.

[13] K. Walker and S. Strassel, "The RATS radio traffic collection system," in *Odyssey 2012: The Speaker and Language Recognition Workshop*, 2012.

[14] Y. Lei, L. Ferrer, A. Lawson, M. McLaren, and N. Scheffer, "Application of convolutional neural networks to language identification in noisy conditions," in *Proc. Odyssey-14*, Joensuu, Finland, June 2014.

[15] "NIST SRE12 evaluation plan," http://www.nist.gov/itl/iad/mig/upload/NIST_SRE12_evalplan-v17-r1.pdf.

[16] "NIST OpenSAD evaluation plan," http://www.nist.gov/itl/iad/mig/upload/Open_SAD_Eval_Plan_v8.pdf.

[17] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Proc. ICASSP*, Brisbane, Australia, May 2015.

[18] O. Abdel-Hamid, A. R Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 10, pp. 1533–1545, 2014.