PLLR Features in Language Recognition System for RATS

Oldřich Plchot [1], *Mireia Diez* [2], *Mehdi Soufifar* [1], *Lukáš Burget* [1]

[1]Brno University of Technology, Speech@FIT and IT4I Center of Excellence, Brno, Czech Republic
[2]GTTS, University of the Basque Country, UPV/EHU, Leioa, Spain
{iplchot,qsoufifar,burget}@fit.vutbr.cz, mireia.diez@ehu.es

Abstract

In this paper, we study the use of features based on frame-by-frame phone posteriors (PLLRs) for language recognition. The results are reported on the datasets developed for the DARPA RATS (Robust Automatic Transcription of Speech) program, which seeks to advance state of the art detection capabilities on audio from highly degraded communication channels. We show that systems based on the PLLRs outperform the standard acoustic system based on PLP2 features. By experimenting with the system combinations, we also demonstrate that the PLLR-based systems contain complementary information with respect to the PLP2 system. Finally we make a comparison between the PLLR and phonotactic systems with the outcome favorable to the PLLR.

Index Terms: language recognition, PLLR, iVector, phone posterior, neural network, RATS

1. Introduction

Building language and speaker recognition (LRE and SRE) systems on top of the iVector [1] paradigm has been a state-of-the-art approach for the last three years. An iVector is a fixed-length low-dimensional vector, which is extracted for each utterance. Using these relatively small vectors representing the whole utterances as input features allows us to build all kinds of simple and powerful classifiers.

In 2011, the iVector approach was successfully applied in the MFCC-based LRE system using a generative linear classifier [2]. In our NIST-LRE2011 submission, we have used multi-class logistic regression and later on, in our RATS Phase 1 LRE submission [3], we have successfully used neural networks (NN) as a classifier with iVectors as inputs.

The success of iVector-based systems built on top of the standard acoustic features led us to adapt the iVector paradigm to the phonotactic approach, which has been dominant in the LRE for many years.

In the classical phonotactic system, a phoneme recognizer is used to tokenize the speech utterances into phone sequences, which are then presented to the generative classifier (e.g. smoothed n-gram language model) or converted to the fixed-length vector of n-gram counts and used as inputs for the discriminative classifiers: e.g. logistic regression (LR) or support vector machines (SVM). The size of these vectors depends on the size of the phone dictionary and grows exponentially with the order of the n-grams.

This work was partly supported by the DARPA RATS Program under Contract No. D10PC20015 and by Ministry of Interior of the Czech Republic, project number, VG20132015129. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

To represent these large vectors of discrete events in the compact iVector form, we have used a regularized subspace n-gram model (SnGM) [4, 5]. This approach has proven to be robust and performs better than previous state-of-the art phonotactic systems. Although these systems are complementary to the acoustic systems, they generally do not achieve their accuracy [6, 3].

Recently, the introduction of the Phone Log-Likelihood Ratio (PLLR) features [7] has shown yet another way to exploit the phonetic information provided by the phoneme recognizer. Unlike the classical phonotactic systems or SnGM-iVector phonotactic systems, PLLR directly use frame-by-frame phone posteriors, which are provided by a neural network trained for frame-by-frame phone classification. Additionally, as it was explored in [8], the nature of PLLR features overcomes the non-Gaussian distributions of the individual phone posteriors. Such properties allow us to treat PLLRs as classical acoustic features and to build a standard iVector-based LRE system.

In this work, we compare a standard acoustic system with phonotactic and PLLR systems using different phone recognizers. We also perform fusion of these systems to explore whether the different modeling and feature extractions in phonotactic and PLLR systems bring complementary information.

We perform all experiments in the (very noisy and difficult) domain of the RATS data. The RATS program focuses on creating technology capable of accurately determining speech activity regions, detecting key words, and identifying language and speakers in highly degraded, weak and/or noisy communication channels. The data sets used in RATS are obtained by retransmitting pre-existing or newly collected telephone conversations in multiple languages over various types of channels, and aim to capture/simulate the acoustic environment present in current radio-based two-way communication systems used by the law enforcement, emergency, air traffic control, etc.

By its nature, these means of radio communication are sensitive to many factors which can degrade or change the quality of the transmission. The most important are the background radio interference, atmospheric conditions, used bandwidth and background additive noise. All of these factors greatly increase the unwanted channel variability present in the audio.

2. Data

The Linguistic Data Consortium (LDC) provided the training and test data. The provided datasets cover 5 target languages (Arabic, Dari, Farsi, Pashto and Urdu) and 10 non-target languages (English, Spanish, Mandarin, Thai, Vietnamese, Russian, Japanese, Bengali, Korean, Tagalog). All recordings were selected from existing databases (Fisher, Callfriend, NIST LRE) or newly collected for the RATS program.

Every audio recording was approximately 2 minutes long

and was retransmitted through 8 different communication channels, labeled by the letters A through H. A "push-to-talk" transmission protocol was used in all channels except G. The communication channels and the equipment used for the retransmission are described by LDC in [9]. The retransmitted data was made available to the RATS participants in three incremental data releases under the codes: LDC2011E95, LDC2011E111 and LDC2012E03.

The LRE systems in the RATS program are evaluated under 120s, 30s, 10s and 3s duration conditions. As only recordings from the 120s condition were released for training and development, we had to construct our own development samples by making cuts of the corresponding durations from the 120s audio files. We partitioned all available data into the training and development sets. Detailed description of the partition can be found in [3].

We have modified the *extended training set* from [3] by creating 10s cuts instead of 30s cuts from the 120s segments, in order to focus the training set more on the shorter durations. Additionally, we have used the smaller *"main training set"* without any modification to train the Universal Background Models (UBM) and iVector extractors.

For the purposes of development and calibration, we are using the *main development set* (DEV), which contains approximately 7120 segments per each duration condition. We tried to keep the number of segments per class balanced as much as possible [3]. The *calibration set* is the same as the DEV set, but with 3s cuts excluded.

We also use the adjudicated version of the LID Phase 1 evaluation data (also called Dev2 within the RATS program) as an independent evaluation set (EVAL). This dataset includes 1,914, 1,782, 1,715, 1,340 samples for the 120s, 30s, 10s and 3s conditions, respectively.

3. Feature Extraction and Frontends

3.1. Voice activity detectors

Voice activity detection is performed by a neural network with the input consisting of a block of Mel filter outputs with context of 300ms. The NN has 18 outputs: 9 for speech and 9 for non-speech, each corresponding to one of the channels (source plus 8 re-transmitted). HMMs with Viterbi decoding are used to smooth out and merge the outputs to speech and non-speech regions. This NN is trained on RATS data defined for the speech activity detection (SAD) task [10].

3.2. PLP2

The PLP2 features are an enhancement of FDLPs [11, 12, 13] using their first stage of the processing to obtain the power spectral estimates for performing the subsequent time-domain linear prediction (TDLP).

The speech signal is divided into 10 second analysis windows. Discrete cosine transform (DCT) is applied in each 10 second analysis window to obtain full-band DCT. The full-band DCT is windowed into 96 linear sub-bands in the frequency range of 125-3800Hz. Linear prediction is performed on each sub-band to obtain parametric sub-band envelopes, which are then stacked to form a two-dimensional time-frequency representation, similar to spectrogram. This representation is decimated to 100Hz sampling rate, providing an estimate of the power spectrum of the signal in the short-term frame level. These power spectral estimates are inverse Fourier transformed to obtain an autocorrelation sequence [14, 15]. This sequence is used for TDLP, using a 19th-order model. The TDLP provides an all-pole approximation of the short-term spectrum. The output TDLP parameters are converted to 20 cepstral coefficients using cepstral recursion. Deltas and double-deltas are appended to generate a 60-dimensional feature vector at each time frame. Before removing the silence, feature warping [16] is applied using a 3s sliding window [14, 17].

We obtain the 600-dimensional iVectors by the means of 2048-component diagonal UBM and iVector extractor, both trained on the *main training set*.

3.3. Phone Decoders

The phone decoders are based on a hybrid NN/HMM approach, where neural networks are used to estimate frame-by-frame posterior probabilities of phones from Mel filter bank log energies using the context of 310ms around the current frame. Each phone is represented by three states. The posterior probabilities for each phone are summed up before the PLLR processing. For the SnGM system, the decoder produces phone lattices.

Two 4-layer NNs are trained on two datasets to produce Czech (CZ) and Leventine Arabic (LE) decoders: Czech CTS data where 30% was artificially corrupted with noise at lowest level 10dB; and RATS LE keyword search data provided to the RATS participants. The LE data are closer to the target data, which are used to train and test the LRE systems. The phone sets for the CZ and LE decoders contain 38 and 36 phones, respectively.

3.4. SnGM System

Expected n-gram counts ("soft-counts") [18] from phone lattices were used during the subspace training. A 600-dimensional subspace over the trigram counts in the *main training set* is trained using the regularized multinomial subspace described in [5].

We use the model along with hard pruning of the low-frequency trigrams to overcome the problem of the data sparsity [4]. The iVectors are the point estimates of the latent variables describing the coordinates of the utterance specific n-gram model in a new low-dimensional subspace. The output is a 600-dimensional iVector for every file.

3.5. PLLR

PLLRs are frame-level features, computed from the frame-by-frame phone posterior probabilities provided by the NN described above. Let us consider a NN that outputs an N-dimensional vector of phone posteriors at each frame: $\mathbf{p} = (p_1, p_2, \ldots, p_N)$, such that $\sum_{i=1}^{N} p_i = 1$ and $p_i \in [0, 1]$ for $i = 1, 2, \ldots, N$. PLLRs are defined as[1]:

$$r_i = \text{logit}(p_i) = \log \frac{p_i}{(1 - p_i)}, \qquad i = 1, ..., N. \quad (1)$$

As explained in [8], PLLRs seem to overcome the non-Gaussian nature of phone posteriors for each individual phone model. Nevertheless, when exploring the distribution of two or more PLLRs, the features show a bounded distribution [19]. To avoid this effect, the PLLRs are projected into a hyper-plane defined by the normal vector:

[1] Note that Eq. 1 differs from the definition used in previous works by missing constant offset [7, 8], which has been dropped from the equation for simplicity.

$$\mathbf{n}|_{r_i=-\log(N-1)} = \frac{(N-1)}{N\sqrt{N}} \cdot \hat{\mathbf{1}} \quad (2)$$

where $\hat{\mathbf{1}} = \frac{1}{\sqrt{N}}[1_1, 1_2, ..., 1_N]$.

The kernel (null space) of the desired projection is $\hat{\mathbf{1}}$, then the matrix $\mathbf{P}$ used to project the data into the selected hyperplane is given by:

$$\mathbf{P} = \mathbb{I} - \hat{\mathbf{1}}' * \hat{\mathbf{1}}, \quad (3)$$

where $\mathbb{I}$ is an identity matrix.

In order to decorrelate the parameters, PCA is applied on the transformed PLLRs. Since the projected features lie on an $(N-1)$-dimensional hyper-plane, the number of non-zero eigenvalues of the PCA projection matrix will be $N-1$. Therefore, the dimensionality of the feature vectors, after PCA, will be reduced by one.

In [7], it was shown that the use of first order dynamic coefficients provided significant performance gains in the systems trained on PLLR features, whereas the use of second order coefficients degraded performance. Therefore, PLLR+δ were also used as features in this work, resulting in the feature vectors of size 74 and 70 for CZ and LE, respectively.

As in the PLP approach, 600-dimensional iVectors were computed using a diagonal covariance UBM and the iVector extractor trained on the *main training set*.

4. Scoring, Calibration and Fusion

4.1. Scoring

The iVector frontends provide input features for training the two classifiers for each system: Multiclass regularized logistic regression [20, 21] (LR) is trained on the *main training set* and the iVectors are conditioned using within-class covariance normalization (WCCN) before training. Three-layer neural network (NN), with the 600-dimensional input, 300 neurons in the hidden layer and 6 outputs (1 nontarget + 5 target languages), is trained on *extended training set*. Stochastic gradient training with L2 regularization[21] is used as the training procedure.

4.2. Calibration and Fusion

Scores were calibrated again by the LR trained on the *calibration set*. The process of training the LR is the same for the calibration as for the recognizer, just the inputs are vectors of scores instead of iVectors:

The L2 regularization penalty weight was chosen prior to training to be proportional to the mean magnitude of the conditioned input vectors (scores) [21].

The calibration uses an affine transform to convert the N_L-dimensional (N_L is the number of target languages) vector of input scores, $\mathbf{s}_t$, for trial t, into a N_L-dimensional calibrated score-vector, $\mathbf{r}_t$:

$$\mathbf{r}_t = \mathbf{C}\mathbf{s}_t + \mathbf{d}. \quad (4)$$

The parameters of logistic regression are given by $\mathbf{C}$, a full N_L-by-N_L matrix and $\mathbf{d}$, a N_L-dimensional vector. These parameters are trained by minimizing the multiclass cross-entropy with equalization of the amount of data for individual classes:

$$F = \lambda \operatorname{tr}(\mathbf{C}^{\mathrm{T}}\mathbf{C}) - \sum_{i=1}^{N_L} \frac{1}{N_L N_i} \sum_{t \in \mathcal{R}_i} \log \frac{\exp(r_{it})}{\sum_{j=1}^{N_L} \exp(r_{jt})}, \quad (5)$$

where r_{it} is the ith component of $\mathbf{r}_t$ and $\mathcal{R}_i$ is the set of N_i training examples of language i. The calibrated scores were fused as:

$$\ell_t = \sum_k \alpha_k \mathbf{r}_{tk} + \beta, \quad (6)$$

where r_{tk} denotes the outputs of kth calibrated system, α_k is a scalar weight and β is N_L-dimensional vector. These parameters are also trained by to minimize cross-entropy objective on the *calibration set*. Here, the regularization is not applied.

5. Experimental Results and Discussion

We reporting the performance of the systems on both our DEV set and EVAL set. Despite the fact that a large portion of the DEV set was used to train the calibration and fusion and the results are therefore optimistic, we still believe that there is a value in showing the performance on this set. The main reasons to use this set is its size. It is approximately four times larger than the EVAL set, and therefore the obtained results, especially for the longer durations, are more reliable. Also, the data for the EVAL set were collected in a different time than the DEV set, which has brought different channel effects compared to the DEV, making this set harder. We report all results in terms of Cavg as defined by NIST for the openset identification scenario [22].

5.1. Logistic Regression vs Neural Networks

We chose to compare the systems using two different classifiers trained on different sets. The LR (see Table 1) is a linear classifier, which by its nature is less prone to over-training and which has been successfully used in the NIST evaluations [23, 21]. Also, the set for the LR training does not contain repeated data in the form of short cuts from the long segments, which are present in the *extended training set* used for the NN training. Adding these cuts into the LR training did not bring any substantial changes in the results.

With the NN (see Table 2), we are showing the effect of the non-linear classifier and its ability to make use of large quantities of examples to extract the useful information out of the training dataset. We have never before observed NNs outperforming LR in our systems for NIST evaluations[23, 21], and we believe that redesigning the training set to contain more segments in a similar way as we did for RATS might enable the NN to outperform the LR on this dataset as well.

Most of the additional training samples with respect to the *main training set* come from cutting the longer segments, effectively showing the NN same data twice, but in the form of several iVectors per original segment. The original segments are still included in the training set.

Indeed, the NNs have achieved overall better results with the *extended training set* with the exception of the 120s condition on the EVAL set, where we can see much lower relative improvements or even degradation. This behavior can be explained by over-weighting the short segments in the training. It

is also important to note that the impressive relative improvements achieved with the NN on the DEV set have to be taken with a grain of salt, as the DEV set is clearly closer to the training data.

5.2. PLLR systems

Given the frame-by-frame phone posterior outputs provided by the NNs from the Leventine Arabic and Czech phone recognizers, we have built two PLLR systems. To provide a comparison with the phonotactic approach, we show the results from the corresponding SnGM systems trained on the outputs of the same recognizers. As the PLLR systems are very similar to the standard acoustic systems using frame-by-frame features, we have included a PLP2 system, which was the best-performing acoustic subsystem in our RATS P2 and P3 submissions.

Both PLLR systems outperformed the PLP2 system on all conditions with the exception of the Czech NN-based PLLR system on 3s condition. Especially the Leventine Arabic system has outperformed the PLP2 by a large margin, as the data used for the phone recognizer training fall into the RATS domain and the phone set is much closer to the majority of target languages.

When comparing the PLLR with the corresponding SnGM systems, the gain in the performance is also substantial, especially on the shorter durations (30s, 10s). The phonotactic systems with respect to the PLP2 are behaving in a typical manner, as they are able to achieve better performance for the longer durations, when losing for the shorter ones. This is most certainly caused by the lack of the decoded phones from the shorter segments and therefore under-estimated n-gram statistics.

5.3. Fusions

Having all of these systems lined up for the comparison invites us to make an analysis of their combinations. The results in the bottom part of Table 2 show combinations of the systems trained by the NN and evaluated on the EVAL set.

From the results, we can immediately observe that having a standard acoustic system in the fusion is still very beneficial in the short duration conditions. The combinations without the PLP2 system are lacking in the performance on 10s and 3s condition.

If we are interested in the performance on the longer (120s, 30s) conditions, it would be beneficial to combine the corresponding PLLR and SnGM systems, in this case especially Leventine Arabic systems (2+4). This choice would bring the advantage of running a single feature extractor, but on the other hand, one would have to run slightly more complex SnGM training.

A favorable combination performing well across all conditions is the fusion of both PLLR systems and a PLP2 system. Adding the PLLR-CZ system brings improvements in some conditions, while the same architecture of all subsystems allows for streamlining the development process and making the cost of including an extra system smaller.

The combination of all systems does not bring any substantial improvements over the smaller fusions.

Table 1: Results with LR - Cavg [%]

DEV set	120s	30s	10s	3s
PLP2	2.08	6.52	11.78	22.46
PLLR-LE	**0.84**	**2.93**	**7.35**	**18.43**
PLLR-CZ	1.66	4.69	9.96	21.74
SnGM-LE	1.42	5.83	13.22	26.53
SnGM-CZ	2.38	8.38	16.54	29.09
EVAL set	120s	30s	10s	3s
PLP2	7.72	11.69	16.39	23.04
PLLR-LE	**4.56**	**7.98**	**12.61**	21.48
PLLR-CZ	6.95	10.76	15.13	**21.32**
SnGM-LE	5.86	12.28	18.53	26.45
SnGM-CZ	8.59	15.76	20.89	27.95

Table 2: Results with NNs - Cavg [%]

DEV set		120s	30s	10s	3s
1	PLP2-NN	1.36	4.61	9.24	21.50
2	PLLR-LE-NN	**0.58**	**2.37**	**6.62**	**17.20**
3	PLLR-CZ-NN	0.90	3.36	7.81	19.70
4	SnGM-LE-NN	1.21	5.08	11.58	23.39
5	SnGM-CZ-NN	1.35	5.09	11.15	24.00
EVAL set		120s	30s	10s	3s
1	PLP2-NN	7.21	9.21	12.43	18.58
2	PLLR-LE-NN	**5.37**	**7.31**	**11.46**	**17.63**
3	PLLR-CZ-NN	5.81	8.83	12.30	19.52
4	SnGM-LE-NN	5.53	9.34	15.61	22.76
5	SnGM-CZ-NN	7.23	10.46	15.38	24.05
Fusions - EVAL set		120s	30s	10s	3s
2+3		5.19	6.79	10.14	16.04
1+2		5.80	6.43	8.69	15.37
2+4		**5.12**	6.61	10.48	16.93
3+5		5.74	8.33	11.28	18.11
1+2+4		5.38	6.31	**8.53**	14.90
1+2+3		5.29	6.43	8.71	14.65
1+2+3+4+5		5.59	**6.21**	8.75	**14.37**

6. Conclusions

In this work, we have demonstrated that building an acoustic system for language recognition based on the frame-by-frame phone posterior features — in our case PLLR features — can bring significant improvements in comparison with standard acoustic systems. We have confirmed this claim by creating two independent PLLR systems and comparing them to the state-of-the-art PLP2 system, our best acoustic system.

By experimenting with the fusions, we have shown, that combining a PLLR system with a standard acoustic system can improve the performance on the short duration segments. Comparison of the PLLR systems to the phonotactic SnGM on the RATS datasets came up strongly in favor of PLLR systems.

Finally, it should be reiterated, that all of the experiments were done in a very noisy domain. The performance gains from the PLLR features observed on the RATS data will most likely be smaller in the standard CTS-based systems tested on relatively clean NIST-LRE data.

7. References

[1] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. PP, no. 99, 2010.

[2] D. Martinez, O. Plchot, L. Burget, O. Glembek, and P. Matějka, "Language recognition in ivectors space," in *Proceedings of Interspeech 2011*, vol. 2011, no. 8. International Speech Communication Association, 2011, pp. 861–864.

[3] P. Matějka, O. Plchot, M. Soufifar, O. Glembek, L. D'Haro, K. Vesely, F. Grezl, J. Ma, S. Matsoukas, and N. Dehak, "Patrol team language identification system for DARPA RATS P1 evaluation," in *Proc. of Interspeech 2012*, Sep. 2012.

[4] M. Soufifar, M. Kockmann, L. Burget, O. Plchot, O. Glembek, and T. Svendsen, "ivector approach to phonotactic language recognition," in *Proceedings of Interspeech 2011*, vol. 2011, 2011, pp. 2913–2916.

[5] M. Soufifar, L. Burget, O. Plchot, S. Cumani, and J. Černocký, "Regularized subspace n-gram model for phonotactic ivector extraction," in *Proceedings of Interspeech 2013*, 2013, pp. 74–78.

[6] F. L. D'Haro, O. Glembek, O. Plchot, P. Matějka, M. Soufifar, R. Cordoba, and J. Černocký, "Phonotactic language recognition using i-vectors and phoneme posteriogram counts," in *Proceedings of Interspeech 2012*, vol. 2012, 2012, pp. 1–4.

[7] M. Diez, A. Varona, M. Penagarikano, L. Rodriguez-Fuentes, and G. Bordel, "On the Use of Log-Likelihood Ratios as Features in Spoken Language Recognition," in *slt*, Miami, Florida, USA, Dec 2012, pp. 274–279.

[8] ——, "Using Phone Log-Likelihood Ratios as Features for Speaker Recognition," in *Proceedings of Interspeech 2013*, Lyon, France, August 2013.

[9] K. Walker and S. Strassel, "The rats radio traffic collection system," in *Proceedings of Odyssey 2012*, 2012.

[10] T. Ng, B. Zhang, L. Nguyen, S. Matsoukas, K. Veselý, P. Matějka, X. Zhu, and N. Mesgarani, "Developing a speech activity detection system for the DARPA RATS program," in *Proc. of Interspeech 2012*, Sep. 2012.

[11] R. Kumerasan and A. Rao, "Model-based approach to envelope and positive instantaneous frequency estimation of signals with speech applications," *JASA*, vol. 105, pp. 1912–1924, 1999.

[12] M. Athineos and D. Ellis, "Autoregressive modelling of temporal envelopes," *IEEE Trans. of Signal Processing*, vol. 55, pp. 5237–5245, 2007.

[13] S. Thomas, S. Ganapathy, and H. Hermansky, "Recognition of reverberant speech using frequency domain linear prediction," *IEEE Signal Processing Letters*, vol. 15, pp. 681–684, 2008.

[14] S. H. Mallidi, S. Ganapathy, and H. Hermansky, "Robust speaker recognition using spectro-temporal autoregressive models," in *Proceedings of Interspeech 2013*, no. 8. International Speech Communication Association, 2013, pp. 3689–3693.

[15] S. Ganapathy, S. Thomas, and H. Hermansky, "Feature extraction using 2-d autoregressive models for speaker recognition," in *ISCA Speaker Odyssey*, 2012.

[16] S. S. J. Pelecanos, "Feature warping for robust speaker verification," in *Proceedings of Odyssey 2006: The Speaker and Language Recognition Workshop*, 2006, pp. 213–218.

[17] S. Ganapathy, J. Pelecanos, and M. Omar, "Feature normalization for speaker verification in room reverberation," in *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011, pp. 4836–4839.

[18] J. Gauvain, A. Messaoudi, and H. Schwenk, "Language recognition using phone lattices," in *Proceedings of Interspeech 2004*, Sep. 2004, pp. 1283–1286.

[19] M. Diez, A. Varona, M. Penagarikano, L. Rodriguez-Fuentes, and G. Bordel, "On the projection of PLLRs for Unbounded Feature Distributions in Spoken Language Recognition," *IEEE Signal Processing Letters*, submitted.

[20] C. Bishop, *Pattern recognition and machine learning*. Springer New York, 2007.

[21] N. Brummer, S. Cumani, O. Glembek, M. Karafiát, P. Matějka, J. Pešán, O. Plchot, M. Soufifar, E. Villiers, and J. Černocký, "Description and analysis of the Brno276 system for LRE2011," in *Proceedings of Odyssey 2012: The Speaker and Language Recognition Workshop*. International Speech Communication Association, 2012, pp. 216–223.

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

[23] N. Brummer, L. Burget, O. Glembek, V. Hubeika, Z. Jančík, M. Karafiát, P. Matějka, T. Mikolov, O. Plchot, and A. Strasheim, "BUT-AGNITIO system description for NIST language recognition evaluation 2009," in *Proceedings NIST 2009 Language Recognition Evaluation Workshop*, 2009, pp. 1–7.