Odyssey 2010
The Speaker and Language Recognition Workshop
28 June – 1 July 2010, Brno, Czech Republic

Data selection and calibration issues in automatic language recognition – investigation with BUT-AGNITIO NIST LRE 2009 system

Zdeněk Jančík[1], *Oldřich Plchot*[1], *Niko Brümmer*[2], *Lukáš Burget*[1], *Ondřej Glembek*[1], *Valiantsina Hubeika*[1], *Martin Karafiát*[1], *Pavel Matějka*[1], *Tomáš Mikolov*[1], *Albert Strasheim*[2], *and Jan "Honza" Černocký*[1]

(1) Brno University of Technology, Speech@FIT, Czech Republic
(2) AGNITIO, South Africa

Abstract

This paper summarizes the BUT-AGNITIO system for NIST Language Recognition Evaluation 2009. The post-evaluation analysis aimed mainly at improving the quality of the data (fixing language label problems and detecting overlapping speakers in the training and development sets) and investigation of different compositions of the development set. The paper further investigates into JFA-based acoustic system and reports results for new SVM-PCA systems going beyond BUT-Agnitio original NIST LRE 2009 submission. All results are presented on evaluation data from NIST LRE 2009 task.

1. Introduction

The goal of this paper is to present a consolidated version of BUT-Agnitio system description for NIST LRE 2009. BUT-Agnitio system included 7 acoustic and phonotactic subsystems, and elaborate calibration and fusion, however, its results were not optimal. The post-evaluation experiments addressed mainly the issues of the data, of which the detection and deletion of segments from speakers overlapping between the training and development was found to be the most important.

We have also investigated into the composition of development data set, as this issue was widely discussed at the evaluation workshop.

Finally, this paper deals with approaches that we have found the most promising for language recognition: joint factor analysis (JFA) for the acoustic part, and SVM with dimensionality reduction for the phonotactics.

The paper is organized as follows: section 2 presents the data used in training and development sets. Section 3 defines the basic scheme of our system while section 4 describes the calibration and fusion. Section 5 gives an overview of the front-ends (or sub-systems) and defines their common parts, and 6 deals with individual front-ends as they come into the fusion. Section 7 presents the experimental results in a structured way, and section 8 concludes the paper.

This work was partly supported by US Air Force European Office of Aerospace Research & Development (EOARD) Grant No. 083066, European project MOBIO (FP7-214324), Grant Agency of Czech Republic project No. 102/08/0707, and Czech Ministry of Education project No. MSM0021630528. We are grateful to Pietro Laface from Politecnico di Torino for allowing us to use the LPT data set

2. Training and development data

The following data (distributed by LDC and ELRA) were used to train our systems:

	CallFriend
*	Fisher English Part 1. and 2.
*	Fisher Levantine Arabic
*	HKUST Mandarin
	Mixer (data from NIST SRE 2004, 2005, 2006, 2008)
	development data for NIST LRE 2007
*	OGI-multilingual
*	OGI 22 languages
*	Foreign Accented English
	SpeechDat-East
*	SwitchBoard
	Voice of America radio broadcast

The VOA data needs further explanation, as there are two parts of this data. Note that only telephone conversations extracted from this data were used in the evaluation.

- VOA2 is raw radio data that originally had no language labels. Before NIST LRE 2009, BUT performed automatic labeling of this data by its production phonotactic system based on Hungarian phone recognizer [3][1] and shipped it to NIST and LDC.
- VOA3 was officially made available by NIST – the labels are given by the sources, some of this data was audited.

2.1. Original training and development data

Our data was split into two independent subsets, which we denoted TRAIN and DEV. The TRAIN subset had 54 languages (including the 23 target languages of NIST LRE2009) and had about 80 000 segments (2500 hours) in total. The DEV subset had 57 languages (including the 23 targets) and a total of about 60 000 segments (195 hours). The DEV subset was split into balanced subsets having nominal durations of 3s, 10s and 30s. The DEV set was based on segments from previous evaluations plus additional segments extracted from longer files from CTS (corpora marked with a star '*' in the table above), VOA3 and human-audited VOA2 data, which were not contained in the TRAIN set.

[1] The system is available through Phonexia, `http://phonexia.com/download/demo-lid`

2.2. LPT data

In the post-evaluation analysis, we have investigated the influence of having additional data for calibrating the system. As Loquendo/Politecnico di Torino (LPT) had excellent results in the evaluation, we experimented with their development set, that was generously provided to us by Prof. Pietro Laface. LPT VOA data contains segments from three different sources: VOA2, VOA3 and FTP[2], see details in [2]. The language labels for each speech segment were the ones provided by NIST or automatically generated by LPT group (with possibility of errors), followed by some auditing. The LPT development set had 34 languages (including the 23 targets) and a total of about 10 000 segments (51 hours).

2.3. Additional VOA2 data

VOA2 data was used in experiments on influence of training and development data (section 7.3). In addition to the original labels provided by us to NIST (obtained by the phonotactic system [3]), this data was also labeled by our JFA-G2048 system (see section 6.1). Only the segments, where the top-level hypothesis from both systems agreed, were selected. In this way, we obtained about 680 hours of speech in about 4800 segments.

3. General System description

In this section we describe the general system architecture that is common to all systems. Each system has three main stages:

Front-ends, of which there may be multiple different ones for a complete system. Each front-end stage maps the input speech segment to a *score-vector*. We denote these front-end outputs as *amorphous scores*. The dimensionality of these scores vary between 23 and 68, as described in more detail later.

Back-end, which performs fusion and calibration. The back-end fuses the amorphous scores from the front-ends and outputs *calibrated scores*. These scores function as multi-class log-likelihoods. In the Closed-set case, there are 23 log-likelihoods per input segment, for each of the 23 target languages. In the Open-set case, there are 24: the 23 target log-likelihoods as well as the log-likelihood for the hypothesis that the input is from some other language. The back-end is further described in the next section.

Decision stage, which takes the (i) back-end output log-likelihoods and (ii) the priors as defined for each trial. These are used in Bayes' rule to obtain the posterior distribution over the language classes. The posterior is then used to make minimum-expected-cost Bayes decisions. For closed-set the prior allows 23 hypotheses, and for open-set 24 hypotheses[3]. For each input segment, there are multiple detection trials, where the prior is varied between trials, as specified in the evaluation plan.

4. Back-end

The back-end maps one or more amorphous input score-vectors to a calibrated output score-vector, for every input segment. There are two back-end variants, for closed-set and open-set respectively. Both variants are composed of separate Gaussian back-ends (GBE's) for different front-ends, followed by a single discriminative fusion and calibration stage:

4.1. Gaussian Back-end (GBE)

The GBE models the amorphous scores with a different Gaussian model in amorphous score-space, for each language class. In the closed-set case, all the class models share the same common within-class covariance (CWCC). In the open-set case, the 23 target languages share the CWCC, but the out-of-set class has a larger covariance. In all cases there are different class-conditional means.

For the closed-set case, we use maximum likelihood (ML) estimates for the parameters. The CWCC was estimated over all 57 languages, while we used the means only for the 23 target languages.

In the open-set case, we take the out-of-set covariance as CWCC+BCC, where BCC is the between-class covariance, estimated from the means of all 57 languages in DEV, so BCC was estimated from 57 data points. The mean for this model was chosen as the mean of the 57 language means.

The output scores of the GBE are the 23 or 24 log-likelihoods of the input score-vector, given each of the class models.

4.2. Fusion and calibration

In contrast to our previous work, where we used three separate back-ends for nominal durations of 3s, 10s and 30s, we built a single duration-compensated fusion and calibration stage for NIST LRE 2009.

Let there be M input systems, where system i produces amorphous score-vector $\mathbf{s}_{it}$ for a given input t. Each system also outputs, as ancillary information, an indication of the duration of the input segment, denoted d_{it}. For acoustic systems, this was the number of 10ms speech frames found by the VAD (voice-activity-detection). For phonotactic systems, this was the expected number of phones in the segment. Let $B()$ denote the mapping effected by the GBE, then the output of the fusion and calibration is:

$$\vec{\ell}_t = \sum_{i=1}^{M} a_{1i} B(\mathbf{s}_{it}) + a_{2i} B(d_{it}^{-0.5} \mathbf{s}_{it}) + a_{3i} B(d_{it}^{-1} \mathbf{s}_{it}) + \mathbf{b} + \mathbf{C}\vec{\gamma}_t \quad (1)$$

where a_{ji} are scalar fusion weights, $\mathbf{b}$ is an offset vector, $\mathbf{C}$ is a matrix and $\vec{\gamma}_t$ is a vector of ancillary data. For systems which fused both acoustic and phonotactic subsystems, we composed $\vec{\gamma}$ of the phone and frame durations, as well as their square roots. In cases where we fused more than one phonotactic system, we used the expected number of phones for each system.

Notice that for each system, we fused in three differently normalized score variants and for each of these variants, a different GBE was trained.

The fusion parameters $(a_{ji}, \mathbf{b}, \mathbf{C})$ were discriminatively trained using multi-class logistic regression. This tends to produce well-calibrated class log-likelihoods. We verified this fact by judging calibration on independent data (see jackknifing below), by comparing C_{avg} (as defined in NIST evaluation plan [1]) and C^*_{avg} [4]. Note that C^*_{avg} was used only in the develop-

[2] Data downloaded from VOA Internet archive.

[3] Note, that only the results for the closed set are reported in this paper.

[4] When we are busy with basic recognizer development (i.e. the front-ends), we want to judge the *discrimination* rather than the calibration of our algorithms. In this case, we prefer not to use the calibration-sensitive C_{avg} as is. Our solution is to discount the effect of calibration

ment, all results are reported on the evaluation data using the conventional C_{avg}.

4.3. Jackknifing

We used our TRAIN data subset to train all front-ends, while we used our DEV data to train all the back-end stages and also to test the performance. To keep back-end training and test separate, we resorted to a jackknifing scheme. We did 5 outer iterations, where in each, we randomly partitioned the DEV data into 5 subsets balanced across all 57 languages. In 5 inner iterations, one subset was held out as test data, while the other 4 were used for back-end training.

We computed C_{avg} and C^*_{avg} on each of the 25 test sets and averaged. We also averaged the 25 back-ends thus obtained for a final back-end which was applied to the LRE'09 evaluation data.

5. Front-end types

There are two types of front-end, *acoustic* and *phonotactic*. Here, we give general descriptions of both types, followed by details of each front-end.

5.1. Acoustic

The acoustic systems are based on MFCC/SDC [6] acoustic features. This paper contains only a brief summary of acoustic feature extraction and UBM training. For more detail, see our previous work [4, 5].

The inputs to the language recognizer are segments of recorded speech of varying duration. The voice activity detection (VAD) is performed by our Hungarian phoneme recognizer [11][5] – we simply drop all frames that are labeled as silence or speaker noises.

All acoustic systems used the popular shifted-delta-cepstra (SDC) [6] feature extraction. The feature extraction is similar to BUT LRE 2005 system [5]. Every speech segment is mapped to a variable-length sequence of feature vectors as follows: After discarding silence portions, every 10ms speech-frame is mapped to a 56-dimensional feature vector. The feature vector is the concatenation of an SDC-7-1-3-7 vector and 7 MFCC coefficients (including C0). Cepstral mean and variance normalization are applied before SDC.

Vocal-tract length normalization (VTLN) performs simple speaker adaptation. We used an efficient VTLN warping-factor estimator based on GMM [7].

A 2048-component, language-independent, maximum-likelihood GMM was trained with the EM-algorithm on the pooled acoustic feature vectors of all 54 languages in the TRAIN data-set. We follow speaker recognition terminology and refer to this language-independent GMM as the *universal background model*, or UBM [8].

5.2. Phonotactic

The phonotactic systems were based on 3 phoneme recognizers: two ANN/HMM hybrids and one based on GMM/HMM context dependent models. All the recognizers are able to produce phoneme strings as well as phoneme lattices. In case of lattices, expected phone counts ("soft-counts") were used in the following processing [9].

5.2.1. Hybrid phoneme recognizers

The phoneme recognizer is based on hybrid ANN/HMM approach, where artificial neural networks (ANN) are used to estimate posterior probabilities of being in given phone-state for given frame. The input to the neural network is a block of Mel filter-bank log energies, with a context of 310ms around the current frame. Frame phone-state posteriors are then used as emission likelihoods in an HMM-based Viterbi decoder producing phone strings or lattices.

Hybrid recognizers were trained for Hungarian and Russian from the SpeechDat-E databases[6]. For more details see [11, 10].

5.2.2. GMM/HMM phoneme recognizers

The third phoneme recognizer was based on GMM/HMM context-dependent state-clustered triphone models, which are trained in similar way as the models used in AMI/AMIDA LVCSR [12]. The models were trained using 2000 hours of English telephone conversational speech data from Fisher, Switchboard and CallHome databases. The features are 13 PLP coefficients augmented with their first, second and third derivatives projected into 39 dimensional space using HLDA transformation. The models are trained discriminatively using MPE criterion [13]. VTLN and MLLR adaptation is used for both training and recognition in SAT fashion. The triphones were used for phoneme recognition with a bi-gram phonotactic model trained on English-only data.

6. Front-end descriptions

This section lists the details of all the different front-end variants.

6.1. JFA-G2048

This is an acoustic system inspired by Joint Factor analysis as introduced to speaker recognition by Patrick Kenny [14, 15]. Unlike "full" JFA, where both inter-session and speaker variabilities are modeled with sub-spaces, we use a simplified version, with only one sub-space representing the inter-session variability. For segment s, the super-vector of GMM means for language $l(s)$ is expressed by:

$$\mathbf{m}_s = \mathbf{t}_{l(s)} + \mathbf{U}\mathbf{x}_s,$$

where $\mathbf{t}_l$ is the location vector of language l, $\mathbf{U}$ is a factor loading matrix with C factors in its columns, and $\mathbf{x}_s$ is a vector of C segment-dependent channel factors. Detailed description can be found in [16]. An important feature of this system is that all we need to estimate its parameters, or to score a segment is a set of sufficient statistics of fixed length.

The channel factor loading matrix $\mathbf{U}$ is trained via an EM algorithm over 500 sessions of each of the 23 target languages. The super-vector dimensionality is about 10^5 and the dimensionality of the channel subspace is 200. The language location vectors $\mathbf{t}_l$ were MAP-adapted with relevance-MAP adaptation from the UBM [8], using also 500 sessions of each of the 23 target languages.

by letting the *evaluator* calibrate every system. That is, the evaluator optimizes calibration on the target data and then reports the value of C_{avg} obtained with this calibration. We denote this measure by C^*_{avg}. MATLAB code to perform this optimization is freely available at `http://niko.brummer.googlepages.com/focalmulticlass`, see also [16].

[5] available from `http://speech.fit.vutbr.cz/en/software`.

[6] see `http://www.fee.vutbr.cz/SPEECHDAT-E`.

For this system, we generated 68 models, to produce 68 front-end scores. We used all of the 54 available languages and trained two separate models for those languages that have both telephone and radio speech.

Test utterance scoring is done by language-independent channel-compensation, followed by linear scoring against each language model [17].

6.2. JFA-G2048-RDLT

Region Dependent Linear Transforms (RDLT) [18] is a discriminatively trained feature extraction, which is a generalization of a technique known in speech recognition as fMPE [19]. In our system, 256 linear transformations (56x56 matrices) take one common 56-dimensional feature vector of SDC+MFCC as input. The outputs of the transformations are linearly combined to form a single 56-dimensional output feature vector. The mixing weights are given by posterior probabilities of 256 components of a GMM, which is trained on the same input features. The transformations are discriminatively trained in similar manner as described in [18, 19] to maximize the expected probability of a segment being correctly recognized by a set of language dependent GMMs, which are ML-trained on the RDLT output features. The average duration of training segments is about 1 second. After training the RDLT, the set of language dependent GMMs is discarded, and the RDLT features are used to generate statistics for the JFA system described in section 6.1.

6.3. RDLT - no channel compensation

This subsystem uses the same RDLT features as system JFA-G2048-RDLT described in the previous section. The difference is that there is no channel compensation – in other words, we use plain MAP-adapted language models trained on RDLT features. The scoring is done by linear scoring against each language model [17].

6.4. MMI-FeaCC-G2048

This subsystem uses GMM models with 2048 Gaussians per language, where mean and variance parameters are re-estimated using Maximum Mutual Information criterion - the same as for BUT LRE2005 [5]. The SDC features are first compensated using eigen-channel adaptation in feature domain [20, 21]. Starting from target language models with means MAP-adapted from UBM using the compensated features, mean and variance parameters are further re-estimated using MMI criterion [4].

6.5. EN-TREE-45-N4, HU-TREE-6133-N4, RU-TREE-50-N4

In all systems, binary decision tree language modeling was based on creating a single language independent tree (referenced as "UBM") and adapting its distributions to individual language training data, as described in Navratil's work [22, 23]. We used English, Hungarian, and Russian phone recognizers to generate lattice-based expected phone 4-gram counts.

6.6. SVM-HU-N3

In this subsystem, the trigram-lattice-counts from Hungarian phone recognizer were used as features for subsequent classification by SVMs, similar to MIT's work [24].

Table 2: *Fixing the data.*

Eval data, [C_{avg}]	30s	10s	3s
JFA-G2048-RDLT	3.62	6.39	16.47
- rename 'pers' segments as 'fars'	3.56	6.36	16.14
- speaker ID filtering	2.33	5.09	15.06

Table 3: *Amount of omitted data by speaker ID filtering*

language	omitted data
bosn	92.8 %
croa	77.9 %
port	17.6 %
russ	30.1 %
ukra	93.8 %

7. Experimental results

All results are presented as C_{avg} on NIST LRE 2009 evaluation data for closed-set condition. The results of all systems and their fusion are summarized in Table 1. We present results for 23 detectors (only models for target languages) and for more detectors (54 detectors - one for each language in TRAIN set or 68 detectors when VOA and CTS data have separate model).

7.1. Fixing problems with the data

The obtained results did not meet our expectations, as we saw big difference between results on the development (30s condition $C_{avg} = 0.66$) and evaluation sets ($C_{avg} = 2.34$). We wanted to investigate, what was wrong with the development set. The biggest difference was observed for JFA-G2048-RDLT, so that we focused our work on this system - see Table 2.

We found two main problems. The first bug was using two different labels: "Persian" and "Farsi" for segments from the same language in our DEV data. After re-labeling, we obtained small, but consistent improvement across all durations.

The second problem was more serious: for some languages with little amount of data, we found that the TRAIN and DEV sets contained large amount of speech from the same speakers. This negatively influenced training of the calibration and fusion parameters. This problem was addressed by training a speaker ID system for each training utterance and scoring all development utterances from the corresponding language. A GMM-UBM based speaker ID system developed by BUT for NIST 2006 SRE evaluation was used [26][7].

Based on the histogram of scores (example for Ukrainian in Figure 1) showing clearly bi-modal structure of identical and different speakers, we chose a language-dependent threshold of speaker ID score for omitting utterances from the development set. The amounts of omitted data are in Table 3. This step brings a nice improvement as we can see in Table 2.

Table 4 shows the results for all systems and fusion after fixing both above mentioned problems. It is obvious that the performance depends heavily on the quality of the data: compared to Table 1, we obtained significant and consistent improvement across all systems and all durations.

[7] This system is available through Phonexia `http://phonexia.com/download/demo-sid`.

Table 1: *Individual systems' and fusion results with the original development data. Outputs denoted '-' were not produced because of too high computational load.*

Eval data, $[C_{avg}]$	30s		10s		3s	
System/Detectors	23	54/68*	23	54/68*	23	54/68*
EN-TREE-45-N4	3.36	-	7.29	-	18.83	-
RU-TREE-50-N4	3.69	3.52	7.02	6.55	16.99	16.81
HU-TREE-6133-N4	4.29	4.05	8.17	8.05	19.14	19.09
SVM-HU-N3	3.82	-	9.28	-	21.54	-
JFA-G2048-RDLT	3.68	3.62*	6.55	6.39*	16.29	16.47*
MMI-FeaCC-G2048	4.47	-	6.46	-	14.92	-
RDLT – no channel comp.	7.07	5.57*	9.97	8.34*	18.66	17.29*
Fusion	2.29	2.34	3.77	3.86	10.29	10.19
Fusion - development data	0.77	0.66	1.70	1.53	6.51	6.14

Table 4: *Individual systems' and fusion results with the fixed development data.*

Eval data, $[C_{avg}]$	30s		10s		3s	
System/Detectors	23	54/68*	23	54/68*	23	54/68*
EN-TREE-45-N4	2.71	-	6.52	-	18.28	-
RU-TREE-50-N4	2.97	2.80	5.80	5.38	16.19	15.93
HU-TREE-6133-N4	3.71	3.79	7.14	7.16	18.30	18.30
SVM-HU-N3	3.08	-	8.50	-	20.93	-
JFA-G2048-RDLT	**2.33**	2.53*	5.09	5.26*	15.06	15.51*
MMI-FeaCC-G2048	2.99	-	**4.78**	-	**13.94**	-
RDLT - no channel comp.	4.92	4.06*	7.81	6.69*	17.23	16.24*
Fusion	**1.93**	2.23	**2.87**	3.01	9.30	**9.28**
Fusion - development data	0.85	0.72	1.87	1.55	6.19	5.62

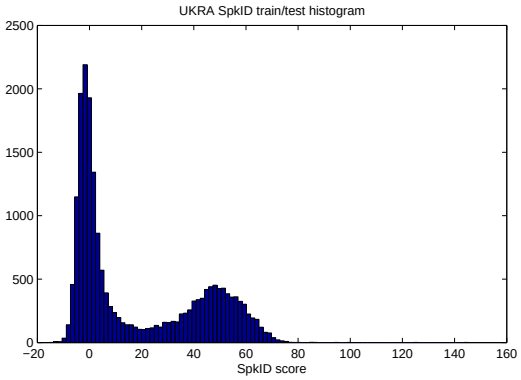

Figure 1: Histogram of speaker ID scores – example for Ukrainian.

7.2. RDLT out and JFA tuning

We have seen good performance of RDLT features on the development set. The comparison of JFA system with and without RDLT (Table 5) shows slight superiority of RDLT for short durations, but an unpleasant hit for the 30s condition. Therefore, RDLT features were dropped from our JFA system for the following experiments.

The next step was improving the JFA. Originally, we used only 500 segments per target language to MAP-adapt language location vectors $\mathbf{t}_l$ before training the factor loading matrix $\mathbf{U}$. Here, we used all available data to train $\mathbf{t}_l$'s, so that the resulting model is much more stable. We have also tuned the optimum number of EM-iterations of $\mathbf{U}$ matrix training. The performance increased for all three durations (last line of Table 5).

Table 5: *JFA system without RDLT features and tuning of the JFA.*

Eval data, $[C_{avg}]$	30s	10s	3s
JFA-G2048-RDLT (cleaned dev)	2.33	5.09	15.06
- drop RDLT features	2.18	5.17	15.16
- JFA model tuning	2.02	4.89	14.57

7.3. Composition of the development set

Excellent results reported by LPT and MIT were partly attributed to the work they invested in the creation and cleaning of their development data. We have therefore compared three datasets for training the calibration of single acoustic JFA-G2048 system:

- our cleaned data, based only on VOA3 and audited VOA2.
- LPT data containing segments from three different sources: VOA2, VOA3 and FTP (see section 2.2)
- additional non-audited VOA2 data (see section 2.3).

The results of three calibration experiments in table 6 indicate that the best performing system uses only the original calibration data. When adding additional VOA2 data to the training, we have seen marginal improvement only for the longest 30s duration.

Sub-optimal performance of LPT data can be probably attributed to repetition of utterances in our TRAIN data-set and

Table 6: *Analysis of LPT and additional VOA2 data.*

Eval data, $[C_{avg}]$	30s	10s	3s
JFA-G2048	2.02	**4.89**	**14.57**
calibration on LPT like data	2.32		
adding VOA2 data to calibration	2.54	5.62	15.03
adding VOA2 data to training	**1.94**	4.95	14.70

Table 7: *SVM-PCA and their fusion.*

Eval data, $[C_{avg}]$	30s	10s	3s
SVM-PCA	1.78	3.86	14.13
MMI-FeaCC-G20486	2.99	4.78	13.94
JFA-2048G	2.02	4.89	14.57
Fusion	**1.57**	**2.76**	**10.22**

in LPT development set (the speaker ID cleaning procedure was not applied here). The other source of worse performance can be quite small number of segments in LPT development set compared to ours (10 000 against 63 000). Adding VOA2 data to the calibration did not perform well, probably because of the quality of automatically generated labels.

7.4. Grand finale with SVM-PCA systems

Recently, excellent results were obtained with SVM-based systems with principal component analysis (PCA) dimensionality reduction [25]. The principle is very simple: vectors of expected n-gram counts are derived from lattices, and after compression by square root, their dimensionality if drastically reduced by PCA. The resulting features are then used for SVM scoring. Reducing the dimensionality allows for several orders of speed up, and possibility to train on entire training data.

The presented SVM-PCA system is a fusion of 13 different SVM systems, based on different level of n-gram language model (3 or 4), different phone recognizers (Hungarian, Russian or English) and with different feature dimensionality. The fusion was done exactly in the same way as described in section 4.2, i.e. by estimating the calibration and fusion parameters on jackknifed DEV set. Except for the shortest duration, the fusion of 13 different SVM systems is by far the best system we have (see Table 7).

When these 13 SVM-PCA systems are fused with the best performing acoustic systems: MMI-FeaCC-G2048 (see 6.4) and post-evaluation version of JFA-G2048, the results are very competitive (last line of Table 7).

8. Conclusions

The presented results show crucial importance of careful work with data in language recognition. Despite using 7 state-of-the-art LRE systems, the original BUT's results were suboptimal due to problems with the data. It seems, that in case of careful pruning of repeated speakers, we can quietly use only the standard VOA3 and human-audited VOA2 data distributed by NIST – we have not seen any significant advantage from using additional data.

Among the acoustic systems, JFA-based one has clearly superior performances, especially for longer durations. With RDLT features, we have seen improved performance on shorter segments (for which RDLT is actually trained), but deterioration for long segments. More work is needed to have them offering stable performance across wider range of durations.

Finally, the recently developed SVM-PCA approach seems to perform better than the acoustic ones, except for the shortest duration, and gives very competitive results when fused with only two acoustic systems. In the same time, this combination is not computationally hungry and could lead to development of highly accurate and fast practical LRE systems.

9. References

[1] "The 2009 NIST Language Recognition Evaluation Plan (LRE2009)", `http://www.itl.nist.gov/iad/mig/tests/lre/2009/LRE09_EvalPlan_v6.pdf`

[2] F. Castaldo, D. Colibro, S. Cumani, E. Dalmasso, P. Laface, and C. Vair, "Loquendo-Politecnico di Torino System Description for NIST 2009 Language Recognition Evaluation", in *Proc. NIST Language recognition workshop 2009*, Baltimore MD, USA

[3] O. Plchot, V. Hubeika, L. Burget, P. Schwarz, and P. Matějka, "Acquisition of Telephone Data from Radio Broadcasts with Applications to Language Recognition", *Proc. 11th International Conference on Text, Speech and Dialogue 2008*, Brno, CZ

[4] V. Hubeika, L. Burget, P. Matjka and P. Schwarz, "Discriminative Training and Channel Compensation for Acoustic Language Recognition", in *Proc. Interspeech 2008*.

[5] P. Matějka, L. Burget, P. Schwarz, and J. Černocký, "Brno University of Technology system for NIST 2005 Language recognition evaluation," in *Proc. IEEE Odyssey: The Speaker and Language Recognition Workshop*, San Juan, Puerto Rico, June 2006, pp. 57–64.

[6] P.A. Torres-Carrasquillo, E. Singer, M.A. Kohler, R.J. Greene, D.A. Reynolds, and J.R. Deller Jr., "Approaches to language identification using Gaussian mixture models and shifted delta cepstral features," in *Proc. International Conferences on Spoken Language Processing (ICSLP)*, Sept. 2002, pp. 89–92.

[7] L. Welling, S. Kanthak and H. Ney, "Improved methods for vocal tract normalization", in *Proc. ICASSP 1999*, Phoenix, March 1999.

[8] D.A. Reynolds, T.F. Quatieri, and R.B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1–3, pp. 19–41, 2000.

[9] J.L. Gauvain, A. Messaoudi and H. Schwenk, "Language Recognition using Phone Lattices," in *Proc. International Conferences on Spoken Language Processing (ICSLP)*, Sept. 2004, pp.1283–1286.

[10] P. Schwarz, P. Matějka, and J. Černocký, "Towards lower error rates in phoneme recognition," in *Proc. International Conference on Text, Speech and Dialogue*, Brno, Czech Republic, Sept. 2004, pp. 465–472.

[11] P. Schwarz, P. Matějka, and J. Černocký, "Hierarchical structures of neural networks for phoneme recognition," in *Proc. ICASSP*, Toulouse, France, May 2006, pp. 325-328.

[12] T. Hain, V. Wan, L. Burget, M. Karafiat, J. Dines, J. Vepa, G. Garau and M. Lincoln: "The AMI System for the Transcription of Speech in Meetings", in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2007)*, Hononulu, 2007, pp. 357-360.

[13] D. Povey, "Discriminative Training for Large Vocabulary Speech Recognition," Ph.D. thesis, Cambridge University, Jul. 2004.

[14] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition", IEEE Transactions on Audio, Speech and Language Processing 15 (4), pp. 1435-1447, May 2007.

[15] P. Kenny, N. Dehak, P. Ouellet, V. Gupta, and P. Dumouchel, "Development of the Primary CRIM System for the NIST 2008 Speaker Recognition Evaluation", in *Proc. Interspeech 2008*, Brisbane, Australia, Sept 2008.

[16] N. Brummer et al., "Discriminative Acoustic Language Recognition via Channel-Compensated GMM Statistics", in *Proc. Interspeech 2009*, Brighton, UK, Sept. 2009.

[17] O. Glembek, L. Burget, N. Dehak, N. Brummer and P. Kenny, "Comparison of Scoring Methods used in Speaker Recognition with Joint Factor Analysis" in *Proc. ICASSP 2009*, Taipei, Taiwan, April 2009.

[18] B. Zhang, S. Matsoukas, R. Schwartz: "Recent progress on the discriminative region-dependent transform for speech feature extraction", in *Proc. Interspeech*, Pittsburgh, PA, September, 2006

[19] D. Povey: "fMPE: discriminatively trained features for speech recognition," in *Proc. ICASSP*, Philadelphia, PA, Mar. 2005, IEEE.

[20] V. Hubeika, L. Burget, P. Matějka and J. Černocký: "Channel Compensation for Speaker Recognition", poster at MLMI 2007, Brno, June 2007.

[21] F. Castaldo, E. Dalmasso, P. Laface, D. Colibro and C. Vair: "Language identification using acoustic models and speaker compensated cepstral-time matrices", in *Proc. ICASSP 2007*, Hononulu, April 2007.

[22] J. Navratil: "Spoken language recognition - a step toward multilinguality in speech processing", in IEEE Trans. on Speech and Audio Processing, Vol. 9, No. 6, pp. 678-685 ISSN: 1063-6676, September 2001.

[23] J. Navratil: "Recent advances in phonotactic language recognition using binary-decision trees," in *Proc. International Conferences on Spoken Language Processing (ICSLP)*, Pittsburgh, PA, October 2006

[24] W.M. Campbell, F. Richardson, and D.A. Reynolds: "Language Recognition with Word Lattices and Support Vector Machines", in *Proc. ICASSP 2007*, Hononulu, April 2007.

[25] T. Mikolov, O. Plchot, O. Glembek, P. Matějka, L. Burget and J. Černocký: "PCA-Based Feature Extraction for Phonotactic Language Recognition", accepted to *Odyssey 2010*, Brno, CZ, July 2010.

[26] P. Matejka, L. Burget, P. Schwarz, O. Glembek, M. Karafiat, F. Grezl, J. Cernocky, D. van Leeuwen, N. Brummer, and A. Strasheim: "STBU system for the NIST 2006 speaker recognition evaluation", in *Proc. ICASSP 2007*, Honolulu, USA, 2007, pp. 221–224.