TRAP Language Identification System for RATS Phase II Evaluation

Kyu J. Han[1], *Sriram Ganapathy*[1], *Ming Li*[2], *Mohamed K. Omar*[1], *Shrikanth Narayanan*[2]

[1]IBM T. J. Watson Research Center, Yorktown Heights, NY, USA
[2]University of Southern California, Los Angeles, CA, USA
{kjhan,sganapa,mkomar}@us.ibm.com,mingli@usc.edu,shri@sipi.usc.edu

Abstract

Automatic language identification or detection of audio data has become an important preprocessing step for speech/speaker recognition and audio data mining. In many surveillance applications, language detection has to be performed on highly degraded audio inputs. In this paper, we present our work on language detection in highly degraded radio channel scenarios. We provide a brief description of the Targeted Robust Audio Processing (TRAP) language detection system built for the Phase II Evaluation of the Robust Automatic Transcription of Speech (RATS) program. This system is a combination of 15 systems with different frontends and speech activity decisions. We also analyze the usefulness of multi-layer perceptron (MLP) based non-linear projection of i-vectors before SVM classification. The proposed backend reduces the Equal Error Rate (EER) by 11%–25% relative compared to the baseline PCA-based feature representation for SVM classification, on the RATS test data consisting of data from eight high-frequency radio communication channels.

Index Terms: Language identification (detection), highly degraded radio channel, RATS, i-vector, multi-layer perceptron.

1. Introduction

This paper describes the recent effort of the Targeted Robust Audio Processing (TRAP) team for the Phase II Language Identification (LID) Evaluation in the DARPA Robust Automatic Transcription of Speech (RATS) program. In the RATS program, noisy speech data transmitted on eight different high-frequency radio communication channels [1] are studied for four tasks; Speech Activity Detection (SAD), Keyword Spotting (KWS), Speaker Identification (SID) and Language Identification (LID). For the LID task, four durations (120s, 30s, 10s and 3s) are considered as data lengths for testing. For each duration, duration-specific testing examples are supposed to be identified as one of five target languages (Arabic Levantine, Farsi, Dari, Pashto and Urdu). To achieve the target performance across durations, it is therefore critical to efficiently handle noisy data and model short-duration segments.

The main components of our LID system for the RATS Phase II Evaluation are (1) two SADs for diversity in system combination, (2) seven frontend features, (3) three combinations of projections and/or classifiers and (4) multi-class linear regression for the system combination of the 15 individual systems. We managed to improve system performance by more than 50% relative in terms of the Equal Error Rate (EER) compared to our Phase I LID submission.

This work was supported in part by Contract No. D11PC20192 DOI/ NBC under the RATS program. The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

This paper is organized as follows. In Section 2, data categorization is given for system training and evaluation. In Section 3, we describe the details of the frontend and backend configurations of our LID system. In Section 4, experimental results are given. We further analyze the performance of feature space expansion using a neural network of one hidden layer and Support Vector Machine (SVM) classification. This backend provides the best performance especially for shorter duration trials among the three backend configurations used in our final submission system. In Section 5, we provide conclusions and future directions.

2. Data

The Linguistic Data Consortium (LDC) distributed the training and development data of the five target languages and ten non-target languages totaling approximately 3,700 hours of recordings (See Table 1). We split the data into three parts for system training, calibration and internal evaluation; TRAIN, COMB, and TEST. The TRAIN data set was used to capture background statistics and train the Universal Background Models (UBMs) [2]. This data set was also utilized to find subspace projections for compact feature representations and backend classifiers. The COMB data set was prepared to calibrate parameters for score combination. The TEST data set is our internal test set to evaluate the system performance. The DEV2 data set is one of the official testing data sets for the RATS Phase I LID Evaluation and was used as an alternative test set in preparation for the Phase II Evaluation. Table 1 shows the data sets used for our system building and testing in terms of the number of recordings and hours.

3. The TRAP System Description

The TRAP team's LID system submission for the Phase II Evaluation of the RATS program consists of 15 system configurations. It is based on two SAD setups, one of which is a channel-dependent (CD) SAD utilizing multi-pass Hidden Markov Model (HMM) Viterbi segmentation and fusion of multiple feature streams [3] and the other is channel-independent (CI) SAD with a two-pass modified Cumulative Sum (CUSUM) approach based on Maximum A Posteriori (MAP) adaptation [4]. Each setup has distinct ingredients as follows:

- CD-SAD: Channel detection with eight channel-dependent Gaussian Mixture Models (GMMs), followed by speech/non-speech HMM Viterbi segmentation using channel-dependent Deep Neural Networks (DNNs) trained on Perceptual Linear Prediction (PLP), voicing and rata-scale features. The second-pass segmentation is then applied for a frame-level score combination of two sets of DNNs trained on PLP, voicing, rate-scale

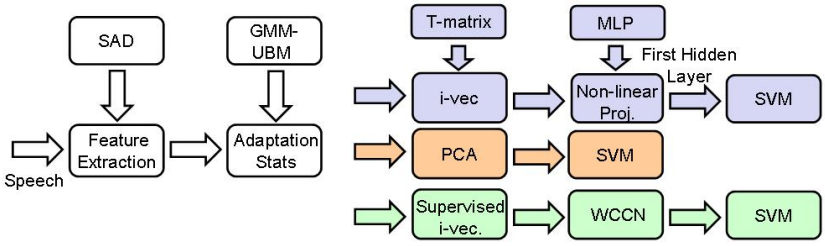

Figure 1: *Block schematic of the TRAP submission for the RATS Phase II LID Evaluation.*

Table 1: Statistics of the TRAIN, COMB, TEST, and DEV2 data sets for system building and testing.

	No. of Recordings	Hours
TRAIN	87,774	2,926
COMB	14,328	324
TEST	9,733	478
DEV2	1,914	64
Total	113,749	3,792

and Frequency-Domain Linear Prediction (FDLP) features and deep Convolutional Neural Networks (CNNs) trained on log-mel spectra [3].

- CI-SAD: The modified CUSUM algorithm is used to MAP-adapt the means of channel-independent speech/non-speech GMMs before segmentation refinement. Each GMM consists of 4,042 diagonal-covariance Gaussian components. The GMM parameters are estimated using maximum likelihood estimation to make sure. Every nine frames of 13-dimensional PLP coefficients are spliced together and then projected to a 40-dimensional vector using Linear Discriminant Analysis (LDA) and Maximum Likelihood Linear Transform (MLLT) [4].

The following sub-sections detail each partner's contribution to our final system.

3.1. IBM

IBM contributed 11 system configurations including 6 frontend features:

- Mel-Frequency Cepstral Coeffiient (MFCC): 13-dimensional base features generated using 37 mel-frequency filter banks, appended with delta and acceleration features, resulting in the total 42-dimensional feature vector for every 32ms frame with a 10ms shift rate. The frequency bands of interest are limited to 300–3300Hz for more robustness to unseen data characteristics.
- Wideband MFCC (WB-MFCC): 19-dimensional cepstral coefficients derived from 24 mel sub-bands in the frequency range of 125–3700Hz for every 32ms frame with a shift of 10ms. Then they are added with delta and acceleration components to yield 57-dimensional features.
- Shifted Delta Cepstrum (SDC): 7-3-3-7 SDC configuration [5] for MFCCs. It is then concatenated with the base MFCC features to a create 56-dimensional feature vector per frame.
- Frequency-Domain Linear Prediction (FDLP): Windowing of the Discrete Cosine Transform (DCT) of a long-term segment $(1,000$ms) for a given signal is followed by the linear prediction of sub-band DCT components to yield temporal envelopes in each band [6]. The sub-band envelopes are then integrated in short-term windows (32ms with a shift of 10ms) to derive a spectrographic representation of the signal which is used as power spectral representation for the second autoregressive (AR) model across the bands [7]. The output of the second AR model is converted to 14-dimensional cepstral features, which are added with delta and acceleration coefficients.
- Cortical modulation (CORT): Two dimensional (2-D) spectrographic representations are derived for a given signal by emulating various processing stages in the periphery of the human auditory system [8]. The auditory spectrogram is then converted to modulation representation using Fourier transforms along the spectral and temporal axis and modulation filtering is applied to extract key dynamics in the scale and rate dimensions, respectively [9]. The modulation filters used in this feature extraction scheme are broad enough to cover a wide range of dynamics (0–2 cycles per octave in the scale dimension and 0.25–25Hz in the rate dimension). Cepstral transformation is applied on the filtered auditory spectrograms and delta/acceleration features are appended to obtain 42-dimensional features.
- Power-Normalized Cepstral Coefficient (PNCC): The power law nonlinearity is applied on temporal envelopes estimated from Gammatone filters [10]. This is followed by a noise supression procedure using assymmetric filters and a power normalization module using a long window span. A frequency smoothing is applied and cepstral features are derived using DCT on the compressed spectrogram. We derive 19 cepstral features and these are used with delta and acceleration components.

CI-SAD was used for MFCC and SDC while CD-SAD for the other features. All the frontends are Wiener-filtered before SAD to suppress channel noise effects [11].

For each feature stream except WB-MFCC[1], two separate projection/classifier backends were developed. One backend consists of PCA-based feature space projection and SVM classification with the 5^{th}-order polynomial kernel [12]. In [12], higher order polynomial kernels such as 5^{th} or 6^{th} were experimentally proven to outperform lower orders like 2^{nd} or 3^{rd} in SVM classification. The other, "*advanced backend*", con-

[1]WB-MFCC has only one backend of PCA projection and SVM classification with the 5^{th}-order polynomial kernel.

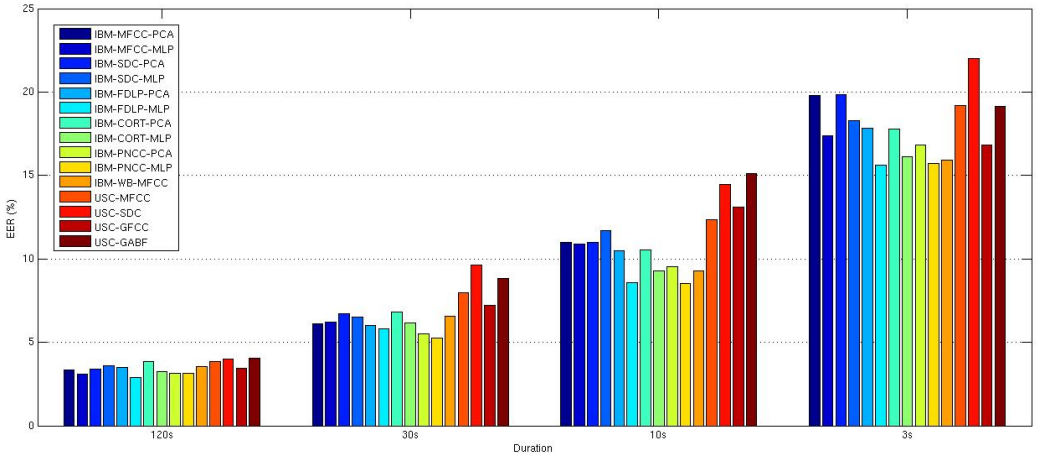

Figure 2: *Individual system performance on DEV2.*

tains i-vector representation followed by feature space expansion using a one hidden layer perceptron and SVM classification with higher-order polynomial kernel functions such as 10^{th}/11^{th}. The advantage of this backend can be seen when comparing each pair of frontend feature systems (Figure 1), especially for shorter duration trials. (We will discuss it in more detail in Section 4.2.) One-versus-all binary SVM classifiers for the target languages were trained using the LIBSVM package [13].

3.2. USC

With CI-SAD, USC implemented 4 sub-systems using simplified and supervised i-vector modeling [14, 15] based on 4 different frontend features, each of which feature warping was applied for:

- MFCC: 25ms Hamming window applied with a 10ms shift. 18-dimensional base features are appended with their delta coefficients, resulting in 36-dimensional feature vector per frame.
- SDC: 7-1-3-7 SDC configuration for MFCCs. It is then concatenated with the base MFCC features including C0 to a 56-dimensional feature vector per frame.
- Gammatone Frequency Cepstral Coefficient (GFCC) [16]: 44-dimensional feature vector per frame generated using 64 Gammatone filter banks (22-dimensional base features without C0, and their first derivatives).
- Gabor Filtering (GABF) [17]: Gabor filters applied for spectro-temporal information to yield 153-dimensional feature vectors.

For these frontends, we adopted the simplified and supervised i-vector modeling framework [14, 15] which not only achieved good results but also reduced a computational time by more than 100 times. In this framework, traditional i-vectors [18] are extended to label-regularized supervised vectors by concatenating GMM mean supervectors (GSVs) and the total variability matrix (T-matrix) with a label vector and a linear classifier matrix, respectively. These supervised i-vectors are optimized to not only reconstruct the GSVs but also minimize mean-squared errors between the original and the reconstructed label vectors, such that they become more discriminative. Also, Factor Analysis (FA) can be performed on pre-normalized GSVs to ensure that each Gaussian component is treated equally during FA, which reduces a computational cost significantly by a factor of 25. Moreover, we can further enhance the efficiency by using a pre-computed table. More details about the simplified and supervised i-vector modeling are provided in [14, 15].

Within-Class Covariance Normalization (WCCN) [19] was applied on the resulting i-vectors before SVM. For fast training of SVM models, we used the 2^{nd}-order polynomial mappings [20] in the LIBLINEAR package [21], which resulted in a multi-class SVM classifier for each duration testing. Moreover, we sub sampled the in the SVM training to make it more balanced and efficient.

Table 2: DEV2 results of the TRAP team's submissions across duration in terms of EER (%). The numbers in the parentheses indicate the total number of sub-systems combined.

System	120s	30s	10s	3s
Primary (15)	1.8	3.2	5.6	10.0
Contrastive I (11)	1.8	3.1	5.8	10.0
Contrastive II (7)	1.9	3.7	6.1	11.4
Phase I (3)	2.9	6.9	12.7	18.9

4. Experimental Results

4.1. Discussions on DEV2 results

Figure 2 shows the performance of individual system configurations on the DEV2 data set, where all individual system performance is compared. For shorter duration testings such as 10s and 3s, the advanced backend is shown to provide significant improvement of 11%–25% especially for FDLP, CORT and PNCC compared to the PCA-SVM setup. (We will analyze this further in the next sub-section.) Among the frontend features, PNCC shows the best results across durations, while FDLP offers similar performance except for the 30s trials. From the USC streams GFCC is the best, which can be expected since, in comparison with MFCCs, GFCCs have higher resolution on low frequency responses.

Table 2 shows our final submission results for the Phase II Evaluation compared with our Phase I submission. The primary submission consists of all the 15 system configurations combined by multi-class logistic regression using the FoCal toolkit

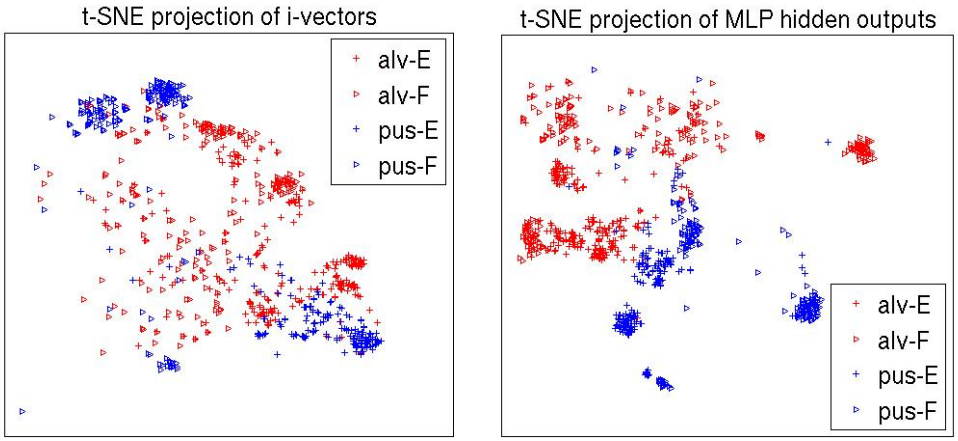

Figure 3: *Scatter plot of two dimensional t-SNE projections for the input i-vectors as well as the MLP hidden layer outputs.*

[22] in a duration-specific manner. For the Contrastive I system, we reduced the number of sub-systems by choosing ten system configurations that could achieve similar performance with the Primary system across duration. The Contrastive II system is a combination of seven IBM individual systems. All the three submission results show that we improved our LID system performance by more or less 50% relative compared to the Phase I submission.

4.2. Analysis on advanced backends

In this section, we analyze the usefulness of MLP-based transformation of i-vectors which are input to SVM classification. As shown in Figure 1, the input features are used to adapt the GMM means and the GSVs are transformed to i-vectors using total variability matrix (T-matrix).

In contrast to the past approach of using i-vectors as features to a MLP classifier for LID [23], our proposal of a "deep" architecture for LID using an MLP-based non-linear projection was inspired by the advances in Deep Belief Networks (DBNs) for Automatic Speech Recognition (ASR) [24]. In ASR applications, the discriminative pre-training of DBNs is done by training a single-hidden layer MLP which is used as an initialization for MLPs with multiple hidden layers [25].

For LID applications, we use a single hidden layer MLP as a feature transformation before SVM classification. The i-vectors are used as features for the MLP and dimensionality of the hidden layer is much higher than the input layer. The MLP is trained with language targets using speech data from all durations and channels. Once the MLP is trained, the non-linear transformation from the i-vectors to hidden layer outputs is alone retained and these features are used for SVM classifiation.

We illustrate the usefulness of MLP-based transformation with the Stochastic Neighborhood Embedding (SNE) based data visualization tool [26]. The input i-vectors as well as the high dimensional MLP hidden layer outputs are projected to two dimensions and this scatter plot is shown in Figure 3. We use 200 random utterances from two different languages (Arabic Levantine (alv) and Pashto (pus)) recorded from two different channels in the RATS development database (channel E and F [1]). As seen in Figure 3, the two-dimensional projection of MLP hidden layer outputs are more discriminative compared to the i-vectors. This explains why in our LID experiments (Section 4.1) systems using MLP-based non-linear features provided significant improvements (relative improvements of 11%–25%) compared to the PCA-SVM setup. It is also observed in Figure 3 that utterances from the same channel tend to form clusters although no channel information was used in the MLP training.

5. Conclusions

In this paper, we discussed the TRAP submission for the RATS LID Phase II evaluation. The submission was a combination of 15 systems with diverse SADs, frontends and backend models. We provided a brief description of each system with emphasis on the new system components compared to our Phase I submission. These new components improved the performance on the RATS DEV2 test set by more than 50% relative. One of these components is the simplified and supervised i-vector modeling framework which not only achieved good results but also reduced the computational load for FA by a factor of 100.

We also provided an analysis of another new component namely the MLP-based non-linear projection of i-vectors before SVM classification. It reduced the EER by 11%–25% relative compared to the baseline PCA-based features for SVM classification. The better performance of these MLP-based representations can be attributed to using the annotation of the training data in estimating the non-linear projection compared to the unsupervised learning of the PCA projection.

6. Acknowledgements

We thank George Saon for providing the CD-SAD setup and Jason Pelecanos, Sibel Yaman, Weizhong Zhu, Todd Ward and Salim Roukos for the suggestions.

7. References

[1] K. Walker and S. Strassel, "The RATS radio traffic collection system," *Proc. of Odyssey: The Speaker and Language Recognition Workshop*, 2012, pp. 291–297.

[2] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Process.*, vol. 10, pp. 19–41, 2000.

[3] G. Saon, S. Thomas, H. Soltau, S. Ganapathy, and B. Kingsbury, "The IBM speech activity detection system

for the DARPA RATS program," *submitted to Interspeech*, 2013.

[4] M. K. Omar, "Speech activity detection for noisy data using adaptation techniques," *Proc. of Interspeech*, 2012.

[5] P. A. Torres-Carrasquillo, E. Singer, M. A. Kohler, R. J. Greene, D. A. Reynolds, and J. R. Deller Jr., "Approaches to language identification using Gaussian mixture models and shifted delta cepstral features," *Proc. of Interspeech*, 2002, pp. 89–92.

[6] S. Ganapathy, S. Thomas, and H. Hermansky, "Feature extraction using 2-D autoregressive models for speaker recognition," *Proc. of Odyssey: The Speaker and Language Recognition Workshop*, 2012.

[7] M. Athineos and D. Ellis, "Autoregressive modelling of temporal envelopes," *IEEE Trans. Signal Process.*, vol. 55, no. 11, pp. 5237–5245, Nov. 2007.

[8] T. Chi, P. Ru, and S.A. Shamma, "Multiresolution spectrotemporal analysis of complex sounds," *J. Acoust. Soc. Am.*, vol. 118, pp. 887–906, 2005.

[9] S. Nemala, K. Patil, and M. Elhilali, "A multistream feature framework based on bandpass modulation filtering for robust speech recognition," *IEEE Trans. Audio, Speech and Lang. Process.*, vol. 21, no. 2, pp. 416–426, Feb. 2013.

[10] C. Kim and R. M. Stern, "Power-normalized cepstral coefficients for robust speech recognition," *Proc. of ICASSP*, 2012.

[11] A. Adami, L. Burget, S. Dupont, H. Garudadri, F. Grezl, H. Hermansky, P. Jain, S. Kajarekar, N. Morgan, and S. Sivadas, "Qualcomm-icsi-ogi features for asr," *Proc. of ICASSP*, 2002.

[12] S. Yaman, J. Pelecanos, and M. K. Omar, "On the use of nonlinear polynomial kernel SVMs in language recognition," *Proc. of Interspeech*, 2012.

[13] C. Chang and C. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intel. Systems Tech.*, vol. 2, no. 3, pp. 27:1–27:27, 2011.

[14] M. Li and S. Narayanan, "Simplified supervised i-vector modeling and sparse representation with application to robust language recognition," *submitted to Comp. Speech Lang.*

[15] M. Li, A. Tsiartas, M. Segbroeck, and S. Narayanan, "Speaker verification using simplified and supervised i-vector modeling," *appear to Proc. of ICASSP*, 2013.

[16] Y. Shao and D. Wang, "Robust speaker identification using auditory features and computational auditory scene analysis," *Proc. of ICASSP*, 2008, pp. 1589–1592.

[17] M. R. Schadler, B. T. Meyer, and B. Kollmeier, "Spectro-temporal modulation subspace-spanning filter bank features for robust automatic speech recognition," *J. Acoust. Soc. Am.*, vol. 131, no. 5, pp. 4134–4151, 2012.

[18] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 19, no. 4, pp. 788–798, 2011.

[19] A. Hatch and A. Stolcke, "Generalized lineer kernels for one-versus-all classification: application to speaker recognition," *Proc. of ICASSP*, 2006.

[20] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin, "LIBLINEAR: A library for large linear classification," *J. Machine Learn. Research*, vol. 9, pp. 1871–1874, 2008.

[21] Y. Chang, C. Hsieh, K. Chang, M. Ringgaard, and C. Lin, "Low-degree polynomial mapping of data for SVM," *J. Machine Learn. Research*, 2009.

[22] N. Brummer, "Application-independent evaluation of speaker detection," *Proc. of Odyssey: The Speaker and Language Recognition Workshop*, 2004.

[23] P. Matejka, O. Pichot, M. Soufifar, O. Glembek, L. F. D'Haro, K. Vesely, F. Grezl, J. Ma, S. Matsoukas, and N. Dehak, "Patrol team language identification system for DARPA RATS P1 evaluation," *Proc. of Interspeech*, 2012.

[24] A. Mohamed, G. Dahl, and G. Hinton, "Deep belief networks for phone recognition," *Proc. of NIPS*, 2009.

[25] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," *Proc. of NIPS*, 2006.

[26] L. Van der Maaten and G. Hinton, "Visualizing high-dimensional data using t-SNE," *J. Machine Learn. Research*, vol. 9, pp. 2579–2605, 2008.