Language Recognition & Diarization on Code-switching Speech

B151448

Word Count: 7994

Master of Science

Speech and Language Processing

School of Philosophy, Psychology & Language Sciences

University of Edinburgh

2020

Abstract

This project investigates language recognition and diarization performance on code-switching speech. The i-vector framework is used, previously established as a suitable method for language recognition, but not yet for language diarization. The i-vector model (UBM and total variability matrix T, in particular) is trained and shared between both tasks; i-vectors are merely utilized differently in each context (with logistic regression for recognition and PLDA with clustering for diarization). Additionally, performance improvement is attempted by modifying the universal background model (UBM) to incorporate explicit phonetic knowledge from a TD-DNN that predicts senones. Results indicate that the TD-DNN UBM improves outcomes slightly, and that the i-vector scheme used for language diarization could be viable but requires further investigation. The overt mismatch between training and testing data is also speculated to be a major source of error.

Table of Contents

List of Acronyms

AHC - agglomerative hierarchical clustering
ASR - automatic speech recognition
DER - diarization error rate
DET - detection error tradeoff
DNN - deep neural network
EER - equal error rate
EM - expectation maximization
FA - false alarm
FBANK - filterbank (features)
GMM - Gaussian mixture model
IPA - international phonetic alphabet
JFA - joint factor analysis
LRE - Language Recognition Evaluation
MAP - maximum a posteriori
MFCC - mel frequency cepstral coefficient
NIST - National Institute of Standards and Technology
OOV - out-of-vocabulary
PLDA - probabilistic linear discriminant analysis
PLP - perceptual linear prediction (features)
SDC - shifted delta cepstra
SVM - support vector machine
TD-DNN - time delay deep neural network
UBM - universal background model
VAD - voice activity detection
WER - word error rate

1. Introduction

Code-switching is a linguistic phenomenon wherein multilingual speakers switch between two or more languages when speaking. This switching behaviour and its cognitive implications are of interest to researchers in both linguistics and psychology, but to study such events it must be quantified for analysis. Given recorded code-switching data, annotating the switching points and respective languages manually is naturally time-intensive and subject to human error. Thus, this project explores more automatic approaches to detecting both the code-switch points and languages.

The detection of code-switch points and languages being spoken is essentially that of language diarization – the task of identifying which language is being spoken and when. This concept is borrowed from speaker diarization, the same task but for speakers. Situations such as meeting recordings and telephone calls are typical targets for speaker diarization, with audio containing at least two speakers, sometimes more. In contrast, the occurrence of multiple languages in the same recording is not too common aside from the code-switching context. Naturally, this makes it an ideal target for language diarization research, which has seen some dedicated work, but is often undertaken as a byproduct of bilingual or multilingual ASR. Thus, current methods are usually based on lexical or phonological knowledge, or other linguistic information easily afforded by building a full ASR system for the data.

As this project is mainly interested in identifying switching points and languages only (without full speech recognition), a more acoustic-phonetic approach is taken. The language recognition aspect is investigated first to gauge performance without interference of poor automatic segmentation. Then, the architecture is extended to the full diarization task, including segmentation and the detection of switching points. The test data for this project is a limited dataset of code-switching speech collected by psychology researchers at the University of Edinburgh. Due to the restricted amount, there is not enough to comprise both a train set and a test set, so already established corpora are used for training instead. This leads to the side-examination of the effect of mismatched train/test data.

2. Related Work

2.1 Approaches to Spoken Language Recognition

The more general problem of language recognition can be approached from different linguistic perspectives (Li et al., 2013). At the most concrete level, the acoustics of a language can help to distinguish between languages as each will have its own phonetic/phonemic inventory and distribution. Similarly, there will be differences in phonology, the patterns and rules which govern the combination of the phonemes. At a more abstract level, the lexical counterpart of this relationship is that of vocabulary and syntax, which can potentially provide an even better distinction between languages, given they are accurately deducible from the spoken signal. As the features increase in abstraction (Figure 2.1), more linguistic knowledge is necessary to build into the system. Thus, for the sole task of language recognition where the full text is not necessary to recognize but merely the language ID, current research directions have largely focussed on the more concrete acoustic and phonetic space. This is the approach taken in this project as well, in hopes of building a relatively straightforward system to detect languages and switching points, without regard to transcription. Further, this method is also more flexible to the addition of new languages, as simple, untranscribed data is sufficient for training.

An early attempt at language identification by modelling phonetic differences was simple Gaussian mixture model classification (Zissman, 1996) in which a GMM is trained for each target language and the one which produces the highest log likelihood for the test utterance is determined as the language. This basic concept is computationally and linguistically simple, fast and easy to implement, yet not the most accurate. It also relies on assumptions of statistical independence that may not be true in practice. However, the idea of using a GMM to model the acoustic space was elaborated on in the development of UBM methods for speaker verification (Reynolds, 1997) to better model acoustic differences in speaker variation, and the extension of joint factor analysis techniques (Kenny, 2005) to

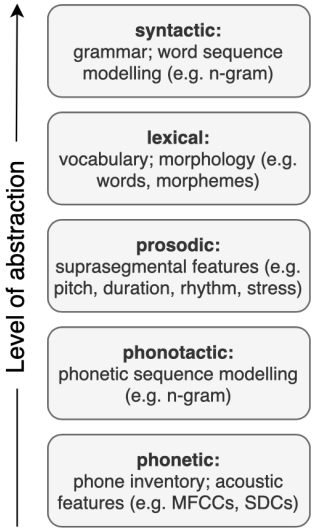


Figure 2.1: Levels of linguistic cues for language recognition.

better characterize the variation into useful components. This concept evolved into the now popular i-vector framework (Dehak et al., 2010), which takes this a step further and distills all significant variation into one set of factors - the i-vector. I-vectors have since been widely adapted for language recognition purposes (Dehak et al., 2011), and continue to be a standard baseline.

Recently with the growing popularity of deep learning, neural networks have been increasingly applied to language recognition. Various aspects of the i-vector approach have been modified and improved upon by incorporating knowledge from a DNN, such as deriving the i-vectors from DNN bottleneck features (Ferrer et al., 2015; Song et al., 2013), or training the UBM based on DNN senone posteriors (Ferrer et al., 2015; Sarma et al., 2018; Tian et al., 2016). DNNs have also been used on their own in directly classifying language (Gonzalez-Dominguez et al., 2014; Lopez-Moreno et al., 2014; Tang et al., 2017), and in the extraction of x-vectors (Snyder et al., 2018), a newer, alternative feature representation scheme to i-vectors.

Due to the powerful ability of the i-vector framework in manipulating acoustic features, and the wide success of its application to language recognition, this project establishes it as a baseline model (with a GMM-UBM). A DNN-UBM is then experimented with to generate senone posteriors to hopefully incorporate more specific phonetic knowledge into the i-vectors. They are then scored with a logistic regression model, further discussed in Section 3.

2.2 Language Diarization

The task of diarization has mainly been investigated in the context of speaker diarization, and addresses the question "who spoke, and when?". Generally, it consists of two main stages: first the audio is segmented (ideally according to speaker changes), and then the segments are clustered - all speech segments attributed to the same speaker are labelled as such (Moattar & Homayounpour, 2012). The general framework of diarization is illustrated in Figure 2.2. This can be adapted to languages by replacing the speaker objective with that of languages ("what language is being spoken, and when?"), wherein the code-switch points correspond to the speaker change points.

Segmentation can be approached from two different perspectives - a naive constant length segmentation, or a more informed speaker or language change detection mechanism. The more naive approach of constant length segmentation splits the speech regions into short, fixed-size segments, in hopes of avoiding overlapped speech. In this case, the true change point detection arises after clustering wherein the cluster identities of the segments help to indicate where there is a switch. In contrast, speaker change detection mechanisms can be based on silence/pauses, segment distance, or even decoded words/phones to help in determining speaker boundaries. Language change detection mechanisms can also be based on acoustic frame classification (Spoorthy et al., 2018), or more commonly, with language modelling and phonotactics (Lyu et al., 2013) in an attempt to incorporate more specific language information. This approach is also found in ASR systems for bilingual or multilingual speech. Due to time constraints, the naive segmentation mechanism is used in this project, a direction not really present in the literature for language diarization, potentially due to poor expected results. However, (Zajíc et al., 2016) investigate both methods (albeit in a speaker diarization context) with conversational telephone speech, and conclude that they can produce very comparable results after a post-resegmentation process.

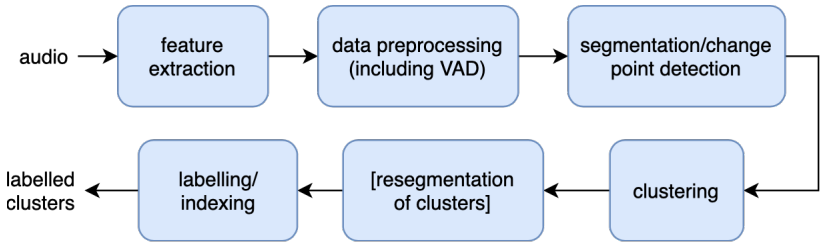


Figure 2.2: The diarization framework.

The subsequent step of clustering can also be addressed in different ways, namely, bottom-up vs. top-down. Bottom-up approaches successively merge similar speech segments until a stopping criterion or a given number of clusters is reached. Top-down methods start with one large cluster and successively splits instead of merges. In general, both have been shown to have largely equivalent performance in speaker clustering (Evans et al., 2012), but it has also been observed that top-down methods are less discriminative, while bottom-up methods are more sensitive to non-speaker variation such as linguistic content (Bozonnet et al., 2011). As this type of diarization scheme (including a clustering component) is not widely done with languages, there does not appear to be any established literature on the effectiveness of either method, but the supposed sensitivity of the bottom-up method to linguistic content may be a sign that this is a more appropriate method for language clustering. Thus, agglomerative hierarchical clustering (a bottom-up procedure) is employed in this project. A similarity metric also must be defined in order to compute the similarity between segments - PLDA scoring is a popular choice in speaker recognition, and is also used here, further discussed in Section 3.

Finally, there is the matter of feature extraction - how to parameterize the speech signal to work well for diarization. Firstly, there is usually a VAD procedure applied at some point in the feature extraction process to detect the speech frames from noise or other non-speech. Then, typical features such as MFCCs/derivatives, PLPs, energy, pitch, etc. have commonly been used directly. But since the inception and success of i-vectors for speaker recognition, they have naturally been applied to the diarization task as well. I-vectors for speaker diarization have been explored (Sell & Garcia-Romero, 2014; Xu et al., 2016) with positive results, yet do not appear to have been investigated for language diarization thus far. This project will attempt to address this disparity and observe how effective they might be in this context.

3. Background

The general structure of speech recognition work can be split into a front end and a back end, the former referring to the feature extraction process, and the latter encompassing the methods used to analyze and interpret the features. This project utilizes additional acoustic features specifically for language recognition (SDCs), as well as i-vectors as the final feature representations. A logistic regression back end is used for recognition purposes and a PLDA/AHC back end is employed for diarization.

3.1 Shifted Delta Cepstra

Typical MFCC features that include deltas and double deltas do a fair job of capturing the immediate context of the current frame. However, in the task of language identification there is more to be gained by including a wider range of context since phones and their phonology are distinguishing features of a language. Thus, the ability to capture enough context so as to at least recognize a phoneme should be a great help. Shifted delta cepstra (SDC) were introduced for this purpose. While deltas could in theory be expanded to cover a wider range of neighbouring frames, this would result in loss of detail for the average is taken over them all (Ambikairajah et al., 2011). SDCs on the other hand are composed of a concatenation of future delta cepstra with the current feature vector, thereby retaining the relevant information over future frames. SDCs have proven to be beneficial in language identification work (Torres-Carrasquillo et al., 2002) and are used in this project.

3.2 I-vectors

At the root of i-vectors is the UBM for speaker recognition - typically a large GMM trained on all available training data (Reynolds, 1997). A popular early method for adapting the UBM to individual speakers was maximum a posteriori (MAP) adaptation - a linear

interpolation of the UBM's mixture components to maximize the likelihood of speaker-specific data (Lei, 2011). Stacking the mean vectors results in a GMM supervector *M(s)* for speaker *s*, given by:

$$M(s) = m + Dz(s) \tag{3.1}$$

where m accounts for the speaker-independent (UBM) model, D is a matrix that represents the possible components to be adapted, and *z(s)* is the set of adaptation factors for speaker s. This mechanism can be visualized as in Figure 3.1, from (Hansen & Hasan, 2015).

Issues with this approach include the treatment of each Gaussian component as statistically independent (Li et al., 2015), as well as the adaptation of components not exclusive to speaker characteristics of speech, such as random noise variations.

To address these issues, factor analysis techniques such as JFA were proposed in which the variance in speaker supervectors is attributed to more specific hidden variables such as speaker and channel factors (Kenny, 2005). The speaker supervector *M(s)* for a given speaker s is now decomposed into different components, as:

$$M(s) = m + Vy(s) + Ux(s) + Dz(s) \tag{3.2}$$

where V and U are the speaker-dependent and channel-dependent components, respectively, and D is left as a residual speaker-dependent component. V and U are low-dimensional matrices that represent the principal dimensions of the corresponding component, often termed eigenvoice and eigenchannel matrices. Thus, *y(s)* and *x(s)* are the sets of factors which operate on these matrices to adapt them to each speaker s.

In further experimentation with JFA, it was found that the channel factors were not only capturing channel effects, but also speaker information (Dehak, 2009). Thus, this

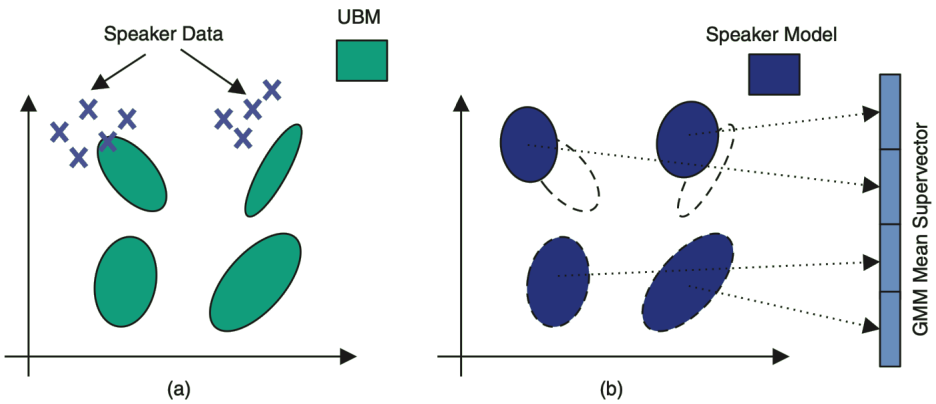


Figure 3.1: GMM-UBM scheme with MAP adaptation.

prompted the concept of a low-dimensional total variability space which encompasses both types of variability (Dehak et al., 2010). So now the supervector is both speaker- and channel-dependent, given by:

$$M(s,c) = m + Tw(s,c) \tag{3.3}$$

where T represents the total variability matrix, and *w(s,c)* are the total factors specific to speaker s, channel c. It is the w that is referred to as the identity-vector, or i-vector, for that speaker/channel.

To extract i-vectors w, first the UBM and total variability matrix T must be trained. The UBM is merely a GMM trained on all training data. Then, Baum-Welch statistics (sufficient statistics) must be extracted from the UBM in order to train the T matrix. Gaussian posterior probabilities (k component occupations) for each frame t of utterance i are required as 0th-order statistics:

$$\sum_t \gamma_{kt}^{(i)} \tag{3.4}$$

to weight the accumulation of means per component to get 1st-order statistics:

$$\sum_t \gamma_{kt}^{(i)} x_t^{(i)} \tag{3.5}$$

and finally taking the centralized 1st-order statistics as 2nd-order statistics:

$$\sum_t \gamma_{kt}^{(i)} x_t^{(i)} x_t^{(i)T} \tag{3.6}$$

(Dehak et al., 2010). Because of the independence of the 0th and 1st order statistics, different datasets or features can be used to produce each, a fact that is exploited in DNN research in particular.

Put in the context of language recognition, i-vectors are extracted for the language-labelled training utterances, as well as for each test utterance. The logistic regression model learns to discriminate between the training set i-vectors based on their language labels, and uses this knowledge to then classify a new test i-vector.

3.3 Role of the UBM

The GMM-UBM is meant to be a generative model of the total speaker-independent (or in this case, language-independent) acoustic space, from which i-vectors can adapt the components to create language-specific representations. Though i-vectors have seen great

success in the language recognition context, it is still true that they were initially designed for speaker verification purposes. Thus, there is no explicit linguistic knowledge built into the UBM or the total variability matrix, though it is true that it could be implicitly learned (Bhattacharjee & Sarmah, 2012; Vaheb et al., 2018).

The concept of incorporating an explicit linguistic feature - phonetics - into the UBM was initially pioneered for speaker recognition, in attempts to capture the variation in speaker pronunciations (Lei et al., 2014). But encouraging the learning of phonetics makes even more sense in a multilingual language ID context, as different languages have different phone inventories and associated phonotactics. Some have tried this approach with positive results (Ferrer et al., 2015; Richardson et al., 2015; Sarma et al., 2018), and it is also attempted in this project.

Modifying the UBM in this way usually involves incorporating a DNN, as a GMM trained in this context did not appear to be as effective in (Lei et al., 2014), albeit for speaker recognition. The basic process involves discriminatively training a DNN as an ASR acoustic model to predict senones (context-dependent phones). Senone predictions are then used as 0th-order statistics, while the regular set of acoustic features (MFCC/SDCs) can still be used for 1st-order statistics. As mentioned previously, the independence of these two allows different features to be used, and different training datasets. One option (as used in this project) is to extract different acoustic features for use with the DNN in computing posteriors (FBANK) and then in accumulating 1st-order stats (MFCC/SDC). The reason for this is to accommodate the fact that the DNN is capable of learning from the more raw features (Sarma et al., 2018). Another option to use differing data is in the training of the DNN - this train set does not have to correspond to the train set used in training the total variability matrix as they are completely separate and have different purposes. The DNN train set is mainly meant to provide senone coverage (and thus requires transcribed data), while the other is actually meant to be used in training and extracting i-vectors. Different DNN training sets is something else explored in this project.

3.4 Recognition – Logistic Regression

Many different back end techniques for i-vectors have been experimented with, including both generative and discriminative methods. Both approaches can work well, and the choice is usually made in consideration of the downstream task at hand. In the context of language recognition, the goal is to determine which language is being spoken in

a particular speech segment. This can be framed as a basic classification task, with a restricted set of possible languages.

In this case, a fairly simplistic, discriminative approach such as logistic regression should be sufficient to model the distinction between languages; SVMs can also work well, but would add complexity. Additionally, (Soufifar et al., 2012) conclude that logistic regression performs slightly better than SVMs without any other i-vector post-processing, as is the case in this project. In contrast, generative approaches such as PLDA are often used in speaker recognition, due to its effectiveness at modelling both the intra- and inter-speaker variability, which is less of a concern in the case of simple language recognition.

3.5 Diarization – PLDA & AHC

The diarization back end consists of PLDA scoring of the i-vectors followed by agglomerative hierarchical clustering. PLDA has been shown to perform better than cosine scoring (Sell & Garcia-Romero, 2014), operating in a low-dimensional space with the i-vectors to compute the likelihood ratio between them. In the case of diarization as in here, PLDA scores are computed between each of the segments in a recording with all of the other segments in that recording, resulting in a matrix of scores. These are the similarities used in decision-making during clustering.

Agglomerative hierarchical clustering is a bottom-up method as described in Section 2.2. In this project, oracle numbers of clusters are given so that the model knows when to stop merging segments. It should be noted that these mechanisms, along with most diarization back ends, do not actually identify the languages or speakers by name. The segments are clustered in such a way so that all of the segments belonging to the same language or speaker will hopefully cluster together, but the groups are labelled otherwise insignificantly. This can be useful in the case of language diarization as one can train the system on many languages mismatched with the test set, since the model is merely trying to learn to distinguish language in general, not any specific ones. Unfortunately, time constraints prevented this from being pursued in the current project, but could make for some interesting future work.

4. Evaluation Metrics

4.1 False Alarms & False Rejections

To evaluate language recognition in this project, it is formulated as a verification task and can be interpreted as a test of hypotheses:

H_0: X is from hypothesized language L
H_1: X is **not** from hypothesized language L

A series of trials is presented to the system wherein a speech segment X is paired with a hypothesized target language L, and it must decide which hypothesis is correct. Each segment is hypothesized as its correct language in target trials, and then as each of the other incorrect languages in nontarget trials. Given the number of segments S and the number of target languages N, this brings the total number of trials to S x N.

As in hypothesis testing, type I and type II error rates can be an indication of how well the system is performing. Type I errors correspond to false rejections (or false positives) wherein a true H_0 is rejected, while type II errors are false alarms (or false negatives), the acceptance of a false H_0. Dividing each of these by the total number of tests gives the respective rates.

4.1.1 C_{avg}

The average cost (C_{avg}) is an evaluation metric introduced by the NIST LRE, incorporating both false alarm and false rejection rates (NIST, 2007). The metric can be formulated as:

$$C_{avg} = \frac{1}{N}\sum_{l=1}^{N} C_{DET}(L_l) \tag{4.1}$$

where $C_{DET}(L_l)$ is the detection cost per language:

$$C_{DET}(L_l) = C_{miss}P_{tar}P_{miss}(L_l) + C_{fa}(1-P_{tar})\frac{1}{N-1}\sum_{m\neq l} P_{fa}(L_l, L_m) \tag{4.2}$$

$P_{fa}(L_l,L_m)$ is the false alarm probability, and P_{miss} is the false rejection rate, or miss probability. The parameters C_{fa} and C_{miss} define the cost of each type of error, both usually set at 1. Finally, P_{tar} controls the prior probability of seeing a target language, usually set at 0.5 in closed-set evaluations for an equal chance of observation or not.

As the C_{avg} metric penalizes misclassifications with added cost, a system is taken to be performing better at a lower C_{avg} and worse at a higher value.

4.1.2 DET Curves

Detection error tradeoff (DET) curves were introduced by NIST (Martin et al., 1997) as one way to visualize the error tradeoff and performance of detection tasks. This is typically the false alarm probability vs. the miss probability, as described above. The normal deviates of each of the probabilities is plotted, so straight lines indicate underlying normal distributions. Curves closer to the lower left quadrant indicate better performance, with a circle indicating the point at which the average of the error rates is minimized.

4.1.3 EER

The equal error rate (EER) is simply the point at which the false alarm rate is equal to the false reject rate. This is the point at which both error rates are simultaneously minimized, as indicated by the circle on the DET curve. Intuitively, a lower EER indicates better system performance.

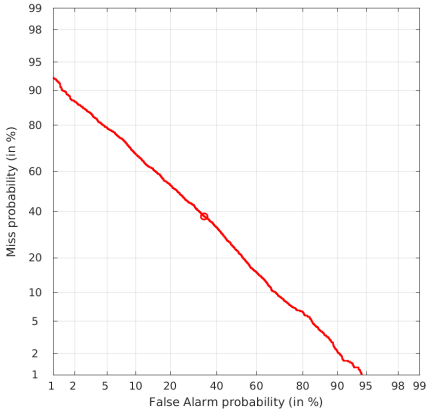


Figure 4.1: Example DET curve.

4.2 DER

Diarization error rate (DER) is another metric devised by NIST, for use in assessing diarization challenges (Galibert, 2013). The intuition of DER is similar to that of word error rate (WER) in ASR, an aggregation of three types of error (incorrect output, missed output, and inserted output) normalized by a common factor in order to be comparable between evaluations. Thus, DER can be formulated as:

$$DER = \frac{confusion + miss + false\ alarm}{total\ reference\ speech\ time} \tag{4.3}$$

where in the context of language diarization, confusion error refers to when the reference language tag and the system-predicted language tag do not match, miss error refers to when speech in the reference is not tagged for language by the system, and false alarm error refers to when there is no speech in the reference but is tagged for language by the system. These individual errors operate on segments of speech, so the durations of segments with an error are summed to give the total time in error.

A greater number of errors will increase the numerator value, and thus increase the overall DER. Hence, a better performing system will minimize the DER.

5. Data

5.1 Evaluation Data

The evaluation data for this project is a collection of wide-band, code-switching conversational recordings. The subjects of the recordings are couples (male/female) in their home, conversing with one another in two or three languages, detailed in Table 5.1. Each person wears their own microphone, resulting in two recordings of the same conversation, to obtain high quality data per speaker. Only one of these recordings is annotated however, as they contain the same content, just recorded from different positions in the room. The original recordings are sampled at 44.1 kHz, a common sample rate for music, but much higher than necessary for speech processing purposes. Thus, all recordings were downsampled to 16 kHz prior to working with them.

Manual annotations were created for only one recording per couple (the male microphone). It is possible to align the two (male and female) recordings to get the offset of the female audio, but since the scope of the project did not involve speaker-specific adaptation, only the male recordings are used as it should still capture the entire dialogue. The annotations include information about who is speaking when (start/end timestamps), and in what language. In addition to the male and female in each recording, non-speech noises are also annotated (such as laughter, coughing, etc.) as well as child speech in a select few.

In order to use the information in the annotation files to evaluate the language recognition and diarization systems, some text processing was done to extract only the segments labelled for man or woman and with a valid language label. This information was used to create the appropriate files for use with Kaldi.

Couple	Code switches	Language 1	Language 2	Language 3	Total duration (hh:mm:ss)
1	332	English 00:39:20	French 00:22:58	Spanish 00:15:23	02:48:40
2	183	English 00:35:23	Spanish 00:46:47	-	02:15:16
3	61	English 00:01:41	German 00:35:24	Spanish 00:01:31	00:58:35
4	36	English 00:51:41	French 00:00:10	Spanish 00:04:57	
5	134	English 00:30:58	Polish 01:17:24	-	02:45:17
6	75	English 00:03:07	Polish 00:29:48	-	00:36:13

Table 5.1: Code-switching data per couple.

Language	Segments	Avg. segment dur.	Total duration (hh:mm:ss)
English (en)	1486	6s	02:42:10
French (fr)	309	4s	00:23:08
German (de)	302	7s	00:35:24
Polish (pl)	1004	6s	01:47:12
Spanish (es)	655	6s	01:08:38
Total	3756	6s	06:36:32

Table 5.2: Code-switching data per language.

5.2 Training Corpora

In an effort to best match the evaluation data, training corpora were chosen to cover the same set of languages and at the same general quality/sample rate (16 kHz). True bilingual or trilingual corpora are scarce, so collections of monolingual data in the specified languages were used. These include the GlobalPhone dataset (Schultz et al., 2013) and

the Euronews corpus (Gretter, 2014). Preliminary experiments with each of these datasets were conducted to determine if one was a better fit to the test data than the other; results in Section 7.1.

5.2.1 GlobalPhone

This data consists of native speaker recordings in environments of minimal background noise. Phrases are read into the microphone, selected from newspapers in the given language. This results in fairly clean data, though it is not spontaneous/conversational. Among the available GlobalPhone languages, four were a match with the evaluation data: French, Spanish, Polish, and German. As there exists a Kaldi recipe for use with this particular corpus, almost all preprocessing was taken care of with pre-written scripts, and all that was left to do was to subset it into the amounts needed. Table 5.3 details the total GlobalPhone data used in this project, sometimes subset into smaller sets for different experiments (specified in Section 7).

Language	**Segments**	**Avg. segment dur.**	**Total duration (hh:mm:ss)**
French (fr)	2000	9s	05:10:23
German (de)	2000	6s	03:39:43
Polish (pl)	2000	8s	04:52:04
Spanish (es)	2000	12s	06:41:50
Total	8000	9s	20:24:00

Table 5.3: GlobalPhone data used.

5.2.2 Euronews

This corpus is compiled from news clips taken from Euronews[1] web and TV. Recordings are a mix of planned/read speech and more conversational interview bits. Much of the speech for languages other than English has been dubbed over in the target language. All five evaluation languages are present in this corpus: English, French, German, Polish, and Spanish. Reference alignment files came included with the data, aligning individual words

[1] https://www.euronews.com

Language	Segments	Avg. segment dur.	Total duration (hh:mm:ss)
English (en)	4000	9s	10:01:36
French (fr)	4000	9s	10:17:55
German (de)	3750	9s	10:12:34
Polish (pl)	3750	9s	09:59:02
Spanish (es)	3750	9s	10:01:33
Total	19250	9s	50:32:40

Table 5.4: Euronews data used.

with their temporal position in the recording. Using these alignments, longer segments were created by concatenating adjacent words - the threshold used in comparing the end of one word to the beginning of another was half a second. As this still left many words on their own or in very short segments, the data is usually subset into segments five seconds or longer for the experiments in Section 7. Table 5.4 outlines the total Euronews data used in this project, sometimes subset into smaller sets for different experiments. Note that the data used for training the DNN is separate from this - as it required transcriptions, shorter segments were used in order to avoid OOVs.

5.2.3 Lexicons

In order to train the DNN to predict senones, the training data must have corresponding transcriptions and a lexicon. Both GlobalPhone and Euronews come with transcriptions, but only GlobalPhone includes lexicons. Since the four languages taken from GlobalPhone are also present in Euronews, the two datasets can share these lexicons, leaving the Euronews English data as the only language without one. In this case, the pre-compiled English lexicon used in (Bell et al., 2015) was used. As all of the languages are combined in training the DNN (only one output layer), all lexicons must be aligned to use the same set of phone labels. The GlobalPhone German, Polish, and Spanish conventions were already consistent with each other, but the French and English lexicons had differing phone set conventions. Luckily, GlobalPhone documentation was included to map the French and Spanish phone sets to the IPA, facilitating an alignment. Similarly, the English lexicon appeared to use ARPABET or similar conventions, which is also aligned to the IPA. The detailed alignment of the aforementioned lexicons is included in Appendix A.

6. Methodology/Implementation

Experiments were conducted using the Kaldi speech recognition toolkit (Povey et al., 2011), with the help of some of the built-in recipes.

6.1 GMM-UBM Setup

The baseline GMM-UBM i-vector system is based on the `lre07/v1` recipe, originally written to evaluate the NIST LRE 2007 data (telephone speech). The general pipeline is as follows (visualized in Figure 6.1):

- Initial feature extraction consists of extracting MFCC+SDC features from both the training and evaluation data, followed by computing VAD decisions for each frame. The MFCC configuration file was modified for wide-band audio, by increasing the sample frequency to 16,000.
- The (original) UBM training involves training a diagonal covariance 2048-component GMM on a subset of the training data to initialize the UBM, and then uses this to bootstrap training a full-covariance GMM on all of the training data. However, since this project involves a relatively small sized training dataset, just the diagonal covariance GMM is used for the UBM, trained on all of the training data. Experiments confirm that using a full covariance GMM results in worse performance.
- An i-vector extractor is trained: sufficient statistics are computed from the UBM and accumulated in order to perform an EM update in estimating the total variability matrix T. This trains for five iterations, which should be sufficient for parameter convergence (Reynolds et al., 2000). The extracted i-vectors are 600-dimensional, found as optimal in (Martinez et al., 2011).
- Once i-vectors have been extracted, a multi-class logistic regression model is trained on the training data i-vectors to discriminate between the possible target languages.

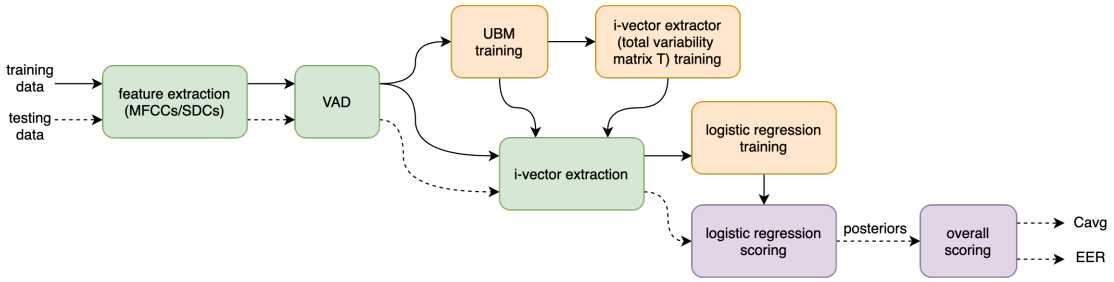


Figure 6.1: Kaldi lre07 recipe pipeline. Solid arrows represent derivatives of training data while dotted lines represent test data. Green boxes apply to both, orange to train data, and purple to test data.

- Finally, the logistic regression model is used with the test set i-vectors to generate posterior probabilities for each trial (as described in Section 4.1). Scoring is completed to produce C_{avg} and EER scores (EER scoring is not included in the recipe but there is a Kaldi binary for it).

6.2 DNN-UBM Setup

The DNN-UBM i-vector system follows the `lre07/v2` recipe, which is the previous recipe adapted for use with a DNN-UBM. This recipe corresponds to the work carried out in (Sarma et al., 2018). The general pipeline of this recipe is similar to the one above, with the additional step of training the DNN, and related modifications.

- A time delay deep neural network is trained prior to beginning. This neural network architecture is based on that in (Snyder et al., 2015). The TD-DNN is trained to predict senones, and in this project the output layer dimension is modified to match the number of Gaussian components in the baseline GMM-UBM (2048).

- Feature extraction includes the extraction of 'high-resolution' MFCCs in addition to the standard MFCC set. The high-resolution MFCCs include all 40 filterbank coefficients, intended for use with the DNN.

6.3 Diarization Setup

The diarization system is based on the `callhome_diarization` recipe, originally written to evaluate Callhome data (telephone speech) for speaker diarization. This recipe mostly

follows the work in (Sell & Garcia-Romero, 2014). The same UBMs and i-vector extractors trained for use in the language recognition experiments are re-purposed here, with other modifications to adapt the recipe to language diarization noted below. A brief description of the differing backend is also included.

- The code-switching test dataset is split into two partitions for use in whitening the PLDA model (they are later combined again before scoring).
- I-vector extraction is done on subsegments of data (3s train i-vectors, 1.5s test i-vectors) to simulate the fixed-length segmentation approach.
- The script that extracts i-vectors was modified to work with the same features as were used in the language recognition experiments, namely, SDCs.
- Data files that map utterances to speakers are modified to map utterances to recording files as is expected by the i-vector extractor script.
- The PLDA model is trained on language labels instead of speaker labels.
- I-vectors are then scored with PLDA and clustered according to agglomerative hierarchical clustering with oracle cluster numbers.
- Finally, the output from the two partitions is combined and scored against the reference file to compute DER.

7. Experiments

7.1 GMM-UBM Baseline

Prior to establishing an official baseline, preliminary experiments were performed to address the question of which training corpus is a better fit to the test data. To do a fair comparison of GlobalPhone to Euronews, both the test set and the Euronews training set were restricted to the only four languages available in GlobalPhone. An additional 'combined' training condition was tested, which comprises an equal amount of data from each corpus. Languages are equally balanced in all training conditions. Results are presented in Table 7.1.

It is interesting to observe that while the C_{avg} is lower for Euronews training data, the EER is lower for the GlobalPhone data. Upon inspecting the confusion matrices (Appendix B), it is evident that in the GlobalPhone system, the predictions are quite skewed towards Polish - most segments are labelled as such even when they aren't. This skew is also highlighted in the DET curve (Figure 7.1), where the GlobalPhone line is not as straight, an indication of an underlying non-normal distribution. As EER does not distinguish between languages, it is possible for it to appear unaffected, while C_{avg} is penalized due to poor performance on a particular language. Euronews on the other hand shows a more moderate distribution of errors (better C_{avg}), if a bit less suited to the test data (worse EER). Naturally then, a combination of the two datasets yields the best results. Here their

Train set	**Test set**	**C_{avg}**	**EER**
GlobalPhone (10hrs)	Code-switching (fr/de/pl/es)	49.29	36.17
Euronews (10hrs)	Code-switching (fr/de/pl/es)	44.48	47.62
Combined (10hrs)	Code-switching (fr/de/pl/es)	**44.19**	**36.12**

Table 7.1: Preliminary results on training corpora suitability.

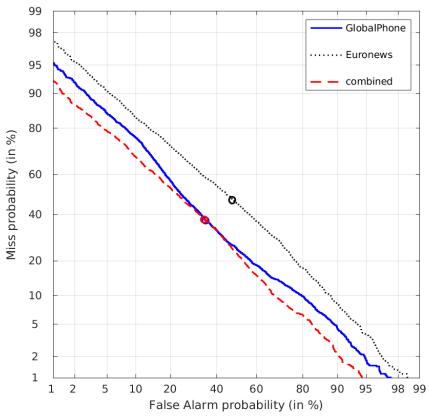


Figure 7.1: DET curves for preliminary baseline experiments.

complementarity is exploited, and perhaps the increased variety also encourages better generalization to unseen test data.

Next, the baseline experiment is extended to the full set of languages (the previous four with English) and a larger amount of training data. Since English is not present in the GlobalPhone corpus, in combined datasets 2x the Euronews data for English is used whereas the other languages continue to be an even split of Euronews/GlobalPhone. Based on the results of the previous tests, the combined training set is expected to be the choice composition, but a Euronews experiment is run just to confirm results under the updated conditions (a GlobalPhone test is excluded due to lack of English). Languages continue to be equally balanced. Results presented in Table 7.2.

As evidenced, the combined training condition continues to be optimal, with further reductions in C_{avg} and EER. The addition of more training data is likely the culprit, as the only other modification was the addition of a fifth language, which adds more challenge in discrimination. The increase in data introduces more variety of language samples, especially in terms of speakers and content, potentially contributing to a more robust system. Another notable result is the more than 10% drop in EER in the Euronews

Train set	**Test set**	C_{avg}	**EER**
Euronews (50hrs)	Code-switching (en/fr/de/pl/es)	45.24	35.38
Combined (50hrs)	Code-switching (en/fr/de/pl/es)	**42.74**	**34.05**

Table 7.2: Extended baseline results.

condition. It's possible that only 10 hours of training data in the initial experiment led to some overfitting.

Thus, using the combined training set, the official baseline result is a C_{avg} of 42.74 and an EER of 34.05%. Overall, these results appear to be quite poor in comparison with published literature (usually under 20 both C_{avg} and EER for short segment (3s) conditions), but it is noteworthy that most language recognition work has been done with standard test datasets such as those of the NIST LREs, which provide a relatively good match in style and content to the training data. Language recognition work outside of the NIST LRE domain tends to either be done with other established corpora with train/test splits within them, or self-collected corpora, also with a train/test split within. So there is usually some consistency within the training and testing data, which is not necessarily the case in this project. To investigate this effect, another test set was compiled with Euronews data (disjoint from the training set), with as close specifications to the code-switching test set as possible, including the same distribution of languages (Table 7.3). This control case yields a C_{avg} of 1.45 and EER of 0.59%. Thus, the baseline system appears to perform very well on a well-matched test set, supporting the notion that the poor performance on the code-switching data is due to a train/test mismatch.

Test set	**Segments**	**Total time (hh:mm)**	**Avg. seg.**	C_{avg}	**EER**
Code-switching	3756	06:36	6s	42.74	34.05
Euronews	3756	05:49	5s	1.45	0.59

Table 7.3: Results compared to control test set.

7.2 DNN-UBM

As the DNN-UBM architecture is based on that of (Sarma et al., 2018), their original training conditions are first investigated (monolingual DNN train set), before experimenting with multilingual data as in the baseline.

7.2.1 Monolingual

Since the sole purpose of a DNN-UBM is to provide senone posterior probabilities, in theory, as long as the training data contains a sufficient representation of senones that are likely to be observed in the test data, it shouldn't matter from which languages they come from (the DNN will not be aware of language anyway). This is the approach taken in (Sarma et al., 2018) which trains the DNN-UBM on 1800 hours of monolingual English data. Following this, the sufficient statistics are computed with data from 50 languages, and i-vectors are extracted for the 14 which appear in the test data. This appears to work relatively well, showing improvements over their baseline GMM-UBM. Thus, this method is attempted here in hopes of seeing similar results.

It should be noted that the DNN output layer dimension (which corresponds to the number of senones, and consequently, the number of UBM components) is not specified in (Sarma et al., 2018), though upon inspection of the corresponding code in Kaldi, it appears to be 7000. In order to be consistent with the baseline and perhaps provide a more fair comparison between methods, this output layer is modified to 2048 (as in the GMM-UBM) in this project.

The training data for the DNN consists of 50 hours of Euronews English speech, followed by i-vector extractor training with the same combined 50 hour dataset as in the GMM-UBM baseline. I-vectors are extracted for the combined training set for logistic regression training. Results on both the code-switching test set and the Euronews test set are presented in Table 7.4.

Curiously, the system evaluated on the code-switching data performs worse than the baseline model, yet tested with the Euronews evaluation set, sees an improvement. The improvement on the Euronews test set (by 0.78 C_{avg} and 0.11 EER) is small in absolute terms, but considering the already very low values of both metrics, may be considered non-trivial. This demonstrates at least some benefit to incorporating the phonetic knowledge contributed by the DNN, if small.

DNN train set	**I-vector train set**	**Test set**	$\mathbf{C_{avg}}$	**EER**
Euronews (en) (50hrs)	Combined (50hrs)	Code-switching	43.69	38.79
Euronews (en) (50hrs)	Combined (50hrs)	Euronews	0.67	0.48

Table 7.4: Monolingual DNN-UBM results.

Language A	Language B	Similarity A	Similarity B	Avg. similarity
French	Polish	63.16%	66.67%	64.91%
French	German	63.16%	58.54%	60.85%
German	Polish	56.1%	63.89%	59.99%
German	Spanish	58.54%	60%	59.27%
Polish	Spanish	61.11%	55%	58.06%
English	Polish	48.94%	63.89%	56.41%
English	French	44.68%	55.26%	49.97%
French	Spanish	50%	47.5%	48.75%
English	German	40.43%	46.34%	43.38%
English	Spanish	38.3%	45%	41.65%

Table 7.5: Similarity scores per language pair, based on phone set overlap (See Appendix A). Similarity A is A's overlap with B, and vice versa - these are different values as the languages have different total numbers of phones.

In contrast, the results on the code-switching data deteriorate by 0.95 in C_{avg} and 4.74 in EER. Upon closer inspection of the confusion matrices (Appendix B), it is interesting to point out first that the baseline model's initial poor performance was mostly due to an inability to recognize French, German, and Polish (each had quite high false reject rates). Segments in these languages were instead identified as Spanish or English, with high frequency. A possible reason for this could be that since these three languages are the most similar to each other, in terms of phonetic overlap (Table 7.5), they require more specific class boundaries to separate them from one another (and the rest of the languages). Thus, it becomes more difficult to classify a test segment (especially one that does not match the train set well) as one of these languages when the boundaries are so confined. On the other hand, English and Spanish - at opposite ends of the spectrum - are likely easier to discriminate from the others and so have more lax criteria. This could be why we see the French, Polish, and German segments corralled into one of these two groups - as the system has not seen the code-switching data before, the more specific information used in discriminating these languages is potentially not present (or it is relying on other cues besides phonetics), leaving them to be classified as one of the more accepting English or Spanish. Regarding the confusion matrix for the DNN experiment, it appears that introducing specific phonetic knowledge has not alleviated any errors, but

rather reallocated them. Since the DNN-UBM was trained specifically on English data, it is no surprise to see that the false alarm rate for English has decreased, by 21%. Conversely, the false alarm rate for Polish has increased. A well-defined phone set for English could mean that the English class boundaries are now more clear, resulting in the same effect previously observed in the baseline experiment. This leaves Polish, the language most similar to English (Table 7.5), to now define that end of the spectrum. Thus, resulting in the largely similar C_{avg} and EER scores.

7.2.2 Multilingual

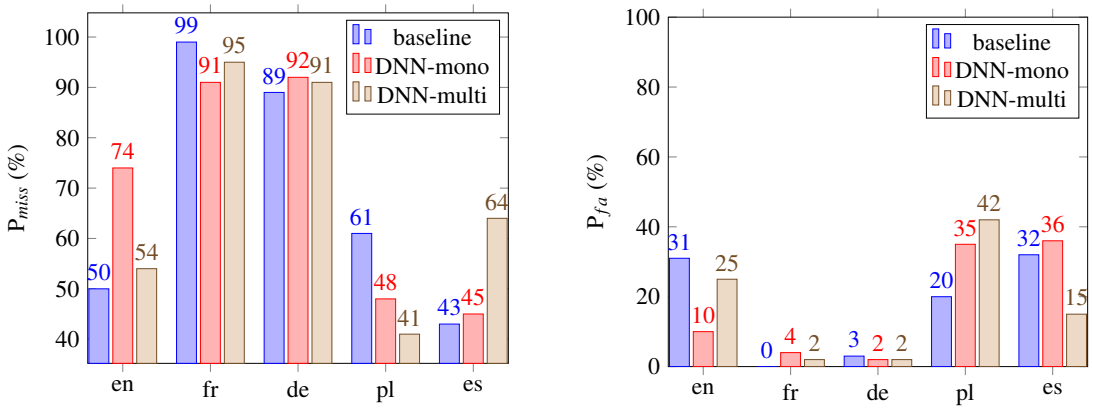
As the previous experiment showed some promise in the incorporation of phonetic knowledge (mainly observed in the Euronews test set), the DNN-UBM is attempted again with a balanced training set - an equal amount of data per language (total 50 hours). This training set has the same partition as the baseline combined set, but contains different audio segments in an effort to minimize OOV words. This was largely successful, with no apparent OOVs at the time of training. The sufficient statistics however are still computed with the same combined set as in the baseline and the previous DNN experiment. Results are given in Table 7.6.

Continuing the trend of improvement on the Euronews test set, the C_{avg} and EER are further reduced with the multilingually-trained DNN-UBM. It is again a small reduction in absolute terms, but an improvement nonetheless in consideration of how low they were previously. This further supports the notion that incorporating specific phonetic knowledge into the UBM aids language identification. In this case, the senone coverage is extended with the addition of four more language phone inventories.

The code-switching test set also shows some improvement this time, with a reduction in EER over both the monolingual DNN-UBM and the baseline GMM-UBM. The C_{avg} is also an improvement over the monolingual DNN-UBM, yet still slightly over that of the baseline (by a mere 0.41). The confusion matrix now shows perhaps a better definition

DNN train set	**I-vector train set**	**Test set**	C_{avg}	**EER**
New combined (50hrs)	Combined (50hrs)	Code-switching	43.15	33.36
New combined (50hrs)	Combined (50hrs)	Euronews	0.41	0.21

Table 7.6: Multilingual DNN-UBM results.



(a) P_{miss} rates.

(b) Average P_{fa} rates.

Figure 7.2: P_{miss} and P_{fa} rates across languages and experiments.

of the spectrum of languages - English and Spanish on opposite ends have lower false alarm rates, while Polish appears to cover the middle with a higher false alarm rate. This is because French and German, also "mid-spectrum" with more than 55% phonetic overlap, are largely being classified as Polish by mistake. This result is perhaps closer to what one might expect to see over the baseline; it seems that overall the addition of phonetic knowledge from the DNN has helped to refine the broad class boundaries. Within that though still requires some improvement (the distinction between phonetically similar languages). It is also possible that there is something inherent in the code-switching data that is causing these results; Polish always appears with English in conversations which could be a reason this language pair gets confused (in addition to their apparent phonetic similarity). French, German, and Polish however never appear in the same conversation, so it is more likely that their phonetic similarity is the cause of confusion rather than speaker or channel effects.

7.3 Diarization

As described in Section 6.3, the diarization system architecture makes use of the previous UBMs and i-vector extractors. The GMM-UBM is used as a baseline again, with just the multilingual DNN-UBM experimented on due to superior performance on the language recognition task. This is followed by PLDA scoring, AHC, and evaluation.

7.3.1 GMM-UBM

First investigated is the diarization performance with the baseline GMM-UBM architecture. It has already been shown that this system does not yield the best results on the code-switching data in the more simple task of language recognition, so increasing the difficulty by adding segmentation will likely add further error.

To attempt a full and true diarization effort, the audio is first put through an energy-based VAD and adjacent detections concatenated to form speech segments. This naive VAD does not discriminate between speech noise and other kinds of noise, so there will inherently be error involved where a detected speech segment does not contain speech at all, but some other kind of noise such as laughter, coughing, humming, etc. Unfortunately time constraints prevented the use of a proper VAD tool to detect speech, although there may have still been inherent error involved with that due to the existence of child speech in the code-switching data, which is not annotated for language but is nonetheless speech.

Next, segmentation is done in a naive fixed-length fashion, as described in Section 2.2. A sliding window of 1.5 seconds was used with a period of 0.75 seconds. As the average segment length in the code-switching data is 6 seconds, it is reasonable to assume that most boundaries will be accurately detected, at least within half a second. The issue is then how well these short segments can be clustered within themselves properly. Two clustering approaches are taken, both using the oracle number of clusters expected - the first is simply by language (the correct number of languages is given per recording), and the second is by speaker-language pairs. Results are in Table 7.7.

System performance is expectedly poor, with DERs above 100%. Considering the individual error rates within, the missed speech and false alarm speech errors are consistent with the VAD strategy used (uninformed). The naive energy-based detector picks up all noise, including non-speech noises- contributing to the dramatically high false alarm rate. This is the main contributor to the poor DERs. For the same reason, the missed speech

VAD	Clusters	Missed speech	FA speech	Confused speech	DER
Auto	Language	2.2%	70.5%	44.1%	116.77
Auto	Speaker-language	2.2%	70.5%	59.2%	131.93

Table 7.7: Baseline diarization results. Language clusters are 2-3 per recording while speaker-language clusters are 4-6 per recording.

VAD	Clusters	Missed speech	FA speech	Confused speech	DER
Oracle	Language	0.5%	0%	42.7%	43.17
Oracle	Speaker-language	0.4%	0%	59.8%	60.12

Table 7.8: Baseline diarization results with oracle VAD.

rate is naturally low. Confused speech on the other hand, refers to speech segments that were not attributed to the correct cluster. As the clustering mechanism operates on the PLDA scores, it would seem that PLDA is not well-characterizing the data. As similar performance was observed on the language recognition task, perhaps it is caused by the same reason - train/test mismatch.

The oracle speaker-language clusters are also tested as a proxy to try and observe the more 'proper' partition of languages - diarization is usually done on a per conversation or recording basis, so it does not make the most sense to try to cluster languages across recordings. Instead, targeting the speaker-language pairs alleviates this assumption. Both the DER and the confused speech rate increase as a result, potentially due to the wider margin of error.

To observe what the DERs might look like without the terrible VAD, the experiments were re-run with the oracle VAD boundaries (as previously manually annotated) (Figure 7.8).

Expectedly, both the missed speech and false alarm rates essentially drop to 0, leaving the confused speech as the only source of real error. DERs drop dramatically due to this. The confused speech rate drops by 1.4% for the language-only clusters, while increasing 0.6% in the speaker-language pairs. This is a curious result; the reduction in error is perhaps due to the removal of interfering false alarm segments, while the increase in error is marginal but could also be caused by the same reason.

7.3.2 DNN-UBM

Finally, the DNN-UBM model is attempted at the diarization task. Showing some improvements over the baseline model in the language recognition experiments, it will hopefully have a similar effect here. Results in Table 7.9.

The same increase in confused speech rate is observed for the oracle speaker-language clusters over the auto VAD speaker-language clusters, by more of a margin this time. A

bit surprisingly, the error rates of the oracle language clusters are both worse than those produced by the baseline GMM-UBM. Since the objective of diarization is to cluster within recordings (not just by languages), perhaps the incorporation of phonetic knowledge helps the speaker-language clustering, the true target groups, and in doing so introduces more error on the solo language task.

VAD	**Clusters**	**Missed speech**	**FA speech**	**Confused speech**	**DER**
Auto	Language	2.2%	70.5%	40.7%	113.35
Auto	Speaker-language	2.2%	70.5%	56.7%	129.36
Oracle	Language	0.6%	0%	45.2%	45.83
Oracle	Speaker-language	0.5%	0%	58.6%	59.03

Table 7.9: Diarization results with the multilingually-trained DNN-UBM.

8. General Discussion & Analysis

A main takeaway from all experimental results is probably: improvements are limited if there is a large train/test mismatch. Potentially the largest source of discrepancy between the train and test sets was the style of speech - conversation between two familiar people will be much more casual and likely a lot quicker than broadcast news or carefully recorded read speech. This kind of data will likely be hard to generalize to quicker speech where the phones and sounds are not as enunciated. Other factors potentially involved include the relatively small size and imbalance in the test set. The languages are not well-balanced and it's possible this is a contributing factor, especially to the confusion in the language recognition systems. Another consideration may be the accent or native speaker status of the speakers in the training data vs. test data. Both of these factors influence pronunciation, and thus the phonetic range. In order to improve performance in this regard, sourcing training data that is more similar to the test set is ideal. Data augmentation and other low-resource techniques could also be tried.

The small improvements from the DNN-UBMs don't reject the hypothesis that they help, but they also aren't overwhelming gains. Perhaps the experimental conditions need to be scaled up by an order of magnitude in order to see more definitive results. There is also the somewhat caveat that a GMM-UBM trained in an ASR way to predict phones was not experimented with, instead opting for a DNN. This option was considered, but ultimately decided that it would not fit within the time frame of the project. Though this setup has been reported on for speaker recognition (with poor results), it may still be worth trying for language recognition as the purpose is mainly to model the general phone set of a language rather than specific variances in speaker pronunciations as is the purpose in speaker recognition.

As for diarization, the scope of this investigation really limited the results and findings. To attempt this again more thoroughly, a competent VAD model should be used to begin. Since there are two audios for each recording, this opens up the possibility of using an audio

synchronization or beamforming technique to take advantage of them both. The CHiME-6[1] challenge contains a speaker diarization task that presents similar audio type and situation. Resources and results from this event (released in Kaldi) may be of relevance to the manner of approach should this code-switching data be revisited. Another technique that could be of use is the resegmentation of clusters after the initial partition. This is the only part present in the corresponding paper that has not yet been implemented in the Kaldi recipe, so was not attempted. Overall, if the language recognition results were any indication, there was already associated error with language characterization, so improvements to the initial train set i-vectors may result in much better DERs.

[1] https://chimechallenge.github.io/chime6/

9. Conclusion

This project investigated the language recognition and diarization performance of an i-vector based architecture trained on general corpora and tested on novel code-switching speech collected locally. The language recognition abilities of the system do not perform well on the mismatched data, but are highly accurate on a control, matched test dataset. This suggests that acquiring more similar training data will help to decrease error.

The positive results on the matched test data confirm that i-vectors are a good way of approaching the language recognition problem without the need to infuse further linguistic knowledge. However, doing so can help, as demonstrated by the DNN-UBM results. They also seem to be adequate for use in a language diarization scheme, given accurate VAD and segmentation. The preliminary experiments done here show some potential in the similarity and clustering operations - error seems to be similar to the error seen in the previous task, which was attributed to train/test data mismatch. Unfortunately, the training data consists of all monolingual recordings so a matched test set cannot be compiled for diarization. Training on more similar data to the test set and adopting more accurate VAD methods will likely yield a better indication of the viability of this approach.

Bibliography

Ambikairajah, E., Li, H., Wang, L., Yin, B., & Sethu, V. (2011). Language identification: A tutorial. *IEEE Circuits and Systems Magazine*, *11*(2), 82–108.

Bell, P., Gales, M. J., Hain, T., Kilgour, J., Lanchantin, P., Liu, X., McParland, A., Renals, S., Saz, O., Wester, M., Et al. (2015). The mgb challenge: Evaluating multi-genre broadcast media recognition, In *2015 ieee workshop on automatic speech recognition and understanding (asru)*. IEEE.

Bhattacharjee, U., & Sarmah, K. (2012). Gmm-ubm based speaker verification in multilingual environments. *International Journal of Computer Science Issues (IJCSI)*, *9*(6), 373.

Bozonnet, S., Wang, D., Evans, N., & Troncy, R. (2011). Linguistic influences on bottom-up and top-down clustering for speaker diarization, In *2011 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE.

Dehak, N. (2009). *Discriminative and generative approaches for long-and short-term speaker characteristics modeling: Application to speaker verification* (Doctoral dissertation). École de technologie supérieure.

Dehak, N., Kenny, P. J., Dehak, R., Dumouchel, P., & Ouellet, P. (2010). Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, *19*(4), 788–798.

Dehak, N., Torres-Carrasquillo, P. A., Reynolds, D., & Dehak, R. (2011). Language recognition via i-vectors and dimensionality reduction, In *Twelfth annual conference of the international speech communication association*.

Evans, N., Bozonnet, S., Wang, D., Fredouille, C., & Troncy, R. (2012). A comparative study of bottom-up and top-down approaches to speaker diarization. *IEEE Transactions on Audio, speech, and language processing*, *20*(2), 382–392.

Ferrer, L., Lei, Y., McLaren, M., & Scheffer, N. (2015). Study of senone-based deep neural network approaches for spoken language recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *24*(1), 105–116.

Galibert, O. (2013). Methodologies for the evaluation of speaker diarization and automatic speech recognition in the presence of overlapping speech., In *Interspeech*.

Gonzalez-Dominguez, J., Lopez-Moreno, I., Sak, H., Gonzalez-Rodriguez, J., & Moreno, P. J. (2014). Automatic language identification using long short-term memory recurrent neural networks, In *Fifteenth annual conference of the international speech communication association*.

Gretter, R. (2014). Euronews: A multilingual benchmark for asr and lid, In *Fifteenth annual conference of the international speech communication association*.

Hansen, J. H., & Hasan, T. (2015). Speaker recognition by machines and humans: A tutorial review. *IEEE Signal processing magazine*, *32*(6), 74–99.

Kenny, P. (2005). Joint factor analysis of speaker and session variability: Theory and algorithms.

Lei, H. (2011). Joint factor analysis (jfa) and i-vector tutorial. *ICSI. Web*, *2*.

Lei, Y., Scheffer, N., Ferrer, L., & McLaren, M. (2014). A novel scheme for speaker recognition using a phonetically-aware deep neural network, In *2014 ieee international conference on acoustics, speech and signal processing (icassp)*.

Li, H., Ma, B., & Lee, K. A. (2013). Spoken language recognition: From fundamentals to practice. *Proceedings of the IEEE*, *101*(5), 1136–1159.

Li, W., Fu, T., & Zhu, J. (2015). An improved i-vector extraction algorithm for speaker verification. *EURASIP Journal on Audio, Speech, and Music Processing*, *2015*(1), 18.

Lopez-Moreno, I., Gonzalez-Dominguez, J., Plchot, O., Martinez, D., Gonzalez-Rodriguez, J., & Moreno, P. (2014). Automatic language identification using deep neural networks, In *2014 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE.

Lyu, D.-C., Chng, E.-S., & Li, H. (2013). Language diarization for code-switch conversational speech, In *2013 ieee international conference on acoustics, speech and signal processing*. IEEE.

Martin, A., Doddington, G., Kamm, T., Ordowski, M., & Przybocki, M. (1997). *The det curve in assessment of detection task performance* (tech. rep.). National Institute of Standards and Technology, Gaithersburg MD.

Martinez, D., Plchot, O., Burget, L., Glembek, O., & Matějka, P. (2011). Language recognition in ivectors space, In *Twelfth annual conference of the international speech communication association*.

Moattar, M. H., & Homayounpour, M. M. (2012). A review on speaker diarization systems and approaches. *Speech Communication*, *54*(10), 1065–1103.

NIST. (2007). The 2007 nist languag recognition evaluation plan (lre07).

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Et al. (2011). The kaldi speech recognition toolkit, In *Ieee 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society.

Reynolds, D. A. (1997). Comparison of background normalization methods for text-independent speaker verification, In *Fifth european conference on speech communication and technology*.

Reynolds, D. A., Quatieri, T. F., & Dunn, R. B. (2000). Speaker verification using adapted gaussian mixture models. *Digital signal processing*, *10*(1-3), 19–41.

Richardson, F., Reynolds, D., & Dehak, N. (2015). A unified deep neural network for speaker and language recognition. *arXiv preprint arXiv:1504.00923*.

Sarma, M., Sarma, K. K., & Goel, N. K. (2018). Language recognition using time delay deep neural network. *arXiv preprint arXiv:1804.05000*.

Schultz, T., Vu, N. T., & Schlippe, T. (2013). Globalphone: A multilingual text & speech database in 20 languages, In *2013 ieee international conference on acoustics, speech and signal processing*. IEEE.

Sell, G., & Garcia-Romero, D. (2014). Speaker diarization with plda i-vector scoring and unsupervised calibration, In *2014 ieee spoken language technology workshop (slt)*. IEEE.

Snyder, D., Garcia-Romero, D., McCree, A., Sell, G., Povey, D., & Khudanpur, S. (2018). Spoken language recognition using x-vectors., In *Odyssey*.

Snyder, D., Garcia-Romero, D., & Povey, D. (2015). Time delay deep neural network-based universal background models for speaker recognition, In *2015 ieee workshop on automatic speech recognition and understanding (asru)*. IEEE.

Song, Y., Jiang, B., Bao, Y., Wei, S., & Dai, L.-R. (2013). I-vector representation based on bottleneck features for language identification. *Electronics Letters*, *49*(24), 1569–1570.

Soufifar, M., Cumani, S., Burget, L., Et al. (2012). Discriminative classifiers for phonotactic language recognition with ivectors, In *2012 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE.

Spoorthy, V, Thenkanidiyoor, V., & Dinesh, D. A. (2018). Svm based language diarization for code-switched bilingual indian speech using bottleneck features., In *Sltu*.

Tang, Z., Wang, D., Chen, Y., Li, L., & Abel, A. (2017). Phonetic temporal neural model for language identification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *26*(1), 134–144.

Tian, Y., He, L., Liu, Y., & Liu, J. (2016). Investigation of senone-based long-short term memory rnns for spoken language recognition., In *Odyssey*.

Torres-Carrasquillo, P. A., Singer, E., Kohler, M. A., Greene, R. J., Reynolds, D. A., & Deller Jr, J. R. (2002). Approaches to language identification using gaussian mixture models and shifted delta cepstral features, In *Seventh international conference on spoken language processing*.

Vaheb, A., Choobbasti, A. J., Najafabadi, S. M., & Safavi, S. (2018). Investigating language variability on the performance of speaker verification systems, In *International conference on speech and computer*. Springer.

Xu, Y., McLoughlin, I., Song, Y., & Wu, K. (2016). Improved i-vector representation for speaker diarization. *Circuits, Systems, and Signal Processing*, *35*(9), 3393–3404.

Zajíc, Z., Kunešová, M., & Radová, V. (2016). Investigation of segmentation in i-vector based speaker diarization of telephone speech, In *International conference on speech and computer*. Springer.

Zissman, M. A. (1996). Comparison of four approaches to automatic language identification of telephone speech. *IEEE Transactions on Speech and Audio Processing*, *4*(1), 31–34.

A. Lexicon Alignment

The FINAL column defines the phone set used in training the multilingual DNN, mapped to the IPA for reference. There was a discrepancy with one Polish (PL-O) phone in particular which was changed accordingly (PL-A). Original French (FR-O) and English (EN-O) conventions are given for reference, along with their new alignments (FR-A, EN-A).

FINAL	IPA	DE	ES	PL-O	PL-A	FR-O	FR-A	EN-O	EN-A
a	a	a	a	a	a	a	a		
a+	aˈ		a+						
a~	ã					A~	**a~**		
aa	ɑ							aa	aa
ae	ɛɪ	ae							
aE	æ							ae	**aE**
ah	ʌ							ah	ah
aI	ai	aI	aI						
aih	aɪ							ay	**aih**
al	aː	al							
ao	ɒ							oh	**ao**
atu	ɐ	atu							
aU	au	aU	aU						
aUU	aʊ							aw	**aUU**
b	b	b	b	b	b	B	**b**	b	b
C	ç	C							
D	ð		D					dh	**D**
d	d	d	d	d	d	D	**d**	d	d
dZ	d͡ʒ			dZ	dZ			jh	**dZ**
dz	d͡z			dz	dz				
dzj	d͡ʑ			dzj	dzj				
e	e	e	e	e	e	e	e		
e+	eˈ		e+						
eetu	eə							ea	**eetu**
eI	ei		eI						
eih	eɪ							ey	**eih**
el	eː	el							
eo5	ɛ			eo5	eo5	E	**eo5**	eh	**eo5**
eo5~	ɛ̃					E~	**eo5~**		
er	ɜˠ							er	er
etu	ə	etu				AX	**etu**	ax	**etu**

eU	eu		eU	eU						
f	f		f	f	f	f	F	**f**	f	f
G	ɣ			G						
g	g		g	g	g	g	G	**g**	g	g
h	h		h		h	h	h	h	hh	**h**
H	ɥ						H	H		
i	i		i	i	i	i	i	i	iy	**i**
i+	iˈ			i+						
i2	ɨ				i2	i2				
ih	ɪ								ih	ih
ihetu	ɪə								ia	**ihetu**
il	iː		il							
j	j		j	j	j	j	J	**j**	y	**j**
k	k		k	k	k	k	K	**k**	k	k
L	ʎ			L						
l	l		l	l	l	l	L	**l**	l	l
l+	l̩								el	**l+**
m	m		m	m	m	m			m	m
M	ɱ						M	M		
m+	m̩								em	**m+**
n	n		n	n	n	n	N	**n**	n	n
N	ŋ								ng	**N**
n+	n̩								en	**n+**
n~	ɲ			n~	n~	n~	NJ	**n~**		
ng	ɴ		ng	ng			NG	**ng**		
o	o		o	o	o	o	o	o		
o+	oˈ			o+						
o~	õ						o~	o~		
o2	ø						EU	**o2**		
oc5	ɔ				oc5	oc5	O	**oc5**	ao	**oc5**
oc5ih	ɔɪ								oy	**oc5ih**
oE	œ						AE	**oE**		
oe	œ		oe				OE	**oe**		
oe~	œ̃						OE~	**oe~**		
oel	œː		oel							
oI	oi			oI						
ol	oː		ol							
oUU	oʊ								ow	**oUU**
p	p		p	p	p	p	P	**p**	p	p
r	r		r	r	r	r				
R	ʁ						R	R		
rf	ɾ			rf						

rr	ɹ							r	**rr**
S	ʃ	S		S	S	SH	**S**	sh	**S**
s	s	s	s	s	s	S	**s**	s	s
sj	ɕ			sj	sj				
T	θ		T					th	**T**
t	t	t	t	t	t	T	**t**	t	t
tS	tʃ		tS	tS	tS			ch	**tS**
ts	ts / t͡s	ts		c	**ts**				
tsj	t͡ɕ			tsj	tsj				
u	u	u	u	u	u	u	u	uw	u
u+	uˈ		u+						
ue	y	ue				y	**ue**		
uel	yː	uel							
ul	uː	ul							
UU	ʊ							uh	**UU**
uu	ɯ					W	**uu**		
UUetu	ʊə							ua	**UUetu**
V	β		V						
v	v	v		v	v	V	**v**	v	v
w	ʊ		w	w	w				
W	w							w	**W**
x	x	x	x						
Z	ʒ			Z	Z	ZH	**Z**	zh	**Z**
z	z	z	z	z	z	Z	**z**	z	z
zj	ʑ			zj	zj				

B. Language Recognition Confusion Matrices

GlobalPhone (10hrs)

ERROR RATES: Pfa(Lt,Ln)

_	Target Language Lt			
Segment Language Ln	FRE	GER	POL	SPA
FRE	---	0.0032	0.9094	0.0874
GER	0.0000	---	0.9238	0.0662
POL	0.0070	0.0578	---	0.1653
SPA	0.0015	0.1405	0.5954	---
Pmiss(Lt)	1.0000	0.9901	0.2301	0.7374
Avg Pfa(Lt)	0.0028	0.0672	0.8095	0.1063

Avg Pmiss = 0.7394

Avg Pfa = 0.2465

Euronews (10hrs)

ERROR RATES: Pfa(Lt,Ln)

_	Target Language Lt			
Segment Language Ln	FRE	GER	POL	SPA
FRE	---	0.0809	0.0032	0.7217
GER	0.1523	---	0.0464	0.3907
POL	0.0867	0.4651	---	0.3705
SPA	0.1710	0.1725	0.0076	---

Pmiss(Lt)	0.8058	0.5894	0.9223	0.3511
Avg Pfa(Lt)	0.1367	0.2395	0.0191	0.4943

Avg Pmiss = 0.6672

Avg Pfa = 0.2224

Combined (10hrs)

ERROR RATES: Pfa(Lt,Ln)

_	Target Language Lt			
Segment Language Ln	FRE	GER	POL	SPA
FRE	---	0.0712	0.6181	0.2654
GER	0.0861	---	0.4934	0.1987
POL	0.0100	0.1375	---	0.1892
SPA	0.0305	0.1298	0.4214	---
Pmiss(Lt)	0.9547	0.7781	0.3367	0.5817
Avg Pfa(Lt)	0.0422	0.1128	0.5110	0.2178

Avg Pmiss = 0.6628

Avg Pfa = 0.2209

Euronews (50hrs)

ERROR RATES: Pfa(Lt,Ln)

_	Target Language Lt				
Segment Language Ln	ENG	FRE	GER	POL	SPA
ENG	---	0.0511	0.0188	0.0067	0.1306
FRE	0.5955	---	0.0129	0.0097	0.3139
GER	0.4636	0.1126	---	0.0497	0.2219
POL	0.7859	0.0299	0.0139	---	0.1066
SPA	0.6260	0.0519	0.0092	0.0092	---
Pmiss(Lt)	0.2073	0.9320	0.8477	0.9363	0.6962

| Avg Pfa(Lt) | 0.6177 | 0.0614 | 0.0137 | 0.0188 | 0.1932 |

Avg Pmiss = 0.7239
Avg Pfa = 0.1810

Combined (50hrs)

ERROR RATES: Pfa(Lt,Ln)

_	Target Language Lt				
Segment Language Ln	ENG	FRE	GER	POL	SPA
ENG	---	0.0054	0.0451	0.1703	0.2806
FRE	0.3139	---	0.0065	0.2039	0.4660
GER	0.3245	0.0000	---	0.2649	0.3013
POL	0.3317	0.0010	0.0339	---	0.2430
SPA	0.2611	0.0046	0.0183	0.1435	---
Pmiss(Lt)	0.5013	0.9903	0.8907	0.6096	0.4275
Avg Pfa(Lt)	0.3078	0.0027	0.0259	0.1956	0.3227

Avg Pmiss = 0.6839
Avg Pfa = 0.1710

DNN (monolingual)

ERROR RATES: Pfa(Lt,Ln)

_	Target Language Lt				
Segment Language Ln	ENG	FRE	GER	POL	SPA
ENG	---	0.0599	0.0363	0.3600	0.2826
FRE	0.0356	---	0.0129	0.3883	0.4757
GER	0.0861	0.0397	---	0.4106	0.3841
POL	0.3317	0.0010	0.0339	---	0.2430
SPA	0.1160	0.0550	0.0305	0.2458	---
Pmiss(Lt)	0.7389	0.9126	0.9205	0.4761	0.4473

Avg Pfa(Lt)	0.0973	0.0426	0.0244	0.3512	0.3583

Avg Pmiss = 0.6991

Avg Pfa = 0.1748

DNN (multilingual)

ERROR RATES: Pfa(Lt,Ln)

_	Target Language Lt				
Segment Language Ln	ENG	FRE	GER	POL	SPA
ENG	---	0.0336	0.0424	0.3351	0.1292
FRE	0.1877	---	0.0065	0.5016	0.2524
GER	0.2351	0.0364	---	0.5066	0.1358
POL	0.3088	0.0050	0.0209	---	0.0767
SPA	0.2611	0.0229	0.0214	0.3328	---
Pmiss(Lt)	0.5404	0.9482	0.9139	0.4114	0.6382
Avg Pfa(Lt)	0.2482	0.0245	0.0228	0.4190	0.1485

Avg Pmiss = 0.6904

Avg Pfa = 0.1726