Odyssey 2014:
The Speaker and Language Recognition Workshop
16-19 June 2014, Joensuu, Finland

Combining Joint Factor Analysis and iVectors for Robust Language Recognition

Brecht Desplanques, Kris Demuynck, Jean-Pierre Martens

ELIS Multimedia Lab
Ghent University - iMinds, Belgium
brecht.desplanques@ugent.be

Abstract

This paper presents a system to identify the spoken language in challenging audio material such as broadcast news shows. The audio material targeted by the system is characterized by a large range of background conditions (e.g. studio recordings vs. outdoor interviews) and a considerable amount of non-native speakers. The designed model-based language classifier automatically identifies intervals of Flemish (Belgian Dutch), English or French speech. The proposed system is iVector-based, but unlike the standard approach it does not model the Total Variability. Instead, it relies on the original Joint Factor Analysis recipe by modeling the different sources of variability separately. For each speaker a fixed-length low-dimensional feature vector is extracted which encodes the language variability and the other sources of variability separately. The language factors are then fed to a simple language classifier. When assessed on a self-composed dataset containing 9 hours of monolingual broadcast news, 9 hours of multilingual broadcast news and 10 hours of documentaries, this classifier is found to outperform a state-of-the-art eigenchannel compensated discriminatively-trained GMM system by up to 20% relative. A standard iVector baseline is outperformed by up to 40% relative.

1. Introduction

Many radio and television programs in Flanders –the Dutch speaking region in Belgium– comprise multiple languages. This is partly due to the fact that Belgium has three official languages, namely Dutch (Flemish), French and German. As a consequence Flemish news broadcasts often comprise French speech segments (German only occurs rarely). Furthermore, international news regularly includes English speech that is subtitled. In documentaries on the other hand, a Flemish narrator frequently clarifies subtitled foreign speech segments.

In an automatic subtitling scenario using Flemish, English and French monolingual speech recognizers, one wants to assign the speech stretches to the right recognizer. This calls for a dedicated language recognition (LR) module as pre-processor.

The LR module itself benefits from a front-end that automatically removes non-speech segments and that segments the speech into speaker turns. Therefore, our LR module is configured as an add-on to a speaker diarization system that conducts non-speech removal, speaker segmentation, speaker clustering and gender detection. In the final stage one can use all of the generated information to produce speech transcriptions.

For this work, we assume that every speaker uses a single language only. We therefore apply LR on the concatenation of all data assigned to a speaker, hereafter called a speaker session. If needed, the operation point of the speaker diarization could be adjusted so that multilingual speaker sessions are likely to be split into monolingual sessions with their own speaker IDs.

Motivated by the periodic NIST Language Recognition Evaluation campaigns e.g. [1], several LR techniques have been developed over the last couple of years. They can be regarded as either phonotactic or acoustic. Phonotactic methods are based on the frequencies of short phone sequences in the output of a phone recognizer. The acoustic methods on the other hand directly classify the speech segments on the basis of their acoustic properties.

As most current state-of-the-art acoustic approaches, our system employs iVectors [2]. A desired property of iVectors is that they provide fixed-length feature vectors per speaker session, giving more freedom in the subsequent language classifier. Contrary to the common practice in iVectors of modeling all important variability in the Total Variability space, we combine the iVectors with Joint Factor Analysis (JFA) [3] in order to separate the language variability from the channel variability. Similar to [4, 5] we use the term channel variability to refer to all sources of variability different from the language. Hence, in an LR setup the channel factors encode both channel and speaker variability. Each speaker session is thus characterized by a set of language factors and channel factors and only the language factors –designed to contain all relevant language information– are fed to a simple language classifier. Our method also replaces the Voice Activity Detector (VAD) adopted in most state-of-the-art systems with a novel frame selection strategy. This frame selection strategy contributes significantly to the high performance of the system.

To evaluate the system, we compiled a dataset containing monolingual and multilingual news broadcasts as well as a large variety of documentaries. The system is judged on its ability to identify the language of the speaker sessions. The collected dataset exposes a large range of different background conditions, ranging from clean studio recordings to noisy outdoor interviews. It also contains a lot of reporters or politicians speaking in a language that differs from their mother tongue, meaning that they must be considered as non-native speakers of the language they speak at that time. Our experiments show performance gains up to 40% relative for the proposed system when compared to a standard iVector system and up to 20% relative when compared to a state-of-the-art eigenchannel compensated GMM system with discriminatively trained models.

The next section briefly describes the acoustic and phonotactic approaches to language recognition. Section 3 handles the state-of-the-art acoustic language recognition techniques in detail. Section 4 details the proposed method. Experimental results are reported in Section 5. The main conclusions of our work are summarized in Section 6.

2. Language recognition

2.1. Problem statement

For this work, we restricted the task to the automatic recognition of the spoken language for each speaker session given a closed set of languages. We therefore limit the literature review to closed-set language recognition. Dealing with out-of-set languages is outside the scope of this paper. A technique to tackle this open-set LR problem by solely relying on data from the target languages can be found for example in [6].

2.2. The phonotactic approach

The standard phonotactic approach, called PRLM (Phone Recognizer followed by Language Model) [7], comprises a single-language phone recognizer followed by an N-gram language model. Training data for each language are tokenized by the phone recognizer, and the resulting symbol sequences are used to estimate the conditional probabilities of the actual phone token given the $N - 1$ previous phone tokens for speech in that language. Note that this approach does not need orthographic nor phonetic labeling of training speech from the target languages. During language recognition the test segment is tokenized and for each target language, the probability that the token sequence was produced for that language is calculated. The speech segment is then assigned to the language yielding the highest score.

More recent phonotactic approaches [8, 9] run multiple PRLM systems in parallel. In this so-called PPRLM approach, each target language is represented by multiple N-gram language models, one for each phone recognizer. In a final stage all language model scores are then fed to a simple language classifier, such as a Gaussian Back-end (see section 3.3.1).

2.3. The acoustic approach

Acoustic LR systems (explained in detail in Section 3) classify speech segments on the basis of acoustic score vectors. One approach is to train one GMM per language to model the speech frames of that language. An alternative is to have one model describing speech in general and a set of transformations to change this language agnostic model into language (and even session) specific speech models. During evaluation, the acoustic models and transformations are employed to extract a fixed length acoustic score vector for each speaker session which contains information concerning the language spoken. This vector is then supplied to a simple language classifier. As will be explained in the next sections, there exist different schemes for extracting suitable acoustic score vectors.

Whereas until recently PPRLM outperformed the acoustic approach, today the acoustic approach reaches a comparable performance, but at a much lower computational expense. That is why our primary focus is on the implementation of an acoustic LR system for the television domain. When computational load is not an issue, one can raise the performance level by fusing the the acoustic and the phonotactic approach in a single system, as in [9, 10] for instance.

3. Baseline acoustic language recognition

In this section we describe the acoustic approach in more detail. We discern three steps in the process: (1) the creation of an acoustic model for the language, (2) the extraction of an acoustic score vector and (3) the classification of that score vector into languages.

3.1. Creation of the acoustic models

3.1.1. SDC feature extraction

Research in GMM-based automatic language recognition suggests that shifted delta cepstral (SDC) feature vectors [11] constitute a richer representation of the signal dynamics than the standard dynamical features derived from the MFCCs. The SDC features are defined by four parameters: N, d, P and k. The first $(2k+1)N$ features of frame t are simply the delta's of the N static MFCCs $c_1 \dots c_N$, computed for the frames $t+iP$ ($i = -k \dots k$). A delta at frame t is computed over the window $(t-d, t+d)$. Most implementations (usually operating on telephone speech) opt for $N = 7$, $d = 2$, $P = 3$ and $k = 3$. We found $N = 10$, $d = 2$, $P = 3$ and $k = 2$ to work slightly better on broadband data at the same level of complexity. Supplementing the SDCs with 19 static MFCCs (the $c_1 \dots c_{19}$ of frame t) and a normalized log-energy finally leads to a feature vector of dimension 70.

3.1.2. Feature normalization and frame selection

Several methods can be applied to make the features more robust against additive noise and channel mismatch, such as Cepstral Mean Subtraction (CMS), Cepstral Variance and Mean Normalization (CMVN) and Feature Warping [12]. The latter technique transforms the individual features via a monotonic non-linear function so that their distribution over the processed time interval fits the standard normal distribution. For this work, we applied Feature Warping on complete speaker sessions rather than using a sliding window approach because we observed that different noise/channel conditions usually give rise to different speaker sessions in our diarization system, meaning that the within session changes will be small.

In most systems, the normalization is applied on the speech frames only. A Voice Activity Detector (VAD) is employed to eliminate the non-speech frames. This VAD can be as complex as a phone recognizer that detects the speech and the non-speech sounds, as in e.g. [5].

3.1.3. GMM training

For each target language a GMM is trained (or a GMM per gender if gender information is available). One usually starts with the training of an Universal Background Model (UBM) on material that comprises data of all target languages and one conducts a number of Maximum Likelihood (ML) training iterations on material from the target language to obtain the envisaged target language specific models.

As was shown in [9], continuing the ML training with some discriminative training such as Maximum Mutual Information (MMI) training, can provide significant performance gains. The objective of the MMI training is to maximize the posterior probability of the correct language per speaker session. Note that to avoid learning the language priors from the training data, the statistics in the MMI re-estimation formulas must be weighed [9].

3.2. Extraction of score vectors

3.2.1. Model evaluation

The most obvious technique is to compute the mean log-likelihood scores (over the selected frames of the speaker session) of the individual language specific models (ML or MMI trained) and to consider these as the elements of the acoustic score vector $\boldsymbol{y}_s$.

3.2.2. Eigenchannel adaptation of the models

The scores obtained with the model evaluation are negatively affected by the variability between speakers: instead of being purely based on the language being spoken, part of the score will reflect the affinity between the voice characteristics and channel conditions of the test speakers and the speakers in the training data for a language. As was shown in [5], this can be effectively compensated for with a simplified version of Joint Factor Analysis (JFA) [3], which the authors termed eigenchannel adaptation in the model domain. The idea underlying JFA is that the inter-class variability and inter-session variability (due to differences in speaker, channel, circumstances, etc.) can to some extent be modeled in different acoustic subspaces.

The eigenchannel adaptation computes the score vector of a speaker session by first converting each language specific model $\mathcal{M}_l$ to a language specific and session specific model $\mathcal{M}_{l|s}$ and by then evaluating these adapted models on the speaker data. The model adaptation is constrained to an adaptation of the Gaussian means, and it maintains the mixture weights and the mixture covariances. The supervector of the mixture means of $\mathcal{M}_{l|s}$ is modeled as

$$\boldsymbol{m}_{l|s} = \boldsymbol{m}_l + \boldsymbol{T}\boldsymbol{x}_{s|l} \tag{1}$$

where $\boldsymbol{T}$ is a low-rank matrix defining the R-dimensional subspace which best explains speaker and channel effects.

The elements of $\boldsymbol{x}_{s|l}$ are called the channel factors and are defined by

$$\boldsymbol{x}_{s|l} = \arg\max_{\boldsymbol{x}} \mathcal{L}(O_s|\mathcal{M}_{l|s}(\boldsymbol{x}))\, \mathcal{N}(\boldsymbol{x};\boldsymbol{0},\boldsymbol{I}), \tag{2}$$

with $\mathcal{L}(O_s|\mathcal{M}_{l|s})$ being the likelihood of speaker data O_s given the session-adapted model $\mathcal{M}_{l|s}$. The factor $\mathcal{N}(\boldsymbol{x};\boldsymbol{0},\boldsymbol{I})$ enforces a normal prior distribution on the channel factors. We call $\boldsymbol{x}_{s|l}$ the MAP point estimate of $\boldsymbol{x}$.

The mathematical procedure for extracting the channel factors $\boldsymbol{x}_{s|l}$ from a model $\mathcal{M}_l$ with M mixtures is explained in [13]. If $\boldsymbol{m}_{l,m}$ represents the component of the mean super vector that corresponds with mixture m and $\boldsymbol{\Sigma}_{l,m}$ is the corresponding covariance matrix, one obtains that

$$\boldsymbol{x}_{s|l} = \boldsymbol{L}_s^{-1} \sum_{m=1}^{M} \boldsymbol{T}_{n,m}^T \boldsymbol{f}_s^m \tag{3}$$

$$\boldsymbol{L}_s = \boldsymbol{I} + \sum_{m=1}^{M} N_s^m \boldsymbol{T}_{n,m}^T \boldsymbol{T}_{n,m} \tag{4}$$

$$N_s^m = \sum_{\boldsymbol{o}_t \in O_s} \gamma_t^m \tag{5}$$

$$\boldsymbol{f}_s^m = \boldsymbol{\Sigma}_{l,m}^{-\frac{1}{2}} \left(\sum_{\boldsymbol{o}_t \in O_s} \gamma_t^m \boldsymbol{o}_t - N_s^m \boldsymbol{m}_{l,m} \right) \tag{6}$$

In these equations, $\boldsymbol{o}_t$ is the feature vector at time t and γ_t^m is the occupation probability of mixture m according to model $\mathcal{M}_l$ at that time. Furthermore, $\boldsymbol{\Sigma}_{l,m}^{-\frac{1}{2}}$ is the Cholesky decomposition of the inverse of $\boldsymbol{\Sigma}_{l,m}$. Matrix $\boldsymbol{T}_{n,m}$ in (3) and (4) is a normalized version of $\boldsymbol{T}_m$, the submatrix of $\boldsymbol{T}$ corresponding to mixture m:

$$\boldsymbol{T}_{n,m} = \boldsymbol{\Sigma}_{l,m}^{-\frac{1}{2}} \boldsymbol{T}_m \tag{7}$$

The channel variability matrix $\boldsymbol{T}$ is obtained by means of Principal Component Analysis (PCA) initialization [14] followed by several iterations of the non-simplified Expectation-Maximization (EM) algorithm described in [13] until it converges. Note that during evaluation one does not know the language of the speaker, and consequently, one also adapts the models that do not correspond to the language spoken by the speaker. Therefore it is of the utmost importance that $\boldsymbol{T}$ does not represent too much language variability. This is realized during the training of matrix $\boldsymbol{T}$. First, by calculating the eigenvectors on the within-language (within-class) covariance matrix during PCA initialization. Secondly, we use the UBM model parameters and occupation probabilities probabilities in the EM-algorithm, but as motivated in [3] we center the first order statistics $\boldsymbol{f}_s$ (identical to (6)) around the language (class) ML means of the annotated speaker language, rather than centering it around the UBM means. This assumes that language effects are common to all speakers of a language. It also assumes that most of the session specific effects are language independent and hence can be learned more efficiently by pooling the data from all languages. A final assumption is that the channel conditions are fixed within a speaker session, otherwise the session would have been split up.

3.2.3. Eigenchannel adaptation in the feature domain

Adapting ML trained models is a logical thing to do, but adapting discriminative MMI models using an ML/MAP based technique seems rather counterintuitive. Therefore, the authors in [5] suggest to perform adaptation in the feature domain and to train discriminative models subsequently in the usual way. Each observation feature vector $\boldsymbol{o}_t$ of speaker s is projected onto a session-independent subspace:

$$\boldsymbol{o}_t' = \boldsymbol{o}_t - \boldsymbol{T}_{m_t} \boldsymbol{x}_s \tag{8}$$

and the channel factors $\boldsymbol{x}_s$ are estimated by using the UBM instead of the language specific models. Now, $\boldsymbol{T}_{m_t}$ is the submatrix of $\boldsymbol{T}$ that corresponds to the best scoring mixture m_t of the considered frame. The channel variability matrix T is identical to the one described in Section 3.2.2. The selection of the best scoring mixture is a straightforward way to convert the shift in the supervector domain to a shift in the feature domain.

3.2.4. iVector extraction

A favored concept in LR is that of iVectors [2] or Total Variability (TV) modeling. Unlike JFA, iVectors model all variability in a single low dimensional subspace. A low rank rectangular matrix $\boldsymbol{U}$, called the TV matrix or the iVector extractor, is used to approximate the session-dependent GMM supervector $\boldsymbol{m}_s$

$$\boldsymbol{m}_s = \boldsymbol{m} + \boldsymbol{U}\boldsymbol{x}_{s,L_s} \tag{9}$$

with $\boldsymbol{m}$ being the UBM supervector and $\boldsymbol{x}_{s,L_s}$ being the iVector which contains information about the session of speaker s and the language L_s spoken by that speaker.

The training of $\boldsymbol{U}$ is similar to the training of $\boldsymbol{T}$ in the eigenchannel approach, but the PCA initialization now calculates the eigenvectors of the covariance matrix instead of the within-class covariance matrix. The EM-algorithm uses the UBM occupation probabilities and the UBM model parameters. The first order moments $\boldsymbol{f}_s$ (see (6)) are now centered around the UBM means, so that matrix $\boldsymbol{U}$ represents all sources of variability.

The technique described in Section 3.2.2 is used to extract the iVector $\boldsymbol{x}_{s,L_s}$, but again we employ an UBM model instead of language specific models. We can interpret $\boldsymbol{x}_{s,L_s}$ as coordinates in the TV subspace of the model parameter space defined by matrix $\boldsymbol{U}$. Instead of evaluating the adapted model, we use the iVectors directly as a feature vector $\boldsymbol{y}_s$ for the language classifier.

3.3. Language classification of the score vectors

In this section we briefly recall two popular methods for classifying the score vectors by means of a classifier needing only a few design parameters.

3.3.1. Gaussian Back-end

Several state-of-the-art LR approaches (e.g. [6], [5], [15]) work with a Gaussian Back-end (GB). It models the distribution of the score vectors $\boldsymbol{y}$ of target language l by means of a multivariate normal distribution $\mathcal{N}(\boldsymbol{\mu}_l, \boldsymbol{\Sigma})$, meaning that one full covariance matrix is shared by all the target languages. The classification is then based on the following language score:

$$y^*_{s,l} = (\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_l)^T\boldsymbol{y}_s - \frac{1}{2}\boldsymbol{\mu}_l^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_l \tag{10}$$

with l indexing the target language. This equation is obtained by calculating the loglikelihood of vector $\boldsymbol{y}_s$. Since the covariance matrix is shared, we can not only drop the constant term but also the language-independent quadratic term.

The output of a GB can be further calibrated with discriminative multi-class logistic regression [6]. However, for our experiments this step was omitted since it did not improve the results.

3.3.2. Cosine distance scoring

As iVector components are coordinates in the Total Variability subspace they have a clear geometric interpretation. This explains why some systems use Cosine Distance Scoring (CDS) as a basis for LR [10]. In order to apply CDS, one first conducts a Linear Discriminant Analysis (LDA) on the iVectors to maximize the inter-class variability while minimizing the intra-class variability. This step reduces the dimensionality of the vectors to the number of languages minus one. The resulting vectors are normalized to have a unit length. Finally, by assuming that each language can be modeled by a von Mises-Fisher distribution, one can retrieve the ML mean of language l from the normalized vectors $\boldsymbol{y}'_s$ as follows:

$$\boldsymbol{\mu}_l = \frac{\sum_{\forall s \in l} \boldsymbol{y}'_s}{\| \sum_{\forall s \in l} \boldsymbol{y}'_s \|} \tag{11}$$

The sum is taken over all speaker sessions belonging to the training data for language l. By further assuming an identical spread for all language distributions we can extract the language score $y^*_{s,l}$ for a test segment $\boldsymbol{y}'_s$:

$$y^*_{s,l} = \boldsymbol{\mu}_l^T\boldsymbol{y}'_s \tag{12}$$

3.3.3. Converting the scores to posteriors

We use the scores $y^*_{s,l}$ (estimated by the GB or the CDS) and priors P_l to obtain the posterior probability distribution of language l given speaker s:

$$P(l|s) = \left(\sum_{j=1}^{N_l} P_j e^{y^*_{s,j}}\right)^{-1} P_l e^{y^*_{s,l}} \tag{13}$$

with N_l the number of target languages. The language l yielding the maximum posterior probability is then selected as the recognition result. We assume an uniform prior distribution. A better estimation of the prior distribution is obviously possible, but is currently not considered.

4. Proposed methods

4.1. Frame selection

In a conventional system, the LR is performed on the speech frames, and these frames are selected by means of a VAD. However, since in the television domain a lot of the speech can be characterized as speech over music and other noises, it is important to use a frame selection scheme that basically selects the same frames irrespective of the characteristics (energy, timbre) of the noise. Therefore, we propose a simple and robust frame selection scheme that mainly selects frames from the syllable nuclei because these most energetic frames will be the least affected by the noise. The selection algorithm tracks the log-energy $\log E(t)$ (natural logarithm) and converts it to a normalized log-energy that is equal to zero when the log-energy is equal to a running mean log-energy $\overline{\log E}(t)$ and positive when it is larger. The running mean is computed by means of a leaky integrator with a time constant of 5 seconds. The normalized log-energy is given by:

$$\log E_{nrm}(t) = \log E(t) - \overline{\log E}(t) \tag{14}$$

A VAD-like configuration discarding all frames with a $\log E_{nrm}(t) < -2$ results in the removal of all frames with an energy level that is more than 8.7 dB below the running mean energy. A syllable nucleus detector-like configuration with a positive threshold of 0.5 on the other hand results in the elimination of more than 50% of the frames in a clean speech utterance. Nevertheless, doing so turns out to improve the LR on our type of data (see experiments).

4.2. Eigenchannel adaptation for MMI-trained models

Inter-session variability can be expected to have a negative influence on the performance of discriminatively trained models, just as it does on ML-trained models (see Section 3.2.2). We therefore apply eigenchannel adaptation on the discriminatively trained models as well. At first, it may seem counterintuitive to apply an ML-based technique on discriminatively trained models. Although the ML-based eigenchannel adaptation can be expected to annihilate some of the extra discriminative power introduced by the MMI-training, we expect that suppressing the negative effects of the channel variability will outweigh the small loss in discriminative power.

Note that the MMI-models are based on ML-trained models which themselves are derived from a common UBM. Hence, it is not unreasonable to assume that the shifts in the Gaussian means related to speaker changes in the ML-trained and MMI-trained models are strongly correlated. As such, one may argue that the ML-based channel variability matrix $\boldsymbol{T}$ derived according to the recipe outlines in Section 3.2.2 can be applied unchanged to the MMI-trained models. We also tested a small variation on this scheme in which the first order moments in (6) are centered around the means $\boldsymbol{m}_l$ of the MMI-trained GMMs rather than around the means of the ML-trained models. Both approaches lead to comparable improvements.

4.3. iVector-based language factor extraction

Compared to systems that compute the scores of language specific models, the iVector approach has two advantages. First, the computational load is substantially lower: iVector systems only have to estimate UBM occupation probabilities, which compares favorable to the evaluation (and re-evaluation when eigenchannel adaptation is used) of a set of language specific models otherwise. Secondly, the dimension of the iVector is a design parameter and hence can be tuned for optimal performance: high enough so that no essential information is lost and low enough to suppress unwanted variability.

In iVector approaches for *speaker identification*, one typically considers the Total Variability, i.e. the matrix $\boldsymbol{U}$ models both speaker and channel variability. This choice can be explained with a combination of factors [2]: it may not be possible to obtain a good separation between channel and speaker, the channel may in many setups carry some (unwanted) information about the speaker, and the final classifier will suppress the non-informative channel variability. When ported to the *language recognition* domain, one can distinguish two major contributors to the total variability: the language and the channel (including speaker variability). In our LR setup, we do not expect that speakers in the training data will reoccur in the test data, hence the channel variability can be expected to carry no language information at all. We therefore tried to separate (factorize) the two sources of variability in order to remove the unwanted one (the channel variability) completely.

In Joint Factor Analysis [3], the contributing factors are both modeled in sub-spaces which are automatically found by an EM algorithm. Let $\boldsymbol{T}$ be the channel variability matrix as defined in Section 3.2.2 and let $\boldsymbol{V}$ be the language variability matrix. Since the number of languages is small, there is no need to rely on the EM algorithm for finding a compact representation $\boldsymbol{V}$ of the language sub-space. Instead, we assign one vector directly to each of the languages and set the values of the column vectors $\boldsymbol{V}_l$ of $\boldsymbol{V}$ equal to the offset between the ML supervector $\boldsymbol{m}_l$ of the corresponding language l (obtained by ML training initialized with the UBM) and the UBM supervector $\boldsymbol{m}$:

$$\boldsymbol{V}_l = \boldsymbol{m}_l - \boldsymbol{m} \tag{15}$$

This ensures that the UBM can be shifted towards the language dependent GMMs when performing language model adaptation with matrix $\boldsymbol{V}$. Combining channel variability and language variability modeling results in the following expression for the mean supervector $\boldsymbol{m}_s$ of the session-dependent GMM:

$$\boldsymbol{m}_s = \boldsymbol{m} + \boldsymbol{V}\boldsymbol{x}_{L_s} + \boldsymbol{T}\boldsymbol{x}_s \tag{16}$$

with $\boldsymbol{m}$ being the UBM supervector. We will refer to $\boldsymbol{x}_{L_s}$ and $\boldsymbol{x}_s$ as the language and channel factors respectively. These factors can be extracted simultaneously by stacking $\boldsymbol{V}$ and $\boldsymbol{T}$ into one matrix and by following the standard procedure as in conventional iVector extraction. In the end we obtain a vector $\boldsymbol{x}$ which can be decomposed as $\boldsymbol{x} = [\boldsymbol{x}_{L_s} \; ; \; \boldsymbol{x}_s]$ and all relevant language information is supposed to be included in $\boldsymbol{x}_{L_s}$. We will feed this low-dimensional score vector to a GB or a CDS language classifier.

5. Experiments and Results

5.1. Experimental setup

In the conducted LR experiments, the speech segments are known to be spoken in one of the three most relevant languages for Flemish broadcast data: English, Flemish and French. The impact of the presence of out-of-set languages, the occurrence of diarization errors and speaker sessions containing speech of multiple spoken languages are beyond the scope of this paper. Also telephone speech segments are currently discarded as our training data largely consists of broadband speech.

5.2. Data

5.2.1. Training and development set

The Flemish training and development data are taken from the CGN corpus [16]: 23 hours of speech (935 speakers) are used for model training and another 6 hours of speech (240 speakers) act as development data. The English models are trained on 63 hours of speech from the 1996 HUB4 Broadcast News training data (3748 speakers). The remaining 3 hours (90 speakers) constitute our development set. Since we had no regular corpus of French broadcast news at our disposal, we harvested 16 hours of speech from public RTBF podcasts[1] (403 speakers) as training data and 7 hours (137 speakers) as development data. RTBF is the public broadcasting organization of the French Community of Belgium and its website offers a wide variety of shows. The semi-automatic annotation of this material started from the outputs of our diarization tool. Segment boundaries and speaker labels were corrected where needed and language labels were added.

5.2.2. Evaluation data

The investigated techniques are evaluated on a custom dataset. The annotations were manually verified and language labels were added for each speaker. In the rare case of a speaker switching between two languages in the evaluation data, the speaker session was manually split into two sessions.

The so-called MONO part of the evaluation data consists of 3 hours of monolingual files per language. The Flemish data is retrieved from Flemish news broadcasts of the Flemish public broadcaster[2] VRT. The English data is the 1997 HUB4 Broadcast News corpus. The French data is extracted from French radio podcasts[3].

The BN (broadcast news) part of the evaluation data consists of 9 hours of news shows of the public and the commercial Flemish broadcaster[4]. The speech is uttered by 658 speakers and Flemish covers 90% of the speech, English accounts for 5% and French makes up for 3%. The remaining 2% represent a large range of out-of-set languages and were currently discarded during testing.

The DOCU part of the evaluation data consists of 10 hours of documentaries, broadcasted by VRT. It holds a completely different language distribution: Flemish 40%, English 22% and French 38%. Out-of-set languages are again discarded.

Details of the language and speaker session distribution of each test set can be found in Tables 1 and 2.

Table 1: Frame-based language distribution (%) of each evaluation subset.

	EN	FL	FR
MONO	29.8	31.0	39.2
BN	5.1	91.9	3.0
DOCU	22.4	39.7	37.9

[1] http://www.rtbf.be/radio/podcast

[2] http://www.vrt.be

[3] http://www.rfi.fr

[4] http://www.vtm.be

Table 2: Number of speaker sessions for each evaluation subset per language.

	EN	FL	FR	total
MONO	92	139	92	323
BN	91	524	45	660
DOCU	53	23	113	189

5.3. Evaluation Measures

To quantify performance, we compute the Session Error Rate (SER) as the percentage of incorrectly classified speaker sessions. However, since the SER strongly depends on the prior language probabilities, we also introduce the ratio between the mutual information $I(X,Y)$ between the recognized and the correct languages and the prior information (entropy) $H(X)$ of the correct language as an evaluation measure that is less affected by the priors. This normalized mutual information is denoted as C_{YX} and is given by

$$C_{YX} = \frac{I(X;Y)}{H(X)} \qquad (17)$$

The perfect classifier yields a $C_{YX} = 1$. The probabilities needed for calculating C_{YX} are retrieved from the confusion matrix summarizing the LR results.

The discussed measures are session-based because the LR algorithms make a decision per speaker session. The Frame Error Rate (FER) defined by the percentage of misclassified frames is also relevant as it is more directly related to the impact of the classification errors on the performance of e.g. the speech recognizers that are called on the basis of the recognized language label. In a few cases we also mention the FER as a performance measure.

5.4. Results for baseline acoustic systems

In this section we report the results for six baseline systems: ML and MMI trained models, ML models with eigenchannel adaptation, MMI models trained on eigenchannel compensated features and two iVector systems, one with a GB and the other with a CDS classifier. The number of GMM mixtures is always 256 and model parameters are always updated in maximum 20 ML and MMI training iterations. The optimal rank R of the channel variability matrix T and the iVector extractor U were both 50. All GMMs are optimized on the training set, the language classifiers on the development set. This avoids too much calibration to the training data.

5.4.1. Feature selection and normalization

Initial experiments showed that the performance of all systems is significantly affected by the scheme that selects the frames on which to base the characterization of a speaker session. Since we want to focus on the differences between the LR methods, we compared all methods in combination with the same frame selection scheme. We opted for the proposed frame selection scheme as this proved to be superior to the standard VAD scheme. The threshold was tuned on the development set using the ML system without eigenchannel adaptation. Figure 1 shows that a VAD-like configuration (negative threshold) performs significantly worse than a syllable nucleus detector-like configuration (positive threshold), as anticipated.

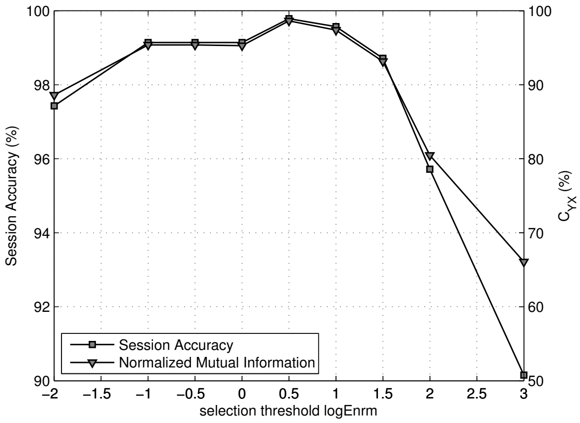

Figure 1: *100-SER (%) and C_{YX} (%) of the basic GMM ML system on the development data as a function of the frame selection threshold.*

Setting the threshold to 0.5 results in the removal of more than 50% of the frames inside a speech segment. Obviously, information of the discarded frames is not completely lost due to the fact that SDC features are calculated over 170 ms windows. We adopt this selection threshold in all further experiments.

The performance of the same ML system on the evaluation data for a VAD scheme vs. the adopted selection scheme is listed in Table 3 and confirms the trend visible in Figure 1.

Table 3: *Performance of the basic ML system in % on the evaluation data in function of the frame selection threshold*

selection threshold		-2.0	0.5
MONO	SER	7.4	3.7
	C_{YX}	75.9	85.3
BN	SER	20.3	11.7
	C_{YX}	33.9	47.0
DOCU	SER	21.7	15.9
	C_{YX}	47.6	57.4

Note that there is a huge performance difference between the different datasets. The SER ranges from 3.7% for MONO to 15.9% for DOCU. This can be explained by the fact that MONO mainly contains well trained speakers speaking a long time in favorable conditions (studio). In DOCU there are a lot of non-native speakers and the well-trained narrative voice is speaking over background noise, including non-native speech.

5.4.2. Comparison of baseline systems

The SER and C_{YX} performance of the six baseline systems are summarized in the upper parts of Tables 4 and 5. Training the language specific models discriminatively (MMI) results in a drop of the SER by 7-8% relative compared to the ML system. This is substantially lower than the 50% relative improvement reported in [9]. When looking at the normalized mutual information, the improvement is even less convincing. This apparently unexpected result can be owed to the large mismatch there is between the training and the evaluation data, a phenomenon that typically has a detrimental effect on the performance of discriminative methods. This may also explain why the discriminative multi-class logistic regression back-end (mentioned in Section 3.3.1) did not lead to better results.

Table 4: *Session Error Rate (%) of the baseline systems and proposed systems on the complete evaluation set. The lower the better.*

	MONO	BN	DOCU
ML + GB	3.7	11.7	15.9
MMI + GB	3.4	10.8	14.8
ML + eigchan. + GB	2.5	7.9	10.6
feat. eigchan. + MMI + GB	2.5	9.5	9.0
iVectors + GB	4.0	11.7	9.5
iVectors + LDA + CDS	2.5	11.8	7.9
MMI + eigchan. + GB	1.9	8.5	9.0
x_{L_s} + GB	2.5	7.6	8.5
x_{L_s} + LDA + CDS	1.5	7.3	9.0

Table 5: *Normalized Mutual Information C_{YX} (%) of the baseline systems and proposed systems on the complete evaluation set. The higher the better.*

	MONO	BN	DOCU
ML + GB	85.3	47.0	57.4
MMI + GB	85.9	52.7	55.5
ML + eigchan. + GB	89.0	59.5	66.1
feat. eigchan. + MMI + GB	89.3	56.5	67.7
iVectors + GB	86.2	55.0	66.7
iVectors + LDA + CDS	89.7	53.1	68.1
MMI + eigchan. + GB	91.5	60.8	68.1
$\boldsymbol{x}_{L_s}$ + GB	90.3	64.5	68.6
$\boldsymbol{x}_{L_s}$ + LDA + CDS	92.6	62.7	66.8

The performance gains obtained by applying eigenchannel adaptation on the other hand, are substantial. The SER reduces by 33% relative on all subsets when applying eigenchannel compensation in combination with ML models. The gains remain comparable in the case of MMI models, indicating that, at least for our test data, suppressing the negative effects of the inter-session variability is more important than increasing the discriminative power via MMI-based training as was anticipated in Section 4.2. Everything considered, in combination with eigenchannel compensation there is no significant difference in performance between the MMI and ML trained models.

Another conclusion is that the iVector methods can compete very well with the other baseline methods which are four to six times more time consuming. The latter is due to the fact that the iVector systems only have to estimate UBM occupation probabilities whereas the MMI eigenchannel system has to evaluate the UBM and the three language models. The ML eigenchannel system even has to evaluate all three language models twice. In terms of SER, the CDS seems to outperform the GB on two of the three evaluation sets, but these differences are not statistically significant.

For the sake of completeness we mention that the Frame Error Rates (FERs) for the iVector system with CDS are 0.7%, 4.8% and 4.9% for MONO, BN and DOCU respectively. These FERs clearly confirm the expectation that LR improves as the length of the speaker session increases.

5.5. Results for the proposed methods

The results for the proposed methods are summarized in the lower parts of Tables 4 and 5. We evaluate model-space eigenchannel adaptation (as opposed to the baseline feature-based eigenchannel adaptation) of the MMI models and language factor extraction (denoted by $\boldsymbol{x}_{L_s}$) in combination with two different language classifiers (GB, LDA + CDS). All GMMs are optimized on the training set and the language classifiers are optimized on the development set. The rank of the channel variability matrix was fixed to R=50, a value that was found optimal for the baseline systems. Figure 2 depicts the performance of the three proposed systems and the two best baseline systems (MMI system with eigenchannel compensation in the feature domain and iVectors with LDA+CDS). In the next subsections we discuss these results in more detail.

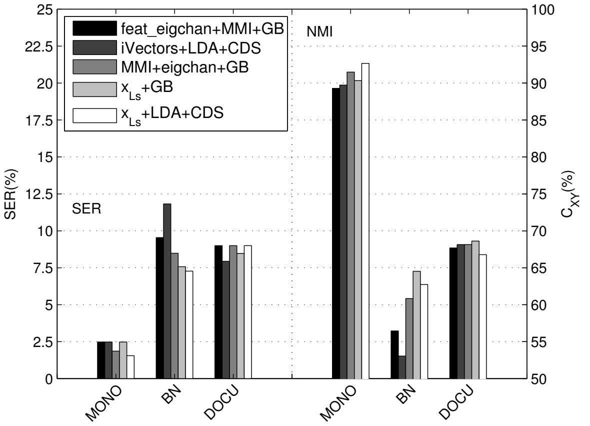

Figure 2: *Session Error Rate (%) and Normalized Mutual Information (%) of the two top performing baseline systems vs all the proposed systems on the complete evaluation set.*

5.5.1. *Eigenchannel adaptation on MMI models*

The result tables show that eigenchannel adaptation of MMI models yields an improvement over the baseline systems on two of the three evaluation sets and does not hurt the performance on the third set. The C_{YX} values improve on all three data sets. The reported results use the channel variability matrix of the ML baseline system. The alternative channel variability matrix which employs the MMI model means instead of the ML model means (see Section 4.2) provided comparable results. This shows that the assumptions underlying Section 4.2 hold: the channel variability subspace computed with an ML-based technique can be readily combined with a discriminatively trained model, and the ML-based eigenchannel adaptation does not completely annihilate the extra discriminative power introduced by the MMI training.

5.5.2. *Language factor extraction*

The results show that significant improvements over baseline iVector systems can be obtained by factorizing the observed variability in language components and channel components and feeding only the language components to the final classifier (GB or LDA+CDS).

The GB and LDA+CDS language factor system yield comparable results and both reduce the SER on the MONO and BN data of their corresponding iVector system by 40% relative. The SER differences on the DOCU data are not statistically significant as they correspond to two errors only. The C_{YX} values confirm the improvement on MONO and BN.

The LDA+CDS system can outperform the top performing loglikelihood-score based system (model-adapted MMI) by 15-20% relative and this at a much lower computational cost. Again, we see no significant performance changes on DOCU.

To conclude this section we also look at the frame error rates of the LDA+CDS system. It yields FERs of 0.3%, 2.7% and 5.4% on the MONO, BN and DOCU datasets respectively. The improvements in FER over the baseline systems on MONO and BN are substantial.

6. Conclusion

In this paper, we proposed a language recognition system for mixed-language TV broadcasts. We made a thorough analysis of some recently developed methods for speaker and language recognition (LR) that have shown to work well in the telephone domain and we introduced two additions to these existing approaches.

First, we improved the frame selection, an indispensable part of any LR system. By means of a simple energy-based selection criterion that can be configured as a selector of speech frames (= voice activity detector) to a selector of mainly syllable nuclei we showed that for the TV broadcast domain, it is extremely important to focus on the syllable nuclei. This reduces the variability since syllable nuclei are more resistant to the presence of background noise and since centering the selected features around the syllable nuclei reduces the variability in how the audio is presented to the subsequent GMM.

Secondly, we separated the variability in a language dependent part and the remainder which is mainly envisioned to represent speaker and channel variability. This separation simplifies the task of the final language classifier. This novelty may not be fundamental in terms of theory but nevertheless it yields a substantial gain in performance on two of the three datasets on which they were evaluated.

7. References

[1] NIST, *The 2009 NIST Language Recognition Evaluation Plan (LRE09)*, 2009, `http://www.itl.nist.gov/iad/mig/tests/lre/2009/LRE09_EvalPlan_v6.pdf`.

[2] Najim Dehak, Patrick Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[3] Patrick Kenny, Gilles Boulianne, Pierre Ouellet, and Pierre Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 4, pp. 1435–1447, 2007.

[4] Niko Brmmer, Albert Strasheim, Valiantsina Hubeika, Pavel Matějka, Lukáš Burget, and Ondřej Glembek, "Discriminative acoustic language recognition via channel-compensated GMM statistics," in *Proc. Interspeech*, 2009, pp. 2187–2190.

[5] Valiantsina Hubeika, Lukáš Burget, Pavel Matějka, and Petr Schwarz, "Discriminative training and channel compensation for acoustic language recognition," in *Proc. Interspeech*, 2008, pp. 301–304.

[6] Mohamed Faouzi BenZeghiba, Jean-Luc Gauvain, and Lori Lamel, "Gaussian backend design for open-set language detection," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 4349–4352, 2009.

[7] Marc A. Zissman, "Comparison of four approaches to automatic language identification of telephone speech," *IEEE Transactions on Speech and Audio Processing*, vol. 4, pp. 31–44, 1996.

[8] Mohamed Faouzi BenZeghiba, Jean-Luc Gauvain, and Lori Lamel, "Context-dependent phone models and models adaptation for phonotactic language recognition," in *Proc. Interspeech*, 2008, pp. 313–316.

[9] Pavel Matějka, Lukáš Burget, Petr Schwarz, and Jan Černocký, "Brno University of Technology system for NIST 2005 language recognition evaluation," in *Proceedings of Odyssey 2006: The Speaker and Language Recognition Workshop*, 2006, pp. 57–64.

[10] E. Singer, P. Torres-Carrasquillo, D. Reynolds, A. McCree, F. Richardson, N. Dehak, and D. Sturim, "The MITLL NIST LRE 2011 language recognition system," in *Proceedings of Odyssey 2012: The Speaker and Language Recognition Workshop*, 2012, pp. 209–215.

[11] Pedro A. Torres-Carrasquillo, Elliot Singer, Mary A. Kohler, and J. R. Deller, "Approaches to language identification using gaussian mixture models and shifted delta cepstral features," in *Proceedings of ICSLP 2002*, 2002, pp. 89–92.

[12] Jason Pelecanos and Sridha Sridharan, "Feature warping for robust speaker verification," in *Proceedings of 2001: A Speaker Odyssey, The Speaker Recognition Workshop*, 2001.

[13] Ondřej Glembek, Lukáš Burget, Pavel Matějka, Martin Karafiát, and Patrick Kenny, "Simplification and optimization of i-vector extraction," in *ICASSP*, 2011, pp. 4516–4519.

[14] Lukáš Burget, Pavel Matějka, Petr Schwarz, Ondřej Glembek, and Jan Černocký, "Analysis of feature extraction and channel compensation in GMM speaker recognition system," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 1979–1986, 2007.

[15] David Martìnez Gonzàlez, Oldřich Plchot, Lukáš Burget, Ondřej Glembek, and Pavel Matějka, "Language recognition in ivectors space," in *Proc. Interspeech*, 2011, pp. 861–864.

[16] Wim Goedertier, Simo M. A. Goddijn, and Jean-Pierre Martens, "Orthographic transcription of the spoken dutch corpus," in *Proc. LREC*, 2000, pp. 909–914.