



Mitigating the Effects of Non-Stationary Unseen Noises on Language Recognition Performance

Luciana Ferrer¹, Mitchell McLaren², Aaron Lawson², Martin Graciarena²

¹ Departamento de Computación, FCEN, Universidad de Buenos Aires and CONICET, Argentina

² Speech Technology and Research Laboratory, SRI International, California, USA

lferrer@dc.uba.ar, {mitch,aaron,martin}@speech.sri.com

Abstract

We introduce a new dataset for the study of the effect of highly non-stationary noises on language recognition (LR) performance. The dataset is based on the data from the 2009 Language Recognition Evaluation organized by the National Institute of Standards and Technology (NIST). Randomly selected noises are added to these signals to achieve a chosen signal-to-noise ratio and percentage of corruption. We study the effect of these noises on LR performance as a function of these parameters and present some initial methods to mitigate the degradation, focusing on the speech activity detection (SAD) step. These methods include discarding the C0 coefficient from the features used for SAD, using a more stringent threshold on the SAD scores, thresholding the speech likelihoods returned by the model as an additional way of detecting noise, and a final model adaptation step. We show that a system optimized for clean speech is clearly suboptimal on this new dataset since the proposed methods lead to gains of up to 35% on the corrupted data, without knowledge of the test noises and with very little effect on clean data performance.

Index Terms: spoken language recognition, non-stationary noise, speech activity detection

1. Introduction

The performance of state-of-the-art language recognition (LR) and speaker recognition (SR) systems on relatively clean microphone or telephone speech has reached impressive levels in recent years. Nevertheless, when noise or channel distortion is present in the signals, the performance degrades drastically. At times, performance can drop to unusable levels, especially in the presence of noises or distortion that were not available for system training. In this work, we begin an effort to study and mitigate the effect of highly non-stationary unseen noises for the LR and SR tasks. Such noises may be encountered in many cellphone conversations in which the speaker is walking or standing on a relatively quiet street, or inside their homes, with noises partially or completely obscuring the speech only in specific regions of the signal. Though we focus here on the LR task, we believe that both the effects of this kind of noise and the methods for mitigating them should be similar across LR, SR and related tasks.

To perform this work, we created a new dataset based on data from the 2009 Language Recognition Evaluation (LRE09) organized by NIST [1]. While the harmful effect of noise on speech processing tasks has been studied extensively over the last few decades using several different datasets, none of these datasets are appropriate for our purpose. Datasets commonly used for studying robustness to noise in automatic speech recognition, such as the Aurora datasets [2, 3, 4, 5] or the SPINE dataset [6], are English only. For speaker recognition, the

PRISM dataset [7] can be used to study the effect of relatively stationary babble noises and reverberation. The dataset for the 2012 Speaker Recognition Evaluation (SRE12) [8] organized by NIST contained three conditions that included stationary noise in the test sample. More recently, the DARPA Robust Automatic Transcription of Speech (RATS) program released a very challenging dataset of noisy and distorted speech [9] for LR, SR, speech activity detection (SAD) and keyword spotting. While this dataset includes different types of degradation, most of the distortion is relatively stationary.

None of these datasets are suitable for studying the effect of extreme non-stationary noise on LR or SR. Furthermore, we wish to be able to control the parameters of the noise; this control will allow us to analyze trends in performance as a function of these parameters. For this reason, we decided to create a new dataset by digitally adding non-stationary noise to a large LR dataset. We chose the LRE09 data, which contains a large number of samples from 23 different languages and is one of the most widely used datasets for evaluation of LR systems. We added 174 distinct car, truck, motorcycle, dog and rooster noises to these signals at different signal-to-noise ratios (SNR) and different rates of corruption. The tools and noises required to create this dataset using the LRE09 data are available to the community upon request.

We show LR performance on this new dataset for a system that has not seen noises of the type present in the dataset during training and find significant degradation in performance due to these noises. We propose some mitigation approaches that focus on the SAD step of the LR system. These approaches result in large gains in LR performance for the corrupted speech.

2. Non-Stationary Noise Data Base

To study the effect of non-stationary noise in language recognition performance, we created a simulated dataset based on the data from LRE09 [1]. The LRE09 data corresponds to telephone bandwidth speech, with some data extracted from radio broadcasts. The data contains 23 languages, with a relatively similar amount of data for each language. The original LRE09 data includes three sets of waveforms with approximate speech durations of 3, 10 and 30 seconds. Only the waveforms containing 30 seconds of speech were selected for noise addition, giving a total of 10,571 original waveforms.

Five types of noises were considered: car, truck, motorcycle, dog and rooster. A set of 258 freely available waveforms containing noises of these types was collected from the internet. These waveforms were edited to contain only the noise of interest, with minimal padding before or after the noise. From this set, we selected waveforms with a maximum length of 10 seconds and a minimum percentage of noise of 70%. The percentage of noise in a waveform was calculated as the percentage of frames with a root mean square value (RMS) within 20dB

from the highest RMS in the waveform. A total of 174 noise signals satisfied these criteria: 45 car, 24 dog, 35 motorcycle, 35 rooster, and 35 truck noises, with an average duration of 4.0 seconds. Each of the waveforms were then faded in (out) by multiplying the first (last) 0.2 seconds with the first (last) half of a Hanning window. This reduces possible effects of the onset and offset of the noise waveforms, which could be distinctive and provide an undesired advantage to SAD algorithms.

The 10,571 original waveforms were randomly split into five equal sets. Each set was then corrupted with one type of noise to generate several new sets with different SNR levels and different percentages of corruption (PC). To this end, we first randomly chose noise samples of the selected type until the number of high-RMS frames from all these signals was approximately equal to the number of frames in the original signal multiplied by the rate of corruption PC/100. The high-RMS frames were defined as those having an RMS within 20dB of the highest RMS value for that waveform. The selected noise samples were then normalized to have the same RMS by dividing the samples by the average RMS over the high-RMS frames. These signals were then concatenated, padding zeroes between them in order to reach the same length as the speech signal. Finally, the corrupted signal was created as $(1 - K)s + K * n$, where s corresponds to the sample values in the speech signal, n corresponds to the concatenated noise signal, and the value of K is determined so that the resulting corrupted signal has the desired SNR. The SNR of the corrupted signal was computed as $20 \log(rms(s) * (1 - K) / (rms(n) * K))$, where $rms(x)$ was computed as the average over the frames with an RMS within 20dB of the highest RMS in the signal. The concatenated noise signal n always has rms value of 1.0 by design.

We call this data set the non-stationary noise data base (NSN-DB). For the experiments in this paper, we divided this dataset in two subsets, dev and eval, splitting the three traffic noises among the two sets. The dev subset contains only the rooster- and motorcycle-corrupted signals, with a total of 4226 signals for each condition given by a combination of SNR and percentage of corruption. The eval subset contains the signals corrupted with the other three types of noises (dog, car and truck), with a total of 6345 signals for each condition.

3. System Description

In this section, we describe the LR system used in this study along with the proposed innovations to make the SAD step more robust to non-stationary unseen noises.

3.1. Language Recognition System

Our spoken language recognition system is a state-of-the-art i-vector-based system [10, 11]. The features are shifted-delta cepstrum (SDC) features given by mel-frequency cepstral coefficient (MFCC) features of 7 dimensions appended with 7-1-3-7 shifted delta cepstra [12], resulting in a final vector of 56 dimensions. A SAD system, later described in Section 3.2, is used to determine the regions where there is speech. The features over these regions are processed with signal-level mean and variance normalization and used to extract sufficient statistics with respect to a GMM. They are then fed to the i-vector extraction step. Given this i-vector, a backend is used to generate the final scores for each language.

The training data for the GMMs, i-vector extractor and backends for all systems was extracted from the CallFriend, NIST LRE 2003, 2005, and 2007, and VOA3 datasets. The data contains a very unbalanced representation of the 23 target languages, ranging from 100 to 7275 samples per language. All

GMMs have 2048 components with diagonal covariances. The i-vectors have 400 dimensions. All components of the system are gender-independent. Note that the NSN-DB is not used to train any part of the LR system. The noises used to generate this dataset are completely unknown to the system, which is trained with relatively clean microphone and telephone data.

A Gaussian backend (GB) is used to generate the final scores [13]. This backend represents the state-of-the-art in i-vector scoring for language recognition and was widely used in recent NIST LREs [11]. In the GB, a Gaussian distribution is estimated for each language, with covariance S shared across languages and language-dependent mean m_l , by maximizing the likelihood on the training data. The scores are computed as the likelihood of the samples given these Gaussian models. Since the LRE09 training data is severely language-imbalanced, we developed a modification of the Gaussian backend approach that provides modest improvements by weighting samples during the computation of the means and covariance of the model such that the sum of weights for each language was identical.

3.2. Baseline SAD System

The baseline SAD system is based on a smoothed log-likelihood ratio between a speech Gaussian mixture model (GMM) and a background GMM [14, 15, 16]. The system was developed for the original LRE09 data and is based on standard 13-dimensional MFCCs, including C0 through C12, with appended deltas and double deltas resulting in a 39-dimensional feature vector. Since C0 is included, gain normalization is performed on the signal prior to feature extraction to make C0 independent of the volume of the signal; all other coefficients are unaffected by gain normalization. To this end, we divide the signal by the average RMS value over the frames with the highest 5% of RMS values. This approach gives us the best performance on the original LRE09 data. In this work we compare this system with one that uses C1 through C13 (C13 is included to keep the number of dimensions unchanged).

All SAD systems in this work are comprised of two 128-component GMMs; one trained on speech frames and one trained on non-speech frames. The data used to train these models comes from our PRISM dataset [7] and includes data from Fisher 1 and 2, Switchboard phase 2 and 3, Switchboard cell-phone phase 1 and 2, and Mixer data. The speech/non-speech labels used to train the GMMs were obtained with our previous HMM-based SAD system. We also tested a set of GMMs trained on the clean data mentioned above and the noisy and reverberated data from the PRISM dataset. Given a test sample, the likelihoods of the speech and the non-speech GMMs given the features are computed for each frame. The logarithm of the ratio of these two likelihoods (the log-likelihood ratio, or LLR) is then smoothed using an averaging filter of length 21 frames. Finally, any frames with LLRs above 0.0 are deemed speech and used for the computation of the sufficient statistics for i-vector extraction, as described in Section 3.1.

3.3. Proposed SAD system

We made two main modifications to the SAD system described above to make it more robust to non-stationary unseen noises. The first modification is based on the observation that some noises might sound more like speech than non-speech to a system that has not encountered them during training. For example, in our simulated data, both rooster cries and dog barks are commonly labeled as speech by our baseline system. While these noises are not well represented by either the speech or the non-speech model in this system, the speech GMM returns a

higher likelihood, resulting in an LLR that exceeds the threshold above which a frame is considered speech. While the LLR might be positive, we expect the two likelihoods involved in the LLR computation to be low. Given this observation, we modified the procedure for deciding whether a frame is speech: a frame is labeled as speech only if both the LLR and the speech likelihood exceed certain (different) thresholds. The two thresholds can be calibrated separately to optimize LR performance. For this purpose, the speech likelihood is smoothed using the same filter length of 21 used for the LLRs.

This new restriction on the speech likelihoods is meant to reduce the number of SAD false alarms (non-speech frames labeled as speech). In the process, the procedure might label as non-speech some noisy speech which does not match the original training data for the speech model. This is acceptable for us, since we are trying to optimize LR performance rather than SAD performance. Noisy speech that is too mismatched with the original training data is also likely to degrade LR performance if those frames are included in the i-vector computation.

The second modification to the SAD system is the addition of an adaptation step. In this approach, two initial sets of speech and non-speech frames are selected using the algorithm described above, potentially using different LLR thresholds for each set to bias the selection to high-confidence frames. These frames are then used to adapt the means of the original speech and non-speech GMMs. The new means are computed as $\tilde{m}_s = (1 - c_s)m_s + c_s M_s$ where \tilde{m}_s is a matrix of mean vectors for the adapted GMM containing one row for each component in the GMM and one column for each dimension, M_s is a similar matrix of means corresponding to the original unadapted GMM, m_s is the matrix of means obtained using the selected frames, and c_s is a vector of factors that interpolates between the original means and the sample-dependent means. If N is the vector of zeroth order statistics and F is the matrix of first order statistics for the selected frames with respect to the original GMM, row i , column j of m_s is given by $m_{s,i,j} = F_{i,j}/N_i$, and $c_{s,i} = w/(N_i + w)$, where w , which is commonly called the relevance factor, controls how much the new means should be adapted to the sample's data. This adaptation method is commonly used in speaker recognition to adapt the universal background model to each speaker's data [17].

The adapted models are then used to recompute the LLRs for each frame which are then smoothed and thresholded as in the baseline system to obtain the final speech regions. The frames originally deemed as non-speech by the speech likelihood threshold are still labeled as non-speech after recomputing the LLRs. This procedure gave significant gains in this data.

4. Experiments

This section presents our results on the NSN-DB. Development experiments showing performance as a function of different system parameters are performed on the dev subset of the dataset for the clean signals and the 50% corruption condition with -20 dB SNR. The last section shows the performance on the eval subset for the parameters selected based on the dev performance for all three percentages of corruption (20, 50 and 80%) and two SNR levels, as well as for the clean data.

Results are presented in terms of average minimum Cdet. The Cdet is defined in the LRE09 evaluation plan [1]. We use minimum Cdet rather than actual Cdet to avoid considering issues of calibration in these initial experiments on this dataset. While calibration is an important issue that might affect performance significantly, it is somewhat orthogonal to the issues we

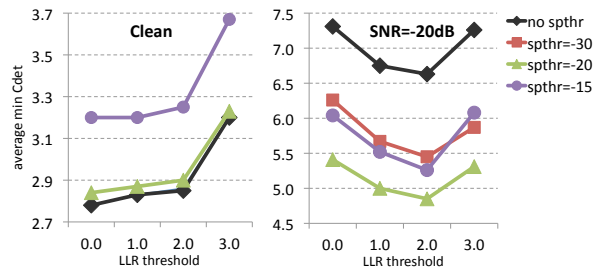


Figure 1: Comparison of LR performance on the dev NSN-DB subset for clean signals and 50% corruption rate with SNR=-20dB, for different LLR thresholds (x-axis) and different speech likelihood thresholds (legend). The spthr=-30 line for the clean signals is hidden by the “no spthr” line.

are studying in this work.

All systems presented here use the same GMM, i-vector extractor and Gaussian backend. These models were trained on the LRE09 training dataset as described in Section 3.1. This is all telephone-bandwidth data, without added noise. Hence, the noises present in the NSN-DB are unseen during training of the system. Note, though, that the models are a good representation of the clean part of the signals in the NSN-DB.

4.1. Development Experiments

In this section we present results on the development set as a function of system parameters.

4.1.1. Effect of Training Data and C0

We tried three different SAD systems: using C0 through C12 and clean and noisy training data; using C1 through C13 and the same training data; and using C1 through C13 and only clean training data. Our results showed that while including C0 gives slightly better performance (1% relative) for the clean condition, it degraded performance on the -20 dB condition by about 15% relative (results not shown for lack of space). This is a very intuitive result, since the gain-normalized C0 reflects the (relative) energy of the signal, which is a good predictor of speech presence when the signal is clean. On the other hand, when the noise level is higher than the speech level, a SAD system that uses gain-normalized C0 will tend to label the noise as speech and the speech as silence. Finally, our results showed that including PRISM noises during training of the SAD models does not give any advantage when testing on mismatched noises (PRISM only contains babble noises, which are quite different from the ones included in NSN-DB). Unless otherwise stated, the experiments in this paper used the SAD models based on C1 through C13 and only clean training data.

4.1.2. Effect of LLR and Speech Likelihood Thresholds

Figure 1 shows the performance on the dev set as a function of the LLR threshold and the speech likelihood threshold (see Sections 3.2 and 3.3). No adaptation of the SAD models was performed for these experiments. Comparing the two sub-figures we can see a close to three-fold degradation on the baseline performance (black markers at an LLR threshold of 0) when -20dB noise is added to the clean signals.

For the clean signals, we can see that baseline SAD setup –no restriction on speech likelihoods, and an LLR threshold of 0.0– is indeed optimal for this data. On the other hand, for the highly corrupted data, we see a large benefit from restricting the speech likelihoods to be above a certain value. A speech likelihood threshold of -20 gives around 25% improvement over not using any threshold. Furthermore, for this data, a higher LLR threshold gives an additional gain over using the 0.0 value

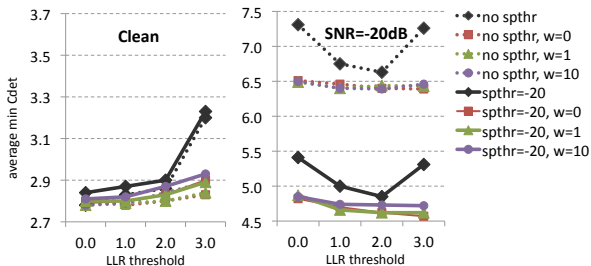


Figure 2: Comparison of LR performance on the dev NSN-DB subset for clean signals and 50% corruption rate with SNR=-20dB, for different LLR thresholds (x-axis), different speech likelihood thresholds (spthr) and different adaptation weights (w).

which is optimal for clean data. In summary, we see that the combination of a speech likelihood threshold of -20 and an LLR threshold of 1.0 or 2.0 gives a good trade-off between the clean and the noisy conditions.

4.1.3. Effect of Adaptation

Figure 2 shows the results when using adaptation for a speech likelihood threshold of -20 (the optimum from Figure 1) or no threshold. For comparison, we also include the results without adaptation for both cases. We set the LLR threshold for selecting non-speech and speech frames for adaptation to 0.0 and 2.0, respectively. These values were independent of the threshold used on the final LLRs computed with the adapted models and were selected based on the performance on the dev data.

The figure shows that adaptation leads to modest gains on both clean and noisy data. Furthermore, on the corrupted data, we see that restricting the detected speech regions to frames with a speech likelihood greater than -20 still gives a significant gain after adaptation, indicating that the two procedures are complementary. Finally, note that adaptation makes the performance of the system much less dependent on the LLR threshold, thus less sensitive to a wrong choice of threshold. In summary, the adaptation setup that gives the best trade-off across the two conditions uses a speech likelihood threshold of -20 and an adaptation weight of 0.0 or 1.0. We selected 1.0 as a safer value, since 0.0 can lead to overfitting. Again, an LLR threshold of 1.0 or 2.0 gives the best trade-off between the clean and the noisy conditions. With this setup, we achieve a gain of 37% on the noisy data with respect to the baseline performance (no spthr, an LLR threshold of 0.0, and no adaptation).

4.2. Results on the NSN-DB Evaluation Subset

Figure 3 shows the results on the held-out subset of the NSN-DB. Note that not only are the speech and the noise signals disjoint between the dev and the eval sets, the types of noises are also different. We can see that the proposed procedures lead to significant gains over the noisy conditions, at the cost of only very small degradation on the clean data. At the -20dB condition, we achieve gains of 21%, 33% and 13% for the 80%, 50% and 20% corruption rates, respectively. At 0dB, the gains are smaller at 15%, 7% and 1%. Note that these are the combined gains from all proposed methods. While the gains from leaving out C0 (second bar compared to first) and the gains from adaptation (last bar compared to second-to-last bar) are very similar for the dev and the eval sets, the gains from the other two approaches are not. Increasing the LLR threshold does not lead to gains on the eval data, except on the 80% corruption rate condition. Furthermore, using a speech likelihood threshold gives a much smaller gain on the eval data than on the dev data. The large gain on the dev data happened for the rooster

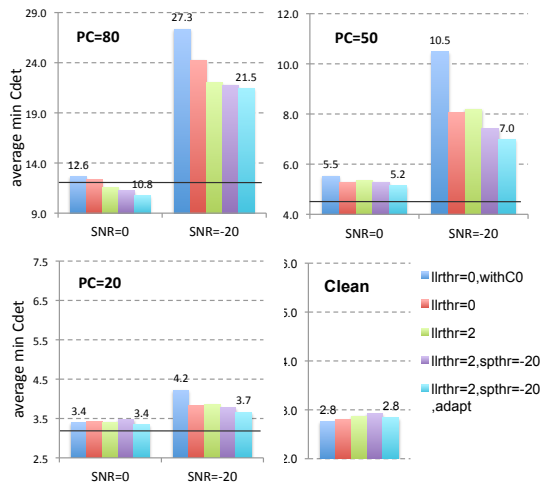


Figure 3: Comparison of LR performance on the eval NSN-DB subset for three corruption rates (PC) with two SNR levels and for clean data when using different SAD systems: the baseline SAD system (llrthr=0,withC0), the baseline system without C0 but including C13 (llrthr=0), that same system with a higher threshold (llrthr=2,0), then adding the speech likelihood constraint of -20 (llrthr=2,spthr=-20), and additionally adapting to each signal's data (llrthr=2,spthr=-20,adapt). The y-axis ranges were chosen to facilitate comparisons across figures. They were set such that the same difference between any bar and the leftmost bar in the same figure represents approximately the same relative gain or loss across all figures. The black horizontal lines in the figures for PC 80, 50 and 20 correspond to the results using the oracle SAD system.

noises, which are not present on the eval data. Nevertheless, overall, the proposed methods are quite safe in that they never lead to significant degradations, while resulting in significant gains for some noises at low SNR levels.

Finally, the black lines in the three figures corresponding to corrupted data show the LR performance using an oracle SAD system. This system labeled all frames where noise was added as non-speech (even if speech is, in fact, somewhat intelligible in those frames). For the remaining frames, it uses the SAD output obtained on the clean signals. As we can see, for PC=80, where the oracle discards 80% of the frames, and intermediate SNR levels, we can outperform the oracle by using the proposed approaches. For the other cases, we can see that the proposed methods allow us to bridge a significant part of the gap between the baseline and the oracle performance.

5. Conclusions

We introduced a new dataset with highly non-stationary noises. We showed that these noises result in a large degradation in the performance of an LR system developed for clean speech which had not encountered these kinds of noises during training. We proposed simple modifications to the SAD stage of the system to mitigate this. The proposed techniques include discarding C0 from the features used for SAD, using a more stringent threshold on the SAD scores, thresholding the speech likelihoods returned by the model, and a model adaptation step. These techniques produced significant gains of up to 33% in LR performance on the noisier conditions. On the clean data, some of these procedures led to small degradations, no greater than 6%, compared to the system optimized for that data. In the future, we plan to test how the proposed approaches behave when stationary noise is present in the signal in addition to the non-stationary bursts. We also plan to explore methods for robustness at other system stages besides SAD.

6. References

- [1] “NIST LRE09 evaluation plan,” http://www.itl.nist.gov/iad/mig/tests/lre/2009/LRE09_EvalPlan_v6.pdf.
- [2] H.-G. Hirsch and D. Pearce, “The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions,” in *ASR2000-Automatic Speech Recognition: Challenges for the new Millenium ISCA Tutorial and Research Workshop (ITRW)*, 2000.
- [3] A. Moreno, B. Lindberg, C. Draxler, G. Richard, K. Choukri, S. Euler, and J. Allen, “SPEECHDAT-CAR. a large speech database for automotive environments,” in *LREC*, 2000.
- [4] N. Parihar and J. Picone, “Aurora working group: DSR front end LVCSR evaluation,” *Inst. for Signal and Information Process, Mississippi State University, Tech. Rep.*, vol. 40, p. 94, 2002.
- [5] H. Hirsch, “Aurora-5 experimental framework for the performance evaluation of speech recognition in case of a hands-free speech input in noisy environments,” *Niederrhein Univ. of Applied Sciences*, 2007.
- [6] A. Schmidt-Nielsen, E. Marsh, J. Tardelli, P. Gatewood, E. Kreamer, T. Tremain, C. Cieri, and J. Wright, “Speech in noisy environments (SPINE) evaluation audio,” *Linguistic Data Consortium*, 2000.
- [7] L. Ferrer, H. Bratt, L. Burget, H. Cernocky, O. Glembek, M. Graciarena, A. Lawson, Y. Lei, P. Matejka, O. Plchot, and N. Scheffer, “Promoting robustness for speaker modeling in the community: the PRISM evaluation set,” in *Proceedings of SRE11 Analysis Workshop*, Atlanta, USA, Dec. 2011.
- [8] “NIST SRE12 evaluation plan,” http://www.nist.gov/itl/iad/mig/upload/NIST_SRE12_evalplan-v17-r1.pdf.
- [9] K. Walker and S. Strassel, “The RATS radio traffic collection system,” in *Odyssey 2012: The Speaker and Language Recognition Workshop*, 2012.
- [10] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Front-end factor analysis for speaker verification,” *IEEE Trans. Audio, Speech, and Lang. Process.*, vol. 19, no. 4, pp. 788–798, May 2011.
- [11] 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 *Interspeech-2012*, 2012, pp. 2049–2052.
- [12] B. Bielefeld, “Language identification using shifted delta cepstrum,” in *Fourteenth Annual Speech Research Symposium*, 1994.
- [13] D. G. Martinez, O. Plchot, L. Burget, O. Glembek, and P. Matejka, “Language recognition in iVectors space,” in *Proc. Interspeech*, Florence, Italy, Aug. 2011.
- [14] T. Ng, B. Zhang, L. Nguyen, S. Matsoukas, X. Zhou, N. Mesgarani, K. Vesely, and P. Matejka, “Developing a speech activity detection system for the DARPA RATS program,” in *Proc. Interspeech*, Portland, USA, Sep. 2012.
- [15] M. Graciarena, A. Alwan, D. Ellis, H. Franco, L. Ferrer, J. H. Hansen, A. Janin, B. S. Lee, Y. Lei, V. Mitra *et al.*, “All for one: feature combination for highly channel-degraded speech activity detection,” in *Proc. Interspeech*, Lyon, France, Aug. 2013.
- [16] L. Ferrer, M. McLaren, N. Scheffer, Y. Lei, M. Graciarena, and V. Mitra, “A noise-robust system for NIST 2012 speaker recognition evaluation,” in *Proc. Interspeech*, Lyon, France, Aug. 2013.
- [17] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, “Speaker verification using adapted Gaussian mixture models,” *Digital Signal Processing*, vol. 10, pp. 19–41, 2000.