Brno University of Technology System for NIST 2005 Language Recognition Evaluation

Pavel Matějka, Lukáš Burget, Petr Schwarz and Jan Černocký

Speech@FIT group, Faculty of Information Technology,
Brno University of Technology, Czech Republic
{matejkap,burget,schwarzp,cernocky}@fit.vutbr.cz

Abstract

This paper presents the language identification (LID) system developed in Speech@FIT group at Brno University of Technology (BUT) for NIST 2005 Language Recognition Evaluation. The system consists of two parts: phonotactic and acoustic. Phonotactic system is based on hybrid phoneme recognizers trained on SpeechDat-E database. Phoneme lattices are used to train and test phonotactic language models. Further improvement is obtained by using anti-models. Acoustic system is based on GMM modeling trained under Maximum Mutual Information framework. We describe both parts and provide a discussion of performance on LRE 2005 recognition task.

1. Introduction

National Institute of Standard and Technology (NIST) regularly organizes language recognition evaluations (LRE), the goals of which are to establish a current baseline of performance for language and dialect recognition of conversational telephone speech using uniform evaluation procedure. The task is the detection of a given target language or dialect. Given a test segment of speech, the system must determine whether or not the speech is from the target language or dialect.

In NIST LRE 1996, approaches using phonotactic information, namely PRLM (phoneme recognizer followed by a language model) and PPRLM (parallel combination of several PRLM), appeared to be the most successful. The following evaluation in 2003 demonstrated that Gaussian Mixture Models (GMM) used to classify acoustic characteristics of speech performed the same or better than phonotactic approach. Singer et al. presented a comparative study dealing with these two approaches in [1]. The phonotactic approach was further improved by Gauvain et al. [2] by recognizing and modeling phoneme lattices instead simple phoneme strings. The importance of high quality phoneme recognition was demonstrated in our previous work [3].

This paper presents Brno University of Technology (BUT) system that participated in NIST LRE 2005. The system combines both phonotactic and acoustic approaches. Two major contribution that made our system successful in the evaluation is use of discriminative techniques to train both acoustic GMM system [14] and language model in phonotactic system [15].

The following section 2 presents NIST LRE 2005 target data, evaluation metrics, and our training data. The acoustic system is described in Section 3. Section 4 presents the phonotactic system. Section 5 describes fusion of scores. Section 6 compares the performance of different systems on LRE 2005 and Section 7 concludes with a brief discussion and summary of results.

2. Data and evaluation

Experiments reported in this paper were performed in the framework of NIST LRE 2005[1]. The test segments contain three nominal durations: 3 seconds, 10 seconds, and 30 seconds from a set of 7 languages and two dialects (English-American, English-Indian, Hindi, Japanese, Korea, Mandarin-Mainland, Mandarin-Taiwan, Spanish and Tamil). Actual speech durations vary but were constrained to be within the ranges of 2-4 seconds, 7-13 seconds, and 25-35 seconds of actual speech contained in segments, respectively. The performance is evaluated separately for test segments of each duration and is done according to NIST [4] per-language, considering each system as a language *detector* rather than recognizer. A standard detection error trade-off (DET) curve is evaluated as a plot of probability of false alarms against the probability of misses with the detection threshold as parameter and equal priors for target and non-target languages. Equal error rate (EER) is the point where these probabilities are equal. The minimum DET is the point on DET curve where the sum of false alarm and miss probabilities is minimal. Fig. 2 presents the DET curve and scalar metrics in graphical form.

The following training data were used: **phoneme recognizers** were trained on Hungarian, Russian and Czech SpeechDat-East [16] which performed the best in our previous work [3]. Only phonetically balanced items were

[1] http://www.nist.gov/speech/tests/lang/2005

used for the training of phoneme recognizers.

Phonotactic language models and **acoustic models** were trained on the CallFriend [17] containing telephone speech of 15 different languages or dialects (each contains 20 complete half-hour conversations) and OGI multilingual and OGI 22 languages corpora [18]. Only seven target languages were used for building models.

Development Data comes from NIST 1996 and 2003 LRE plus 40 additional segments from OGI Foreign Accented English database (Hindi part) to compensate for the lack of Indian accented English.

3. Acoustic system

Gaussian mixture models are used to represent distributions of cepstral features of individual languages. Parameters of the models are trained discriminatively using Maximum Mutual Information estimation (MMI). In comparison to conventionally used maximum likelihood (ML) training, MMI allows to model languages using much less parameters (256 mixture components in our system) and at the same time it provides significant improvement in recognition accuracy. In our previous work [14], we compared ML and MMI trained models on NIST LRE2003 evaluation set: for 30 sec. segments, MMI trained GMMs with 128 mixture components provided more that 50% relative improvement over state-of-the-art ML trained GMM with 2048 components. Similar trend can be seen also in the results presented in this paper, which were obtained on NIST LRE2005 evaluation data.

3.1. Features

The most widely used features for LID (as well as for other speech processing techniques) are Mel-Frequency Cepstral Coefficients (MFCC). The works of Torres-Carasquillo [5] and others have however shown the importance of broader temporal information for LID. The shifted delta cepstra (SDC) features are created by stacking delta-cepstra computed across multiple speech frames. The SDC features are specified by a set of 4 parameters: N, d, P and k, where N is the number of cepstral coefficients, d is the advance and delay for the delta-computation, k is the number of blocks whose delta-coefficients are concatenated to form the final feature vector, and P is the time shift between consecutive blocks. In case we denote the original features $o_h(t)$[2], shifted deltas are defined:

$$\Delta o_h(t) = o_h(t + iP + d) - o_h(t + iP - d)$$

for $i = 0, P, 2P, \dots, (k-1)P$.

The features in our system are 7 MFCC coefficients (including coefficient C0) concatenated with SDC 7-1-3-7, which totals in 56 coefficients per frame. RASTA filtering of cepstral trajectories is used to alleviate channel mismatch [6] and Vocal-tract length normalization (VTLN) [7] performs simple speaker adaptation. VTLN warping factors are estimated using single GMM (512 Gaussians), ML-trained on the whole CallFriend database (using all the languages). The model was trained in standard speaker adaptive training (SAT) fashion in four iterations of alternately re-estimating the model parameters and the warping factors for the training data.

The effect of the individual techniques (such as SDC, VTLN and RASTA) on language detection performance is analyzed in section 6.

3.2. Model training

One GMM with 256 mixture components was created for each of 7 target languages (English, Hindi, Japanese, Korean, Mandarin, Spanish and Tamil). Models were trained using data of target languages from Call Friend, OGI Multilingual, OGI22 and OGI Foreign Accented English[3]. We have however seen that the OGI databases did not provide much benefit, mainly because of their smaller size compared to CallFriend.

Initial set of models was trained under conventional ML framework. These models served only as a starting point and were further discriminatively re-trained using Maximum Mutual Information estimation in about 20 iterations of MMI training.

Unlike in the case of ML training, which aims to maximize the overall likelihood of training data given the transcriptions, MMI objective function to maximize is the posterior probability of correctly recognizing all training segments (utterances):

$$\mathcal{F}_{MMI}(\lambda) = \sum_{r=1}^{R} \log \frac{p_\lambda(\mathcal{O}_r|s_r)^{K_r} P(s_r)}{\sum_{\forall s} p_\lambda(\mathcal{O}_r|s)^{K_r} P(s)}. \qquad (1)$$

where $p_\lambda(\mathcal{O}_r|s_r)$ is likelihood of r-th training segment, $\mathcal{O}_r$, given the correct transcription (in our case the correct language identity) of the segment, s_r, and model parameters, λ. R is the number of training segments and the denominator represents the overall probability density, $p_\lambda(\mathcal{O}_r)$ (likelihood given any language). We consider the prior probabilities of all classes (languages) equal and drop the prior terms $P(s_r)$ and $P(s)$. Usually, segment likelihood $p_\lambda(\mathcal{O}_r|s)$ is computed as simple multiplication of frame likelihoods incorrectly assuming statistical independence of feature vectors. Factor $0 < K_r < 1$, which is increasing the confusion between hypothesis represented by numerator and denominator, can be considered as a compensation for underestimating segment likelihoods caused by this incorrect

[2] $o_h(t)$ denotes the h-th element of feature vector $\mathbf{o}(t)$

[3] From OGI Foreign Accented English database, only Hindi and Tamil English data were used as representatives of Indian accented English.

assumption. In our experiments, this factor was empirically determined as $K_r = C/T_r$, where C is a constant dependent on type of features (6 in our case) and T_r is number of frames in r-th segment.

It can be shown [9] that MMI objective function (1) can be increased by re-estimating model parameters using extended Baum-Welch algorithm with the following formulae for updating mean and variances:

$$\hat{\boldsymbol{\mu}}_{sm} = \frac{\theta^{num}_{sm}(\mathcal{O}) - \theta^{den}_{sm}(\mathcal{O}) + D_j \boldsymbol{\mu}'_{sm}}{\gamma^{num}_{sm} - \gamma^{den}_{sm} + D_{sm}} \tag{2}$$

$$\hat{\boldsymbol{\sigma}}^2_{sm} = \frac{\theta^{num}_{sm}(\mathcal{O}^2) - \theta^{den}_{sm}(\mathcal{O}^2) + D_{sm}(\boldsymbol{\sigma}'^2_{sm} + \boldsymbol{\mu}'^2_{sm})}{\gamma^{num}_{sm} - \gamma^{den}_{sm} + D_{sm}} - \hat{\boldsymbol{\mu}}^2_{sm}$$

where s and m are identities of model (language) and its mixture component, $\boldsymbol{\mu}'_{sm}$ and $\boldsymbol{\sigma}'^2_{sm}$ are old means and variances and D_{sm} is smoothing constant controlling speed of convergence, which is set to be greater than (i) twice the value ensuring all variances to be positive (ii) $E\gamma^{den}_{sm}$ where E is another constant (we use $E = 2$). The terms:

$$\begin{aligned}\theta^{num}_{sm}(\mathcal{O}) &= \sum_{r=1}^{R}\sum_{t=1}^{T_r} \gamma^{num}_{smr}(t)\mathbf{o}_r(t) \qquad (3)\\ \theta^{num}_{sm}(\mathcal{O}^2) &= \sum_{r=1}^{R}\sum_{t=1}^{T_r} \gamma^{num}_{smr}(t)\mathbf{o}_r(t)^2\\ \gamma^{num}_{smr} &= \sum_{r=1}^{R}\sum_{t=1}^{T_r} \gamma^{num}_{smr}(t)\end{aligned}$$

are mixture component specific first and second order statistics and occupation counts corresponding to numerator of the objective function (1). Denominator statistics can be expressed by similar equations, where all superscripts num are merely replaced by den. Note that the numerator statistic are ordinary ML statistics. Therefore the numerator posterior probability of occupying mixture component sm by t-th frame of training segment r,

$$\gamma^{num}_{smr}(t) = \begin{cases} \gamma_{smr}(t) & \text{for } s = s_r \\ 0 & \text{otherwise} \end{cases}, \tag{4}$$

is nonzero only for mixture components corresponding to correct class (language). To estimate the posterior probabilities for denominator:

$$\gamma^{den}_{smr}(t) = \gamma_{smr}(t)\frac{p_{\acute{\lambda}}(\mathcal{O}_r|s)^{K_r}}{\sum_{\forall q} p_{\acute{\lambda}}(\mathcal{O}_r|q)^{K_r}}, \tag{5}$$

likelihoods $p_{\acute{\lambda}}(\mathcal{O}_r|q)$ are evaluated using old parameters $\acute{\lambda} = \{\boldsymbol{\mu}', \boldsymbol{\sigma}'^2\}$. The factor K_r is discussed above. Finally,

$$\gamma_{smr}(t) = W_s \frac{c_{sm}\mathcal{N}(\mathbf{o}_r(t); \boldsymbol{\mu}'_{sm}, \boldsymbol{\sigma}'^2_{sm})}{\sum_{j=1}^{J_s} c_{sj}\mathcal{N}(\mathbf{o}_r(t); \boldsymbol{\mu}'_{sj}, \boldsymbol{\sigma}'^2_{sj})} \tag{6}$$

where c_{sm} is mixture component weight, $\mathcal{N}(\cdot\,; \boldsymbol{\mu}'_{sm}, \boldsymbol{\sigma}'^2_{sm})$ is evaluated for old mean and variance estimates and J_s is number of mixture components in model s.

The drawback of MMI is that it also learns the (undesirable) language priors from training data. We equalize the amounts of training data per language but rather than throwing out training data, we appropriately weigh statistics in MMI re-estimation formulae using factor W_s, which is set to be inversely proportional to amount of training data available for language s.

The derivation of parameter update formulae is described in detail for example in [9]. Formulae for discriminative update of mixture component weights can be also derived [9], however, we train these weights only in ML iterations and keep them fixed in MMI iterations as their discriminative update is not expected to bring any significant improvement.

The advantage of MMI is that it concentrates on precise modeling of decision boundary and does not waste the parameters on highly overlapped parts of feature distributions with low discriminative power.

Another advantage is that MMI optimizes parameters for good recognition of whole segments (not individual frames as in the case of ML) and therefore it takes into account speech segmentation. We used segmentation generated by our phoneme recognizer (see section 4.1), where all phonemes are linked to ‘speech’ class and pause and speaker noises (breath, hesitation, etc.) to ‘silence’ class. The silence segments are not used for training. Each speech segment generated by phoneme recognizer is taken as an individual segment for MMI training and only segments longer than half second are used for training (about 4/5 of available speech). Compared to the segments used in testing, the segments generated by phoneme recognizer are rather short (usually between 1 and 2 seconds). The influence of length of training segments on the quality of MMI training is still to be investigated.

3.3. Scoring

As the test score, which should represent the confidence that test segment, $\mathcal{O}$, is the target language, s, we use the log posterior probability:

$$\log P(s|\mathcal{O}) = \log \frac{p_{\lambda}(\mathcal{O}|s)^K}{\sum_{\forall q} p_{\lambda}(\mathcal{O}|q)^K}, \tag{7}$$

Here, we set the factor K to the inverse value of number of speech frames in the test segment. Note that non-speech frames (detected again by our phoneme recognizer) are not considered in evaluation of all likelihoods in equation (7)

4. Phonotactic system

The phonotactic system uses three phoneme recognizers and lattice based training and scoring. To further improve this system we use language anti-models to correct mistakes of the target language model.

4.1. Phoneme recognizer

Our phoneme recognizer (Fig. 1) is a hybrid system based on Neural Networks (NN). The feature extraction makes use of long temporal context. First, Mel filter bank energies are obtained in conventional way. After sentence mean normalization in each band, temporal evolutions of critical band spectral densities are taken around each frame. Based on our previous work in phoneme recognition [10, 11], the context of 31 frames (310 ms) around the current frame was selected. This context is split into 2 halves: Left and Right Contexts. This allows for more precise modeling of the whole trajectory while limiting the size of the model (number of weights in the NN) and reducing the amount of necessary training data. Both parts are processed by discrete cosine transform to decorrelate and reduce dimensionality. Two NNs are trained to produce phoneme-state posterior probabilities for both context parts. Third NN functions as a merger and produces final set of phoneme-state posterior probabilities. All neural networks[4] have 1500 neurons in hidden layer.

For each frame, outputs of the merger are converted to estimates of HMM state output probabilities and used in the following Viterbi algorithm to produce phoneme lattices (or strings).

Each phonotactic system has its own segmentation derived from recognized phoneme strings. All silence classes, speaker and intermittent noises are merged together to form one silence class. If the silence is longer than 5 sec., the system flushes the previous segment and starts a new one.

4.2. N-gram language modeling

Smoothed trigram back-off language model was used to capture phonotactic statistics of each language. Training speech of all target languages was processed by phoneme recognizer and trigrams were counted for each language separately. Phoneme insertion penalty (PIP) in the decoder was tuned on our development set with the best LID performance as criterion. We used standard Witten-Bell discounting [12] implemented in SRI LM toolkit[5] [13].

Since phoneme recognizer is not 100% accurate on 1 best decision, it is advantageous to use richer structure at the end of decoder: lattices instead of strings. Lattices were generated with segmentation described above and without any language model. At first, acoustic likelihoods contained in the lattices are converted to phoneme posteriors. Then, the LM is computed from N-gram estimates weighted by these posteriors, similarly to Gauvain et al. [2].

4.3. Anti-models

Anti-models are inspired by boosting training techniques and discriminative-like training. Anti-model is a language model modeling the space where target model makes mistakes [15]. Its training works in the following way: we will denote all utterances belonging to language s as set S_s^+ and all utterances not belonging to language s as set S_s^-. First, the training of phonotactic model LM_s^+ of each language s is done in standard way using only set S_s^+. Then, all *training* utterances are scored by all phonotactic models and posteriors of utterances are derived:

$$P(\mathcal{O}_r|s) = \frac{p(\mathcal{O}_r|LM_s^+)^K}{\sum_{\forall q} p(\mathcal{O}_r|LM_q^+)^K}, \tag{8}$$

where $\mathcal{O}_r$ is the r-th training utterance[6], $p(\mathcal{O}_r|LM_s^+)$ is the likelihood provided by phonotactic model LM_s^+ and $K_r = 100/T_r$ is empirically determined scaling factor, where T_r is the expected number of phonemes in r-th speech segment computed as sum of posterior probabilities of links in the corresponding phoneme lattice.

For language s, the parameters of anti-model LM_s^- should be trained on all segments from S_s^- mis-recognized as s. However, we can use *all* utterances $\mathcal{O}_r \in S_s^-$ and weigh their trigram counts by posteriors $P(\mathcal{O}_r|s)$. Obviously, an utterance from S_s^- with high probability to be mis-recognized as s will contribute more to the anti-model than an utterance correctly recognized as language q where $q \neq s$.

Final score of test utterance $\mathcal{O}$ is then obtained by subtracting the weighted log-likelihood provided by the anti-model from log-likelihood provided by the target model:

$$\log \mathcal{S}(\mathcal{O}|s) = \log p(\mathcal{O}|LM_s^+) - k\ \log p(\mathcal{O}|LM_s^-), \tag{9}$$

where the constant k needs to be set experimentally. While tuning it, we obtained a stable peak at $k = 0.3$ for all three durations and with all 3 phoneme recognizers.

4.4. Scoring

Scores for test segments are computed from phoneme lattices. Triphone expanded phoneme lattices are generated without any language model. Partial scores are given by trigram probabilities corresponding to triphone links weighted by their respective posteriors. The total score of segment is then computed as a sum of partial scores.

[4] All nets are trained using QuickNet from ICSI http://www.icsi.berkeley.edu/Speech/qn.html

[5] http://www.speech.sri.com/projects/srilm/

[6] Note, that in section 3, $\mathcal{O}$ denoted features of an utterance. Here it stands for its phoneme lattice representation.

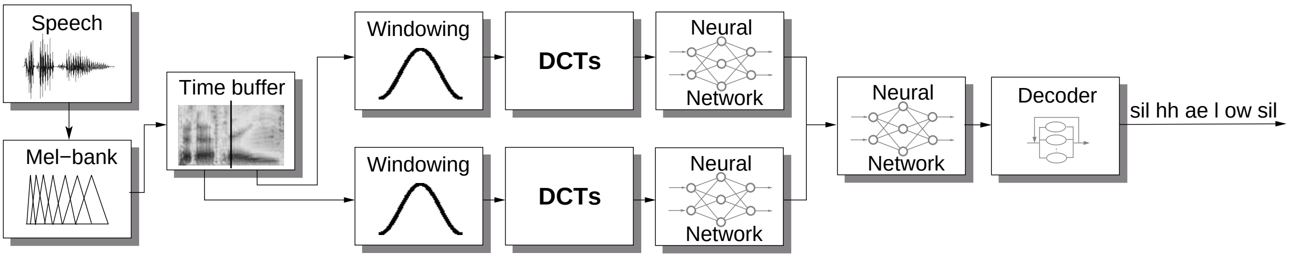

Figure 1: Phoneme recognizer with split temporal context and three neural nets.

To compute "evaluation" scores for unseen trigrams, we backed-off to bigrams or unigrams according to Witten-Bell discounting (this applies to both LM_s^+ and LM_s^-). Although backing-off worked well in our earlier experiments, it turned out to cause some serious problems and degradation in performance on LRE 2005 data. The explanation of such behavior is still under investigation. So far, a better alternative seems to be to ignore trigram not seen in language model of any target language.

Normalization is performed to obtain the final score of language s. Similarly to the acoustic part (Eq. 7), it makes use of scores from all individual language detectors:

$$\log P(s|\mathcal{O}) \approx \log \mathcal{S}(\mathcal{O}|s)/T - \log \sum_{\forall q} \mathcal{S}(\mathcal{O}|q)/T, \quad (10)$$

where $\log \mathcal{S}(\mathcal{O}|s)$ is the score given by equation 9 and T is the expected number of phonemes in speech segment computed as a sum of posterior probabilities of links in the phoneme lattice $\mathcal{O}$.

5. Fusion of scores

To fuse scores from separate systems, a simple linear combination is done according to:

$$\begin{aligned} score \;=\; & \alpha\, GMM_{MMI} \;+\; \beta\, PRLM_{HU} \;+ \quad (11) \\ & +\, \gamma\, PRLM_{RU} \;+\; \delta\, PRLM_{CZ} \end{aligned}$$

where weights $\alpha, \beta, \gamma, \delta$ are tuned by simplex method on the development set. We are aware that linear combination is quite primitive and that better results should be obtained with more elaborate methods, such as GMM- or NN-based merging. We were however severely limited by the amounts of development data, especially by Indian-English (no development data at all).

6. Results

To simplify the comparison of the performance of our different systems, we will disuse only results obtained for NIST LRE2005 primary condition (30 sec. segments) throughout this section. Tables 1, 2 and 3, however, present results also for the other two conditions (10 and 3 sec. segments).

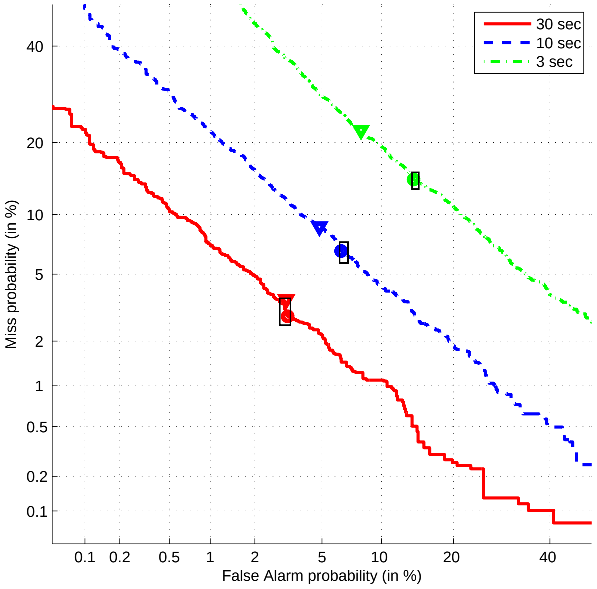

Figure 2: DET curve for NIST LRE 2005 primary system for three test durations. The triangle represents our actual decision, the circle is the minimum DET and the rectangle is the EER with its confidence interval.

For comparison of GMM system trained using MMI (256 mixture components) and GMM trained under ML framework (2048 components) see the first two rows of table 1. MMI training provided more than 50% relative improvement compared to ML training, which confirmed our previous good results and superiority of discriminatively trained models over ML-trained ones [14].

The following lines of table 1 show the effect of different feature extraction techniques. Since MMI training is very time-consuming process, all these results are reported only for ML-trained system. A significant degradation (7% relative) can be seen for the system without VTLN and even more prominent degradation (27% relative) is caused by further omitting the RASTA processing. Our features are 7 MFCC coefficients concatenated with 49 SDC coefficients. The usual practice is to use only SDC coefficients alone. Doing so, however, leads

System EER [%]						Duration		
						30sec	10 sec	3 sec
MMI	MFCC		SDC	VTLN	RASTA	4.6	8.6	17.2
ML	MFCC		SDC	VTLN	RASTA	10.8	13.9	21.0
ML	MFCC		SDC		RASTA	11.6	14.5	21.8
ML	MFCC		SDC			14.8	17.8	24.0
ML			SDC	VTLN	RASTA	14.0	17.8	25.0
ML	MFCC	noC0	SDC	VTLN	RASTA	10.9	14.4	21.2
ML	MFCC		$\Delta, \Delta\Delta$	VTLN	RASTA	18.2	20.0	26.2

Table 1: EER for different GMM systems for NIST LRE 2005.

System EER [%]	Duration		
	30sec	10 sec	3 sec
PRLM+lattice	9.8	14.9	23.4
PRLM+lattice+anti.m.	8.5	13.8	23.0
PPRLM+lattice+anti.m.	7.3	11.4	19.5
GMM-MMI 256	4.6	8.6	17.2
Fusion	**3.1**	**6.5**	**14.1**

Table 2: EER for different system for NIST LRE 2005 – submission.

System EER [%]	Duration		
	30sec	10 sec	3 sec
PRLM (string)	6.8	13.9	24.5
PRLM+lattice	5.7	10.7	21.2
PRLM+lattice+anti.m.	5.3	10.7	21.4
Fusion	**2.9**	**6.4**	**14.1**

Table 3: EER for PRLM system for NIST LRE 2005 – post-evaluation.

again to significant degradation in performance (23% relative). For the purpose of the language or speaker identification the zero'th MFCC coefficient (C0), which is carrying information about short term signal energy, is often discarded. We have obtained slightly worse results without C0 coefficient, however, according to our experience, discarding C0 coefficient would be important without RASTA processing. The last line in table 1 shows the results for features that are 13 MFCC augmented with their delta and delta-delta coefficients – the features commonly used for speech recognition. The performance with these features is far behind those with any of SDC based features, which confirms the superiority of SDC for language identification task.

Our system submitted to LRE 2005 was built as a combination of several individual systems. Performance of these systems is presented in Table 2. The presented PRLM system is based only on Hungarian phoneme recognizer (the best out of our 3 PRLM systems). PRLM with trigram language model derived from lattices and tested using lattices performs with EER=9.8%. Use of anti-models results in relative improvement of 14%. Fusion of three phonotactic systems (PPRLM) performs with EER=7.3%. Compared to our previous work on NIST 2003 data [15], the results with PRLM system were worse than we expected, which lead us to carry out the post-evaluation analysis discussed below.

Fusion of GMM-MMI and PPRLM systems gives 33% relative improvement over the best separate system and the final EER reaches 3.1%. Unfortunately, this is a post-evaluation results – due to use of wrong fusion weights in Equation 11, the EER of our submitted system on 30 sec. condition was 5.0% (even worse than GMM-MMI 256 itself). The second and third columns in Table 2 show results for 10 sec. and 3 sec. conditions (here, the table contains the submitted results – the weights were correct for these durations). Figure 2 presents the DET curve for fused system and all three conditions. The triangle on the curve represents our actual decision, the circle represents the minimum DET and the rectangle is the EER with its confidence interval on the level of 95%.

Table 3 shows the results of post-evaluation work. We analyzed the results of PRLM system and found two important problems. First, we found a bug in the implementation of language model back-offing. Fixing this bug resulted in decrease of EER from 9.8% to 7.7% on 30 sec. segments. Second problem was the segmentation of test data. For generation of lattices, we segmented test utterances to chunks smaller than the original segments. We tuned the length of chunks for optimal performance on Indian accented English as this was the only example of LRE 2005 data that we had available. This was clearly wrong decision. On chunk boundaries, we lost information about phoneme context for estimation of trigram statistics. Without segmentation of test utterances, the systems preforms with EER=5.7%. For comparison, Table 3 contains also results for PRLM based on strings (language model are trained and test trigram statistics are estimated only using the best path through the lattice). The anti-models improved the final lattice based PRLM, however the gain we obtained was smaller compared to our previous work. It seems that anti-models mostly corrected our second problem in the submitted system. Since NIST 2003 test data match well to the training (CallFriend) data and do not contain long silences, we were not able to identify these problems before the evaluation.

The performance of the fused system making use of the corrected PPRLM system and original GMM-MMI system is EER=2.9%. The weights for fusion were tuned again on our development set.

7. Conclusion

Compared to other systems in the NIST 2005 evaluations, our system performed well. Especially the discriminative training brought substantial improvement over the standard ML scheme. In the phonotactic approach, good performance obtained on NIST 2003 did not fully generalize; several problems were found and their solutions were suggested in post-evaluation experiments.

8. Acknowledgment

This work was partially supported by EC project Augmented Multi-party Interaction (AMI), No. 506811, Grant Agency of Czech Republic under project No. 102/05/0278 and Ministry of Defence of Czech Republic. Lukáš Burget was supported by post-doctoral grant of Grant Agency of Czech Republic No. 102/06/P383.

9. References

[1] E. Singer, P.A. Torres-Carrasquillo, T.P. Gleason, W.M. Campbell, and D.A. Reynolds, "Acoustic,phonetic,and discriminative approaches to automatic language identification," in *Proc. Eurospeech*, Sept. 2003, pp. 1345–1348.

[2] 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.

[3] P. Matějka, P. Schwarz, J. Černocký, and P. Chytil, "Phonotactic language identification using high quality phoneme recognition," Sept. 2005, pp. 2237–2241.

[4] A.F. Martin and M.A. Przybocki, "NIST 2003 language recognition evaluation," in *Proc. Eurospeech*, Sept. 2003, pp. 1341–1344.

[5] 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.

[6] M.A. Zissman, "Comparison of four approaches to automatic language identification of telephone speech.," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 4, no. 1, pp. 31–44, 1996.

[7] J. Cohen, T. Kamm, and A.G. Andreou, "Vocal tract normalization in speech recognition: Compensating for systematic speaker variability," *J. Acoust. Soc. Am.*, , no. 97, pp. 2346, 1995.

[8] S. Young et al., *The HTK Book*, Cambridge University Engineering Department, 2005.

[9] D. Povey, *Discriminative Training for Large Vocabulary Speech Recognition*, Ph.D. thesis, Cambridge University, July 2004.

[10] P. Schwarz, P. Matějka, and J. Černocký, "Hierarchical structures of neural networks for phoneme recognition," *accepted to Proc. ICASSP*, Toulouse, France, May 2006.

[11] 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.

[12] I.H. Witten and T.C. Bell, "The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression," *IEEE Trans. Inform. Theory*, vol. 4, no. 37, pp. 1085–1094, 1991.

[13] A. Stolcke, "SRILM - an extensible language modeling toolkit," in *Proc. International Conferences on Spoken Language Processing (ICSLP)*, Denver, Colorado, Sept. 2002, pp. 901–904.

[14] B. Burget, P. Matějka, and J. Černocký, "Discriminative Training Techniques For Acoustic Language Identification," *accepted to Proc. ICASSP*, Toulouse, France, May 2006.

[15] P. Matějka, P. Schwarz, B. Burget, and J. Černocký, "Use of anti-models to further improve state-of-the-art prlm language recognition system," *accepted to Proc. ICASSP*, Toulouse, France, May 2006.

[16] "SpeechDat-East project," http://www.fee.vutbr.cz/SPEECHDAT-E.

[17] "Callfriend corpus, telephone speech of 15 different languages or dialects," www.ldc.upenn.edu/Catalog.

[18] "OGI multi language telephone speech," http://www.cslu.ogi.edu/corpora/mlts/.