EXPLORING THE ROLE OF PHONETIC BOTTLENECK FEATURES FOR SPEAKER AND LANGUAGE RECOGNITION

Mitchell McLaren[1]*, Luciana Ferrer*[2]*, Aaron Lawson*[1]

[1]Speech Technology and Research Laboratory, SRI International, California, USA
[2] Departamento de Computación, FCEN, Universidad de Buenos Aires and CONICET, Argentina
{mitch,aaron}@speech.sri.com, lferrer@dc.uba.ar

ABSTRACT

Using bottleneck features extracted from a deep neural network (DNN) trained to predict senone posteriors has resulted in new, state-of-the-art technology for language and speaker identification. For language identification, the features' dense phonetic information is believed to enable improved performance by better representing language-dependent phone distributions. For speaker recognition, the role of these features is less clear, given that a bottleneck layer near the DNN output layer is thought to contain limited speaker information. In this article, we analyze the role of bottleneck features in these identification tasks by varying the DNN layer from which they are extracted, under the hypothesis that speaker information is traded for dense phonetic information as the layer moves toward the DNN output layer. Experiments support this hypothesis under certain conditions, and highlight the benefit of using a bottleneck layer close to the DNN output layer when DNN training data is matched to the evaluation conditions, and a layer more central to the DNN otherwise.

Index Terms— Bottleneck Features, Deep Neural Networks, Speaker Recognition, Language Recognition

1. INTRODUCTION

Recently, deep neural networks (DNNs) have been applied to many speech applications. In this work, we focus on the tasks of speaker identification (SID) and language identification (LID) using features extracted from a DNN. Specifically, we analyze bottleneck (BN) features extracted from a 5-layer DNN trained to discriminate tied triphone states (also referred to as senones), as typically used in automatic speech recognition (ASR) systems [1].

Research has shown that bottleneck features are perhaps the most effective feature for LID [2, 3]. And when combined with Mel frequency cepstral coefficients (MFCCs), they result in one of the most powerful combined features for SID [4, 3]. Though these features are extremely useful for both the SID and LID task, the role of the bottleneck output representation has not been deeply explored. For example, LID is thought to benefit from the dense phonetic content represented in the bottleneck features extracted from a layer close to the DNN output layer where phone-discriminative information should be most salient. In contrast, such BN features are assumed to be somewhat speaker-independent (relative to the features that are input to the DNN), and yet they still greatly assist SID [4, 3]. This finding is particularly true when the BN features are combined with traditional SID features such as MFCCs. One hypothesis is the bottleneck features provide information that enables the universal background model (UBM) to better align frames to phonetic content rather than to the clusters formed purely based on acoustic sounds. In doing so, speaker-dependent pronunciations can be leveraged to improve SID comparisons. In both SID and LID, the actual impact of the bottleneck feature and its relative balance between phonetic and speaker information (extracted closer to, or farther from, the output layer, respectively) has yet to be quantified with respect to detection performance.

In this analysis study, we aim to demonstrate how the phonetic information contained in bottleneck features impacts the performance of both SID and LID. To this end, we vary the position at which the bottleneck layer is placed, to understand the impact of trading contextualized filter energy information toward the start of the network for phonetic information at the end of the network. We evaluate the sole use of bottleneck features in both the SID and LID task. For SID, we also evaluate feature-level fusion of BN features and MFCC features, with the MFCCs appended with deltas and double deltas. Additionally, we subset results for meaningful analysis over a variety of datasets to observe language-dependence and performance under mismatched train vs. test conditions and duration.

2. BOTTLENECK FEATURES FOR DETECTION TASKS

Several methods of using DNNs for SID and LID have been published in literature. These include DNN-based i-vectors [1, 2], DNN-posterior features [5, 2], and, more recently, bottleneck features [6, 7, 8]. The former methods use the output layer of the DNN to aid in detection, while the bottleneck features are extracted from a layer prior to the output layer. Given that previous comparisons of these approaches have led to the conclusion that the BN-based approach gives the best results [4] and [2], this study will be constrained to bottleneck features extracted from a DNN trained to discriminate tied tri-phone states (senones).

This section provides a brief overview of bottleneck features and how they are applied in both SID and LID tasks.

2.1. Bottleneck Feature Extraction

Bottleneck features [6, 7, 8] are a set of activations of nodes over time from a bottleneck layer in a trained DNN. The bottleneck layer

This material is based on work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract HR0011-15-C-0037 and a development contract with Sandia National Laboratories (SNL) (Subcontract # 1046087/ DO 1568990). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the view of SNL, nor DARPA or its contracting agent, the U.S. Department of the Interior, National Business Center, Acquisition & Property Management Division, Southwest Branch. "A" (Approved for Public Release, Distribution Unlimited).

is a hidden layer in the DNN of reduced dimension relative to the other layers (i.e., 80 nodes compared to 1200 in this work). This DNN can be trained to discriminate between different output classes such as senones, speakers, conditions, etc. Using a bottleneck layer in the DNN ensures that all information required to ultimately determine the posteriors at the DNN's output layer is restrained to a small number of nodes. When the DNN is trained to predict senone posteriors at its output layer, the features extracted as activations of the nodes in the bottleneck layer lend themselves well to tasks in which phonetic content is beneficial, such as LID. Consider now that the position of the information bottleneck (or bottleneck layer) can be close to the input layer, where the phonetic information in the resulting features is assumed to be relatively low compared to that of a bottleneck layer placed close to the output layer (assuming the DNN classes are senones or other phonetic units). In this work, we focus on how the position of the bottleneck layer affects LID and SID performance in order to help understand the role they play in these detection tasks.

2.2. Language Identification Using Bottleneck Features

Determining the spoken language from audio can typically be broken down into two methodologies: phonotactic and acoustic. Phonotactic approaches attempt to model the permissible combinations of phones and their frequencies in the languages of interest. Many phonotactic approaches involve collecting the probabilities for phone sequences as a representation of the signal by using the output of one or several open-phone loop recognizers [9, 10, 11]. Language models or support vector machines are then used to generate the final scores. Another phonotactic approach uses the phoneme posteriogram counts from the phone recognizer to create bigram conditional probabilities, which are then used to create features for LID (e.g., [12]). These phonotactic approaches work with a relatively small set of units (approximately 50) representing the individual phones of the language being modeled. Information about the frequency of different phone sequences is collected through n-gram generation. Acoustic approaches directly model the acoustic features such as MFCCs rather than trying to explicitly determine sub-units of what was spoken [13].

Bottleneck features provide a middle-ground between the two main approaches to language identification, since the information required to obtain senone posteriors is contained in the bottleneck activations. However, the meaning of each node is obscure. A major benefit of bottleneck features in LID over most phonotactic approaches is the ability to use them in long-standing, robust modeling techniques, such as the i-vector framework [13], followed by simple classifiers, such as the Gaussian backend or a neural network. In recent literature, bottleneck features extracted from DNNs were successfully applied to language identification in [6], with later works supporting the strength of these feature in severely degraded audio conditions [7, 8, 2].

2.3. Speaker Identification Using Bottleneck Features

Strictly speaking, bottleneck features extracted from a DNN trained for ASR should be relatively speaker-independent, as this enables for better senone prediction across all speakers. In light of this, one pressing question is why bottleneck features are suitable for speaker recognition at all. In [4], for example, they were shown to outperform MFCCs for SID under ideal conditions. In the same work, when combined with MFCCs on a feature level, they set a new state-of-the-art performance level on the National Institute for Standards and Technology (NIST) Speaker Recognition Evaluation (SRE) 2012. One hypothesis as to why bottleneck features contribute to speaker recognition in this feature-fusion approach is that they provide assistance to the UBM during unsupervised clustering, such that the components of the UBM align better with phonetic units (senones) compared to when purely based on acoustic sound. In turn, this improved alignment provides a better basis for exploiting pronunciation differences in the SID system. We attempt to support this hypothesis by showing through a series of experiments in Section 4 that moving the bottleneck layer toward the DNN output layer reduces the speaker information available to SID.

3. EXPERIMENT PROTOCOL

In this work, we trained several DNNs and used them to extract BN features for both the SID and LID experiments. In all experiments, speech activity detection (SAD) was based on 13-D MFCC features contextualized with deltas and double deltas, and modeled using 128-component Gaussian mixture models (GMMs) for both speech and non-speech. A median filter of 21 frames was used to smooth the SAD output before applying a threshold.

3.1. Extraction of Bottleneck Features

A 5-layer DNN was initially trained without a bottleneck layer by using 1200 nodes in each hidden layer to predict 3494 senone outputs. This DNN served as a initial DNN from which a selected hidden layer (1 through 5) was randomly re-initialized, and the DNN retrained to convergence, to obtain the bottleneck layer for the SID and LID experiments. The input features for the DNN consisted of 40 log Mel filter bank energies along with the energies from seven frames either side of a frame for a contextualized feature of 600 dimensions. The DNNs were trained by using the same dataset as used in [4]. Similarly, the input features were mean and variance normalized over the full waveform to improve channel robustness.

3.2. Speaker Recognition Protocol

For speaker recognition, MFCCs were used both alone and fused with BN features. These were 20-dimensional MFCCs contextualized with deltas and double deltas. An i-vector/PLDA framework was used throughout [14]. The UBM and i-vector subspace were trained by using the non-degraded portion of the PRISM dataset [15]. All i-vectors were processed with mean and length normalization and LDA prior to PLDA. Both LDA and PLDA were trained using the previously mentioned dataset along with a subset of microphone recordings corrupted with noise, reverb, and audio compression to obtain additional robustness to these artifacts per [16]. Evaluation data consisted of part 1 (short sentences) of the RSR2015 dataset [17] to analyze speaker recognition for matched or different prompts and prompt identification, and a subset of conditions from the 14-condition Evaluation Corpus used in [18] to enable analysis of text-independent SID for same- or cross-language trials under mixed channel conditions.

3.3. Language Recognition Protocol

For language recognition, the BN features were extracted from the range of aforementioned DNNs. Based on our previous work in [2], we used an i-vector Gaussian Backend (GB) approach to LID. A 2048 Gaussian UBM and 400-D i-vector subspaces were used for the evaluation of both Language Recognition Evaluation (LRE) 2009

Table 1. The effect of varying the position of the BN layer in a phonetic DNN from which features for LID were extracted for the LRE'09 and RATS LID datasets and evaluated in terms of Cdet across different test durations. Unlike the LRE'09 telephone data, the RATS LID data is severely mismatched to the DNN training conditions. Results from traditional MFCC-SDC features are also presented for comparison. Layer 1 denotes the first hidden layer of the DNN, while layer 5 denotes the last hidden layer.

System	LRE'09			RATS		
	3sec	**10sec**	**30sec**	**3sec**	**10sec**	**30sec**
MFCC-SDC	14.8	5.7	3.3	28.9	18.3	12.7
BN Layer 1	13.5	6.2	4.0	**24.1**	16.4	10.3
BN Layer 2	11.9	5.1	3.0	25.2	15.6	9.3
BN Layer 3	9.3	3.4	1.9	25.9	**15.0**	**8.9**
BN Layer 4	**8.9**	**2.8**	**1.6**	28.4	17.1	9.4
BN Layer 5	9.2	2.9	1.7	26.9	15.5	9.1

and the RATS LID task. For LRE'09 experiments the system was trained on development data consisting of around 50k target class utterances. Approximately 40k i-vectors were then used to model the 23 target languages using a weighted GB as described in [2]. Similarly, calibration was applied using cross-validation on the test set with results reported in terms of the official LRE metric, Cdet [19]. For the RATS LID task, we used the same development datasets as defined in [20] to model five target languages and an out-of-set class. The RATS corpus [21] consists of retransmitted audio from eight analog push-to-talk channels covering HF, VHF, UHF frequencies and AM, FM and single-side band transmissions. This resulted in high levels of channel distortion and wideband noise. All channels except channel "D" were used in this experiment. Results are also reported in terms of Cdet.

4. RESULTS

The following analysis aims to observe the effect of bottleneck layer position on the accuracy of SID and LID detection tasks. We commence with LID experiments on the LRE'09 and RATS LID corpora, followed by SID experiments on several corpora.

4.1. Language Recognition Experiments

We evaluated five different systems on the LRE'09 and RATS LID datasets, corresponding to the five DNNs with the BN layer respectively positioned at layers 1 through 5. Performance from these systems are presented in Table 1. Focusing first on the LRE'09 results, a trend of improved LID performance as the bottleneck layer moves from layer 1 to layer 4 is seen, with significant relative gain of 34–60%. This finding indicates that BN features extracted closer to the phonetic DNN output layer provide improved LID performance across all evaluated durations, thus demonstrating the benefit that dense phonetic information brings to LID. Interestingly, a marginal loss in performance exists between use of layer 4 and layer 5. This finding might indicate that predicting the 3494-dimensional output layer of the DNN directly from the 80 nodes in the BN layer is suboptimal. The presence of an intermediate hidden layer of size 1200 when the BN layer is at position 4 allows more flexibility in the model to properly map 80 values to 3494 posteriors.

The conditions of LRE'09 and the DNN training dataset are closely matched, which is likely a contributing factor to the significant gains observed using bottleneck features with respect to traditional MFCC-SDC features. To evaluate the strength of this hypothesis, the same experiments were run on the RATS LID dataset with i-vectors being extracted from the same clean English DNNs, and a RATS-specific UBM, i-vector extractor and GB. In this way only the input features and speech activity components were mismatched to the RATS conditions while the remainder of the system was tailored toward RATS data. In contrast to LRE'09 results, we can observe that use of a bottleneck layer closer to the DNN input layer (layer 2 or 3) is more suitable for the RATS data that is severely mismatched to the DNN training set. We can also observe that, while still significant, the relative improvement over MFCC-SDC is 12-30%; about half that observed for LRE'09. These results indicate that bottleneck features extracted from a DNN trained on mismatched conditions are still useful for the task of LID, although reducing mismatch should provide additional performance gains.

4.2. Speaker Recognition Experiments

In the previous section, we showed that increasing the phonetic information in the bottleneck features, by using a layer close to the output of the DNN, provided considerable improvements in LID performance under conditions in which DNN training data matched those of the evaluation data. In this section, we perform a similar analysis for the SID task. This is of particular interest given the results in the previous section and our hypothesis that the BN features contain less speaker information as they move toward the output layer.

We start by focusing on text-independent SID by using part of the 14-condition evaluation corpus previously used in [18] which is a culmination of several different corpora for understanding the efficacy of SID across many common conditions. From this corpus, we evaluated three distinct conditions with results illustrated in Figure 1. These conditions (from left to right) were English telephone recordings, cross-language telephone trials, and finally, cross-language and cross-channel (cell vs. studio microphone) trials. In all examples, we observe that BN+MFCC consistently outperforms the BN systems, and in the majority of cases, it also outperforms the MFCC system. The English telephone trials (left) show that a bottleneck layer central in the DNN provides the best performance. In the cross-language telephone trials of the middle plot, however, we observe two distinct trends. First, using BN features alone dramatically increases SID error as the features become more phonetically-rich toward the output of the DNN. This finding is expected in cross-language trials due to the speaker using disjoint sets of senones between the two languages, providing the system with no common ground in which the same speaker traits can be analyzed. In contrast, however, appending MFCCs to BN features (BN+MFCC) appears to normalize this issue, providing performance slightly better than that of the MFCC system, irrespective of the BN-layer position. This finding tends to suggest that BN features provide limited information for SID in the context of cross-language trials when the channel between enrollment and test is matched. The right plot then introduces cross-channel conditions to the cross-language trials. Here, we observe similar trends as the BN layer moves toward the DNN output layer. The performance of the BN-only system is rather degraded compared to that of MFCC, indicating a mismatch between the DNN training data and the evaluation set. Additionally, the best-performing layer of any system relying on BN features is layer 2, thus suggesting that a BN+MFCC system with mismatched conditions between the DNN training data and the expected evaluation data should use a BN layer closer to the DNN input, to prevent this mismatch from degrading SID performance.

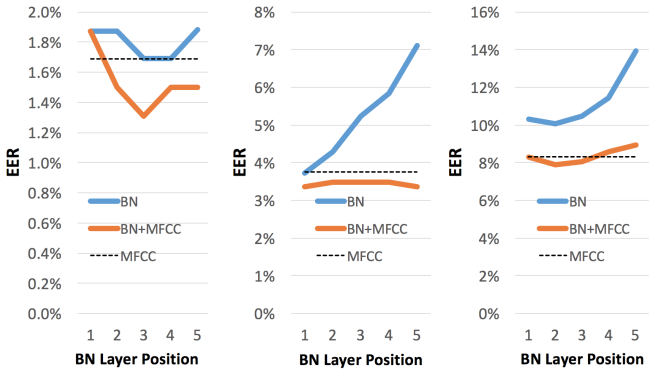

Fig. 1. The effect of varying BN layer position (1 and 5 denote the first and last hidden layer, respectively) in a phonetic DNN from which SID features were extracted for a subset of English telephone trials (left), cross-language telephone trials (center), and cross-channel, cross-language trials (right) of the 14-condition evaluation corpus previously evaluated in [18]. Baseline MFCC performance is also shown with results reported in terms of Equal Error Rate (EER).

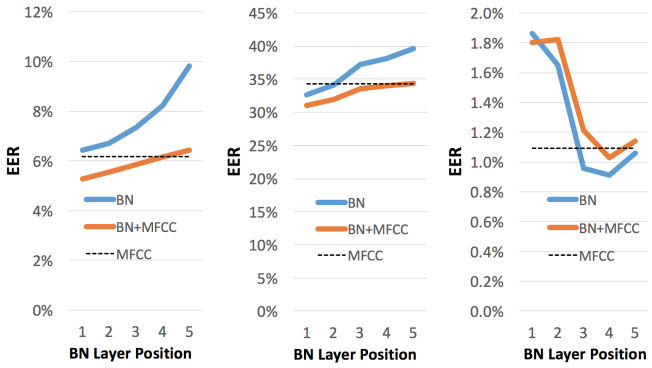

Fig. 2. The effect of varying BN layer position (1 and 5 denote the first and last hidden layer, respectively) in a phonetic DNN from which features were extracted for matched prompt speaker recognition (left), different prompt speaker recognition (center), and prompt identification (right) using the RSR2015 dataset. Baseline MFCC performance is also shown with results reported in terms of Equal Error Rate (EER).

4.3. Phonetic Dependence in Speaker Recognition

The final set of experiments focuses on speaker recognition under text-dependent conditions, where using phonetically rich features may be of particular benefit. For this purpose, we used part 1 of the RSR2015 dataset involving short sentences, with a focus on two conditions: 1) matched prompt and 2) different prompt trials. Figure 2 shows three plots. We first focus on the left and center plots, which indicate the effect of changing the BN layer position for the matched and different prompt trials, respectively. In contrast to the previous datasets, we observe that a bottleneck layer situated at the entry point of the DNN is most suitable for both conditions. Appending MFCC to the BN features is also consistently beneficial, with BN+MFCC outperforming MFCC whenever the BN features held limited phonetic information. This trend may be due either to the DNN being mismatched to the data conditions (e.g., mobile microphone recordings of non-native English) or to an exact match of phonetic content starting to hinder the SID task, and the SID classifier beginning to operate as a prompt classifier. Specifically, the phonetic content may start to dominate the i-vector rather than speaker content. To determine the strength of this hypothesis, we evaluated prompt identification using the RSR2015 dataset with the systems trained for SID.

We performed the prompt-identification experiments in similar way as for SID: enroll a speaker-dependent phrase using three utterances from three devices and then score against the remaining 180 utterances of 30 distinct sentences from the same person using different devices in the RSR2015 dataset. Note that the task of speaker recognition has been removed for this analysis by excluding trials across speakers. This resulted in a trial set of over 17k and 500k target and impostor trials, respectively. Figure 2 plots the results for the different BN layer positions where the 4th hidden layer as a bottleneck can be seen to provide the best prompt-detection performance. In contrast, however, the BN features outperform BN+MFCC. This finding is expected, because MFCC brings additional speaker dependence. Noteworthy is the very low error rates despite the system being trained for SID and not specifically tailored for prompt ID. This finding supports the fact that *speaker* discrimination using PLDA still maintains phonetic content information as reported in [22].

5. CONCLUSIONS

In this work, we analyzed how bottleneck layer position in a phonetic DNN impacts SID and LID performance when based on bottleneck features. Five different DNNs with five hidden layers were trained, each with the bottleneck layer at a different position, and bottleneck features for all tasks were extracted from each of the DNNs. Language recognition analysis showed that greater phonetic information, invoked by having the bottleneck layer close to the output layer, provided better performance across duration of the LRE'09 dataset. On the RATS LID task, however, a bottleneck layer more central to the DNN provided additional robustness when evaluating the system on conditions mismatched to the DNN training data. The same trend was observed for SID. In particular, under mismatched conditions, a bottleneck layer closer to the input features (log Mel filter bank energies) was more robust, which is consistent with the hypothesis that proper alignment of test phones with UBM components is crucial to reducing the impact of phonetic content on SID. A final experiment in prompt identification showed that a SID i-vector/PLDA system achieves very high accuracy without tailoring PLDA to the task with BN features closer to the output layer significantly improving performance over an MFCC-based system.

This study suggests several feasible avenues of research. For LID, adding additional layers prior to using the second-to-last hidden layer as a bottleneck feature may enable additional phonetic discrimination in the corresponding features by making it relatively closer to the output layer. For SID, the feature-level fusion of BN+MFCC still offers considerably better performance over each individual feature. Given the hypothesis that BN features better align frames to pronunciation instead of acoustic sounds, one might expect that using BN features to align MFCC frames may provide the same benefit; that is, train the UBM on BN features, while generating first-order statistics on MFCCs aligned using the corresponding BN features. Finally, given the strong phonetic information of i-vectors based on BN features from short speech segments, continuing research in the direction of [22] to help suppress this information may further improve the robustness of SID based on BN features.

6. REFERENCES

[1] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically aware deep neural network," in *Proc. ICASSP*, 2014.

[2] 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.

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

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

[5] L. Ferrer, Y. Lei, M. McLaren, and Scheffer N., "Language identification based on senone posteriors," in *Proc. Interspeech*, 2014.

[6] Y. Song, B. Jiang, Y. Bao, S. Wei, and L. Dai, "i-Vector representation based on bottleneck features for language identification," *Electronics Letters*, vol. 49, no. 24, pp. 1569–1570, 2013.

[7] 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. Speaker Odyssey*, 2014.

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

[9] P. Matejka, P. Schwarz, J. Cernocky, and P. Chytil, "Phonotactic language identification using high-quality phoneme recognition," in *Proc Interspeech*, 2005.

[10] W. Shen, W. Campbell, T. Gleason, D. Reynolds, and E. Singer, "Experiments with lattice-based PPRLM language identification," in *Proc. Odyssey*, 2006.

[11] A. Stolcke, M. Akbacak, L. Ferrer, S. Kajarekar, C. Richey, N. Scheffer, and E. Shriberg, "Improving language recognition with multilingual phone recognition and speaker adaptation transforms," in *Proc. Odyssey*, 2010.

[12] Luis Fernando D'haro Enríquez, Ondřej Glembek, Oldřich Plchot, Pavel Matějka, Mehdi Soufifar, Ricardo de Córdoba Heralde, and Jan Černockỳ, "Phonotactic language recognition using i-vectors and phoneme posteriogram counts," in *Proc. Interspeech*, 2012.

[13] M. Penagarikano, A. Varona, M. Diez, L. J. Rodriguez-Fuentes, and G. Bordel, "Study of different backends in a state-of-the-art language recognition system," in *Proc. Interspeech*, 2012.

[14] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. on Speech and Audio Processing*, vol. 19, pp. 788–798, 2011.

[15] L. Ferrer, H. Bratt, L. Burget, H. Cernocky, O. Glembek, M. Graciarena, A. Lawson, Y. Lei, P. Matejka, O. Plchot, and Scheffer N., "Promoting robustness for speaker modeling in the community: The PRISM evaluation set," in *Proc. NIST 2011 Workshop*, 2011.

[16] Y. Lei, L. Burget, L. Ferrer, M. Graciarena, and N. Scheffer, "Towards noise-robust speaker recognition using probabilistic linear discriminant analysis," in *Proc. ICASSP*, 2012, pp. 4253–4256.

[17] Anthony Larcher, Kong-Aik Lee, Bin Ma, and Haizhou Li, "RSR2015: Database for text-dependent speaker verification using multiple pass-phrases.," in *Proc. Interspeech*, 2012.

[18] M. McLaren, A. Lawson, L. Ferrer, Scheffer N., and Lei, "Trial-based calibration for speaker recognition in unseen conditions," in *Odyssey 2014: The Speaker and Language Recognition Workshop*, 2014.

[19] *The 2009 NIST language recognition evaluation plan*, 2009, `http://www.itl.nist.gov/iad/mig/tests/lre/2009/`.

[20] A. Lawson, M. McLaren, Y. Lei, V. Mitra, N. Scheffer, L. Ferrer, and M. Graciarena, "Improving language identification robustness to highly channel-degraded speech through multiple system fusion," in *Proc. Interspeech*, 2013.

[21] K. Walker and S. Strassel, "The rats radio traffic collection system," in *Proc. Odyssey*, 2012.

[22] T Stafylakis, Patrick Kenny, P Ouellet, J Perez, M Kockmann, and Pierre Dumouchel, "Text-dependent speaker recognition using PLDA with uncertainty propagation," in *Proc. Interspeech*, 2013, p. 36843688.