The MITLL NIST LRE 2011 Language Recognition System

Elliot Singer, Pedro Torres-Carrasquillo, Douglas Reynolds, Alan McCree, Fred Richardson, Najim Dehak, and Doug Sturim*

Massachusetts Institute of Technology
Lincoln Laboratory
{es,ptorres,dar,mccree,frichard,sturim}@ll.mit.edu
*Computer Science and Artificial Intelligence Laboratory
najim@csail.mit.edu

Abstract

This paper presents a description of the MIT Lincoln Laboratory (MITLL) language recognition system developed for the NIST 2011 Language Recognition Evaluation (LRE). The submitted system consisted of a fusion of four core classifiers, three based on spectral similarity and one based on tokenization. Additional system improvements were achieved following the submission deadline. In a major departure from previous evaluations, the 2011 LRE task focused on closed-set pairwise performance so as to emphasize a system's ability to distinguish confusable language pairs. Results are presented for the 24-language confusable pair task at test utterance durations of 30, 10, and 3 seconds. Results are also shown using the standard detection metrics (DET, minDCF) and it is demonstrated the previous metrics adequately cover difficult pair performance. On the 30 s 24-language confusable pair task, the submitted and post-evaluation systems achieved average costs of 0.079 and 0.070 and standard detection costs of 0.038 and 0.033.

1. Introduction and Task

The National Institute of Science and Technology (NIST) has conducted formal evaluations of language detection algorithms since 1994. The emphasis in NIST's 2011 Language Recognition Evaluation (LRE) was the performance of submitted systems on the most confusable language pairs, where "most confusable" was system dependent rather than predefined. The task in LRE11 was to decide which of two languages was spoken in a speech segment for a given pair of languages. The 24 languages in LRE11 were themselves selected from within language clusters so as to maximize confusions within the recognizers. Languages appearing in previous evaluations were: Bengali, Dari, English-American, English-Indian, Farsi/Persian, Hindi, Mandarin, Pashto, Russian, Spanish, Tamil, Thai, Turkish, Ukrainian, and Urdu. New languages for LRE11 were Arabic-Iraqi, Arabic-Levantine, Arabic-Maghrebi, Arabic-MSA, Czech, Lao, Panjabi, Polish, and Slovak. (Although the targets should properly be referred to as classes, this paper will follow NIST usage and employ the term "languages.") As in 2009, evaluation utterances were drawn by NIST from both conversational telephone speech recordings collected specifically for NIST and "found" segments drawn from narrowband segments identified within foreign language broadcast sources, such as the Voice of America.

The metric used to evaluate performers is based on the language pair cost function given by

$$C\left(L1, L2\right) = 0.5 * (P_{Miss}(L1) + P_{Miss}(L2)) \quad (1)$$

From the submitted scores for the 276 possible pairs, the N=24 pairs with the highest minimum costs were selected for the 30 s segments, where the minimum costs were determined by varying decision thresholds. The overall performance measure was computed as the average across the N worst performing pairs using the hard decisions supplied by the participants to compute the language pair costs. More details are available in the NIST LRE11 Evaluation Plan [1]. In addition to this new metric, this paper will also show results using the standard detection metrics (Detection Error Tradeoff curves and decision cost functions) obtained from pooling scores from the collection of 24 language detection systems. It is important to maintain this standard metric since it allows comparison to previous LREs and has a direct interpretation of performance of an actual working system (the pair-wise metric is interesting as a diagnostic, but merely reflects the average performance of a set of two-language detectors).

The organization of this paper is as follows: Section 2 describes the development data used for the MITLL submission. Section 3 describes the core classifiers and score fusion method employed in the submitted and subsequently improved post-evaluation systems. Section 4 presents system performance on the NIST 2011 LRE task and a discussion of results, and Section 5 presents conclusions.

2. Development Data

Data for training and development testing was obtained by augmenting existing LRE09 resources with data from additional sources to cover shortfalls in the new and/or under-represented languages in LRE11. The LRE09 resources consisted of

- Telephone data from previous LREs (1996, 2003, 2005, 2007, 2009): CallFriend, CallHome, Mixer, OHSU, and OGI-22 collections.
- Narrowband segments from VOA broadcasts.

The new data sources consisted of

- NIST 2011 development data (Telephone and narrowband broadcast segments).

This work was sponsored by the Department of Defense under Air Force contract F19628-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

- Narrowband segments from Radio Free Asia, Radio Free Europe, and GALE broadcasts.
- Arabic corpora from LDC and Appen (telephone and some interview data)

No effort was made to remove duplicate speakers or to create gender balancing in the development corpus. A breakdown of the amount of data available for training is shown in Table 1. Post-evaluation experiments determined that the interview data source was not useful in training and could be removed. For development testing and fusion/calibration training we created 30 s, 10 s, and 3 s test sets comprising segments from previous LREs and segments extracted from longer files. We had 100-200 test segments per duration per source (CTS or BNBS) per language when available.

Table 1: Hours of speech and number of segments per language in initial training corpus

	CTS		BNBS		INT	
	hrs	#segs	hrs	#segs	hrs	#segs
Arabic-Iraqi	16.5	289	13.2	800	18.7	483
Arabic-Levantine	40.8	732	0	0	38.9	976
Arabic-Maghrebi	7.7	100	0	0	0	0
Arabic-MSA	0	0	108.3	7326	0	0
Bengali	4.1	55	0	0	0	0
Czech	13.5	100	0	0	0	0
Dari	0	0	15.1	800	0	0
English-American	78.4	633	0	0	0	0
English-Indian	7.6	359	0	0	0	0
Farsi	35.6	160	22.2	800	0	0
Hindi	34.6	164	9	516	0	0
Lao	0	0	1	100	0	0
Mandarin	119.8	1020	11	800	0	0
Pashto	0	0	13.4	800	0	0
Polish	13.1	100	0	0	0	0
Punjabi	9.3	100	0	0	0	0
Russian	24.7	374	20.1	800	0	0
Slovak	12.7	100	0	0	0	0
Spanish	99.4	625	13.7	800	0	0
Tamil	15.7	80	0	0	0	0
Thai	1.6	20	8.7	661	0	0
Turkish	0	0	16.6	747	0	0
Ukrainian	0	0	6.1	249	0	0
Urdu	1.7	22	14.3	800	0	0

There were five languages (Arabic-Maghrebi, Czech, Punjabi, Polish, and Slovak) for which insufficient data was available to create well populated train, dev, and test partitions. Consequently, a cross-validation scheme was employed in which the data for these five languages was partitioned into five non-overlapping folds, with each fold using 80% of the segments for training and 20% for testing. The remaining 19 well provisioned languages used a fixed train/test data partition across the folds. Furthermore, the backend was trained by cross-validation over the aggregated scores from the five folds.

Late in the development cycle, data was obtained from the Special Broadcast Services (SBS) in Australia for 13 of the languages, including the five low-resource languages. A breakdown of the amount of data available in the extra training set is shown Table 2. Only one system (SVM-GSV) made use of the extra training data in the primary submission. Also, the availability of the extra data made it unnecessary to employ cross-validation for training the backend. Results with extra training and without cross-validation were obtained after the submission deadline and are discussed below.

Table 2: Hours of speech and number of segments per language in extra training corpus

	CTS		BNBS	
	hrs	#segs	hrs	#segs
Begnali	0	0	1.7	157
Czech	0	0	3.1	330
Dari	0	0	1.2	65
English-American	15.0	174	0	0
English-Indian	7.4	88	0	0
Hindi	6.1	72	3.8	259
Lao	0	0	7.2	975
Polish	0	0	2.8	165
Punjabi	0	0	2.2	183
Slovak	0	0	1.2	88
Tamil	8.4	92	0.3	34
Ukrainian	0	0	2.3	105
Urdu	0	0	2.9	137

3. Classifiers

As in previous LREs, the Lincoln language recognition system consisted of the fusion of spectral and token based classifiers. For LRE11 we introduced two i-vector system. In this section we briefly describe the core classifiers and the fusion/calibration system.

3.1. Spectral Classifiers

Four systems were used for spectral based recognition: two discriminately trained classifiers (GMM-MMI, SVM-GSV) and two generative i-vector systems.

3.1.1. Features

The spectral based systems used a common set of features and processing. The main processing chain consisted of:

- Speech windowing of 20 ms length and 10 ms shift. The windowed signal mean is subtracted and a low energy dither is added to the signal to avoid runs of digital zeros.
- Mel-scale filterbank analysis over the band 0-4000 Hz producing 24 log-filterbank energies. Per-file vocal tract length normalization (VTLN) warps are applied to the filterbank centers and the first filterbank energy is removed to reject out-of-band signaling. RASTA filtering is then applied to the log-energy filterbank trajectories.

- Conversion to cepstral coefficients via DCT. The first seven cepstral coefficients (c0-c6) are retained.
- Shifted Delta Cepstra (SDC) features are extracted using the conventional 7-1-3-7 scheme. The static cepstra are appended producing a 56-dimensional feature vector.
- Non-speech frames are gated out using speech activity detection marks derived from a GMM-based speech/non-speech detector.
- Each feature element is normalized to zero mean, unit variance by subtracting the mean and dividing by the standard deviation computed from either a 3 s window of speech frames or from the entire file.
- The features are then compensated using feature domain Nuisance Attribute Projection (fNAP) inspired by the work in [2].

3.1.2. GMM-MMI

The GMM-MMI system [3] used for the 2011 LRE is similar to the system that was used by Lincoln in recent evaluations [4]. For each language, weights, means and variances for a 2048 order Gaussian mixture model are trained to maximize the likelihood on its training data starting from a common, language-independent background model. Next, means and variances for the set of models are jointly updated to optimize the maximum mutual information criterion on all the training data.

For LRE11, the MMI training time was significantly decreased by approximating the log-likelihood computation using a fixed set of sufficient statistics derived from a common background model. With this decrease in computation, the number of training iterations was increased from 20 to 40, which improved performance. For recognition, standard frame-by-frame scoring was used. Although experiments on development data found that there was no decrease in accuracy in using sufficient statistics training, some decrease in accuracy on the evaluation data was observed.

3.1.3. SVM-GSV

The SVM-GSV system [5] was also similar to that used in previous evaluations. Maximum *a posteriori* (MAP) adapted GMMs derived for each file were used as input observations to train a Support Vector Machine (SVM) classifier. The configuration for this system is as follows:

- GMMs for each segment are adapted from a 1024 mixture UBM.
- GMM means and variances are adapted with a relevance factor of 0.001.
- SVM input vectors are the stacked mean and variance supervectors from the adapted GMMs.
- SVM training is accomplished using a KL divergence-based kernel and a one-versus-rest strategy.
- SVM models are converted ("pushed") into GMMs by normalizing the support vectors by the sum of the support vector weights resulting in two models (target and non-target) per language. Earlier work had shown that the conversion to GMMs improved language recognition performance.
- Scoring is performed by computing the log likelihood ratio between a language's pushed models. The SVM-GSV system used features processed with fNAP and VTLN compensation.

The submitted SVM-GSV system contained a bug that resulted in a mismatch between the scores generated for development and the NIST evaluation set. This discrepancy was corrected in the post-evaluation version of the system.

3.1.4. I-vector systems

The i-vector framework has become very popular in speaker recognition and language identification [6][7]. This approach, based on factor analysis, is an elegant way to capture the majority of useful variabilities between GMM supervectors in low dimensional space. In the i-vector formulation, each speech utterance has a corresponding GMM supervector that is assumed to be generated as follows:

$$M = m + Tw \tag{2}$$

where *m* is the speaker independent and channel independent supervector (which can be taken to be the UBM supervector), *T* is a rectangular matrix of low rank, and *w* is a random vector having a prior standard normal distribution *N*(0, I). Analysis of this generation model, along with several assumptions, leads to a maximum *a posteriori* estimator of the i-vector, *w*, using the Baum-Welch statistics for a given utterance. The *T* matrix is estimated using statistics from development data via iterative EM training or principle component analysis (PCA). We developed two i-vector systems for LRE11.

3.1.4.1 First I-vector system (IVEC1)

The first i-vector system (IVEC1) uses linear discriminant analysis (LDA) and cosine scoring following the setup used for speaker verification [6]. LDA consists of finding the basis that maximizes the between-language variability while minimizing the intra-language variability. The LDA axes are then defined by a projection matrix *A*, which is trained using the training data from all languages. In speaker recognition, within-class covariance normalization (WCCN) is also used, but it was found in development experiments that it provided no performance gain over LDA for the IVEC1 system.

It is well known that cosine scoring is based only on i-vector directions, discarding vector lengths. Using directional statistics, we can formalize this further by assuming the languages are modeled by a Von-Mises-Fisher distribution (which is the analog of a Gaussian distribution on the unit sphere). The Von-Mises-Fisher distribution operates on unit norm vectors and is defined as

$$f_d(\hat{w} \mid m, \kappa) = C_d(\kappa)\exp(\kappa m^T \hat{w}) \tag{3}$$

where *m* is the mean, κ is the spread parameter, *d* is the vector dimension, and *C()* is a normalization constant. The maximum likelihood estimate of the mean for a language is given as

$$m_l = \frac{\sum_{j=1}^{N_l} \hat{w}_j}{\left\| \sum_{j=1}^{N_l} \hat{w}_j \right\|} \tag{4}$$

where, N_l is the number of utterances for each language *l* and the unit norm LDA i-vectors are

$$\hat{w} = \frac{A^T w}{\left\| A^T w \right\|} \tag{5}$$

The spread parameter κ can also be estimated, but during development experiments we found that using a fixed spread parameter for all languages worked best. A constant κ has no effect on the final decision and thus its exact value is irrelevant. We also note that we experimented with mixtures of Von-Mises-Fisher distributions per language, but found that a single distribution always performed best.

Scoring is then simply the log-likelihood of a test i-vector after LDA and unit normalization. Discarding constants, this is merely a dot product with the language model mean

$$score_l = \hat{w}_{test}{}^T m_l \tag{6}$$

Note that this scoring differs from cosine scoring only in that the language mean is estimated using unit normalized i-vectors.

The IVEC1 system used a 2048 order GMM UBM and i-vectors of dimension 600. In the submission to NIST, VTLN and fNAP compensation was not applied to the raw features prior to i-vector extraction. Following the submission deadline, an updated version of this system was created that used the extended training set and VTLN+fNAP feature compensation, and eliminated a bug in the cosine scoring.

3.1.4.2 Second I-vector system (IVEC2)

The second i-vector system (IVEC2) used Gaussian scoring in the i-vector space, as in [8]. Training a language model is then simply a matter of computing the mean of the training vectors, and testing involves a Gaussian likelihood evaluation using the shared within-class covariance:

$$m_l = \frac{1}{N_l} \sum_{j=1}^{N_l} w_j$$

$$\Sigma_w = \frac{1}{L} \sum_{l=1}^{L} \frac{1}{N_l} \sum_{i=1}^{N_l} (w_i^l - m_l)(w_i^l - m_l)^T \tag{7}$$

Raw scores were normalized to a UBM in the Gaussian space, i.e. a mean of zero:

$$score_l = w_{test}{}^T \Sigma_w^{-1} m_l - \frac{1}{2} m_l{}^T \Sigma_w^{-1} m_l \tag{8}$$

During development, experiments were conducted with LDA and discriminative training of the Gaussians using MMI, but no significant performance improvements from these enhancements were obtained so they were not used in the IVEC2 system. The initial version of this system did not use VTLN or fNAP and also had a defective SAD algorithm, and so was not used in the primary system submission. Following the submission deadline, an updated version of this system was created that used the same SAD algorithm and fNAP+VTLN feature compensation as used by the other spectral systems.

In addition to modeling and scoring, IVEC1 and IVEC2 also differ in how the T matrix was estimated. IVEC1 used iterative EM estimation with minimum divergence steps and updates to the UBM covariance matrix, while IVEC2 used PCA estimation.

3.2. Token Classifier

3.2.1. TRAPS/NN Tokenizer

Tokenization of speech was performed using a system based on the Brno University (BUT) TRAPS/NN design [9]. The tokenizer used three-state left-to-right HMMs with a null grammar and consisted of two key components for generating HMM state posteriors: TRAPS, which are long time-span time-frequency features, and feedforward artificial neural nets. The tokenizer was trained on approximately 10 hours of English Switchboard2 Cell data. The data was phonetically segmented using an STT system, and the resulting system used 49 monophones including silence.

3.2.2. SVM N-gram Language Modeling

The SVM token system [10] used a bag-of-*N*-grams. For a sequence of tokens, (joint) probabilities of the unique *N*-grams on a per conversation basis are calculated and weighted by a token dependent scale factor D_j. The general weighted probability vector is then combined to form a kernel between two token sequences. For two token sequences, *W* and *V*, the kernel is

$$K(W,V) = \sum_j D_j^2 \, p(\hat{w}_j, w_j \,|\, W) \, p(\hat{w}_j, w_j \,|\, V) \tag{9}$$

SVM training and scoring require a method of kernel evaluation between two objects that produces positive definite kernel matrices (the Mercer condition). We use the package SVMTorch and a one-versus-rest strategy for training.

A 4-gram system was used in the 2011 LRE (4GR-SVM). Using the full set of 4-grams from the English tokenizer is impracticable due to the large number of 4-grams (as many as 49^4). Instead, a subset of the 4-grams was selected using the alternating filter-wrapper feature selection method [11]. This approach starts by selecting a fixed number of 3-grams that have the highest and lowest SVM weights from the SVM model, and extending them at front or back by each token in the tokenizer's lexicon. The new subset of 4-grams is then used to train the SVM.

3.3. Fusion/Calibration

The backend processing consisted of per-system calibration and duration normalization followed by linear fusion with a zero offset. Calibration used a discriminatively-trained (MMI) Gaussian with shared covariance for each system, followed by a multiclass logistic regression across systems for the final score. The backend was trained on the development data set and then applied to the evaluation data.

Two types of backends were developed: a single closed-set multiclass backend and pairwise backends. The primary submission used the single closed-set backend followed by a simple application of Bayes' rule to extract the pair-wise likelihood ratios. For backend identification likelihoods C_i, identification posteriors are given by

$$P_{ID}(C_i \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid C_i) P(C_i)}{\sum_{j=1}^{M} p(\mathbf{x} \mid C_j) P(C_j)} \qquad (10)$$

and pairwise likelihood ratios by

$$LR_{PAIR(m,n)}(C_m \mid \mathbf{x}) = \frac{P_{ID}(C_m \mid \mathbf{x})}{P_{ID}(C_n \mid \mathbf{x})} \frac{P(C_n)}{P(C_m)} \qquad (11)$$

Experiments were run on the development and evaluation data for LRE09 using the pairwise backends, but performance was inferior to that obtained using the multiclass backend. Therefore, the submitted system used the single multiclass backend and applied Equation 11 to compute pairwise scores.

4. Results and Discussion

This section presents the results for the primary system submitted for NIST LRE11 and the final post-evaluation system which contained a number of bug fixes and enhancements.

4.1. Official NIST Submission

Results for the submitted system, along with a breakout by classifier, are shown in Figure 1. The leftmost two bars for each system show the new NIST LRE11 metric – the minimum average pair detection cost (minAPD) of the 24 worst pairs and the actual average pair detection cost (actAPD) of those pairs. For individual classifiers, the actAPD ranges from 0.114 (IVEC1) to 0.134 (GMM-MMI), and 0.079 for fusion. Note that due to the nature of the new metric, the language pairs used for each system's results may be different since the worst pairs are system dependent. It is apparent that there are calibration issues for these pairs as evidenced by the disparities between the actAPD and minAPD values, which are likely due to uncompensated mismatches between the development and evaluation data.

For reference, the rightmost two bars show performance based on the standard multiclass detection performance metric used by NIST for evaluations prior to LRE11. With the standard metric, performance for systems is computed over the same languages and data. The new and standard metrics track closely, with the actual DCF for the primary submission (FUSE) being 0.0382.

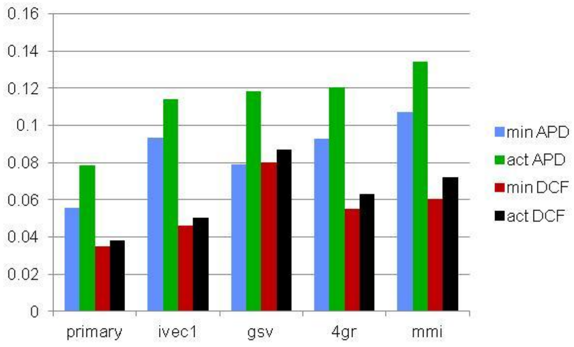

Figure 1: Performance of submitted system ("primary') and its component classifiers on the 30 s LRE11 task.

Figure 2 shows the actual DCF for the worst pairs of the submitted system as ordered by Equation 1. Not surprisingly, the pairs with the highest DCF tend to be languages spoken in countries that are geographically and culturally quite close (e.g., Lao in Laos and Thai in Thailand). Somewhat unexpected is the fact that Pashto, Punjabi, and Bengali are confused across a wide range of other languages, behavior that was common to many systems submitted to NIST for LRE11 [12].

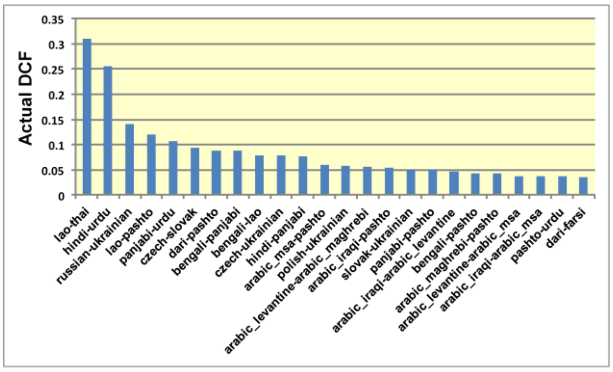

Figure 2: Actual DCF (Eq. 1) for worst pairs in the primary submission.

Due to time pressures and system implementation issues, the submission to NIST was handicapped in several ways, all of which were corrected in the subsequent post-evaluation system:

- A complete IVEC2 i-vector system was not available in time for submission. Consequently, the submitted system was a fusion of the IVEC1, GSV, 4GR-SVM, and GMM-MMI core classifiers.
- The GSV system contained a bug due a mismatch in models used for scoring the development and evaluation sets.
- The IVEC1 i-vector system inadvertently excluded test vector normalization. The system also did not include VTLN+fNAP feature compensation.
- None of the systems, except for SVM-GSV, was trained using the extra training data.
- All systems used cross-validation to generate backend training data.

4.2. Post-evaluation System

This section presents results for the final post-evaluation system, which was not completed in time for submission to NIST for LRE11. The system has the following characteristics:

- All classifiers were trained with the full complement of training data.
- All spectral classifiers used the full feature extraction and normalization sequence outlined in Section 3.1.1.
- Bugs that were identified in the primary submission were corrected.
- With the addition of IVEC2, a total of five classifiers were included.
- The backend was trained using the full development set rather than with cross-validation.

Results for the post-evaluation system, along with a breakout by classifier, are shown in Figure 3. In general, performance has improved with the inclusion of the extra training data, the elimination of cross-validation, and the removal of known bugs. The leftmost two bars for each system show the NIST LRE11 metric (actAPD) based on the 24 worst pairs as determined by the minimum closed set pairwise costs, and the average minimum cost of those pairs. For individual classifiers, the actAPD now ranges from 0.089 (IVEC1) to 0.125 (SVM-4GR), and 0.070 for 5-way fusion. Again, calibration issues are apparent in the disparities between the actAPD and minAPD values. For reference, the rightmost two bars show performance based on NIST's standard metric based on multiclass detection performance, with the two metrics again tracking well. The actual DCF for the post-evaluation fusion system is 0.033. DET plots for the post-evaluation fusion system are shown in Figure 4.

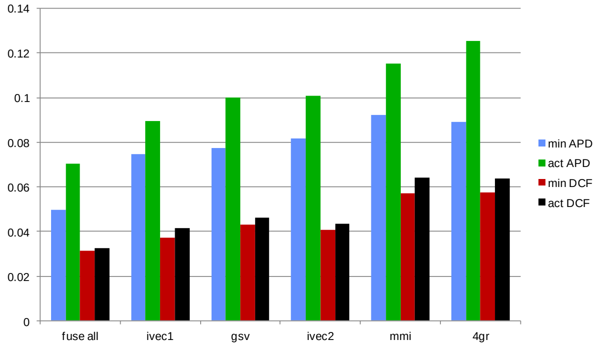

Figure 3: Performance of the final post-evaluation system and its component classifiers on the 30 s LRE11 task.

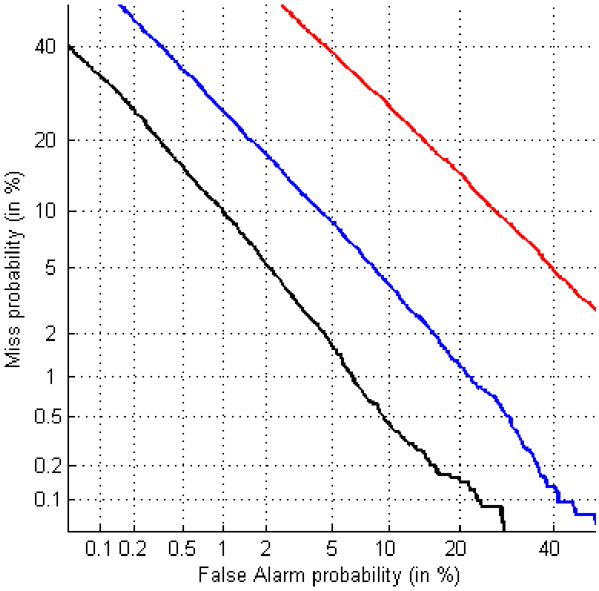

Figure 4: DET plots for the fused post-evaluation system for test durations 30 s (black), 10 s (blue), and 3 s (red).

4.3. Comparison of Multiclass and Pairwise Fusion

As discussed in Section 3.3, two types of backends were evaluated for generating pairwise scores, the first being a single closed-set multiclass backend from which scores were derived via Bayes' rule, and the second a set of *N*(N-1)/2* =276 individual pairwise closed-set backends. Results shown in Figure 5 indicate that multiclass class training provides better performance, as well as being much simpler to manage.

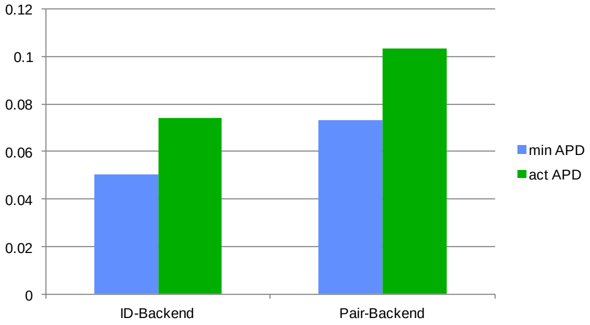

Figure 5: Performance of fusion using a jointly trained multiclass backend and pair-specific backends.

4.4. Confusion Matrix

One of the aims of the new metric introduced for LRE11 was to focus on confusable pairs of languages. As shown in the results in previous sections, the standard metrics are adequate for measuring relative performance of systems in a consistent manner with more difficult language pairs present. Here we show that the information contained in the multiclass confusion matrix adequately focuses on confusable language sets. A bubble plot[1] of the confusion matrix from the final MITLL system is shown in Figure 6. From this plot it is easy to find the "hot spots" of confusable languages. In fact the confusion plot shows more information than artificial pair-wise scores since it is easier to see input languages which cause problems over multiple models or models which false alarm over multiple input languages.

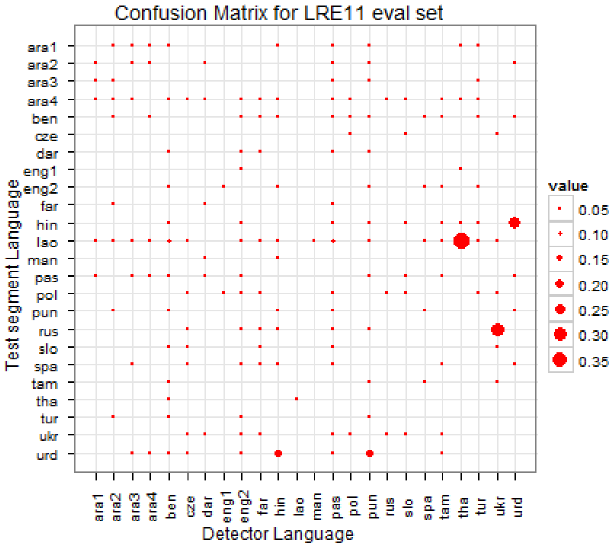

Figure 6: Bubble plot of confusion matrix from final MITLL system.

[1] Thanks to Dave Farris for providing the code for producing the bubble plot from a confusion matrix.

4.5. Classifier Fusions

In this section we report on different system combinations using core systems from the post-evaluation core classifiers. Table 3 shows multiclass pooled minimum and actual decision cost function values for some interesting combinations. The first line is for fusion of all classifiers in the post-evaluation system. The second line shows that the best classifier combination, which used the n-gram and both i-vector classifiers. In the combination sweep, it was found that the top combinations always included the n-gram classifier, which is consistent with previous experience that language recognition systems benefit by combining spectral and phonotactic based information. The third line shows results when using a single i-vector classifier with the other classifiers. In the fourth line we show the best combination not using the n-gram classifier. Finally, rows five and six show results for the best pair and single combinations.

While these results indicate limited value in fusion of the GMM-MMI and SVM-GSV systems, this was not observed during development and further experimentation is required to verify this conclusion.

Table 3: Standard multi-class detection cost function results for different system combinations

	min DCF	**act DCF**
fuse 5	0.0312	0.0327
4gr+ivec1+ivec2	0.0286	0.0301
4gr+gsv+mmi+ivec1	0.0312	0.0331
gsv+ivec1+ivec2	0.0336	0.0361
4gr+ivec1	0.0286	0.0307
ivec1	0.0371	0.0415

5. Conclusion

In this paper we have described the MITLL submission to the 2011 NIST Language Recognition Evaluation. The submission consisted of spectral, token, generative, and discriminative classifiers fused using a joint backend. The i-vector systems introduced this year provided very good performance alone and fused well with other systems. We found that using standard pooled class detection metrics for system optimization and a single multiclass backend was sufficient for addressing the new pair metric introduced in LRE11.

For a historical perspective, Figure 7 shows performance (EER %) of Lincoln Laboratory systems on the NIST LRE test data beginning in 1996 and continuing to LRE11. It is apparent that the task has increased in difficulty since LRE09, most likely due to the deliberate choice of confusable languages and perhaps also because of the wider range of broadcast sources used for the evaluation utterances.

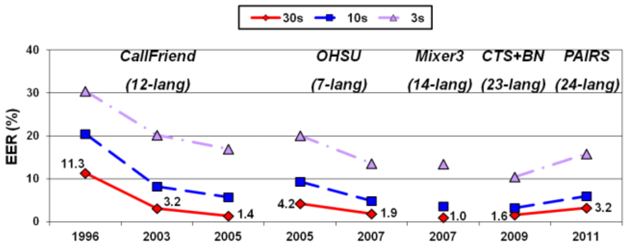

Figure 7: Performance trends of MITLL language recognition systems on NIST evaluation corpora at three durations. Dates on the horizontal axis indicate the system vintage.

6. References

[1] 2011 Language Recognition Evaluation, http://www.nist.gov/itl/iad/mig/lre11.cfm

[2] C. Vair, D. Colibro, F. Castaldo, E. Dalmasso and P. Laface, "Channel Factors Compensation in Model and Feature Domain for Speaker Recognition", *IEEE Odyssey 2006*, San Juan , PR.

[3] L. Burget, P. Matejka and J. Cernocky, "Discriminative Training Techniques for Acoustic Language Identification",in *Proc. ICASSP*, 2006, pp. 209-212, Toulouse, France.

[4] P. A. Torres-Carrasquillo, E. Singer, T. Gleason, A. McCree, D. A. Reynolds, F. Richardson and Douglas Sturim, "The MITLL NIST LRE 2009 Language Recognition System", in *Proc. ICASSP*, 2010, pp. 4994-4997, Dallas, TX, USA.

[5] W. M. Campbell, D. E. Sturim, D. A. Reynolds, A. Solomonoff, "SVM Based Speaker Verification Using A GMM Supervector Kernel and NAP Variability Compensation", in *Proc. ICASSP*, 2006, pp. 97-100, Toulouse, France.

[6] N. Dehak, P. Kenny, R. Dehak, P. Ouellet, and P. Dumouchel, "Front end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 4, pp. 788–798, May 2011.

[7] N. Dehak, P. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language recognition via ivectors and dimensionality reduction," in *Proc. Interspeech*, 2011, pp. 857–860.

[8] A. McCree, D. Sturim, and D. Reynolds, "A New Perspective on GMM Subspace Compensation Based on PPCA and Wiener Filtering," *Proc. Interspeech*, 2011, pp. 145-148.

[9] P. Matejka, P. Schwarz, J. Cernocky and P. Chytil, "Phonotactic Language Identification using High Quality Phoneme Recognition", in *Proc EuroSpeech 2005*, pp. 2237-2240, Lisbon, Portugal.

[10] F. Richardson and W. M. Campbell, "Language recognition with discriminative keyword selection", in *Proc. ICASSP*, 2008, pp. 4145-4148, Las Vegas,NV, USA.

[11] Campbell, W. M., Richardson, F., "Discriminative Keyword Selection Using Support Vector Machines", in *Neural Information Processing Systems*, Vancouver, BC Canada. Dec. 3-8, 2007.

[12] A. Martin, personal communication.