Improvements in Language Identification on the RATS Noisy Speech Corpus

Jeff Ma[1], *Bing Zhang*[1], *Spyros Matsoukas*[1],
Sri Harish Mallidi[2], *Feipeng Li*[2], *Hynek Hermansky*[2]

[1] Raytheon BBN Technologies, USA
[2] The John Hopkins University, USA

{jma,bzhang,smatsouk}@bbn.com, mallidi@jhu.edu, fli12@jhmi.edu, hynek@jhu.edu

Abstract

This paper presents a set of techniques that we used to develop the language identification (LID) system for the second phase of the DARPA RATS (Robust Automatic Transcription of Speech) program, which seeks to advance state-of-the-art detection capabilities on audio from highly degraded radio communication channels. We report significant gains due to (a) improved speech activity detection, (b) special handling of training data so as to enhance performance on short duration audio samples, and (c) noise robust feature extraction and normalization methods, including the use of multi-layer perceptron (MLP) based phoneme posteriors. We show that on this type of noisy data, the above techniques provide on average a 27% relative improvement in equal error rate (EER) across several test duration conditions.

Index Terms: language identification, noisy speech, robust feature extraction

1. Introduction

The goal of the DARPA RATS (Robust Automatic Transcription of Speech) program is to create technology capable of accurately determining speech activity regions, detecting key words, and identifying language and speakers in highly degraded, weak and/or noisy communication channels. The Patrol team, led by BBN, participates in all the RATS tasks. The LID system that the Patrol team built for the RATS Phase 1 evaluation was described in [1]. That paper mainly focused on the individual systems built by members of the Patrol team, as well as on the calibration and the fusion of the individual systems. In this paper we concentrate on the work that was conducted to improve the BBN system for the RATS Phase 2 evaluation.

This paper is organized as follows: Section 2 describes the RATS LID data corpus that we used for developing our systems; Section 3 briefly reviews the BBN LID Phase-1 system; Section 4 presents improvements on various components of the BBN system; Section 5 concludes this paper.

2. The RATS LID Data Corpus

The Linguistic Data Consortium (LDC) provided training and test data for the RATS evaluation tasks. For the LID evaluation task, the provided audio recordings cover the 5 target languages (Arabic, Dari, Farsi, Pashto, and Urdu) and the 10 non-target languages (English, Spanish, Mandarin, Thai, Vietnamese, Russian, Japanese, Bengali, Korean, Tagalog). These recordings were selected from both existing data resources and new data collected specifically for RATS (more details can be found in [1]). All recordings were about 2 minutes long and were retransmitted through 8 different communication channels, labeled by the letters A through H. The retransmitted data was released to the RATS participants for developing their evaluation systems. LDC issued 3 incremental data releases for the LID task: LDC2011E95, LDC2011E111, and LDC2012E03. We only used the first two releases for developing our LID systems.

All LID systems were evaluated under four testing conditions in which test samples are 120, 30, 10, and 3 seconds long, respectively. In this paper we use "120s", "30s", "10s", and "3s" to represent the four conditions. The RATS program does not provide development data for the short-duration conditions, "30s", "10s" and "3s". Hence the participants need to find ways to develop systems for the short-duration conditions.

As described in [1], we partitioned the first two data releases into training and development sets. To construct development samples for the shorter durations, on each of the 120s development audio samples we started from the beginning of the audio, cut out the first 30, 10 and 3 seconds of speech (based on BUT's voice activity detection) to obtain "30s", "10s" and "3s" development samples, respectively. Right before the Phase 1 evaluation, a small official development set was released for the purpose of a "dry-run" evaluation. By examining the "dry-run" test samples, we saw that the cut-outs of the short-duration samples did not always start from the beginning of the 120s audio samples. For the Phase-2 evaluation we used our Phase-1 training-development partition, but re-cut the short-duration development samples by starting at randomly selected time points, rather than always starting from the beginning. Another change was that the re-cutting was done based on BBN's speech activity detection (SAD) system. As we did in Phase-1, we added the official "dry-run" test set to the development set. The total number of test samples in the development set was kept the same as before, about 7,120 samples for each condition. In the rest of the paper, we use "Dev" to denote this development set.

We also measured LID performance on one adjudicated version of the LID Phase 1 evaluation data (also called Dev2 within the RATS program), which includes 1,914, 1,782, 1,715, 1,340 samples for the "120s", "30s", "10s" and "3s" conditions, respectively. We use "Eval" to denote this evaluation set.

3. Baseline System Description

We first briefly review the LID system that BBN built for the phase-1 evaluation, which will serve as the baseline system for measuring improvements we report in this paper. The BBN LID system consisted of 4 major components, speech activity detection (SAD), feature extraction, I-vector estimation, and neural network (NN) LID classifier.

3.1. Speech Activity Detection (SAD)

The SAD system that BBN built for RATS was carried out in three main steps [2]. First, the input frame-level acoustic features are projected to a lower-dimensional space using heteroscedastic linear discriminant analysis (HLDA). Second, the reduced features are used to compute per-frame log likelihood scores with respect to speech and non-speech classes, each class being represented separately by a Gaussian mixture model (GMM). Third, the frame-level log likelihood scores are mapped to speech/non-speech classification decisions to produce final segmentation outputs. The mapping is done by computing the average per-frame log likelihood ratios, based on a sliding window, and comparing the smoothed likelihood ratios to a threshold. Frames with scores higher than the threshold are classified as speech, and the rest as non-speech.

3.2. Feature extraction

The commonly used feature extraction method for LID is the shifted delta cepstra (SDC) algorithm [3, 4], applied to MFCC or PLP feature vectors. The shifted delta can be considered as a special case of an approach that projects concatenated cepstral frames down to sub-space features through HLDA [5]. The most commonly used SDC scheme, [7-1-3-7], extracts features from a span of 39 successive frames, and essentially projects 39 frames of cepstral features down to a 56-dimensional sub-space using a fixed pre-determined projection matrix. In our LID experiments we estimated HLDA by maximizing the likelihood of a model that consisted of a single Gaussian for 5 classes represent the 5 target languages and one additional class representing all the non-target languages. We used the PLP cepstral feature for training our LID systems. During the Phase-1 system development we explored various HLDA configurations and found that the best configuration, in terms of the LID performance, was to project 11 frames, each frame including energy and the first 8 PLP coefficients, down to 60-dimensional sub-space features. We also found that this configuration produced better LID performance compared to the SDC method. Therefore, we continued to use the above HLDA configuration during our Phase-2 system development.

3.3. I-vector estimation

We estimated I-vectors according to the algorithm described in [6]. As done in [7], during the estimation we do the minimum divergence (MD) update [8, 9] to speed up the convergence. Besides the MD, we also revised the estimation with an "iterative" procedure that adapts the universal background model (UBM) to speakers at each iteration and then uses the adapted model to re-compute the posterior probabilities that are used for collecting the sufficient statistics for the I-vector estimation. We estimated 400-dimension I-vectors for the training of LID systems.

As mentioned before, all the LID training samples had at least 2 minutes of audio. So, in terms of audio length these training samples do not match the short-duration (30s, 10s and 3) test samples. To make the training better match the shorter duration test conditions, during the Phase-1 system development, we estimated the "20s" I-vectors for audio chunks of approximately 20 seconds of speech. The chunks were generated by grouping adjacent speech regions within each training audio file. We them combined these "20s" I-vectors with the regular I-vectors estimated on the entire audios to train the LID classifier. Our results showed (as expected) that adding the "20s" I-vectors to training improved the performance on the short-duration testing conditions.

3.4. NN LID classifiers

We compared three approaches for building the LID classifiers. One was the generative model approach [10] - training one single Gaussian model (SGM) for each language; The second one was the approach used in [6] that first applies linear discriminative analysis (LDA) on I-vectors, then carries out within-class covariance normalization (WCCN) and finally computes cosine distances as the final outputs. The third approach, developed at BBN, was to use neural networks as the LID classifier. We used the ICSI Quicknet NN tools [11] to train our NNs so as to map the I-vectors into language posteriors. We configured all our NNs with 3 layers (input, hidden, and output) with the input layer taking the I-vectors as inputs and the output layer generating posteriors for the 6 language classes. In all our classifier training we trained 4 NNs with the number of hidden nodes setting at 400, 500, 600 and 700, respectively, and took the arithmetic mean of the 4 NN output posteriors as the final outputs. Our comparison revealed that the NN classifier performed significantly better than the other two classifiers. For the phase-2 evaluation we kept using NNs as the LID classifier.

4. System Enhancements and Results

The RATS program sets higher accuracy targets for each successive phase evaluation. The LID Phase-2 targets are much more challenging, especially on the "10s" and "3s" test conditions. Therefore, for the Phase-2 system development, we focused on improving the LID system performance on the "10s" and "3s" test conditions. We will only report the performance measured on these two short-duration conditions in this paper. We measured the LID performance according to three metrics, EER, Cavg (computed the same way as in the NIST LRE evaluation), and one of the RATS Phase-2 operating points – miss rate at false alarm rate equal to 5% (we denote it as "Pmiss_5"). We report the EER, Cavg and Pmiss_5 scores as percentage numbers in this paper.

4.1. Modeling of short-duration conditions

As described before, the "20s" I-vectors estimated on chunks of 20s of speech improved our LID system performance on the short-duration test conditions in the Phase-1 evaluation. We would expect further gains on the "10s" and "3s" test conditions if we estimate I-vectors on speech chunks shorter than 20 seconds. Since our focus for the Phase-2 evaluation was the "10s" and "3s" test conditions, we resumed the work in this direction. We estimated I-vectors on chunks that have 10s and 3s of speech and added the I-vectors to the training of the NN LID classifiers, respectively. Performance of the systems trained with different sets of I-vector is shown in Table 1. In the "System" column, the "+20s", "+10s" and "+3s" represent the systems trained with the regular I-vectors (estimated on the entire audios) plus the "20s", "10s" and "3s" I-vectors, respectively. I also trained a LID system using the "3s" I-vectors only, which is denoted as "3s only" in the table, hoping it achieve the best performance on the "3s" test condition.

Table 1: LID performance (EER/Cavg/Pmiss_5), measured on the Dev set, for systems trained with different sets of I-vectors

System	"10s" condition	"3s" condition
+20s	10.18/10.75/18.18	23.10/24.44/53.99
+10s	9.79/10.23/16.29	21.85/23.41/50.40
+3s	9.52/10.20/16.10	21.50/22.50/49.21
3s only	9.70/10.49/16.15	21.89/23.00/ 49.75

As can be seen, adding I-vectors estimated on shorter speech chunks to the training kept improving the performance on the shorter test conditions. Adding the "3s" I-vectors to the training produced the best performance on both the "10s" and "3s" test conditions.

It is a little out of expectation to observe that the "3s only" system did not produce better performance on the "3s" testing condition than the "+3s" system did. Based on these results, we always added the "3s" I-vectors to our LID system training later.

4.2. Improving SAD

For the phase-2 evaluation we re-trained the GMM SAD system, used for the LID task, with changes in the following dimensions:

- trained on the full RATS SAD training set, about 1900 hours
- increased size of GMMs (from 512 to 2048 mixture components)
- Gaussian parameters estimated using maximum mutual information (MMI) criterion rather than maximum likelihood (ML)
- The size of the sliding window used in the last step changed from 81 frames to 41 frames.

All these changes helped improving the SAD performance. We observed that the last change (reducing the sliding window size from 81 to 41) improved the SAD performance on the short-duration test conditions greatly. We re-ran speech detection with this improved SAD on both the LID training and test data and re-trained the "+3s" system. We denote this re-trained system as "+P2-SAD" and its performance is shown in Table 2. Compared to the "+3s" system, the "+P2-SAD" produced small gains on the "10s" condition and large gains (about 10% relative) on the "3s" condition.

4.3. Using MLP phoneme posterior features

The primary focus of the RATS program is the challenge of highly degraded and/or noisy voice signals. The Patrol team had taken great efforts in extracting robust features. One of such efforts was the multi-stream (MS) feature developed in The John Hopkins University (JHU), one of the Patrol team members. The multi-stream feature is created by decomposing the narrow-band speech into 5 sub-bands (or streams), each of which covers about 3 critical bands along the frequency. To improve the speech information being extracted from individual streams, the filter bank of uniform bandwidth in the single stream, the frequency-domain linear prediction (FDLP) [12], is replaced with a multi-resolution filter bank that consists of two sets of filters of different bandwidths. A five-layer multi-layer perceptron (MLP) of size (390,1500,1500,1500,39) was trained in each stream to classify the narrow-band speech signal using the extended FDLP feature. Then the posterior probabilities from the five streams are integrated by a fusion MLP of size (125,1500,1500,1500,39) to produce the multi-stream feature.

For the RATS key word spotting (KWS) task, LDC released Levantine Arabic data with annotated transcriptions (about 380 hours of speech). On this data JHU trained MLPs with the MS feature to produce posterior probabilities for 39 phonemes. Then the posteriors were transformed into tandem feature by applying a logarithm followed by a Karhunen-Loève transform (KLT), which reduces the feature dimensionality from 39 to 25. For simplicity of notation, we use "MS-MLP" to denote these 25-dimensional features. These "MS-MLP" features helped improve our KWS system when they were used together with our regular PLP features. Hence, we tried to use them in our LID system training. We appended these 25-dimension "MS-MLP" features to the concatenated 11 PLP frames and then used HLDA to project them down to 60-dimensional sub-space features. We re-trained the "+P2-SAD" system with these new features. This re-trained system is denoted as "+MS-MLP" in Table 2. Comparing these two systems, we can see that the use of the "MS-MLP" features improved the LID performance significantly – about 20-30% relatively on both the "10s" and "30s" test conditions.

Table 2: LID performance (EER/Cavg/Pmiss_5), measured on the "Dev" set, of systems trained with the improved SAD, MLP phoneme posteriors and language-dependent UBMs

System	"10s" condition	"3s" condition
+20s	10.18/10.75/18.18	23.10/24.44/53.99
+3s	9.52/10.20/16.29	21.50/22.50/49.21
+P2-SAD	8.75/9.28/13.87	18.73/19.37/42.34
+MS-MLP	6.14/6.75/7.47	14.94/15.58/31.31
+LDUBM	4.84/5.32/4.69	12.64/13.31/24.76

Table 3: LID performance (EER/Cavg/Pmiss_5), measured on the "Eval" set, of systems trained with the improved SAD, MLP phoneme posteriors and language-dependent UBMs

Systems	"10s" condition	"3s" condition
+20s	14.13/16.88/26.78	20.59/24.01/46.87
+3s	12.45/16.21/22.58	19.52/22.60/43.86
+P2-SAD	12.66/15.80/22.21	17.90/20.66/40.55
+MS-MLP	10.09/13.63/16.82	15.40/17.32/28.68
+LDUBM	8.60/12.55/12.07	12.76/14.91/25.68

4.4. Training language dependent I-vectors

The universal background model (UBM) used for I-vector estimation is usually trained with data from all classes, such as speakers and languages, and then the estimated I-vectors tend

to represent variations from each class to the UBM. In our LID training there are 6 classes – 5 target languages and the non-target languages. If we train the UBM with data only from one language, the estimated I-vectors would tend to reflect the variations from the language to the other languages. We expected the I-vectors estimated with the language-dependent UBM (LDUBM) would bring complementary information to each other when we use them together.

Motivated by this idea, for each of the 6 language classes we train one set of I-vectors with its LDUBM. Then for each set of the I-vectors we trained one NN LID classifier. During the LID testing, we combined the posteriors from the 6 NN classifiers by taking the geometric mean as the final LID outputs. Each set of the I-vectors and its corresponding NN classifier were trained in the same way as the "+MS-MLP" system was trained. We use "+LDUBM" to denote this language-dependent system and its performance is shown in Table 2. Compared to the "+MS-MLP" system, this language-dependent system improved the performance by 15-20% relative on the two test conditions. There gains are significant.

The performance of all the above systems measured on the "Eval" test set is shown in Table 3. As can be seen, the incremental improvements observed on the "Dev" set are also observed on the "Eval" set.

5. Analysis on the MS-MLP features

As shown in Section 4, the largest improvements came from the use of the MS-MLP phoneme posterior features. As described before, the MLP used to generate the phoneme posterior features was trained with the RATS KWS Levantine Arabic data, which we thought possibly had some overlaps with the Levantine Arabic data included in our "Dev" set (it is non-trivial to identify the overlaps due to the inconsistence between the audio file names). To completely eliminate effects that could be caused by the possible data overlap, we excluded the Levantine Arabic language from both the LID training and test. Hence, all our following analysis experiments were set up to do LID on the 4 remaining target plus the 10 non-target languages.

We first re-trained the "+P2-SAD" and "+MS-MLP" systems with Levantine language excluded. They are denoted as "PLP" and "PLP+MS-MLP" in Table 4 (the 2nd and 4th rows), respectively. Comparing these two systems, we can see that the use of the "MS-MLP" feature produced similar gains as observed before ("+P2-SAD" vs. "+MS-MLP"). This confirms that the gains from the "MS-MLP" feature did not result from the data overlap on the Levantine language.

Next, we checked if the "MS-MLP" feature itself out-performs the regular PLP feature. We re-trained the "PLP" system by replacing the PLP with the "MS-MLP" feature. As we did for the PLP feature, we tried different HLDA configurations and found that the HLDA projecting 3 concatenated "MS-MLP" frames down to 60-dimension features produced the best performance. This system is denoted as "MS-MLP" in Table 4 (the 3rd row). Compared to the "PLP" system, it produced the similar performance. So the "MS-MLP" feature itself did not out-perform the PLP feature.

Lastly, we tried to verify if the MS feature helped generating better quality of phoneme posterior features, which resulted in the LID gains. As described before, the MS feature was only used in the training of the MLP that generated the phoneme posteriors. Therefore, we re-generated the phoneme posterior features, but replacing the MS feature with the single-stream FDLP feature and the regular PLP feature in the MLP training step, respectively. We denoted these two new sets of phoneme posterior features as "FDLP-MLP" and "PLP-MLP". We then re-trained the "MS-MLP" system with replacing the "MS-MLP" feature with the "FDLP-MLP" and "PLP-MLP" features, respectively. These two systems are denoted as "PLP+FDLP-MLP" and "PLP+PLP-MLP" in Table 4 (the 5th and 6th rows). As can be seen, these two systems produced similar performance as the "PLP+MS-PLP" system did. This indicates the MS front-end did not outperform other front-ends in the MLP posterior feature generation.

Table 4: LID performance (EER/Cavg), measured on the "Dev" set, for systems trained with different MLP phoneme posteriors

Features	"10s" condition	"3s" condition
PLP	8.99/9.38	17.82/18.37
MS-MLP	8.65/9.20	17.86/18.40
PLP + MS-MLP	7.01/7.25	15.47/15.90
PLP + FDLP-MLP	7.05/7.28	16.00/16.40
PLP + PLP-MLP	6.95/7.21	14.75/15.02

From the above analyses we can conclude that the MLP phoneme posterior features themselves do not outperform the regular PLP feature in the training of the LID system. But, together with the PLP feature, the MLP features always improved the LID performance, no matter what cepstral features are used to train the MLPs.

6. Conclusions

We have presented the work we conducted for improving our LID system for the RATS Phase-2 evaluation. We had improved our speech activity detection, feature extraction and LID modeling. The improved SAD helped improving our LID system performance by about 10% relative, the use of the MLP phoneme posterior features improved the LID performance by 20-30% relative, and the language-dependent system trained with the LDUBMs improved the performance by about 20% relative.

The largest improvement came from the use of the MLP posterior features. Our further analyses on the MLP feature revealed that the improvements were due to the complementary information the MLP features carry in and the choice of the front-end used to train the MLP was not crucial.

7. Acknowledgments

This work was supported by the DARPA 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.

8. References

[1] Matejka, P., et al., “Patrol Team Language Identification System for DARPA RATS P1 Evaluation”, ISCLP2012.

[2] Ng, T., Zhang, B., Nguyen, L., et al., “Developing a Speech Activity Detection System for the DARPA RATS Program”, in Interspeech 2012.

[3] Bielefeld, B., "Language identification using shifted delta cepstrum," In Fourteenth Annual Speech Research Symposium, 1994.

[4] Torres-Carrasquillo, P. A., Singer, E., et al., "Approaches to language identification using Gaussian mixture models and shifted delta cepstral features", Proc. ICSLP\'02, pp.90 -93 2002.

[5] Kumar, N. and Andreou, A.G., "Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition", Speech Communication, Vol. 26, No. 4, December, 1998.

[6] Dehak, N., Kenny, P., Dehak, R., Dumouchel, P. and Ouellet, P., “Front-end Factor Analysis for Speaker Verification”, IEEE Transactions on Speech and Audio Processing, Vol. 19, No.4, 2011.

[7] Glembek, O., Burget, L., Matejka, P., Karafiat, M. and Kenny, P., “Simplification and optimization of I-vector extraction”, pp. 4516-4519, ICASSP 2011.

[8] Brummer,N., “The EM algorithm and minimum divergence”. Agnitio Labs Technical Report (http://niko.brummer.googlepages.com/EMandMINDIV.pdf), Oct. 2009

[9] Kenny, P., “Joint factor analysis of speaker and session variability: Theory and algorithms”, Technical report CRIM-06/08-13, Montreal, CRIM, 2005.

[10] Martinez, D., Plchot, O., Burget, L., Glembek, O. and Matejkea, P., “Language Recognition in iVectors Space”, in the proceedings of Interspeech 2011.

[11] http://www.icsi.berkeley.edu/Speech/icsi-speech-tools.html

[12] Ganapathy, S., Thomas, S., and Hermansky, H., “Temporal envelope compensation for robust phoneme recognition using modulation spectrum”, J. Acoust. Soc. Amer. 128(6):3769--3780, 2010.