Factors Affecting i-Vector Based Foreign Accent Recognition: a Case Study in Spoken Finnish

Hamid Behravan\textsuperscript{a,b,*}, Ville Hautamäki\textsuperscript{a}, Tomi Kinnunen\textsuperscript{a}

\textsuperscript{a}School of Computing, University of Eastern Finland, Box 111, FIN-80101 Joensuu, Finland
\textsuperscript{b}School of Languages and Translation Studies, University of Turku, Turku, Finland

Abstract

\textit{i-vector} based recognition is a well-established technique in state-of-the-art speaker and language recognition but its use in dialect and accent classification has received less attention. In this work, we extensively experiment with the spectral feature based i-vector system on Finnish foreign accent recognition task. Parameters of the system are initially tuned with the CallFriend corpus. Then the optimized system is applied to the \textit{Finnish national foreign language certificate} (FSD) corpus. The availability of suitable Finnish language corpora to estimate the hyper-parameters is necessarily limited in comparison to major languages such as English. In addition, it is not immediately clear which factors affect the foreign accent detection performance most. To this end, we assess the effect of three different components of the foreign accent recognition: 1) recognition system parameters, 2) data used for estimating hyper-parameters and 3) language aspects. We find out that training the hyper-parameters from non-matched dataset yields poor detection error rates in comparison to training from application-specific dataset. We also observed that, the mother tongue of speakers with higher proficiency in Finnish are more difficult to detect than of those speakers with lower proficiency. Analysis on age factor suggests that mother tongue detection in older speaker groups is easier than in younger speaker groups. This suggests that mother tongue traits might be more preserved in older speakers when speaking the second language in comparison to younger speakers.

*Corresponding author

Email addresses: behravan@cs.uef.fi (Hamid Behravan), villeh@cs.uef.fi (Ville Hautamäki), tkinnu@cs.uef.fi (Tomi Kinnunen)
1. Introduction

Foreign spoken accents are caused by the influence of one’s first language on the second language (Flege et al., 2003). For example, an English-Finnish bilingual speaker may have an English accent in his/her spoken Finnish because of learning Finnish later in life. Non-native speakers induce variations in different word pronunciation and grammatical structures into the second language (Grosjean, 2010). Interestingly, these variations are not random across speakers of a given language, because the original mother tongue is the source of these variations (Witteman, 2013). Nevertheless, between-speaker differences, gender, age and anatomical differences in vocal tract generate within-language variation (Witteman, 2013). These variations are nuisance factors that adversely affect detection of the mother tongue.

Foreign accent recognition is a topic of great interest in the areas of intelligence and security including immigration and border control sites. It may help officials to detect travelers with a fake passport by recognizing the immigrant’s actual country and region of spoken foreign accent (GAO, 2007). It has also a wide range of commercial applications including services based on user-agent voice commands and targeted advertisement.

Similar to spoken language recognition (Li et al., 2013), various techniques including phonotactic (Kumpf and King, 1997; Wu et al., 2010) and acoustic approaches (Bahari et al., 2013; Scharenborg et al., 2012; Behravan et al., 2013) have been proposed to solve the foreign accent detection task. The former uses phonemes and phone distributions to discriminate different accents; in practice, it uses multiple phone recognizer outputs followed by language modeling (Zissman, 1996). The acoustic approach in turn uses information taken directly from the spectral characteristics of the audio signals in the form of mel-frequency cepstral coefficient (MFCC) or shifted delta cepstra (SDC) features derived from MFCCs (Kohler and Kennedy, 2002). The spectral features are then modelled by a ”bag-of-frames” approach such as universal background model (UBM) with adaptation (Torres-Carrasquillo et al., 2004) and joint factor analysis (JFA) (Kenny, 2005). For an excellent
recent review of the current trends and computational aspects involved in
general language recognition tasks including foreign accent recognition, we
point the interested reader to (Li et al., 2013).

Among the acoustic systems, total variability model or \textit{i-vector} approach
originally used for speaker recognition (Dehak et al., 2011a), has been suc-
cessfully applied to language recognition tasks (González et al., 2011; Dehak
et al., 2011b). It consists of mapping speaker and channel variabilities to
a low-dimensional space called \textit{total variability space}. To compensate inter-
session effects, this technique is usually combined with \textit{linear discriminant
analysis} (LDA) (Fukunaga, 1990) and \textit{within-class covariance normalization}
(WCCN) (Kanagasundaram et al., 2011).

The \textit{i-vector} approach has received less attention in dialect and accent
recognition systems. Caused by more subtle linguistic variations, dialect
and accent recognition are generally more difficult than language recognition
(Chen et al., 2010). Thus, it is not obvious how well \textit{i-vectors} will perform
on these tasks. However, more fundamentally, the \textit{i-vector} system has many
data-driven components for which training data needs to be selected. It
would be tempting to train some of the hyper-parameters on a completely
different out-of-set-data (even different language), and leave only the final
parts — training and testing a certain dialect or accent — to the trainable
parts. This is also motivated by the fact that there is a lack of linguistic
resources available for languages like Finnish, comparing to English for which
corpora from NIST\footnote{http://www.itl.nist.gov/iad/mig/tests/spk/} and LDC\footnote{http://www.ldc.upenn.edu/} exist.

The \textit{i-vector} based dialect and accent recognition has previously been ad-
dressed in (DeMarco and Cox, 2012) and (Bahari et al., 2013). (DeMarco
and Cox, 2012) addressed a British dialect classification task with fourteen
dialects, resulting in 68 \% overall classification rate while (Bahari et al.,
2013) compared three accent modeling approaches in classifying English ut-
terances produced by speakers of seven different native languages. The accu-
gravity of the \textit{i-vector} system was found comparable as compared to the other
two existing methods. These studies indicate that the \textit{i-vector} approach is
promising for dialect and foreign accent recognition tasks. However, it can
be partly attributed to availability of massive development corpora includ-
ing thousands of hours of spoken English utterances to train all the system
hyper-parameters. The present study presents a case when such resources are not available.

Comparing with the prior studies including our own preliminary analysis (Behravan et al., 2013), the new contribution of this study is a detailed account into factors affecting the i-vector based foreign accent detection. We study this from three different perspectives: parameters, development data, and language aspects. Firstly, we study how the various i-vector extractor parameters, such as the UBM size and i-vector dimensionality, affect accent detection accuracy. This classifier optimization step is carried out using the speech data from the CallFriend corpus (Canavan and Zipperle, 1996). As a minor methodological novelty, we study applicability of heteroscedastic linear discriminant analysis (HLDA) for supervised dimensionality reduction of i-vectors. Secondly, we study data-related questions on our accented Finnish language corpus. We explore how the choices of the development data for UBM, i-vector extractor and HLDA matrices affect accuracy; we study whether these could be trained using a different language (English). If the answer turn out positive, the i-vector approach would be easy to adopt to other languages without recourse to the computationally demanding steps of UBM and i-vector extractor training. Finally, we study language aspects. This includes three analyses: ranking of the original accents in terms of their detection difficulty, study of confusion patterns across different accents and finally, relating recognition accuracy with four affecting factors such as Finnish language proficiency, age of entry, level of education and where the second language is spoken.

Our hypothesis for the Finnish language proficiency is that recognition accuracy would be adversely affected by proficiency in Finnish. In other words, we expect higher accent detection errors for speakers who speak fluent Finnish. For the age of entry factor, we expect that the younger a speaker enters a foreign country, the higher the probability of fluency in the second language. Thus, we hypothesize that it is more difficult to detect the speaker’s mother tongue in younger age groups than in older ones. This hypothesis is reasonable also because older people tend to keep their mother tongue traits more often than younger people (Munoz, 2010). Regarding the education factor, we hypothesize that mother tongue detection is more difficult in higher educated speakers than in lower educated ones. Finally, We also hypothesize that mother tongue detection is more difficult for the person who consistently use their second languages for social interaction as compared to the speakers who do not use their second language in regular
basis for social interaction.

2. System Components

Figure 1 shows the block diagram of the method used in this work. The i-vector system consists of two main part: front-end and back-end. The former consists of cepstral feature extraction and UBM training, whereas the latter includes sufficient statistics computation, training of the T-matrix, i-vector extraction, dimensionality reduction and scoring.

2.1. i-Vector System

I-vector modeling (Dehak et al., 2011a) is inspired by the success of joint factor analysis (JFA) (Kenny et al., 2008) in speaker verification. In JFA, speaker and channel effects are independently modeled using eigenvoice (speaker subspace) and eigenchannel (channel subspace) models:

\[ M = m + Vy + Ux, \]  \hspace{1cm} (1)
where $M$ is the speaker supervector, $m$ is a speaker and channel independent supervector created by concatenating the centers of UBM and low-rank matrices $V$ and $U$ represent, respectively, linear subspaces for speaker and channel variability in the original mean supervector space. The latent variables $x$ and $y$ are assumed to be independent of each other and have a standard normal distributions, i.e. $x \sim \mathcal{N}(0, I)$ and $y \sim \mathcal{N}(0, I)$. (Dehak et al., 2011a) found that these subspaces are not completely independent, therefore a combined total variability modeling was introduced.

In the i-vector approach, the GMM supervector ($M$) of each accent utterance is decomposed as (Dehak et al., 2011a),

$$M = m + Tw,$$

where $m$ is again the UBM supervector, $T$ is a low-rank rectangular matrix, representing between-utterance variability in the supervector space, and $w$ is the i-vector, a standard normally distributed latent variable drawn from $\mathcal{N}(0, I)$. The $T$ matrix is trained using a similar technique which is used to train $V$ in JFA, except that each training utterance of a speaker model is treated as belonging to different speakers. Therefore, in contrast to JFA, the $T$ matrix training does not need speaker or dialect labels. To this end, i-vector approach is an unsupervised learning method. The i-vector $w$ is estimated from its posterior distribution conditioned on the Baum-Welch statistics extracted from the utterance using the UBM (Dehak et al., 2011a).

The i-vector extraction can be seen as a mapping from a high-dimensional GMM supervector space to a low-dimensional i-vector that preserves most of the variability. In this work, we use 1000-dimensional that are further length normalized and whitened (Garcia-Romero and Espy-Wilson, 2011).

Cosine scoring is commonly used for measuring similarity of two i-vectors (Dehak et al., 2011a). The cosine score $t$ of the test i-vector, $w_{test}$, and the i-vectors of target accent $a$, $w_{target}^a$, is defined as their inner product $\langle w_{test}, w_{target}^a \rangle$ and computed as follows:

$$t = \frac{\hat{w}_{test}^T \hat{w}_{target}^a}{\| \hat{w}_{test} \| \| \hat{w}_{target}^a \|},$$

where $\hat{w}_{test}$ is,

$$\hat{w}_{test} = A^T w_{test},$$

and $A$ is the HLDA projection matrix (Loog and Duin, 2004) to be detailed below in section 2.2. Further, $\hat{w}_{target}^a$ is the average i-vector over all the
training utterances in accent \( a \), i.e.

\[
\hat{w}_\text{target}^a = \frac{1}{N_a} \sum_{i=1}^{N_a} \hat{w}_i^a,
\]

(5)

where \( N_a \) is the number of training utterances in accent \( a \) and \( \hat{w}_i^a \) is the projected i-vector of training utterance \( i \) from accent \( a \), computed the same way as (4).

Obtaining the scores \( \{t_a, a = 1, \ldots, L\} \) for a particular test utterance compared with all the \( L \) target accent models of accent \( a \), those scores are further post-processed as (Brümmer and van Leeuwen, 2006):

\[
t'(a) = \log \frac{\exp(t_a)}{\sum_{k \neq a} \exp(t_k)},
\]

(6)

where \( t'(a) \) is the detection log-likelihood ratio or final score used in the detection task.

2.2. Reducing the i-Vector Dimensionality

As the extracted i-vectors contain both intra- and between-accent variations, the aim of dimensionality reduction is to project the i-vectors onto a space where between-accent variability is maximized and intra-accent variability is minimized. Traditionally, LDA is used to perform dimensionality reduction where, for \( R \)-class classification problem, the maximum projected dimension is \( R - 1 \).

As (Loog and Duin, 2004) argue, these \( R - 1 \) dimensions do not necessarily contain all the discriminant information for the classification task. Moreover, LDA separates only the class means and it does not take into account the discriminant information in the class covariances. In recent years, an extension of LDA, heteroscedastic linear discriminant analysis (HLDA), has gained popularity in speech research community. HLDA, unlike LDA, deals with discriminant information presented both in the means and covariance matrices of classes (Loog and Duin, 2004).

HLDA was originally introduced in (Kumar, 1997) for auditory feature extraction, and later applied to speaker (Burget et al., 2007) and language (Rouvier et al., 2010) recognition with the purpose of reducing dimensionality of GMM supervectors and acoustic features, respectively. In this work, we also use it to reduce the dimensionality of extracted i-vectors. For completeness, we briefly summarize the HLDA technique below.
In the HLDA technique, the i-vectors of dimension $n$ are projected into first $p < n$ rows, $d_j=1...p$, of $n \times n$ HLDA transformation matrix denoted by $A$. The matrix $A$ is estimated by an efficient row-by-row iteration method (Gales, 1999), whereby each row is iteratively estimated as,

$$
\hat{d}_k = c_k G^{k-1} \sqrt{\frac{N}{c_k G^{k-1} c_k^T}}.
$$

(7)

Here, $c_k$ is the $k^{th}$ row vector of the co-factor matrix $C = |A|^A^{-1}$ for the current estimate of $A$ and

$$
G^k = \begin{cases} 
\sum_{j=1}^J \frac{N_j}{d_k \tilde{\Sigma}^{(j)} d_k} \tilde{\Sigma}^{(j)} & k \leq p \\
\frac{N}{d_k \hat{\Sigma}} \hat{\Sigma} & k > p,
\end{cases}
$$

(8)

where $\hat{\Sigma}$ and $\hat{\Sigma}^{(j)}$ are estimates of the class-independent covariance matrix and the covariance matrix of the $j^{th}$ model, $N_j$ is the number of training utterances of the $j^{th}$ model and $N$ is the total number of training utterances. To avoid near-to-singular covariance matrices in HLDA training process, principal component analysis (PCA) is first applied (Loog and Duin, 2004; Rao and Mak, 2012) and the PCA-projected features are used as the inputs to HLDA. The dimension of PCA is selected in such a manner that most of the principal components are retained and within-models scatter matrix becomes non-singular (Loog and Duin, 2004).

2.3. Within-Class Covariance Normalization

To compensate for unwanted intra-class variations in the total variability space, within-class covariance normalization (WCCN) (Hatch et al., 2006) is applied to the extracted i-vectors. To this end, a within-class covariance matrix, $\Lambda$, is first computed using,

$$
\Lambda = \frac{1}{L} \sum_{a=1}^L \frac{1}{N_a} \sum_{i=1}^{N_a} (w^a_i - \bar{w}_a)(w^a_i - \bar{w}_a)^T,
$$

(9)

where $\bar{w}_a$ is the mean i-vector for each accent $a$, $L$ is the number of target accents and $N_a$ is the number of training utterances for the accent $a$. The inverse of $\Lambda$ is then used to normalize the direction of the projected i-vectors in the cosine kernel. This is equivalent to projecting the i-vector subspace by the matrix $B$ obtained by Cholesky decomposition of $\Lambda^{-1} = BB^T$. 

8
3. Experimental Setup

3.1. Corpus

We use Finnish national foreign language certificate (FSD) corpus (University of Jyväskylä, 2000) to perform foreign accent classification task. The corpus consists of official language proficiency tests for foreigners interested in Finnish language proficiency certificate for the purpose of applying for a job or citizenship. All the data has been recorded by language experts. Generally, the test is intended for evaluating test-takers’ proficiency in listening comprehension, reading comprehension, speaking, and writing. This test can be taken at basic, intermediate and advanced levels. The test-takers choose the proficiency level at which they wish to participate. The difference between the levels is the extent and variety of expression required. At the basic level, it is important that test-takers convey their message in a basic form, while in the intermediate level, richer expression is required. More effective and natural expressions should be presented in the advanced level. However, communication purposes, i.e. functions and questions, are more or less the same at all levels. Table 1 shows the grading scale at each level of the tests in this corpus.

<table>
<thead>
<tr>
<th>Levels</th>
<th>Grades</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic</td>
<td>0 1 2</td>
</tr>
<tr>
<td>Intermediate</td>
<td>3 4</td>
</tr>
<tr>
<td>Advanced</td>
<td>5 6</td>
</tr>
</tbody>
</table>

For our purposes, we selected Finnish responses corresponding to 18 foreign accents. Unfortunately, as the number of utterances in some accents was not large enough, a limited number of eight accents — Russian, Albanian, Arabic, English, Estonian, Kurdish, Spanish, and Turkish — with enough data were chosen for the experiments. However, the unused accents were utilized in training the hyper-parameters of the i-vector system, the UBM and the T-matrix.

---

3 The FSD corpus is available by request from http://yki-korpus.jyu.fi/. Filelists used in this study are available by request from the first author.
To perform the recognition task, each accent set is randomly partitioned into a training and a test subset. To avoid speaker and session bias, the same speaker was not placed into the test and train subsets. The test subset corresponds to (approximately) 40% of the utterances, while the training set corresponds to the remaining 60%. The original audio files, stored in MPEG-2 Audio Layer III (mp3) compressed format, were decompressed, resampled to 8 kHz and partitioned into 30-second chunks. Table 2 shows the distribution of train and test files in each target accent.

Table 2: Train and test files distributions in each target accent in the FSD corpus.

<table>
<thead>
<tr>
<th>Accent</th>
<th>No. of train files</th>
<th>No. of test files</th>
<th>No. of speakers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spanish</td>
<td>47</td>
<td>25</td>
<td>15</td>
</tr>
<tr>
<td>Albanian</td>
<td>56</td>
<td>29</td>
<td>19</td>
</tr>
<tr>
<td>Kurdish</td>
<td>61</td>
<td>32</td>
<td>21</td>
</tr>
<tr>
<td>Turkish</td>
<td>66</td>
<td>34</td>
<td>22</td>
</tr>
<tr>
<td>English</td>
<td>70</td>
<td>36</td>
<td>23</td>
</tr>
<tr>
<td>Estonian</td>
<td>122</td>
<td>62</td>
<td>38</td>
</tr>
<tr>
<td>Arabic</td>
<td>128</td>
<td>66</td>
<td>42</td>
</tr>
<tr>
<td>Russian</td>
<td>556</td>
<td>211</td>
<td>235</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>1149</strong></td>
<td><strong>495</strong></td>
<td><strong>415</strong></td>
</tr>
</tbody>
</table>

The NIST SRE 2004\(^4\) corpus was chosen as the out-of-set-data for hyperparameter training. For our purposes, 1000 gender-balanced utterances were randomly selected from this corpus to train the UBM and T-matrix. We note that this is an American English corpus of telephone-quality speech.

Unlike UBM and T-matrix, training the HLDA projection matrix requires labeled data. Since accent labels are not represented in the NIST corpus, we use the CallFriend corpus (Canavan and Zipperle, 1996) to train HLDA. This corpus is a collection of unscripted conversations of 12 languages recorded over telephone lines. It includes two dialects for each target language available. All utterances are organized into training, development and evaluation subsets. For our purposes, we selected all the training utterances from dialects of English, Mandarin and Spanish languages and partitioned them into 30-second chunks, resulting in approximately 4000 splits per each subset. All audio files have 8 kHz sampling rate.

\(^4\)http://catalog.ldc.upenn.edu/LDC2006S44
3.2. Front-end Configuration

The front-end consists of concatenation of MFCC and SDC coefficients (Kohler and Kennedy, 2002). To this end, speech signals framed with 20ms Hamming window with 50% overlap are filtered by 27 mel-scale filters over 0-4000 Hz frequency range. RASTA filtering (Hermansky and Morgan, 1994) is applied to log-filterbank energies. Seven first cepstral coefficients (c0-c6) are computed using discrete cosine transform. The cepstral coefficients are further processed using utterance-level cepstral mean and variance normalization (CMVN) and vocal tract length normalization (VTLN) (Lee and Rose, 1996), and converted into 49-dimensional shifted delta cepstra (SDC) feature vectors with 7-1-3-7 configuration parameters (Kohler and Kennedy, 2002). These four parameters correspond to, respectively, the number of cepstral coefficients, time delay for delta computation, time shift between consecutive blocks, and number of blocks for delta coefficient concatenation. Removing non-speech frames, the 7 first MFCC coefficients (including c0) are further concatenated to SDCs to obtain 56-dimensional feature vectors.

In a preliminary experiment on our evaluation corpus FSD (Behravan, 2012), the combined feature set is shown to give a relative decrease in EER of more than 30% as compared to the only SDC feature based technique.

3.3. Objective Evaluation Metrics

System performance is reported in terms of both average equal error rate (EER_{avg}) and average detection cost (C_{avg}) (Li et al., 2013). EER indicates the operating point on detection error trade-off (DET) curve (Martin et al., 1997) at which false alarm and miss rates are equal. EER per target accent is computed in a manner that other accents serve as non-target trials. Average equal error rate (EER_{avg}) is computed by taking the average over all the \( L \) target accent EERs.

\[ C_{avg} = \frac{1}{L} \sum_{a=1}^{L} C_{DET}(L_a), \]  \hspace{1cm} (10)

where \( C_{DET}(L_a) \) is the detection cost for subset of test segments trials for which the target accent is \( L_a \):

\[ C_{DET}(L_a) = C_{miss} P_{tar} P_{miss}(L_a) + C_{fa}(1 - P_{tar}) \frac{1}{L - 1} \sum_{m \neq a} P_{fa}(L_a, L_m). \]  \hspace{1cm} (11)
\( P_{\text{miss}} \) denotes the miss probability (or false rejection rate), i.e. a test segment of accent \( L_a \) is rejected as not being in that accent. \( P_{fa}(L_a, L_m) \) is the probability when a test segment of accent \( L_m \) is detected as accent \( L_a \). It is computed for each target/non-target accent pairs. \( C_{\text{miss}} \) and \( C_{fa} \) are costs of making errors and are set to 1. \( P_{\text{tar}} \) is the prior probability of a target accent and is set to 0.5.

4. Results

We first optimize the i-vector parameters in the context of dialect and accent recognition tasks. For this purpose, we utilize the CallFriend corpus. The results are summarized in Table 3.

Table 3: The i-vector system’s optimum parameters as reported in (Behravan et al., 2013).

<table>
<thead>
<tr>
<th>i-vector parameters</th>
<th>Search range and optima</th>
</tr>
</thead>
<tbody>
<tr>
<td>UBM size</td>
<td>256, 512, 1024, 2048, 4096</td>
</tr>
<tr>
<td>i-vector dimensionality</td>
<td>200, 400, 600, 800, 1000</td>
</tr>
<tr>
<td>HLDA dimensionality</td>
<td>50, 100, 150, 180, 220, 300, 350, 400</td>
</tr>
</tbody>
</table>

In Figure 2, we show EER as a function of HLDA output dimension. We find that the optimal dimension of the HLDA projected i-vectors is 180 and too aggressive reduction in dimension decreases accuracy. We also find that accuracy improves with the increase of i-vector dimensionality as Table 4 shows. Furthermore, our results showed that the UBM with smaller size outperforms larger UBM as Table 5 shows. Based on these previous findings, UBM size, i-vector size and output dimensionality are set to 512, 1000 and 180, respectively.

Table 4: Performance of the i-vector system in the CallFriend corpus for selected i-vector dimensions (EER in %, form). UBM has 1024 Gaussians as reported in (Behravan et al., 2013).

<table>
<thead>
<tr>
<th>i-vector dim.</th>
<th>English</th>
<th>Mandarin</th>
<th>Spanish</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>23.20</td>
<td>20.49</td>
<td>20.87</td>
</tr>
<tr>
<td>400</td>
<td>22.60</td>
<td>19.11</td>
<td>20.21</td>
</tr>
<tr>
<td>600</td>
<td>21.30</td>
<td>18.45</td>
<td>19.63</td>
</tr>
<tr>
<td>800</td>
<td>19.83</td>
<td>16.31</td>
<td>18.63</td>
</tr>
<tr>
<td>1000</td>
<td><strong>18.01</strong></td>
<td><strong>14.91</strong></td>
<td><strong>16.01</strong></td>
</tr>
</tbody>
</table>
Figure 2: Equal error rates at different dimensions of the HLDA projected i-vectors in the CallFriend corpus as reported in (Behravan et al., 2013).

Table 5: Performance of the i-vector system in the CallFriend corpus for five selected UBM sizes (EER in %, form). i-vectors are of dimension 600 as reported in (Behravan et al., 2013).

<table>
<thead>
<tr>
<th>UBM size</th>
<th>English</th>
<th>Mandarin</th>
<th>Spanish</th>
</tr>
</thead>
<tbody>
<tr>
<td>256</td>
<td>21.12</td>
<td>17.93</td>
<td>19.00</td>
</tr>
<tr>
<td>512</td>
<td>21.61</td>
<td>17.91</td>
<td>19.15</td>
</tr>
<tr>
<td>1024</td>
<td>21.30</td>
<td>18.45</td>
<td>19.63</td>
</tr>
<tr>
<td>2048</td>
<td>23.81</td>
<td>21.15</td>
<td>22.01</td>
</tr>
<tr>
<td>4096</td>
<td>23.89</td>
<td>21.57</td>
<td>22.66</td>
</tr>
</tbody>
</table>

4.1. Effect of Development Data on i-Vector Hyper-parameters Estimation

Table 6 shows the results on the FSD corpus when the hyper-parameters are trained from different datasets. Here, WCCN and score normalization are not applied. By considering the first row with matched language as a baseline (13.37% EER$_{avg}$), we observe the impact of each of the hyper-parameter training configurations as follows:

- Effect of HLDA (row 1 vs row 2): EER$_{avg}$ increases to 18.28% (relative increase of 37%)
- Effect of T-matrix (row 1 vs 3): $\text{EER}_{\text{avg}}$ increases to 20.98% (relative increase of 57%)

- Effect of UBM (row 1 vs 4): $\text{EER}_{\text{avg}}$ increases to 23.85% (relative increase of 78%)

- Effect of UBM and T-matrix (row 1 vs 5): $\text{EER}_{\text{avg}}$ increases to 26.76% (relative increase of 101%)

Table 6: $\text{EER}_{\text{avg}}$ and $C_{\text{avg}} \times 100$ performance for effect of changing datasets in training the i-vector hyper-parameters. (WCCN and score normalization turned off.)

<table>
<thead>
<tr>
<th>UBM</th>
<th>T_matrix</th>
<th>HLDA</th>
<th>Database used for training</th>
<th>$\text{EER}_{\text{avg}}$</th>
<th>$C_{\text{avg}} \times 100$</th>
<th>Id$_{\text{error}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSD</td>
<td>FSD</td>
<td>FSD</td>
<td>FSD</td>
<td>13.37</td>
<td>7.04</td>
<td>33.65</td>
</tr>
<tr>
<td>FSD</td>
<td>FSD</td>
<td>CallFriend</td>
<td>FSD</td>
<td>18.28</td>
<td>7.49</td>
<td>38.29</td>
</tr>
<tr>
<td>FSD</td>
<td>NIST</td>
<td>FSD</td>
<td>FSD</td>
<td>20.98</td>
<td>7.83</td>
<td>40.30</td>
</tr>
<tr>
<td>NIST</td>
<td>FSD</td>
<td>FSD</td>
<td>FSD</td>
<td>23.85</td>
<td>8.15</td>
<td>42.91</td>
</tr>
<tr>
<td>NIST</td>
<td>NIST</td>
<td>FSD</td>
<td>FSD</td>
<td>26.76</td>
<td>8.41</td>
<td>44.67</td>
</tr>
</tbody>
</table>

In the light of these findings, it seems clear that the ‘early’ system hyper-parameters (UBM and T-matrix) have a much larger role and they should be trained from as closely matched data as possible; we see that when all the hyper-parameters are trained from the FSD corpus, the highest accuracy is achieved. The most severe degradation (101%) is attributed to the joint effect of UBM and T-matrix and the least severe (37%) to HLDA, T-matrix (57%) and UBM (78%) falling in between. It is instructive to recall the order of computations: sufficient statistics from UBM $\rightarrow$ i-vector extractor training $\rightarrow$ HLDA training. Since all the remaining steps depend on the “bottleneck” components, i.e. UBM and T-matrix, it is not surprising that they have the largest relative effect.

The generally large degradation relative to the baseline set-up with matched data is reasonably explained by the large differences between type of data of evaluation corpus (FSD) and hyper-parameter estimation corpora (NIST SRE and CallFriend). FSD consists of Finnish language data recorded with close-talking microphones in a classroom environment. Even though speech is very clear, background babble noise from the other students is evident in all the recordings. This is contrast to the NIST SRE and CallFriend corpora.
where most of the speech files are recorded over telephone line and babble noise is less common.

The results of Table 6 were computed with WCCN and score normalization turned off. Let us now turn our attention to these additional system components. Firstly, Table 7 shows the effect of score normalization when all the hyper-parameters are trained from the FSD corpus (i.e., row 1 of Table 6). EER$_{avg}$ decreases from 13.37% to 13.01%, which indicates a slightly increased recognition accuracy when the scores are normalized in the backend.

Table 7: Effect of score normalization on the recognition performance. (HLDA and WCCN turned on and off, respectively.)

<table>
<thead>
<tr>
<th>Score normalization</th>
<th>EER$_{avg}$%</th>
<th>C$_{avg}$×00</th>
<th>Id$_{error}$%</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>13.37</td>
<td>7.04</td>
<td>33.65</td>
</tr>
<tr>
<td>Yes</td>
<td>13.01</td>
<td>6.94</td>
<td>32.85</td>
</tr>
</tbody>
</table>

Secondly, Table 8 shows the joint effect of WCCN and HLDA on the recognition performance when all the hyper-parameters are trained from the FSD corpus (i.e., row 1 of Table 6). In addition to that, score normalization is also applied. EER$_{avg}$ decreases from 17.10% to 12.60% when both HLDA and WCCN are applied. The worst case is when HLDA is turned off and WCCN is turned on. This is because turning off HLDA leads to inaccurate estimation of covariance matrix in higher dimensional i-vector space.

Table 8: The joint effect of WCCN and HLDA on the recognition accuracy. (Score normalization turned on.)

<table>
<thead>
<tr>
<th>HLDA</th>
<th>WCCN</th>
<th>EER$_{avg}$%</th>
<th>C$_{avg}$×00</th>
<th>Id$_{error}$%</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>No</td>
<td>17.70</td>
<td>7.04</td>
<td>39.58</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>13.01</td>
<td>6.94</td>
<td>32.85</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>19.00</td>
<td>7.31</td>
<td>41.55</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>12.60</td>
<td>6.85</td>
<td>30.85</td>
</tr>
</tbody>
</table>

4.2. Comparing i-Vector and GMM-UBM Systems

In order to have a baseline comparison between the i-vector approach and the classical accent recognition systems, we used conventional GMM-UBM system with MAP adaptation similar to the work presented in (Torres-Carrasquillo et al., 2004). GMM-UBM system is simpler and computationally more efficient in comparison to the i-vector systems. Map adaptation
consists of single iteration for adapting the UBM to each dialect model using SDC+MFCC features. It requires updating only centers of UBM. The testing is a fast scoring process described in (Reynolds et al., 2000) to score the input utterance to each adapted foreign accent models by selecting top five Gaussians per speech frame.

Table 9 shows the result of GMM-UBM system with four different UBM sizes. Increasing the number of Gaussians results in higher recognition accuracy. Table 10 further compares the best recognition accuracies achieved by both recognizers. In the i-vector system, the best recognition accuracy, i.e. $EER_{avg}$ of 12.60%, is achieved with all the hyper-parameters trained from the FSD corpus and HLDA, WCCN and score normalization being turned on. On the other hand, the best GMM-UBM recognition accuracy, $EER_{avg}$ of 17.00%, is achieved with UBM order 2048 when score normalization is applied. The results indicate that the i-vector system outperforms the conventional GMM-UBM system with 25% relative improvements in terms of $EER_{avg}$ at the cost of higher computational time and additional development data.

Table 9: Recognition performance of GMM-UBM system with different UBM sizes.

<table>
<thead>
<tr>
<th>UBM size</th>
<th>$EER_{avg}$%</th>
<th>$C_{avg} \times 100$</th>
</tr>
</thead>
<tbody>
<tr>
<td>256</td>
<td>19.94</td>
<td>11.02</td>
</tr>
<tr>
<td>512</td>
<td>19.03</td>
<td>10.56</td>
</tr>
<tr>
<td>1024</td>
<td>18.20</td>
<td>10.12</td>
</tr>
<tr>
<td>2048</td>
<td>17.00</td>
<td>9.46</td>
</tr>
</tbody>
</table>

Table 10: Comparison between the best recognition accuracy in the GMM-UBM and i-vector system. (Score normalization turned on for the both cases.)

<table>
<thead>
<tr>
<th>Recognition system</th>
<th>$EER_{avg}$%</th>
<th>$C_{avg} \times 100$</th>
<th>Id_{error}$%$</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMM-UBM</td>
<td>17.00</td>
<td>9.46</td>
<td>43.65</td>
</tr>
<tr>
<td>i-vector</td>
<td>12.60</td>
<td>6.85</td>
<td>30.85</td>
</tr>
</tbody>
</table>

4.3. Detection Performance per Target Language

In the previous section, we analysed the overall average recognition accuracy. Now, here we focus on performance for each individual foreign accent. In order to compensate the lack of sufficient development data in reporting these results, we used the previously unused accents in the FSD corpus to
train UBM, T-matrix and HLDA. These unused accents are Chinese, Dari, Finnish, French, Italian, Somali, Swedish and Misc\textsuperscript{5} corresponding to 210 speakers and 1110 utterances in total. Further, to increase the number of test trials in the classification stage, we report the results using a leave-one-speaker-out (LOSO) protocol. As demonstrated in the pseudo code below, for every accent, each speaker’s utterances are held out one at a time and the remaining utterances are used in modeling the $\hat{w}_{\text{target}}$ as in Eq. (5). The held-out utterances are used as the evaluation utterances.

**Algorithm** Leave-one-speaker-out (LOSO)

Let $A = \{a_1, a_2, \ldots, a_L\}$ be the set of $L$ target accents
Let $S(a_i)$ be the set of speakers in target accent $a_i$
$\hat{w}_{\text{target}}^a$ defines the i-vectors of target accent $a$ after HLDA and WCCN.
for $a_i \in A$ do
  for $s_j \in S(a_i)$ \{Held-out test speaker\} do
    Let $S' = S(a_i) - s_j$ \{Remove the speaker being tested\}
    Form $\hat{w}_{\text{target}}^a$ using the i-vectors in set $S'$, Eq. (5)
    Compute cosine scores $\langle w_{\text{test}}^{s_j}, w_{\text{target}}^a \rangle$ \{$w_{\text{test}}^{s_j}$ are the test i-vectors of speaker $s_j$\}
  end for
end for
Normalize scores per each target accent, Eq. (6)

Table 11 shows the language wise results. The results suggest that certain languages which do not belong to the same sub-family as Finnish are easier to detect. Turkish achieves the highest recognition accuracy, whereas English shows highest error rate. The recognition accuracy is consistent among Albanian, Arabic, Kurdish and Russian languages. $C_{\text{avg}}$ is bigger than the results already given in Table 10. Note that in Table 11, the unused accents are used to train UBM, T-matrix and HLDA. This induces mismatch between model training data and the hyper-parameter training data. Which is not the case in Table 10.

Figure 3 further exemplifies the distribution of scores for three selected languages of varying detection difficulties. The histograms are plotted with the same number of bins, 50. For visualization purposes, the width of bins

\footnote{\textsuperscript{5}refers to those utterances in which the spoken foreign accent is not clear.}
Table 11: Per language results in terms of EER% and $C_{DET} \times 100$ for the i-vector system.

<table>
<thead>
<tr>
<th>Accents</th>
<th>EER%</th>
<th>$C_{DET} \times 100$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turkish</td>
<td>11.90</td>
<td>6.35</td>
</tr>
<tr>
<td>Spanish</td>
<td>16.49</td>
<td>6.92</td>
</tr>
<tr>
<td>Albanian</td>
<td>18.76</td>
<td>7.00</td>
</tr>
<tr>
<td>Arabic</td>
<td>18.98</td>
<td>7.17</td>
</tr>
<tr>
<td>Kurdish</td>
<td>19.37</td>
<td>7.19</td>
</tr>
<tr>
<td>Russian</td>
<td>19.68</td>
<td>7.21</td>
</tr>
<tr>
<td>Estonian</td>
<td>20.05</td>
<td>7.52</td>
</tr>
<tr>
<td>English</td>
<td>23.60</td>
<td>8.00</td>
</tr>
</tbody>
</table>

in the non-target score histogram was set smaller than in the target score histogram. The score distribution explains the differences between EERs. For example, in case of Turkish as the easiest and English as the most difficult detected accent, the overlap between the target and the non-target scores is higher in the latter.

Here, the problem is treated as foreign accent identification task. Table 12 displays the confusion matrix corresponding to Table 11. In all the cases, majority of the detected cases corresponds to the correct class (i.e., the entries in the diagonal). Taking Turkish as the language with the highest recognition accuracy, out of the 11 misclassified Turkish test segments, 7 were misclassified as Arabic. This might be because Turkey is bordered by two Arabic countries, Syria and Iraq, and Turkish shares common features with Arabic. Regarding Spanish, out of the 27 misclassified test segments, 9 were detected as Arabic. It is possibly due to the major influence of Arabic on Spanish. In particular, numerous words of Arabic origin are adopted in the Spanish language.

To analyze further reasons why some languages are harder to detect, we first compute the average target language score on a speaker-by-speaker basis. To measure the degree of speaker variation, we show the standard deviation of these average scores in Table 13, along with the corresponding EER and $C_{DET}$ values. The results indicate that languages with more diverse speaker populations, having speaker-dependent biases in the detection scores, are more difficult to handle. It does not yet explain why certain languages, such as Russian, have a larger degree of speaker variation, but suggests that there will be space for further research in speaker normalization techniques.
Figure 3: Distribution of scores for Turkish, Russian and English accents.
Table 12: Confusion matrix of the results corresponding to Table 11.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Turk.</td>
<td>50</td>
<td>0</td>
<td>1</td>
<td>7</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Span.</td>
<td>1</td>
<td>58</td>
<td>1</td>
<td>11</td>
<td>2</td>
<td>3</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>Alba.</td>
<td>1</td>
<td>0</td>
<td>61</td>
<td>9</td>
<td>1</td>
<td>5</td>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>Arab.</td>
<td>4</td>
<td>2</td>
<td>14</td>
<td>110</td>
<td>7</td>
<td>7</td>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>Kurd.</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>50</td>
<td>6</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>Russ.</td>
<td>51</td>
<td>21</td>
<td>51</td>
<td>26</td>
<td>2</td>
<td>369</td>
<td>13</td>
<td>28</td>
</tr>
<tr>
<td>Esto.</td>
<td>5</td>
<td>5</td>
<td>7</td>
<td>15</td>
<td>1</td>
<td>6</td>
<td>117</td>
<td>15</td>
</tr>
<tr>
<td>Engl.</td>
<td>7</td>
<td>3</td>
<td>3</td>
<td>6</td>
<td>3</td>
<td>7</td>
<td>9</td>
<td>59</td>
</tr>
</tbody>
</table>

Table 13: The standard deviation of the average target language score on a speaker-by-speaker basis along with the corresponding EER and $C_{DET}$ results.

<table>
<thead>
<tr>
<th>Accents</th>
<th>Standard deviation</th>
<th>EER%</th>
<th>$C_{DET} \times 100$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turkish</td>
<td>0.1205</td>
<td>11.90</td>
<td>6.35</td>
</tr>
<tr>
<td>Spanish</td>
<td>0.1369</td>
<td>16.49</td>
<td>6.92</td>
</tr>
<tr>
<td>Albanian</td>
<td>0.1380</td>
<td>18.76</td>
<td>7.00</td>
</tr>
<tr>
<td>Arabic</td>
<td>0.1505</td>
<td>18.98</td>
<td>7.17</td>
</tr>
<tr>
<td>Kurdish</td>
<td>0.1392</td>
<td>19.37</td>
<td>7.19</td>
</tr>
<tr>
<td>Russian</td>
<td>0.1402</td>
<td>19.68</td>
<td>7.21</td>
</tr>
<tr>
<td>Estonian</td>
<td>0.1621</td>
<td>20.05</td>
<td>7.52</td>
</tr>
<tr>
<td>English</td>
<td>0.1667</td>
<td>23.60</td>
<td>8.00</td>
</tr>
</tbody>
</table>

4.4. Factors Affect Foreign Accent Detection

We are interested to find out what factors affect the foreign accent recognition accuracies. The rich metadata available in the FSD corpus includes language proficiency, speaker’s age, education and the place where the second language is spoken. In the following analysis, we used the whole set of scores from the LOSO experiment and grouped them to different categories according to each metadata variable at a time.

Language Proficiency

To find out the impact of language proficiency, we take the sum of spoken and written Finnish grades in the FSD corpus as a proxy of the speaker’s Finnish language proficiency. The objective was to find out how speakers’ language proficiency and their detected foreign accent are related. Figure
Figure 4 shows $C_{avg}$ for each grade group. As hypothesized, the lowest $C_{avg}$ is attributed to speakers with the lower grade (5) and the highest accuracy to speakers with the higher grade (8). This indicates that detecting the foreign accents from speakers with higher proficiency in Finnish is considerably more difficult than speakers with lower proficiency.

In addition, we looked at language proficiency across different target languages. We study the average language proficiency grade across the speakers in different languages (Table 14). For the three most difficult languages to detect, Russian, Estonian and English, the average language proficiency grades are higher than the rest of languages, supporting the preceding analysis.

![Figure 4: $C_{avg} \times 100$ for different grade groups in the language proficiency measurement.](image)

**Table 14:** The average language proficiency grade across the speakers in different languages along with the corresponding EER and $C_{DET}$ results.

<table>
<thead>
<tr>
<th>Accent</th>
<th>Grade</th>
<th>EER%</th>
<th>$C_{DET} \times 100$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turkish</td>
<td>6.09</td>
<td>11.90</td>
<td>6.35</td>
</tr>
<tr>
<td>Spanish</td>
<td>6.20</td>
<td>16.49</td>
<td>6.92</td>
</tr>
<tr>
<td>Albanian</td>
<td>5.78</td>
<td>18.76</td>
<td>7.00</td>
</tr>
<tr>
<td>Arabic</td>
<td>5.73</td>
<td>18.98</td>
<td>7.17</td>
</tr>
<tr>
<td>Kurdish</td>
<td>5.71</td>
<td>19.37</td>
<td>7.19</td>
</tr>
<tr>
<td>Russian</td>
<td>6.30</td>
<td>19.68</td>
<td>7.21</td>
</tr>
<tr>
<td>Estonian</td>
<td>7.02</td>
<td>20.05</td>
<td>7.52</td>
</tr>
<tr>
<td>English</td>
<td>6.34</td>
<td>23.60</td>
<td>8.00</td>
</tr>
</tbody>
</table>
Age of entry

Age is one of the most important effective factors in learning a second language (Krishna, 2008). The common notion is that younger adults learn the second language more easily than older adults. (Larsen-Freeman, 1986) argues that during the period of time between birth and the age when a child enters puberty, learning a second language is quick and efficient. In the second language acquisition process, one of the affecting factors relates to the experience of immigrants, such as the age of entry and the length of residence (Krishna, 2008). We analyse the relationship between the age of entry and the foreign accent recognition results. To analyse the effect of age to foreign accent detection, we categorized the detection scores into six age groups with 10 years age interval (Figure 5). Our hypothesis was that mother tongue detection is easier in older people than younger ones. The results support this hypothesis. $C_{avg}$ decreases from 5.30 (a relative decrease of 16%) to 4.45 from the age group [11-20] to [61-70]. This indicates that the mother tongue detection in older age groups could be easier than in the younger age groups.

![Figure 5: $C_{avg} \times 100$ for different age groups. Age refers to age of entry to foreign country. Number of utterances for the age group [11-20], [21-30], ..., [61-70] is 46, 342, 535, 239, 100, 12, respectively.]

Level of Education

According to Gardner’s socio-educational model (Gardner, 2010), intrinsic motivation to learn a second language is strongly correlated to educational
achievements. The objective was to find out how speakers’ level of education and their detected foreign accent might be related. To analyse the effect of education, we categorized the detection scores into different levels of education groups. We hypothesized that people with higher level of education speak the second language more fluently than lower educated people. As a consequence, mother tongue detection for higher educated people is relatively difficult. But the results in Figure 6 in fact show the opposite; the highest $C_{avg}$ belongs to elementary school and the lowest to university education. However, $C_{avg}$ is somewhat similar for the high school, vocational school, and polytechnic level of education.

![Bar chart showing $C_{avg} \times 100$ for different level of education groups.](image)

Figure 6: $C_{avg} \times 100$ for different level of education groups.

**Where the Second Language is Spoken**

Finally, we were also interested to observe whether the place or situation, where the second language is spoken, affects foreign accent detection or not. To this end, we categorized the scores into four groups based on the level of social interaction: home, hobbies, study and work. We hypothesized that the places with more social interactions between people, the mother tongue traits will be less in the second spoken language, therefore making it more difficult to detect the mother tongue. Figure 7 shows the $C_{avg}$ for different places where the second language is spoken. The results indicate no considerable sensitivity to the situation where the second language is spoken.
5. Conclusion

In this work, we studied how the various i-vector extractor parameters, data set selections and the speaker’s language proficiency affects foreign accent detection accuracy. Regarding parameters, highest accuracy was achieved using UBMs with 512 Gaussians, i-vector dimensionality of 1000 and HLDA dimensionality of 180. These are similar to those reported in general speaker and language recognition literature, except for the higher-than-usual i-vector dimensionality of 1000.

Regarding data, we found that the choice of the UBM training data is the most critical part, followed by T-matrix and HLDA. This is understandable since the earlier system components affect the quality of the remaining steps. In all cases, the error rates increased unacceptably high for mismatched sets of hyper-parameter training. Thus, our answer to the question whether hyper-parameters could be reasonably trained from mismatched language and channel is negative. The practical implication of this is that the i-vector approach, even though producing reasonable accuracy, requires careful data selection for hyper-parameter training — and this is not always feasible.

Applying within-class covariance normalization followed by score normalization technique further increased the i-vector system performance by 6% relative improvements in terms of $C_{avg}$. We also showed that the i-vector system outperforms the conventional GMM-UBM system by 28% relative decrease in terms of $C_{avg}$.
In our view, the most interesting contribution of this work is the analysis of language aspects. The results, broken down by the accents, clearly suggested that certain languages which do not belong to the same sub-family as Finnish are easier to detect. Turkish was the easiest ($C_{DET}$ of 6.35) while for instance Estonian, a language similar to Finnish, yielded $C_{DET}$ of 7.52. The most difficult language was English with $C_{DET}$ of 8.00. In general, confusion matrix revealed that phonetically similar languages are more often confused.

Our analysis on affecting factors suggested that language proficiency and age of entry affect detection performance. Specifically, accents produced by fluent speakers of Finnish are more difficult to detect. Speaker group with the lowest language grade 5 yielded $C_{avg}$ of 4.75 while the group with grade 8 yielded $C_{avg}$ of 6.76. Analysis of the age of entry, in turn, indicated that mother tongue detection in older speakers is easier than younger speakers. The age group [61-70] years yielded $C_{avg}$ of 4.45 while the group with age interval [11-20] years old yielded $C_{avg}$ of 5.31.

After optimizing all the parameters, the overall EER$_{avg}$ and $C_{avg}$ were 12.60% and 6.85, respectively. These are roughly an order of magnitude higher compared to state-of-the-art text-independent speaker recognition with i-vectors. This reflects the general difficulty of the foreign accent detection task, leaving a lot of space for future work on new feature extraction and modeling strategies. While these values are unacceptably high for security applications, the observed correlation between language proficiency and recognition scores suggests potential applications for automatic spoken language proficiency grading.

6. Acknowledgements

We would like to thank Ari Maijanen from University of Jyväskylä for an immense help with the FSD corpus. This work was partly supported by Academy of Finland (projects 253000, 253120 and 283256) and Kone foundation.

References

Bahari, M. H., Saeidi, R., hamme, H. V., Leeuwen, D. V., 2013. Accent recognition using i-vector, Gaussian mean supervector and Gaussian posterior


