Odyssey 2018 The Speaker and Language Recognition Workshop
26-29 June 2018, Les Sables d'Olonne, France

Approaches to Multi-domain Language Recognition

Mitchell McLaren[1], *Mahesh Kumar Nandwana*[1], *Diego Castan*[1], *Luciana Ferrer*[2]

[1]Speech Technology and Research Laboratory, SRI International, California, USA
[2] Instituto de Investigación en Ciencias de la Computación, UBA-CONICET, Argentina
{mitchell.mclaren, mahesh.nandwana, diego.castan}@.sri.com, lferrer@dc.uba.ar

Abstract

Multi-domain language recognition involves the application of a language identification (LID) system to identify languages in more than one domain. This problem was the focus of the recent NIST LRE 2017, and this article presents the findings from the SRI team during system development for the evaluation. Approaches found to provide robustness in multi-domain LID include a domain-and-language-weighted Gaussian backend classifier, duration-aware calibration, and a source normalized multi-resolution neural network backend. The recently developed speaker embeddings technology is also applied to the task of language recognition, showing great potential for future LID research.

1. Introduction

The NIST LRE series of evaluation has typically focused on a single domain during evaluation of unseen data. In 2017, however, the LRE considered the evaluation of both telephone speech and speech audio extracted from videos [1] and the recognition of 14 languages, which were categorized into five closely-related language groups. This problem motivated the development of novel approaches to language identification (LID) to cope with multiple domains at test time. For LRE'17, a large pool of out-of-domain data was supplied for system training along with smaller partitions of data from both target domains of the evaluation. It was this data partitioning that forced teams to consider how best to use each data source with the knowledge that the out-of-domain source would not be observed during evaluation. The evaluation considered both the *fixed* scenario, in which teams could not use data beyond that supplied for that condition, as well as an *open* condition in which any data (public or proprietary) could be used to improve the system.

In this article, we investigate the novel aspects of the systems developed by the SRI team for LRE'17 to cope with the new context of multi-domain LID. The systems relied on three types of vectors to represent speech from an audio recording: bottleneck i-vectors, simple senone posterior zero-order statistics processed with probabilistic principal component analysis (PPCA) [2], and language embeddings based on the application of a speaker embedding framework [3, 4] to LID. Two backend scoring options were used: a language-and-domain balanced Gaussian Backend (GB) and a source-normalized multi-resolution neural network (MRNN). The novel aspects of each of these backends were shown to provide significant robustness in the context of multi-domain LID. The complementarity offered from multiple systems was explored based on a fusion of up to four subsystems using multi-class linear regression and duration-aware calibration, which helped to account for varying durations in the evaluation's test segments.

Table 1: The 14 languages and corresponding clusters defined for the NIST LRE'17.

Cluster	Languages
Arabic	Iraqi, Levantine, Maghrebi, Egyptian
Chinese	Mandarin, Min Nan
English	British, General American
Slavic	Polish, Russian
Spanish/Iberian	Caribbean, European, Latin American Continental, Brazilian Portuguese

We first describe the development data setup, followed by a description of the features, vector representations, backend classifiers, and fusion process. Results and analysis are then presented focusing on the aspects that added robustness to the task of multi-domain LID.

2. System Training and Development Data

The LRE'17 dataset included 14 languages from five clusters of closely-related languages (or dialects). These languages are provided in Table 1. NIST provided several LDC databases for use in the LRE'17, of which all but the BABEL dataset were allowed for use in the *fixed* condition. These datasets can be broken down as follows:

- **LRE17Train:** 16,202 audio files sourced from previous LRE's, Fisher, and Switchboard datasets. Transcriptions were provided for Fisher and Switchboard datasets to allow for training of phonetically based DNNs. Additionally, 17 languages from the BABEL dataset were provided for use in the *open* training condition.

10.21437/Odyssey.2018-13

- **LRE17Dev:** 3,660 audio files (from 1806 unique audio files) sourced from the target domains of the evaluation. These included 873 files containing video speech audio from the VAST dataset and 933 files from a single telephone conversation side of the MLS14 database [1]. The 933 files each contained 30 seconds of speech, and each had a 3 and 10 sec segment produced by NIST; however, the file of origin for these segments was not provided. The duration of the VAST files varied from a few seconds of speech to several minutes.
- **LRE17Eval:** The evaluation set consisted of 25,451 audio cuts sourced from 10,831 original files of the MLS14 and VAST datasets.

The Fisher and Switchboard datasets were used to train senone DNNs used in both the *fixed* and *open* conditions, while the BABEL data provided for LRE'17 was used to train additional DNNs for use in the *open* condition only. For training system components (SAD, i-vector sub-spaces, etc.), we defined our *fixed* system training data as the full 16,202 files from the provided LRE'17 train set as well as the development set. The development set was, however, divided into two partitions to perform cross-validation. These sets were defined by determining which 3, 10, or 30 sec MLS14 cuts were from the same original audio file. This was done using automatic analysis of the cross-correlation of energy from each 3 and 10 sec signal with the longer 30 sec samples. It was our understanding that the VAST files were of full duration and so this "nesting" detection process was not performed on the VAST files.

In the systems developed for the *open* training condition, we leveraged two additional sources of data:

- **LRE15eval:** We used 5632 files from the MLS14 part of the LRE'15 evaluation dataset. These segments were found by first locating the longest file for each unique LDC ID in the set, then removing approx. 1000 files deemed to overlap heavily with the 933 MLS14 30 sec files from the LRE'17 development set. We removed this presumed overlap to ensure that files from LRE'15 would not positively bias cross-validation results.
- **Languages in the Wild (LITW):** This data was a collection of 2180 audio files sourced from open-source videos and represented 29 languages. The collection, currently not publicly available, was largely automatically sourced, in contrast to the similar Speakers in the Wild (SITW) [5]. The audio length varied from less than a minute to over two hours in duration. As dialect information was not available, we excluded from this set any language that was closely related to the 14 LRE'17 target languages to prevent potential confusion in language discrimination.

Table 2: Datasets used in training the major system components. Trn, Dev, LITW, and E15 are lre17train, lre17dev, Languages in the Wild, and lre15eval, respectively. The * refers to the use of cross-validation, which was removed for preparation of the final systems after development. The subscript 1 and 2 refer to different methods of chunking audio as explained in the text.

Component	Fixed	Open Supplement
1) I-vector subspaces	Trn, Dev	+ LITW E15
2) Senone PPCA	Trn_1, Dev_1*	+ LITW_1 E15_1
3) Gaussian backends	Trn_1, Dev_1*	+ LITW_1 E15_1
4) MRNN pre-proc.	Trn_1, Dev_1*	+ LITW_1 E15_1
5) MRNN backends	Trn_2, Dev_2*	+ E15_2
6) Calibration/Fusion	Dev_1*	

Benchmarking and tuning of system parameters were conducted only on the development data. All data was down-sampled to 8 kHz prior to any processing during feature extraction.

2.1. Data Use

For development, the backends and fusion parameters were trained and applied using cross-validation. For evaluation, the development splits were pooled to retrain the backends, calibration, and fusion parameters.

The *fixed* and *open* systems differ only in terms of the data used during training and the additional multi-lingual BN extractors for the *open* condition. The subsystem architectures and algorithms remained constant. The flow diagram for the SRI systems is detailed in Figure 1. Table 2 indicates which datasets were used to train the major system components. For training system components in this table, apart from the i-vector subspaces, only files with more than five seconds of speech were used. The subscript 1 indicates single-chunk extraction for each duration of 10, 30, 60, 90, 180, 360, and 720 seconds, and 2 indicates using 10 to 30 chunks per duration of 8 and 16 seconds. To give an example of total numbers, the Trn_1, the Dev_1 combined set contained 70,921 out-of-domain chunks, and 2,979 and 3,417 chunks from the in-domain MLS14 and VAST development datasets. For development scoring, we used all 873 VAST and 2787 MLS14 development files.

3. Vector Representations

We considered several acoustic features in our systems, each of which are referenced in the following sections: power normalized cepstral coefficients [6] (**PNCC**), Mel frequency cepstral coefficients (**MFCC**), log Mel spectra (**LMS**), and 5th-order Legendre polynomials of pitch and voicing [7] (**PROS**).

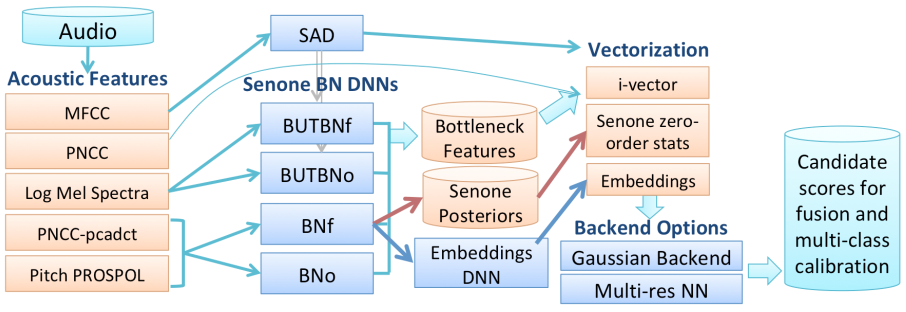

Figure 1: Flow diagram of components considered in the development of systems for multi-domain LID.

3.1. Bottleneck i-vectors

We extracted 80-dimensional bottleneck (BN) features from several DNNs [8, 9, 2]. Post-fixed with *f* or *o* to indicate suitability for the *fixed* and *open* training conditions respectively, these DNNs included:

- **BNf**: Target of 5302 senones learned using Fisher and Switchboard data. Hidden layers had 1200 nodes, with the last hidden layer being a bottleneck of 80 nodes.
- **BNo**: Target of 4993 senones learned using a subset of Fisher and Switchboard data, and 14 languages from the BABEL set. Hidden layers had 1200 nodes, with the last hidden layer being a bottleneck of 80 nodes.
- **BUTBNf**: Target of 2423 senones learned using Fisher data. The model was a stacked BN architecture [10] with two hidden layers of 500 nodes before each BN layer.
- **BUTBNo**: Target of 3096 phoneme states (phoneme states per language stacked together) learned using 17 languages from the BABEL datasets. The model was a stacked BN architecture [10] with two hidden layers of 1500 nodes before each BN layer.

The BNf and BNo DNNs used as input the concatenation of two features: PROS and PNCC. Pitch and voicing estimates were extracted using the open-source Kaldi package [11] from which 12-dimensional PROS feature vectors were extracted as the coefficients of the Legendre polynomial approximation of order 5 of the pitch and energy signals over a 25 ms sliding window [7]. These were concatenated with PNCC features that had been contextualized with principal component analysis-discrete cosine transform (pca-DCT) [12] to form the 102-dimensional input to the senone DNNs. The aim of the data-driven pca-DCT contextualization is to represent a dense “speech” space, and the transformation in this work was learned from the Fisher and Switchboard development sets.

The BUT-supplied bottleneck extractors [10] (BUTBNf and BUTBNo) for LRE’17 were based on LMS features with 10 stacked frames with Hamming windows applied on the trajectories prior to taking six DCT bases, to give a 144-dimensional vector that was finally normalized with mean and variance normalization (MVN) on the speech frames. We passed our own speech activity detection to the extractors for the purpose of MVN.

BN features from each of the above DNNs were used to train an i-vector extractor [13] consisting of a 2048-component universal background model (UBM) with diagonal covariance and a subspace of rank 400. For the *fixed* condition, i-vector extractors were trained using the LRE17train and complete LRE17dev sets. For the *open* condition, all data from LRE15eval and LITW was added during training.

3.2. Senone Posterior Vectors

In recent work [2], we proposed a very simple language recognition system based on zero-order statistics of DNN senone posteriors. Specifically, the counts for each senone q on the ith file are given by

$$Z_q(i) = \sum_{t|t\in S} \gamma_q(i,t),\ q \in Q \tag{1}$$

where $\gamma_q(i,t)$, the posterior of the senone q given the tth frame of the ith file, is obtained by the DNN, Q is the set of senones, and S corresponds to the set of speech frames as determined by a speech activity detector. Similar to the lattice counts used by phone recognition and language modeling (PRLM) approaches, this count does not depend on hard decisions from the recognizer. To create the log-posterior features, we divide these values by the number of speech frames N_S and take the logarithm.

$$\hat{W}_q(i) = \log Z_q(i)/N_S \tag{2}$$

Finally, we form a feature vector to represent waveform i by concatenating these values for all senones,

$$\hat{W}(i) = \left[\hat{W}_0(i),\ \hat{W}_1(i),\ \cdots,\ \hat{W}_Q(i)\right]^T \tag{3}$$

The dimension of the resulting vector is equal to the number of senones (around 5k), which is much larger than the usual size of the i-vector modeled by standard backend approaches for LID. We use PPCA to reduce the dimension of $\hat{W}(i)$ to 400D, which was trained on the available training data for either the *fixed* or *open* training conditions. Mean and variance normalization is applied prior to PPCA. These vectors can then be modeled with standard backends used for i-vectors.

3.3. Embeddings

Recent work in [3, 4] has shown significant advances in the related field of *speaker recognition* by replacing the i-vector extraction process with speaker embeddings extracted from a DNN trained to directly discriminate speakers. Several of the best systems submitted to LRE'17 leveraged this technology in the context of language recognition by replacing speaker labels with language labels during DNN training. We decided to apply our findings on what makes a good speaker embeddings extractor [14] to this task.

We start with the Kaldi recipe for speaker embeddings [15], which involves five frame-level hidden layers of 512 or 1500 nodes, a statistics pooling layer and two segment-level hidden layers of 512 nodes. Apart from the statistics pooling layer, hidden layers were based on a rectified linear unit (ReLU) activation and batch normalization, and the first three layers incremented time context in the network. More details can be found in [15]. As there are only 14 languages compared to thousands of speakers, we modified training to allow for 5000 examples per language in each training data partition (archive) instead of the default 35. For training the embeddings network, we used bottleneck features from the fixed condition BNf extractor (based on PNCC-pcadct + PROS features). Training data included LRE17train data only, as the inclusion of LRE17dev data was found to cause the backend classifier to be wildly miscalibrated. This is due to seeing and discriminating dev data within the embeddings network itself.

Embeddings networks have been shown to respond surprisingly well to artificial degradation of audio during training [14, 16]. In addition to the raw audio, we artificially degraded the data four-fold using each of the following:

- **Transcoding:** The random application of one of 32 codecs ranging from very low (2.4 kb/s) to high bitrate (>32 kb/s) codecs.
- **Noise:** Degrading audio using the FaNT toolkit with a random non-vocal noise sample of more than 100 collected from `opensound.org` with a random SNR of 8, 15, or 20 dB.
- **Music:** The same process as for noise, but with 100 instrumental music samples from a wide range of genres.
- **Reverb:** Using fconv to reverberate audio samples using a randomly selected sample from 45 real impulse responses. These samples ranged from low reverb to very high reverb.

Finally, for each audio file, we extracted 512-dimensional embeddings from the first layer after the statistics pooling layer.

4. Speech Activity Detection

We trained a DNN-based SAD model for the LRE'17 evaluation that conformed with the *fixed* condition restrictions. Using the BNf DNN described above, we generated senone posteriors for our DNN-SAD training data, which consisted of Fisher, Switchboard, and LRE17train data, along with a five-minute Dual Tone Multi Frequency (DTMF) audio file. The three silence senones (silence onset/offset/remain) were then summed for each frame. Those frames for which the silence sum was above 0.5 were labeled as non-speech, and the other frames as speech. This formed the ground truth labels used to train the DNN-SAD model. This model had two hidden layers with 500/100 nodes, respectively, and used 20-dimensional MFCC features, stacked with 31 frames, and MVN norm over a 201-frame window. A threshold of 0.0 was used to threshold SAD of any training, development, and evaluation audio used in our systems.

We performed a benchmark of SRI's NIST OpenSAT SAD system submission on the LRE'17 development set and found it to offer comparable or slightly worse performance than the approach detailed above. This OpenSAT system was rather complex and shown to be robust across many conditions, and this benchmark helped to justify our approach to training SAD from pseudo-ground truth using the LRE'17 fixed data and the languages of interest in LRE'17.

5. Backend Scoring

Vector representations were processed with one or two scoring backends. In both cases, the use of training chunks with less than five seconds of speech was found to degrade performance on the development set. These backends are described below:

5.1. Domain-and-Language-Weighted Backend

We started with a traditional Gaussian backend and implemented class-weighting as in [17] to normalize for language and domain imbalance in the training data. The domains to be balanced were "train" for lre17train, "mls14" for MLS14 data, "vast" for VAST data, and additionally for the *open* systems, a "litw" domain for LITW data. The within-class covariance for the GB was estimated from all training languages (including more than 20 languages in LITW for the *open* condition) prior to subsetting to the 14 target languages for determining the GB class means. The model was trained using a single chunk for each duration of approximately 10, 30, 60, 90, 180, 360, and 720 seconds for each file where possible.

The means and the covariance of the GB were obtained by weighting each training sample with a weight that depends on its language and its domain such that the sum of all weights from each language-domain pair is the same. This way, the estimated mean for each language is not dominated by the most frequent domain for that language, and the estimated covariance metric (shared across all languages) is not dominated by the most frequent languages or domains. See [17] for the equations used to estimate the parameters given the weight and vector representation of each sample.

5.2. Source-Normalized Multi-Resolution Neural Network (MRNN)

The multi-resolution neural network (MRNN) backend was developed during our LRE'15 development phase [18]. The MRNN is not only trained on chunked audio, but test samples are also chunked at multiple durations before taking a duration-weighted average to form a final score. The motivation behind this backend was to improve dialect discrimination by leveraging the short spoken cues that differentiate dialects. Consequently, this backend was applied to bottleneck features as these contain rich phonetic information when i-vectors are extracted from short speech samples [5].

The MRNN was trained using 10—30 chunks per file of durations of 8 and 16 sec. An i-vector was extracted for each audio chunk. Prior to training the neural network, the i-vectors were normalized using source-normalized within-class-covariance estimation [19] (based on the same domains as the GB). In this context, source normalization (SN) aims to remove cross-source (or domain) variability from the feature space (i-vectors in our case) to allow backend models to better model the target domain while leveraging information from the out-of-domain data. SN was performed using all available training languages, including more than 20 languages from the LITW data in the *open* condition. The MRNN backend was trained using a multi-class objective function over the 14 target languages and included 500 hidden nodes. In contrast to the GB, which had a weighting to normalize for domain bias, it was found that limiting the influence of LRE17train data to approximately 10k chunks was found to allow the limited MLS14 and VAST development data to contribute more to training and offered better performance.

During testing, test files were cut to a target of 10 chunks of durations of 2, 4, 8, 16, and 32 seconds of speech where possible. An i-vector for each chunk was then evaluated by the NN, and the scores for a single test file were merged using a weighted average given by $\frac{d_i}{d}$ where d was the duration of speech in the longest test chunk, and d_i was the duration of speech that went into the test chunk i. This average weight has the effect of normalizing for the dominant shorter chunks from test files, but also allows the chance for more confident decisions (ideally due to more distinct phonetic content of the detected language/dialect) on the short segments to be realized. The major performance benefit to the MRNN over using a single i-vector for testing comes from the "chunking" of test samples, quantified by a relative improvement of 5—10% on the development set.

6. Score-level Fusion and Calibration

We applied duration-dependent calibration and fusion by training duration-specific parameters using the strategy below. Only development data from MLS14 and VAST were used in calibration, from which a single chunk per duration of 10, 30, 60, 90, 180, 360, and 720 seconds for each file was used where possible. Use of speech audio less than five seconds was not found to benefit calibration and fusion. Our calibration and fusion strategy is as follows:

1. Set duration boundaries to 0, 1000, 1500, 2000, 2500, 3000, 4000, 5000, 6000, 12000, 18000, 24000, 30000, 42000, 54000, 66000, 78000, Inf which are in terms of frames (100 per sec)
2. Set M the minimum samples per class to 75 (tuned on the dev set using cross-validation)
3. For each duration window (for instance, 1500—2000), locate the corresponding chunked i-vectors with that much speech for a given language. If the number of samples is less than M, increase the pool to M by selecting the samples closest to the duration-window boundaries.
4. Train a standard multi-class calibration model for each duration window, having a shared scale and language-dependent shift.

This paradigm naturally provides a level of regularization when limited samples are found within a duration window. It has one caveat: it also naturally relies on the longer VAST files for the longer duration windows due to the construct of the development set in which MLS14 data is limited around the 30 sec duration. For the evaluation of test audio, all that is needed is the duration of speech to decide which calibration model parameters to apply.

6.1. Fusion Selection

Each input feature was processed using each backend where possible to generate a set of scores on the development set, which formed candidates for score fusion. Fusion was done using multi-class linear regression with cross-validation (xval) of the development set, with a split based on estimated unique ID. The exhaustive fusion of the candidate score sets was explored and the best fusion of up to four systems under certain constraints was selected based on Cost$_{act}$ performance on the development set. These constrains were as follows: *Primary* had no constraint, *Secondary* was constrained to the weighted GB to remove potentially overfitted NN backends, and *Tertiary* additionally restricted the bottleneck extractors to DNNs trained in-house.

7. Results

This section analyzes results of SRI's systems on the development and evaluation sets for LRE'17. We start by presenting the systems SRI used in the evaluation, provide remarks on the leading aspects of the systems, then discuss the impact of language embeddings based on the experiments performed after the evaluation.

7.1. Performance Metrics

NIST defined the LRE task to focus on the pair-wise LID performance for all same/different language pairs using a linear cost model, which was based on the cost of missed detections and false alarms and the *a priori* probability of the target language P_{tar} [1]. While costs were held constant at 0.5, P_{tar} took on values of 0.5 and 1.0, averaged and normalized to provide a metric that could easily be interpreted and based on more than one operating point. Finally, the primary metric was formed by averaging this dual-cost model metric across 28 partitions by dividing the test data into languages (14 of) and domains (2 of). In addition to reporting actual Cost$_{act}$ [1], we report the minimum cost value obtainable, Cost$_{min}$ reflecting the case of having perfect system calibration.

7.2. SRI *Fixed* Submissions

The final *fixed* subsystems selected and their use in each of the SRI *fixed* submissions are detailed in Table 3a. The primary system made use of all possible feature extractors and backends. The secondary system excluded the neural network backend from the fusion pool to leave purely GB scoring. The tertiary submission was based on a single, traditional i-vector system (PNCC) and a single BNf bottleneck extractor from which the i-vectors or senone posterior vectors were subject to all backend scoring methods.

Comparing across subsystems in the table, we see that the traditional shifted-delta cepstra (SDC) i-vector system PNCC-GB performed significantly worse than the BN-based i-vector systems. The BNf i-vector system performed over 20% better than the equivalent senone system (BNf-GB vs. BNf-Senone-GB), however, the senone-based system was consistently selected in fusion due to its complementarity. The BUT BN extractors performed quite well, and on par with the BNf extractor when used with the GB. The complementarity offered through these extractors was around 9% relative as observed when comparing Cost$_{act}$ of the secondary and tertiary systems.

Table 3: Development results for SRI *fixed* and *open* subsystems and submissions using cross-validation on the original lre17dev set as provided by NIST and the NIST scoring package (equalized results).

(a) *Fixed* training condition

Submission	Cost$_{min}$	Cost$_{act}$	EER
1) PNCC-GB	0.364	0.383	10.55%
2) BNf-Senone-GB	0.274	0.282	7.65%
3) BNf-NN	0.238	0.245	6.80%
4) BUTBNf-NN	0.216	0.222	6.05%
5) BNf-GB	0.211	0.220	6.19%
6) BUTBNf-GB	0.209	0.220	5.85%
Primary (2+4+5+6)	0.173	0.178	4.97%
Secondary (2+5+6)	0.179	0.186	4.87%
Tertiary (1+2+3+5)	0.197	0.204	5.40%

(b) *Open* training condition

Submission	Cost$_{min}$	Cost$_{act}$	EER
A) BNf$_o$-Senone-GB	0.251	0.256	7.44%
B) BNf$_o$-GB	0.190	0.199	5.20%
C) BUTBNf$_o$-NN	0.189	0.193	5.54%
D) BNo$_o$-GB	0.188	0.190	5.04%
E) BUTBNf$_o$-GB	0.178	0.181	5.04%
F) BUTBNo$_o$-GB	0.169	0.174	4.69%
Primary$_o$ (A+C+D+F)	0.139	0.140	3.96%
Secondary$_o$ (A+D+E+F)	0.143	0.145	4.01%
Tertiary$_o$ (A+B+D)	0.165	0.171	4.49%

7.3. SRI *Open* Submissions

All subsystems used in the open condition were trained with the additional data sources detailed in Section 2, including LRE15eval data and LITW. These subsystems are denoted with subscript o. The available scores for fusion matched the restrictions imposed in Section 7.2 to form the primary, secondary, and tertiary submissions. Observing the results in Table 3b, we see similar trends to the *fixed* systems in Section 7.2. One major difference, however, is the significant gain of more than 20% relative in the primary submission for the *open* condition compared to the *fixed* condition. The major contributing factor to this gain was the **LRE15eval data** in the open condition systems which positively impacted both MLS14 development results and VAST results. While not shown here, the LITW data was found to provide only a marginal gain in VAST performance on the dev set, perhaps due to its use only in covariance estimation

Table 4: Evaluation results for SRI *Fixed* and *Open* submissions provided by NIST (equalized results).

Submission	Cost_{min}	Cost_{act}	EER
Primary	0.170	0.173	4.51%
Secondary	0.170	0.173	4.46%
Tertiary	0.201	0.209	5.38%
Primary_o	0.147	0.151	3.99%
Secondary_o	0.147	0.148	4.07%
Tertiary_o	0.165	0.167	4.51%

and not having any of the 14 target languages. Also in contrast to the fixed condition, the best-performing subsystem (BUTBNo_o-GB) was based on the stacked phonetic states of multiple languages instead of the universal phone set used in the BNo_o-GB system. While not entirely comparable due to originating in two separate development environments, an 8% relative difference was observed, suggesting the stacked approach may be superior.

7.4. Evaluation Results

The official evaluation results for the SRI submissions are detailed in Table 4. In general, comparing development results from Tables 3a and 3b, we can see that the development results were quite a good gauge as to the expected performance on the evaluation dataset (i.e., less than 10% relative difference).

Analyzing the results in Table 4, we can observe that using the MRNN in the fusion (primary submissions) instead of restricting to just the GB backends (secondary submissions) provided comparable performance. This suggests the MRNN generalized well to the evaluation data, but due to the computational expense and sensitivities during training, the simplicity of the language-and-domain-weighted GB appears to be a solid-performing backend. We can also observe that the calibration performance of the systems was quite reasonable, such that there is a limited loss between Cost_{act} and Cost_{min}.

7.5. Development Observations

Aspects believed to be of considerable impact in the SRI submissions included:

- **Language/domain weighting** in the GB provided up to 10% relative gain in Cost_{act} over no weighting and removed the sensitivity of the GB to the amount of out-of-domain data that was used in training (i.e., LRE'17 "train" set).
- **Duration-aware calibration** provided 10–15% relative gain in Cost_{act} for our fusion systems compared to using a single global calibration model.
- **Source normalization in NN i-vector preprocessing** provided a 20% relative gain in Cost_{act} over not using it. When evaluated in the estimation of the GB covariance, it provided similar performance to language/domain weighting, however, it was more sensitive to the amount of out-of-domain data used in GB training.

Table 5: The impact of language embeddings on development (white rows) and evaluation (shaded rows) results for the *fixed* training condition using cross-validation on the LRE'17 datasets (equalized results).

System	Cost_{min}	Cost_{act}	EER
BNf i-vector GB	0.211	0.220	6.19%
BNf embeddings GB	0.211	0.218	5.85%
Secondary (3-way fused)	0.179	0.186	4.87%
+ Embeddings	0.165	0.177	4.42%
Secondary (3-way fused)	0.170	0.173	4.46%
+ Embeddings	0.159	0.164	4.19%

8. Post-evaluation: Language Embeddings

After the evaluation, and inspired by recent work in speaker embeddings, we evaluated language embeddings in combination with our submissions. Results comparing individual systems from Section 7 and the language embeddings on both the dev and evaluation sets are provided in Table 5. We observe that the embeddings system provided marginal gains in EER compared to the equivalent bottleneck i-vector framework, while it's addition to the fusion of the secondary system provided a 5% and 9% relative gain in Cost_{act} and EER, respectively. The evaluation results also followed a similar trend in terms of complementarity. Given this was our first attempt using embeddings for LID, the embeddings technology appears to be a very promising area of research in the future, offering complementary information to current i-vector technology.

9. Conclusions

This work focused on the task of language recognition across multiple domains in the context of the NIST LRE'17. System training data was heavily concentrated on out-of-domain audio, and limited data from each of the two target domains was also available. The SRI team focused on three novel aspects to improve robustness to multi-domain LID. These included a language-and-domain-weighted Gaussian backend; a source-normalized, multi-resolution neural network backend; and duration-aware calibration with natural regularization. These aspects led to a very competitive submission in the LRE'17. Additionally, the promising technology of language embeddings was demonstrated in the multi-domain LID context in which high complementarity was shown.

10. Acknowledgments

The team would like to thank: Martin Graciarena and Nikoletta Basiou from the SRI STAR lab; Emre Yilmaz, a visiting postdoctoral researcher from Radboud University, Nijmegen, The Netherlands; and members of the BRNO BUT speech lab for providing the bottleneck extractors to the research the community for both the *fixed* and *open* LRE'17 conditions.

This material is based upon work supported partly by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-15-C-0037. The views, opinions, and/or findings contained in this article are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. Distribution Statement A: Approved for Public Release, Distribution Unlimited.

11. References

[1] *NIST 2017 Language Recognition Evaluation Plan*, 2017, `https://www.nist.gov/sites/default/files/documents/2017/09/29/lre17_eval_plan-2017-09-29_v1.pdf`.

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

[3] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and Sa Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 165–170.

[4] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Proc. Interspeech*, 2017, pp. 999–1003.

[5] M. Mclaren, L. Ferrer, and A. Lawson, "Exploring the role of phonetic bottleneck features for speaker and language recognition," in *Proc. ICASSP*, 2016.

[6] C. Kim and R. Stern, "Power-normalized cepstral coefficients (PNCC) for robust speech recognition," in *Proc. ICASSP*, 2012, pp. 4101–4104.

[7] N. Dehak, P. Dumouchel, and P. Kenny, "Modeling prosodic features with joint factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2095–2103, 2007.

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

[9] P. Matejka, L. Zhang, T. Ng, S.H. Mallidi, O. Glembek, J. Ma, and B. Zhang, "Neural network bottleneck features for language identification," in *Proc. Speaker Odyssey*, 2014.

[10] R. Fer, P. Matejka, F. Grezl, O. Plchot, K. Vesely, and J. Cernocky, "Multilingually trained bottleneck features in spoken language recognition," *Computer Speech & Language*, vol. 46, no. Supplement C, pp. 252 – 267, 2017.

[11] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011.

[12] M. McLaren and Y. Lei, "Improved speaker recognition using DCT coefficients as features," in *Proc. ICASSP*, 2015.

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

[14] M. McLaren, D. Castan, M. Nandwana, L. Ferrer, and E. Yilmaz, "How to train your speaker embeddings extractor," in *Submitted to Speaker Odyssey*, 2018.

[15] *NIST SRE 2016 Xvector Recipe*, 2017, `https://david-ryan-snyder.github.io/2017/10/04/model_sre16_v2.html`.

[16] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *Submitted to ICASSP*, 2018.

[17] L. Ferrer, Y. Lei, M. McLaren, and N. Scheffer, "Study of senone-based deep neural network approaches for spoken language recognition," *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. 24, no. 1, pp. 105–116, 2016.

[18] M. McLaren, D. Castan, and L. Ferrer, "Analyzing the effect of channel mismatch on the SRI language recognition evaluation 2015 system," in *Proc. Odyssey: Speaker Recognition Workshop*, 2016, pp. 188–195.

[19] M. McLaren and D. Van Leeuwen, "Source-normalized LDA for robust speaker recognition using i-vectors from multiple speech sources," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 3, pp. 755–766, 2012.