Frame Level LID in Code-Mixed Speech

Sridhara Manideep

Ref: [Probability and Statistics](https://docs.google.com/document/u/0/d/1gMh2JRQpoNt3K35CJShe5lhTdBMSOBiFRboIECzC1SI/edit)

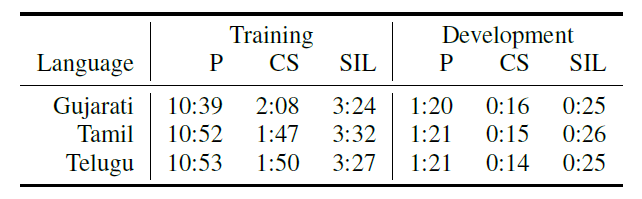
horizontal line

# [Vocapia-LIMSI Submission](https://drive.google.com/file/d/1mLhajdwZEIdNyU7CvDHuVchLwyYYKPAi/view?usp=sharing)

* The primary system combines an acoustic approach based on i-vector modelling of audio segments with a phonotactic approach that focuses on sequences of language-independent phone units.
* Both modelling approaches provided comparable performance, and a gain was obtained by a simple linear combination of their scores, showing their complementarity.
* The average accuracy was 81.2% on the development set and 78.7% on the evaluation set.

### Dataset

* Code-switched sentences were provided with the corresponding 200ms frame-level annotations.



### 

### 

### iVectors

* The main idea is that the language- and channel-dependent supervectors of concatenated Gaussian Mixture Model (GMM) means can be modelled as

***S = m + Tw***

* **w** → standard normally distributed latent variable
* **T** → Total Variability Space Matrix  
  **m** → Gaussian supervector from language-independent model (UBM)  
  **S** → Gaussian supervector from language-dependent model
* For each observation sequence representing an utterance, our iVector is the Maximum A Posteriori (MAP) point estimate of the latent variable w

### Method

* For each frame, a 32 ms window and a 10 ms offset are used to extract a 32-band Mel scale spectrogram concatenated with log-pitch, delta-log-pitch and voicing probability.
* TRAP-DCT features are estimated on 100 ms windows (11 features), retaining the first 6 coefficients including the DC component.
* The resulting TRAP-DCT features with 210 dimensions (35 x 6) are passed as input to a bottle-neck DNN that has 3 hidden layers with 2000 units and 1 bottle-neck hidden layer with 400 units. Each hidden layer is followed by a non-linearity p-norm unit which reduces the dimension of the layer to 200 and 40 respectively.
* The Phonetic bottle-neck DNN was trained on about 1000 hours of English Broadcast Data.
* The bottle-neck features are extracted without a cepstral mean or variance Normalization (CMVN).
* The MLR model is estimated on all training utterances/segments using an expectation-maximization algorithm

### Details

* The full covariance GMM with 2048 components, the UBM, and an i-vector extractor are estimated using the training data using the Kaldi toolkit.
* A 600-dimension i-vector is extracted for each training utterance or segment which is then normalized to unity.
* A language-specific i-vector is obtained by averaging the normalized i-vectors for each training utterance.
* All audio files of the training, development and evaluation data are analyzed using 600ms-long overlapping windows with a 200ms step (a frame).
* The label of the segment was associated to the frame at the centre of the window
* For each of 3 the Indian languages, an MLR classifier is estimated on all training i-vectors (one/frame) for each class (silence, native language and English).
* For this task, the use of an explicit i-vector class for silence trained on the target dataset significantly improved the silence frame detection performance over using either GMM or DNN pre-trained VAD models.

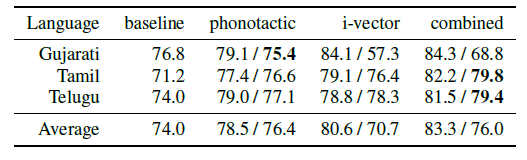
### Testing Phase

* I-vector is extracted for each utterance and is processed to compensate for session variability.
* Multi-class logistic regression (MLR) is used to compute test utterance scores.
* In the test phase, an i-vector is extracted for each test frame and scored using the MLR model.

### Phonotactic Identification

* Phone decoders using phonetic models from several languages are used to decode the training data and to estimate phone n-gram statistics on the resulting phone lattices for each target class.
* Then, given a new utterance, the expectation of its phonetic log-likelihood is computed according to each target model, resulting in a set of posterior scores.
* The implementation relies on the VoxSigma Software Suite.
* Frames labelled as silence in the reference annotation were discarded.

### Results



# [JFA (Joint Factor Analysis) and I-Vectors](https://www1.icsi.berkeley.edu/Speech/presentations/AFRL_ICSI_visit2_JFA_tutorial_icsitalk.pdf) ([Paper](https://drive.google.com/file/d/1mLjoCXSEwzD7IMpz0CCIdWCa6wtrttgd/view?usp=sharing))

* The problem with the Relevance MAP adaptation of GMM-Supervectors is it adapts to not only speaker-specific characters of speech but also channel and other nuisance factors.

# [Relevance MAP for GMM-Supervector Space](https://drive.google.com/file/d/1mJ_-V7oJ-sdCznLCb8kRMYTm3BHPrhw0/view?usp=sharing)

* In the GMM approach, a language model is obtained by maximum a posteriori (MAP) estimation from a universal background model (UBM). [Link](https://drive.google.com/file/d/1mKNTbjBsAdIPNui-rJOTc5ms4iGBKY5q/view?usp=sharing)
* The UBM has usually trained through an expectation-maximization (EM) algorithm from a background dataset covering a wide range of languages, speakers and channels.
* It is observed that Supervector formed by the GMM encounters a shifting problem in the super vector space due to the varying duration of the utterance.

# 

# 

# [Language Recognition in iVectors Space](https://drive.google.com/file/d/1i3S3p1KdDNBvPvGjHAm5pRKYzqVFqUJ_/view?usp=sharing)

* iVector is a fixed-length low-dimensional vector, which is extracted for each utterance based on the JFA-like idea of estimating latent variables corresponding to high variability subspace.
* The principal difference from JFA is that we are not interested in evaluating the adapted model. Instead, the latent variables - iVectors - are used as features for another (possibly very simple) classifier.
* Model for iVector extraction can be trained in an unsupervised manner
* To recognize language in the iVector space, 3 different linear classifiers are used:
  + Generative model, where classes are modelled by Gaussian distributions with a shared covariance matrix.
  + Linear Support Vector Machine (Discriminative Model)
  + Logistic Regression (Discriminative Model)

## iVectors

* The main idea is that the language- and channel-dependent supervectors of concatenated Gaussian Mixture Model (GMM) means can be modelled as

***S = m + Tw***

* **w** → standard normally distributed latent variable
* **T** → Total Variability Space Matrix  
  **m** → Gaussian supervector from language-independent model (UBM)  
  **S** → Gaussian supervector from language-dependent model
* For each observation sequence representing an utterance, our iVector is the Maximum A Posteriori (MAP) point estimate of the latent variable w