Rapid Language Identification

Maarten Van Segbroeck, Member, IEEE, Ruchir Travadi, Student Member, IEEE, and Shrikanth S. Narayanan, Fellow, IEEE

Abstract—A critical challenge to automatic language identification (LID) is achieving accurate performance with the shortest possible speech segment in a rapid fashion. The accuracy to correctly identify the spoken language is highly sensitive to the duration of speech and is bounded by the amount of information available. The proposed approach for rapid language identification transforms the utterances to a low dimensional i-vector representation upon which language classification methods are applied. In order to meet the challenges involved in rapidly making reliable decisions about the spoken language, a highly accurate and computationally efficient framework of i-vector extraction is proposed. The LID framework integrates the approach of universal background model (UBM) fused total variability modeling. UBM-fused modeling yields the estimation of a more discriminant, single i-vector space. This way, it is also a computationally more efficient alternative than system level fusion. A further reduction in equal error rate is achieved by training the i-vector model on long duration speech utterances and by the deployment of a robust feature extraction scheme that aims to capture the relevant language cues under various acoustic conditions. Evaluation results on the DARPA RATS data corpus suggest the potential of performing successful automated language identification at the level of one second of speech or even shorter duration.

Index Terms—i-vector, noise robustness, rapid language identification, short-duration speech, total variability modeling, universal background model (UBM) fusion.

I. INTRODUCTION

LANGUAGE identification (LID) is the task of identifying the spoken language of, from, speech recordings. Numerous research efforts have contributed to significant technological advances in automated LID, providing viable alternatives to human audition and processing. The perceptual baseline experiments conducted in [1] have shown that human listeners are able to distinguish between familiar languages with high accuracy. When the languages are unfamiliar to the human listeners, subjective decisions are made. Furthermore, these decisions become less reliable when they need to distinguish between languages from the same family. The amount (duration) of data available for decision making also plays a critical role. From the study of [1] that involved 10 native English listeners having to distinguish between 9 non-English languages, a steep performance drop was observed when the duration of the audio excerpts was decreased from 6 to 1 seconds. An average human accuracy of 49.7%, 37.4% and 20.7% respectively for speech utterances with a duration of 6, 2 and 1 seconds was reported.

Over the years, several machine learning approaches have been proposed to develop algorithms for automated language identification. Automated LID systems could have potential advantages over humans in that they can be trained much faster and on a larger number of languages simultaneously. Moreover, these systems can provide valuable support to humans in retrieving the spoken language, especially when the language is unknown to them. With the expanding internationalization and continuing growth of technology-driven public and commercial services, the implementation of robust, fast and accurate language identification systems can improve the customer experience in many ways. Examples include technologies for security and defense applications [2], and multi-language translation devices where LID serves as a front-end to discover the input language before the appropriate translation process can be initiated. When used in emergency call centers, the impact of LID systems could be crucial by rapidly dispatching calls in order to make the operator’s responses more effective. This paper aims to specifically contribute to the domain of Rapid Language Identification in which reliable decisions about the spoken language need to be made quickly with as few seconds of speech as possible.

In order to retrieve the identity of the spoken language, various information cues are available in speech [3]. Languages differ from each other in (i) the makeup of their acoustic phonetic inventory, (ii) the combinatorial rules between the phones, called phonotactics, (iii) the characteristic patterns in prosody, and (iv) language-specific vocabulary and grammar. Research in the field of automated LID system development has emerged along two main directions related to the language information cues they exploit. Phonotactic systems use phone recognition and language modelling (PRLM). A phone recognizer first converts speech signals into a sequence of phone symbols or tokens. The tokenization of speech is subsequently followed by training a language model to extract language-specific phonotactic information from the statistics derived from the token strings [4]–[7].

The second category of LID methods attempts to classify languages by using Gaussian mixture models (GMMs) to capture the acoustic properties of speech. The potential of GMM-based language identification was shown in [8], [9] and significant progress in LID performance has been made by employing supervector modeling [10] and the introduction of Joint Factor Analysis (JFA) [11], [12]. JFA involves the adaptation of a large general speech GMM, named the Universal Background Model (UBM), with the aim to reduce the variability from non-language related effects, such as differences in recording channels,
session aspects and background noises in the data. This way, JFA retains a subspace that ideally captures the variability of the desired factor of interest, e.g., in the present case, the spoken language identity. Although originally applied to the problem of speaker verification, the factor analysis formulation can be easily generalized to language identification.

The method of JFA has led to its successful variant, namely total variability or i-vector modeling, which was introduced in [13] and has since become popular due to its excellent performance, reduced complexity and small model size. The success of performing LID in the i-vector framework has been shown in [14]–[17]. A review of GMM-based LID, will be provided in Section II.

This work contributes to the challenging problem of rapid language identification (RLID) where decisions about the spoken language need to be made on short duration utterances as well as in a real-time mode to enable online or interactive processing. Achieving a high accuracy on a small amount of speech data is the main goal of this paper. Therefore, we also attempt to address a fundamental question in the field of RLID: what is the minimum amount of data we need in order to make reliable decisions about the language? When the utterance duration is in the order of a few seconds or shorter, key informational cues such as the prosodic patterns, vocabulary or grammatical structure, tend to become less evident to be algorithmically extractable. Furthermore, with a decreasing number of phones expected in a word-based context, the language models of phonotactic-based approaches do not always guarantee providing sufficient discriminating rules to distinguish between languages. Moreover, inconsistency in phone recognition due to mismatched conditions between training and testing will have a relatively greater impact on language identification accuracy. In GMM-based LID, the statistical probabilities derived from the acoustical representation of the utterance will be accumulated over time and propagated until the final classification stage. Although their performance also increases when more statistics can be accumulated, these systems tend to be more robust on short utterances as they do not rely on rule-based approaches applied on phonetic transcription. Various attempts of accurate language identification on short-duration sentences using the i-vector framework have been proposed in [15], [18]–[22].

Section III presents a novel algorithmic framework for rapid language identification and describes the proposed technological advances to improve robustness and performance. To deploy rapid LID in real-life application, it is desirable that the computational time required for making the decision about the language is small. To address the computational requirements, the proposed RLID approach adopts the simplified i-vector framework proposed in [23] and exploits the computational benefits of UBM-fused total variability modeling [24]. In the simplified i-vector approach, the complexity of i-vector extraction is drastically reduced by defining a well-chosen prenormalization of the first order Baum-Welch statistics allowing fewer computations in the factor analysis. UBM-fused total variability modeling is a novel technique that combines multiple UBMs trained on diverse feature representations into a single combined UBM, with the goal of making the extracted Baum-Welch statistics more equally distributed along the UBMs Gaussian components, assuring an estimation of the i-vector space that is more discriminant between the language classes. Therefore, a multi-feature extraction scheme is designed that is robust to changing acoustic conditions in the recording environment. The paper also contributes to reduce the computational complexity of conventional language identification systems where accuracy and robustness improvements are obtained by applying system level fusion. Instead of training, evaluating and combining the output probabilities of multiple systems, the proposed UBM-fused LID system only requires the estimation of a single i-vector space. We will show that UBM-fusion achieves a better accuracy and is computationally less complex as compared to conventional system level fusion.

The RLID system will be evaluated in Section V on the LID data corpus collected by the Linguistic Data Consortium (LDC) under the DARPA Robust Automatic Transcription of Speech (RATS) program. The main goal of the RATS program is to accurately separate the target speech from interfering background sources, to identify the language and the speaker, and to apply keyword detection on a data corpus that consists of highly degraded speech recordings. The RATS data collection contains conversational telephone recordings that were retransmitted through eight different noisy radio communication channels [25]. In this work, the challenging requirements of the RATS LID task were further compounded by the constraints of performing language identification on speech utterances with durations shorter than the 3 seconds utterance duration, while maintaining a high level of robustness and keeping the computational demands low. When tested on short speech utterances of 5, 3 and 1 second duration, an equal error rate of respectively 6.61%, 8.36% and 14.49% is achieved, demonstrating the potential of rapid automated language identification (as well as the current limits of performance).

This paper concludes in Section VI by summarizing the presented work and suggesting future directions in the domain of RLID.

II. GMM-BASED LANGUAGE IDENTIFICATION

This section provides a review of language identification based on acoustic representations of spoken languages. It requires the training of a prior model that forms an adequate basis for representing the variability present in speech signals, the adaptation of this model into a low-dimensional subspace to discriminate between the target languages, the projection of speech utterances onto this subspace, and finally a classification strategy applied on the parameter representation of the utterances.

A. Universal Background Modeling

The first step in training a language-specific model is to train a general prior model that represents a generic language-independent, statistical distribution of the underlying acoustical characteristics captured by the feature vectors extracted from various languages. Research on acoustic modeling has shown the success of Gaussian Mixture Models (GMM) as probabilistic models that are able to adequately represent the acoustic variability of speech. The prior model that is deployed in our language identification system is a GMM composed of a number of
mixture components, and trained on all available training data. This GMM is commonly referred to as the Universal Background Model (UBM). The UBM plays a fundamental role in achieving the desired language classification accuracy of the final system. Adaptation techniques against the target language data are applied to the UBM in order to discriminate between the languages of interests.

A well-designed UBM should take the following considerations into account. Firstly, the UBM should be trained independent of the spoken language. A training set with a balanced amount of data from each language prevents the UBM from being biased toward a specific language. The mixture components of the UBM need to accurately model the subtle acoustic differences between the languages of interest. Furthermore, ideally the UBM should be acoustically matched with the data expected to be observed in the testing phase; this requirement motivates the robust feature extraction proposed in Section III-A. Robust modeling also yields consistency in the set Gaussian components that are dominant over time for utterances spoken in the same language. As will be shown in Section III-C, a uniform occupancy distribution of the Gaussian components will have a beneficial influence on the performance.

Let us define a UBM composed of $C$ Gaussian mixture components as $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_C\}$ where each mixture component is characterized by $\lambda_c = \{\rho_c, \mu_c, \Sigma_c\}$ with mixture weight $\rho_c$, Gaussian mean $\mu_c$ and (diagonal) covariance matrix $\Sigma_c$. Given the training data, Maximum Likelihood Estimation (MLE) [26] is applied to estimate the GMM. Here, the model parameters are iteratively found by means of the Expectation-Maximization (EM) algorithm [27].

### B. Total Variability Modeling

The system for Rapid Language Identification (RLID) used in this work uses the total variability or i-vector modeling approach, originally proposed in [13]. Total variability modeling is based on, and motivated by, the technique of Joint Factor Analysis (JFA) [11], [12] and was first applied to the task of speaker verification. The aim of JFA is to jointly capture the desired variability of the predefined signal factor of interest (e.g., gender, speaker, language, etc.), and the undesired session variability originating from other factors, such as the transmission channel, the recording environment or the affective speaker state, to name a few.

Given the Universal Background Model of Section II-A, applying JFA to model spoken languages implies representing each utterance as a language- and session-dependent supervector $M_j$:

$$M_j = m + Vv_j + Uu_j + Dd_j$$

where $m$ is a language- and session-independent supervector constructed by stacking all mean vectors of the Gaussian mixture components of the UBM. Matrices $V$ and $U$ can be seen as the eigenspace of the language and the eigenspace of the session, modeling respectively the desired (between-language) and undesired (within-language) variability. Matrix $D$ is a diagonal matrix and contains the residual of the language subspace that is not captured by $V$. The utterance representation of (1) allows extracting a low-dimensional vector for both the language and the session information that is present in each utterance. These vectors are all normally distributed and referred to respectively the language factors, $v_j$ and $u_j$, and the session factors $u_j$.

The between-language and within-language variability are ideally captured in eigenspaces that are maximally decorrelated to prevent the loss of important language information in the session eigenspace. However, the assumption of zero mutual information between these subspaces does not hold in practice when real life speech signals are observed [28]. Therefore, an alternative factor analysis framework, namely total variability modeling, was presented in [13] to mitigate this type of information loss by the estimation of a single low-dimensional subspace, i.e., the identity or i-vector space, modeling all variability together.

In the total variability framework, the supervector $M_j$ of equation (1) is now reformulated as

$$M_j = m + Tw_j$$

where the matrix $T$ spans a low-dimensional total variability subspace of rank $K$. The utterance $j$ is now represented by a normally distributed vector $w_j$, containing the corresponding $K$ total factors, commonly referred to as the identity- or i-vector. Note that the probability function of the feature vectors given $w_j$ is a Gaussian mixture model with mean supervector $M_j$ and super covariance matrix denoted by $\Sigma$ that explains the residual variability not captured in the eigenspace defined by the column vectors of $T$.

The low-dimensional eigenspace spanned by the i-vectors, i.e., the i-vector eigenspace, yields an intermediate, suboptimal representation of language variability. Hence, post-processing techniques are required to compensate for the undesired variability of the session factors and will be briefly mentioned in Section II-C.

Let $y_{j,t}$ denote the $D$-dimensional acoustic feature vector at a time frame $t$ of utterance $j$. The i-vectors are then estimated from the acoustic feature representation of the utterance, using the corresponding UBM $\Lambda$ as a prior. The zeroth order Baum-Welch statistics of UBM mixture component $\lambda_c$ for utterance $j$ are then given as

$$N_{j,c} = \sum_{t=1}^{T} P(c|y_{j,t}, \lambda_c)$$

where the sum of the occupancy probabilities is taken over all $T$ frames that are present in the utterance. The centralized first order Baum-Welch statistics are computed as

$$F_{j,c} = \frac{1}{N_{j,c}} \sum_{t=1}^{T} P(c|y_{j,t}, \lambda_c)(y_{j,t} - \mu_c).$$

Rearranging the statistics (3)–(4) over all $C$ mixture components, we stack all vectors $F_{j,c}$ into the supervector $F_j$, and we define the $CD \times CD$ diagonal matrix $N_j$, which is composed of $C$ diagonal blocks of respectively $N_{j,c}I$, with $I$ being the identity matrix of dimension $D$.

The total variability framework (2) can now be restated as

$$F_j = Tw_j$$

1120 IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, VOL. 23, NO. 7, JULY 2015
and associated probability distributions

\[ P(w_j \Lambda) = N(0, I) \]
\[ P(F_j w_j, \Lambda) = N(Tw_j, N_j^{-1} \Sigma_j) \] (6)

Note that the distribution of \( F_j \) is both conditioned on \( w_j \) and the UBM \( \Lambda \).

The total variability matrix \( T \) is iteratively trained by the EM-algorithm described in [29] for only one factor in the JFA and by considering each training utterance as being produced by a new speaker. The Expectation-step involves the computation of the posterior probability \( P(w_j|F_j, \Lambda) \) using Bayes’ rule and the priors given in (6). The estimated i-vectors are explained as the expected values of the posterior distribution and given as

\[ w_j - \beta_j^{-1} T^T \Sigma_j^{-1} N_j F_j \] (7)

with

\[ \beta_j^{-1} = I + T^T \Sigma_j^{-1} N_j T. \] (8)

The Maximization-step of the EM algorithm updates the total variability matrix \( T \) and the supervector covariance matrix \( \Sigma \) such that the global likelihood defined over all \( \Gamma \) training utterances:

\[ \prod_{j=1}^{\Gamma} P(F_j, w_j \Lambda) \] (9)

is maximized. The updated matrices are found by linear regression of (9) using the estimated i-vectors (7) as explanatory variables. The total variability matrix \( T \) is randomly initialized, while \( \Sigma \) is initialized by the covariance matrices of the UBM. For further algorithmic details on the training procedure, we refer to [11].

Experimental evidence of the success of total variability modeling as compared to JFA was given in [13] for speaker verification, while [16] presents its benefits when applied on language identification. Despite the simplification of the JFA equations, the method of total variability modeling is still computationally expensive. The iterative EM procedure in the training stage and the i-vector extraction during testing are both dominated by the computationally expensive matrix products of equations (7) and (8). This yields a combined complexity of \( O(K^2 + CK^2 + CDK) \) for each speech signal of any duration that needs be evaluated to retrieve the spoken language.

C. Language Classification

Total variability modeling transforms each test utterance into a single low-dimensional i-vector representation of fixed dimension \( K \). To perform language identification on these utterance i-vectors, various classification techniques have been proposed. Generative modeling approaches [14] attempt to model the language classes by training a Gaussian distribution on the i-vectors, while discriminative methods seek to find the language decision boundaries in the i-vector space. For the latter, a distinction is further made between classifiers that require training, such as Support Vector Machine (SVM) or Neural Networks [21], and direct scoring approaches against a target i-vector for each language, e.g. Cosine Distance Scoring (CDS) [13] and sparse representation classification (SRC) [30].

As mentioned above, the presence of the undesired factors in the variance of the i-vector should be compensated prior to classifier training. Therefore, variability compensation methods such as Within-Class Covariance Normalization (WCCN) [31], Linear Discriminative analysis (LDA) and Nuisance Attribute Projection (NAP) [10], are typically applied within the i-vector space.

Our previous research on LID [17], [23], also confirmed by the work of [15], [17], has shown that best results are obtained when WCCN is applied prior to training an SVM classifier with polynomial kernel of high order. In this paper, the WCCN feature transformation matrices and the SVM are both trained on the same training set that was used in total variability modeling.

III. RAPID LANGUAGE IDENTIFICATION

Obtaining a high accuracy at a low computational cost is essential for making rapid and reliable decisions about the spoken language on utterances with short duration. This section gives an overview of recent advances that have been made in the context of the RATS LID project, all of which are steps to enable systems of rapid language identification (RLID). We start by explaining the importance of a robust front-end module, together with the proposal of an acoustic feature set capturing multiple discriminative speech characteristics. Next, we restate the modification to the i-vector modeling that was proposed in [17], [23] to improve LID performance in terms of computational load. This simplified i-vector system is further extended to the framework of UBM-fused total variability modeling [24]. It will be shown that significant improvements in accuracy, while maintaining the system’s complexity, are achieved when the i-vector space is estimated in this framework and by training on utterances with long duration.

A. Robust Feature Extraction

The feature vectors that are extracted from the audio data should capture the acoustic properties that are relevant to discriminate between the languages. Furthermore, deployment of LID in real-life scenarios requires the features to be relatively invariant for a wide range of adverse acoustic conditions, such as variations in the background noise environment and changes in the audio transmission channels or recording devices. To this end, an ideal front-end for language identification contains a Voice Activity Detection (VAD) to prevent non-speech audio segments from interfering with the classification decision, a speech enhancement method [32] to compensate for noise distortions and a robust feature extraction module followed by a normalization step to further reduce the sensitivity of the features to the acoustic variability. A schematic overview of such a robust front-end for a LID system is shown in Fig. 1.

As discussed in Section II-A, language (identification) learning involves the adaptation of Universal Background Model that is trained on the chosen feature representation. It is clear that the accuracy of the Baum-Welch statistics (3) and (4) which are derived from the UBM will have an important impact on the entire system performance. Since UBM components are assumed to have diagonal covariance Gaussians allowing their evaluation to be computationally tractable, feature components require to be sufficiently decorrelated to ensure accurate modeling.
By replacing (10) in (5), the occupancy, now computed as, will be factored out from the covariance matrix in (7) and (8) are replaced by the scaling factor, which implies that all super-
during training, the system.

Fig. 1. Overview of pre-processing steps of a robust front-end for a LID system.

In this work, the front-end feature extraction as depicted in Fig. 1, involves denoising of the audio files by standard Wiener Filtering [33], trimming the data to speech-only audio segments by applying Voice Activity Detection (VAD) [34], and the extraction of robust, decorrelated features vectors which are finally normalized to zero mean and unit variance on a per utterance basis.

Finding a feature representation that is optimal for the task of language identification is challenging. Different approaches have been proposed [15], [17], [20], [21], [32], all exploiting various feature sets. Among them are standard speech features (Mel-Frequency Cepstral Coefficients (MFCC) [35], Perceptual Linear Prediction (PLP) coefficients [36], Gammatone Frequency Cepstral Coefficients [37]), shifted-delta-cestral (SDC) acoustic features [38], spectral modulation features (cortical features [39], Gabor features [40], Medium Duration Modulation Cepstral [41]), robust acoustic features (Mean Hilbert Envelope Coefficients [42], Power-normalized coefficients [43], Frequency-Domain Linear Prediction [44], [45]), and pitch-based features. Rather than finding a single representation that captures most relevant and discriminative language properties, most of these research studies suggest exploiting the diversity of multiple feature streams to be more beneficial.

The RLID front-end exploits the combined usage of four acoustic feature representations. Three standard features were extracted: Mel-Frequency Cepstral Coefficients (MFCC), Perceptual Linear Prediction (PLP) coefficients, and Gammatone Frequency Cepstral Coefficients (GFCC). Each of these standard representations yields a 44-dimensional feature vector obtained by adding the first order derivatives to the 22 static components. A fourth feature representation was constructed by combining the speech streams proposed in [34], each modeling a different spectral cue in the auditory spectrum that is related to (and constrained by) the human speech production process. Hence, their combination could have high potential to also reveal language-specific acoustic differences. These streams model:

- the steady state of the articulators which is reflected in the spectral shape of the acoustics and extracted by Gammatone filtered spectra,
- the complex interaction and movement of the articulators, acoustically represented by spectro-temporal amplitude modulations, and modeled by Gabor filtered Mel-scaled spectra,
- the harmonic structure due to the quasi-periodic vibrations of the vocal folds, calculated by the autocorrelation value at the estimated pitch period, normalized by signal energy at frame level and context-integrated by applying a 5-dim DCT-transform on moving analysis windows of long duration, e.g. 500 ms,
- the successive phone generation corresponding to the produced syllable rate inherent to the language and which manifest itself by spectral energy fluctuations at short and long-term time scales. These specific temporal variations are captured by the long-term signal variability (LTSV) measure [46], [47], defined as the variance of the entropy over all spectral bins, and is followed by the context-integration step mentioned above.

These streams are stacked in one feature vector and subsequently decorrelated and dimensionality reduced by applying Principal Component Analysis (PCA). The principal components are computed on all training data, and only the components that correspond to the 88 largest singular values are retained, concentrating around 90% of the variance. We will refer to this feature representation as the Fused Speech Stream (FuSS) feature vector. All deployed feature representations are subsequently mean variance normalized (MVN) on a per utterance basis.

B. Efficient I-Vector Extraction

Rapid language identification requires the transformation from the utterance acoustics into a single low-dimensional i-vector representation to be computationally tractable. To this end, the simplified i-vector modeling of [23] was adopted in both training and decoding stages of the proposed RLID system. The simplified framework requires a particular prenormalization step in the factor analysis that allows a complexity reduction in the i-vector extraction. The prenormalization is obtained by re-weighting the first order Baum-Welch statistics (4) as follows:

$$F^*_j \leftarrow \frac{1}{\sqrt{N_j^c}} F^*_j$$

with $n_j = \sum_{i=1}^{N_j^c}$. By replacing (10) in (5), the occupancy probabilities $N_j^c$ will be factored out from the covariance matrix of $F_j$, now computed as $n_j^{-1} \Sigma$, which implies that all super-vector dimensions are equally treated in the i-vector modeling. As a result of this approximation, the matrix multiplications that involve $N_j$ in (7) and (8) are replaced by the scaling factor $n_j$. This reduces the total complexity to $O(K^3 + C DK)$.

To avoid the costly matrix inversion of $\beta_j$ during training, the use of a precomputed cache table was also proposed in [23]. The table is updated at each iteration after the Maximization-step of the EM algorithm and decodes the matrix product $\beta_j^{-1} T^* \Sigma^{-1}$ into a set of table entries that are selected based on their value.
for \( n_1 \). It was shown that a limited quantization error can be assured for a table size of the order of a few hundred entries, which is typically much smaller than the number of utterances in the training set. The table look-up strategy further reduces the complexity in training mode to \( O(CDK) \).

As shown in [17], the simplification of the i-vector system slightly reduces the performance of the conventional i-vector baseline. However, the sacrifice in performance is often negligible and tolerated given the measured computation speed increase of more than 100 times compared to the baseline.

### C. Accurate Modeling

It was shown in [24] that accurate modeling of the i-vector space is highly related to the extracted UBM Baum-Welch-statistics. An improved accuracy of Baum-Welch statistics can be achieved by (i) training on long duration speech utterances, i.e. containing multiple conversational sentences, since they activate more components per utterance and hence accumulate more statistically relevant acoustic language cues, and (ii) by fusion of multiple UBMs that are trained on various feature representations capturing diverse and complementary acoustic information. Hence, both approaches will be exploited in the training stage of the proposed RLID system.

The concept of UBM fused total variability modeling was recently proposed in [24] where its efficiency, especially when evaluated on short-duration speech utterances, was demonstrated. The technique of UBM-fusion increases the number of dominant (and non-redundant) UBM components by combining various feature representations into a single i-vector model and yields significant gains in classification accuracy without increasing its computational complexity.

The prenormalization strategy of Section III-B allows a straightforward implementation of UBM-fused total variability modeling, which involves the following steps prior to i-vector training:

1. Define \( L \) as the total number of UBMs that will be jointly exploited in the fused i-vector training. In order to be effective, each UBM should be trained on a different acoustical feature representation.

2. Construct the supervector \( \tilde{\mathbf{F}}_j \) by combined stacking of all UBMs (unweighted) first order Baum-Welch statistics:

\[
\tilde{\mathbf{F}}_j = \left[ \mathbf{F}_1^T, \mathbf{F}_2^T, \ldots, \mathbf{F}_L^T \right]^T.
\] (11)

3. Reweight each UBM-specific component \( \mathbf{F}_j^c \) of (11) according to formula (10), with \( n_j \) summed over all \( C_j \) UBM components, hence

\[
n_j = \sum_{i=1}^{L} \sum_{c=1}^{C_j} N_{ij}^c.
\] (12)

4. Initialize \( \Sigma \) from the covariance matrices of all Gaussian components in (11).

With the above steps, the UBM-fused i-vector modeling framework is then formulated as:

\[
\tilde{\mathbf{F}}_j = \tilde{\mathbf{T}} \mathbf{w}_j
\] (13)

with probability distributions

\[
P(\mathbf{w}_j | \lambda) = N(0, \mathbf{I})
\]

\[
P(\tilde{\mathbf{F}}_j | \mathbf{w}_j, \lambda) = N(\tilde{\mathbf{T}} \mathbf{w}_j, n_j^{-1} \Sigma).
\] (14)

Finally, the system is trained by the EM-procedure of Section II-B, where the i-vectors are found as the posterior expectations of \( P(\mathbf{w}_j | \tilde{\mathbf{F}}_j, \lambda) \), and computed as:

\[
\mathbf{w}_j = \tilde{\mathbf{b}}_j^{-1} (\tilde{\mathbf{T}}^T \Sigma^{-1} \tilde{\mathbf{F}}_j)
\] (15)

with

\[
\tilde{\mathbf{b}}_j = 1/n_j + \mathbf{T}^T \Sigma^{-1} \mathbf{T}.
\] (16)

Note that when all UBMs have a number of \( C_j = C_j L \) components, the computational complexity during i-vector training remains \( O(CDK) \). Hence, UBM-fused i-vector modeling reduces the total complexity by a factor \( L \) when compared to conventional system level fusion of the same number of systems. The latter requires the estimation of an i-vector space for each system in the fusion, which yields a total complexity of \( O(LCDK) \) for i-vector modeling. In a real time implementation, the multi-feature extraction scheme of the LID front-end extracts features simultaneously during the recording of the utterance, leaving only a few frames unprocessed when the utterance ends. Therefore, an efficiently implemented feature extraction module only adds a negligible computational bias to the language identification operation that is dominated by the i-vector extraction. During testing phase, the fused UBM is treated as if it was a single UBM of the same size and hence the total number of computations remains \( O(K^3 + CDK) \). Note that system level fusion is of \( L \) systems would have required \( O(LK^3 + LCDK) \) computations during testing.

Experimental evidence for the beneficial impact on long duration training and UBM-fused total variability modeling is given in Section V-B and Section V-E, respectively.

### IV. Data Corpus

The DARPA Robust Automatic Transcription of Speech (RATS) data corpus [25] was chosen as it allows assessing the robustness and performance of the proposed RLID system under various use conditions and to compare it with many state-of-the-art LID systems that have been recently reported on this corpus. The Linguistic Data Consortium (LDC) collected conversational recordings from public telephone networks of five target languages (Arabic Levantine, Dari, Farsi, Pashto, and Urdu) and 10 non-target languages. The recordings were about 2 minutes long and were retransmitted through eight different radio communication channels using different transmitter and receiver systems. The process of rebroadcasting introduces various aspects of heavy speech degradation such as nonlinear speech distortions and channel noise, frequency shift, band limitation and a variable signal-to-noise ratio (SNR) ranging from 30 dB to values lower than 0 dB. A training and development set of these retransmitted data was distributed by the LDC to all participants of the DARPA RATS program. The official development set, denoted by DEV2, was split into four test sets containing utterances with durations of 120, 30, 10 or 3 seconds.
In this work, we evaluate RLID performance under various utterance durations, ranging from 10 seconds to shorter than 1 second. The official RATS data sets are heavily out of balance in the amount of utterances per language. To reduce the bias in modeling due to the inequality in distribution, we constructed for each duration a train and test set that was balanced equally along the 5 target and 1 non-target languages. The training utterances were obtained by extracting utterances of equal duration from the RATS LID 120 seconds training database, using all 8 transmission channels and including the original source recordings (9 channels). The starting time of each extracted utterance was randomly chosen on a per-utterance basis. Each language contains 16,000 utterances, hence a total amount of 96,000 utterances were used for RLID system training. The results that are shown next were obtained on a training set of short utterance lengths (1, 3, 5, 10 seconds) and long duration utterances of 30 seconds.

A similar approach was followed for creating the test sets of various durations, but here the utterances were extracted from the official DEV2-10 seconds merged with DEV2-120 seconds to obtain a total of 2000 test utterances, quasi equally distributed along the 6 language classes (~335 utterances per language class) and 9 channels (~38 utterances per channel). The same set of test utterances were used for each duration-specific test set, although the starting time of the extracted utterances randomly vary along each utterance. The details of the training and test set derived from the RATS LID corpus are summarized in Table I.

V. EVALUATION

The performance of the LID system is assessed in terms of the following evaluation metrics that are commonly used to report on the RATS LID corpus:

- equal error rate (EER) as derived from the Detection Error Trade-off (DET) curve [48],
- minimum detection cost function (DCF), proposed by National Institute of Standards and Technology (NIST) [49], and
- minimum average cost (Cavg), the official evaluation measure for RATS and defined in the 2009 NIST language recognition evaluation plan [50].

The LID system design parameters, such as the size of the UBM and the dimension of the i-vectors, are evaluated in this section. The parameter values were optimized and selected to perform well on the RATS data set. By following a step-by-step implementation of the techniques presented in Section III, we illustrate their contributions to relative improvements in LID performance. Finally, we discuss the minimum utterance length required for RLID to operate at an acceptable system performance.

A. UBM Size

As explained in Section III-B, the computational requirement of extracting i-vectors in the testing phase is $O(K^3 + CDK)$. The feature dimensionality $D$ is considered to be optimally designed by the front-end extraction and typically of an order smaller than the i-vector dimension $K$. Therefore, an optimal trade-off between system performance and computational time is largely determined by $K$ and the number of Gaussian components $C$ in the UBM.

The sensitivity of language identification accuracy to the size of the UBM is plotted in Fig. 2. The figure presents the LID performance measured in terms of EER for varying sizes of the UBM. The LID front-end extracts 22-dimensional standard MFCC feature vectors augmented with their first order delta coefficients, hence $D = 44$. We trained an i-vector system in the simplified framework of Section III-B on utterances of various durations, i.e. 1, 3, 5 and 10 seconds. The LID performance of the system was assessed on the test sets described in Section IV containing utterances with a duration matching those of the training models. Except for the experiments conducted in Section V-C, the rank $K$ of the total variability matrix, and hence the dimension of the i-vectors, is fixed to 400. We applied WCCN on the extracted i-vectors to suppress the undesired variability caused by the speaking style, speaker differ-
Fig. 3. Variance of the normalized zeroth order Baum-Welch statistics of a 2048 component UBM trained on MFCC features and evaluated on training sets with variable utterance duration ranging from short length to up to 30 seconds.

ences and acoustic characteristics of the transmission channel. A Support Vector Machine (SVM) with 5th order polynomial kernel was trained on the normalized i-vectors to obtain the language output probabilities for each utterance of the test set.

As can be seen from Fig. 2, the EER decreases as the UBM size increases and the performance improvement is more profound when the utterances are shorter. The relative improvement obtained by using 2048 Gaussian components instead of 128 ranges from 11% to 20%. From the figure we can conclude that a good trade-off between complexity and accuracy is obtained with a UBM with 512 or 1024 components. However if achieving a high performance is a key system requirement, it is beneficial to train a UBM of a larger size. It can be expected that the performance would (slightly) increase for UBM sizes larger than 2048, however this will inevitably come at the cost of significant increase in UBM training time, and increased complexity of i-vector extraction. For the remainder of this paper, the total number of Gaussian components of the UBM will be fixed to 2048.

B. Short Versus Long Duration Training

Section III-C motivated that accurate extraction of UBM Baum-Welch-statistics achieves a better estimate of the total variability matrix during training. A decreased variance of the occupancy counts can be obtained by evaluating the UBM on a training set of utterances that activate more UBM components over time, e.g. using utterances of long duration. Fig. 3 shows the variance, taken over all utterances, of the zeroth order Baum-Welch statistics (divided by the number of frames in the utterance) for a UBM with 2048 components trained on MFCCs. The evaluation was performed on utterances with a length varying from 1 second up to 30 seconds duration and the variance values were sorted by value for illustrative purposes. From the figure it is clear that the variance of the occupancy counts decreases with the length of the utterances. Experimental evidence for the beneficial impact of long duration training on the estimation of the total variability matrix is given in Table II. The table shows the EER on the short utterance test sets when the i-vector space is trained on matched utterance duration, or longer duration of 10 and 30 seconds. The relative improvement in EER obtained by i-vector training on utterance of 30 seconds duration as compared to matched duration training for the 5, 3 and 1 seconds test set, is 10.7%, 12.1% and 12.7%, respectively. Since the impact increases when the utterances are shorter, long duration training is an important aspect in designing the training phase of the RLID system.

It is important to note that, while the model parameters, i.e. the total variability matrix $T$ and (super) covariance matrix $\Sigma$, are being trained on utterances of long duration, the SVM classifier is trained on i-vectors extracted from the training set with matched utterance length. Experiments not reported here have shown that this improves the classification performance, however it is not crucial for achieving high accuracies. Hence, we will train the total variability matrix from Baum-Welch statistics derived on 30 seconds duration and the SVM on the i-vector representation of training utterances matching the utterances of the test set in duration.

C. Dimensionality of the I-Vector Space

The rank $K$ of the total variability matrix corresponds to the dimensionality of the i-vector space that optimally discriminates between the target language classes. Fig. 4 shows the constellation plot of EER against Cavg value for the LID system of Table II trained on utterances of 30 seconds duration, for different values of $K$. The plot suggests that the LID system yields an optimized performance around a certain i-vector dimension. Empirically, a value of $K = 260$ seems to achieve a near-optimal system performance when testing on 10 seconds utterances, and this value gradually evolves to $K = 400$ when the utterances become shorter. Further increasing $K$ to 600 does not reduce the LID error rate. Hence, this motivates the use of an i-vector dimension equal to 400 for all other experiments conducted in this paper.

D. Robust Features

In this section, we explore the LID performance for different feature representations that are extracted by the robust front-end of Section III-A. The three standard feature sets are compared to the FuSS features. The results in terms of the evaluation metrics, are presented in Table III for the 1, 3, 5 and 10 seconds duration tasks. Rows 1-4 correspond to the systems using a single UBM

<table>
<thead>
<tr>
<th>Test utterance duration</th>
<th>Duration of Training utterances</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 sec</td>
</tr>
<tr>
<td>1 sec</td>
<td>21.35</td>
</tr>
<tr>
<td>3 sec</td>
<td>-</td>
</tr>
<tr>
<td>5 sec</td>
<td>-</td>
</tr>
<tr>
<td>10 sec</td>
<td>-</td>
</tr>
</tbody>
</table>
TABLE III
EER, CAVG and DCF METRICS (ALL IN %) FOR INDIVIDUAL LID SYSTEMS TRAINED ON DIFFERENT FEATURE REPRESENTATION (ROW 1-4) EVALUATED ON THE TEST SETS WITH SHORT UTTERANCE DURATION. THE PERFORMANCE OF FEATURE FUSION (ROW 5) USING FEATURES OF SYSTEMS 1-4, AND SYSTEM LEVEL FUSION (ROW 6) OF THE INDIVIDUAL LID SYSTEMS, ARE COMPARED TO THE PROPOSED UBM-FUSED RLID SYSTEM (ROW 7)

<table>
<thead>
<tr>
<th>System</th>
<th>Feature(s)</th>
<th>Feature Fused</th>
<th>Feature Fusion</th>
<th>UBM-fused</th>
<th>UBM-fused</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PLP</td>
<td>1x2048</td>
<td>8.06 7.66 7.57</td>
<td>8.72 8.59 8.55</td>
<td>11.97 11.66 11.60</td>
</tr>
<tr>
<td>2</td>
<td>MFCC</td>
<td>1x2048</td>
<td>7.22 7.27 7.14</td>
<td>8.00 8.04 7.92</td>
<td>10.52 10.55 10.30</td>
</tr>
<tr>
<td>3</td>
<td>GFCC</td>
<td>1x2048</td>
<td>7.58 7.44 7.29</td>
<td>8.54 8.67 8.35</td>
<td>11.12 11.37 11.04</td>
</tr>
<tr>
<td>4</td>
<td>FuSS</td>
<td>1x2048</td>
<td>6.79 6.49 6.36</td>
<td>7.64 7.74 7.58</td>
<td>10.22 10.16 10.00</td>
</tr>
<tr>
<td>5</td>
<td>feature</td>
<td>1x2048</td>
<td>8.46 8.78 8.35</td>
<td>9.79 10.09 9.71</td>
<td>13.51 13.93 13.47</td>
</tr>
<tr>
<td>6</td>
<td>system</td>
<td>1-4</td>
<td>1x2048</td>
<td>6.25 6.05 6.05</td>
<td>7.28 7.33 7.10</td>
</tr>
<tr>
<td>7</td>
<td>UBM-fusion</td>
<td>4x512</td>
<td>5.77 5.45 5.40</td>
<td>6.61 6.59 6.46</td>
<td>8.36 8.31 8.33</td>
</tr>
</tbody>
</table>

Fig. 4. EER versus Cavg for the UBM-fused i-vector system evaluated on the 3 seconds TEST set. The system was trained on training utterances of 3, 10 and 30 seconds duration. The trend towards higher accuracies for long-duration training experimentally motivates the discussion related to Fig. 3. Different i-vector dimensions of 200, 400 and 600 were used to tune this value for reporting on the DEV-2 set.

with 2048 components trained on the features of Section III-A. The LID system trained of the FuSS features performs 3-6% better compared to the systems trained on standard features. This difference can be explained since the FuSS features captures, beside spectral shape, information about spectro-temporal modulations, voicing and long-term spectral variability.

Feature level fusion has been reported to be efficient when the number of target classes are large, such as in speaker verification. Here, concatenated feature vectors can be processed by a Linear Discriminant Analysis (LDA) step to discriminate the feature along the class labels. Due to the low number of language classes, LDA discrimination is not effective for the RATS LID task. Feature fusion was applied by concatenating all four feature streams followed by PCA to decorrelate and reduce the dimensionality to 120 components retaining around 95% of the total variance. The evaluation metrics are shown in row 5 of Table III and illustrate the limitations of feature fusion in our LID system. Similar as in [51], exploiting the diversity of the features by means of stacking does not guarantee performance gains as it could introduce inconsistencies in the fusion that result in inaccurate UBM modeling. When multiple LID systems are available during testing, linear fusion of the individual systems is typically applied instead of feature fusion to further improve performance [14], [15], [17], [21], [32]. In this paper, the fusion is done by running four systems in parallel and computing the linear combination of the SVM output probabilities. The results of system level fusion are given in row 6 of Table III and show a 5-8% relative improvement compared the best individual system, i.e. the FuSS feature LID system of row 4.

E. UBM-Fusion

The impact of the UBM Baum-Welch statistics on LID performance was experimentally shown in Section V-B by means of long utterance duration training. Another approach to improve i-vector space modeling by manipulation of UBM counts can be done through the UBM-fusion technique of Section III-C.

Fig. 5 shows how the DET-curves of the individual LID systems of Table III-row 2 are moved downward lowering error rates. The dashed lines correspond to the DET-curves
for the 3 and 10 seconds duration test set by the RLID system where the i-vector model was derived from the 30 seconds duration training set, using MFCC features and a UBM of 2048 components. The effect of UBM-fusion on the RLID error rate is illustrated by the solid DET-curves of Fig. 5. These curves are obtained by the RLID system where the i-vector training was performed by combining four UBMs of 512 components, each trained on one of the four feature representations of Section III-A, into a single UBM. The explanation of the positive effect UBM-fusion lies in the estimation of a better, more discriminant i-vector space that is derived from Baum-Welch statistics extracted from an increased number of dominant and diverse UBM components.

The evaluation metrics on the 1, 3, 5 and 10 seconds duration tasks are given in row 7 of Table III. As mentioned in Section III-C, the computational complexity of the UBM-fused RLID system during testing is identical to those of the individual systems, while this is not the case for system level fusion, i.e. the complexity is multiplied by the number of systems in the fusion. It is interesting to observe that UBM-fusion also outperforms system level fusion in LID accuracy, with relative improvements obtained around 13-18% of the best individual LID system.

F. Exploring the Utterance Duration Limits of RLID

The results of Table III show how the language identification performance decreases when the duration of the test set utterances reduces from 10 seconds to 1 second. Therefore, it would be interesting to explore the bounds on amount of (language-specific) information as a function of speech duration. From information theory, it is known that Fano’s inequality [52] relates Bayes error rate with mutual information [53] and provides a lower bound on the error probability for any classification framework that tries to infer labels as a function of feature vectors. In the current LID framework, the amount of mutual information between an utterance \( j \) and its language label is defined by the level of statistical dependency between the label variable and the corresponding i-vector representation \( \mathbf{w}_j \) of the utterance.

Similar to the approach in [54], [55], the feature space is first quantized into a finite number \( Z \) of quantization bins by means of K-means clustering, denoting the quantization function by \( Q(\cdot) : \mathbb{R}^K \rightarrow \mathbb{R} \). This allows to apply the discrete version of the mutual information criterion by using frequency counts to estimate the distributions \( P(Q(w)) \), \( P(x) \) and the joint distribution \( P(Q(w), x) \). The mutual information, measured in bits, is then given as

\[
\sum_{x=1}^{Z} \sum_{z=1}^{Z} P(Q(w) = z, x = x) \log_2 \frac{P(Q(w) = z, x = x)}{P(Q(w) = z)P(x = x)}
\]

It was demonstrated in [56] that \( I(Q(w), x) \leq I(w, x) \) and converges to \( I(w, x) \) when the number of quantization bins increases.

The mutual information results estimated on the RATS training corpus are given in Fig. 6(a). Here, the i-vector space of the single UBM and UBM-fused RLID system of Section V-E is quantized into \( Z = 2048 \) bins. Note that according to Table I, \( P(x) \) is uniformly distributed along the

\( X = 6 \) target language classes. Fig. 6(a) shows that error rate improvements, either by algorithmic advances or longer utterance durations, are related to amount of information extracted from the speech. A significant gain in mutual information is observed from 250 milliseconds to 2.3 seconds, followed by an important improvement up to 5 seconds and convergence towards 10 seconds. These findings on information content versus utterance duration also relate to the human performance bounds in language identification on the perceptual experiments conducted in [1] using speech sentences of increasing length.

The performance of the UBM-fused RLID system described above is also evaluated on test sets with decreasing utterance length. Fig. 6(b) presents the EER and accuracy plot for duration ranging from 10 seconds to as low as 250 milliseconds. From the figure, one can infer that the drop in accuracy tends to be more steep when the duration of the utterances becomes smaller than 2-3 seconds.

For utterances that are about half a second long, the accuracy is 63.4%, which decreases to 57.9% when the duration becomes a quarter of a second. Although the accuracy is much lower compared to longer test durations (around 80% for utterances longer than 2.5 seconds), a considerable amount of those very short speech utterances can still be classified correctly. Note that the
effect of silent frames arising from pauses during speaking contributes minimally to the discrimination between spoken languages when the utterances are long. The proportional presence of these frames might become more influential when the speech utterances become shorter.

Relating both subplots of Fig. 6(a) clearly shows how the RLID system accuracy is related to the amount of language-specific information available in speech. These results also suggest that there is still relevant acoustic information available in the very short speech utterances to discriminate between languages.

VI. CONCLUSIONS

This paper presents our recent technological advances in the domain of rapid language identification. The proposed RLID system extracts relevant acoustic cues of the spoken languages and deploys an i-vector based framework to discriminate between the target languages of interest. A robust front-end scheme, that includes a novel feature representation set, was proposed that significantly lowers the LID error rates when compared to standard features. To make fast and almost instant decision about the spoken language, a simplified modeling framework was integrated within the RLID system to increase the efficiency of the i-vector extraction process during training and testing. Techniques for accurate i-vector space modeling were proposed to further improve the identification performance on short duration speech utterances. Experimental analysis on the RATS data corpus showed the potential of i-vector training procedures such as UBM-fused total variability modeling and long duration training, especially when the duration of the speech utterances decreased to only a few seconds or even shorter lengths. Finally, we compared language identification accuracy to the amount of language information as a function of the utterance duration.

Future research work on rapid language identification involves rapid learning of new language identities, system adaptability to changing acoustic environments and integration of phonotactic language information to further lower the error rates on short duration speech utterances. The source code in support of this paper is available at www.mvansegbroeck.com/rlid licensed under Apache v2.0.

ACKNOWLEDGMENT

The authors thank Ming Li of SYSS-CMU Joint Institute of Engineering for useful discussions and software distribution.

REFERENCES


