



Multilingual Bottleneck Features for Language Recognition

Radek Fér, Pavel Matějka, František Grézl, Oldřich Plchot and Jan "Honza" Černocký

Brno University of Technology, Speech@FIT and IT4I Center of Excellence, Brno, Czech Republic

{ifer,matejkap,grezl,iplchot,cernocky}@fit.vutbr.cz

Abstract

In this paper, we investigate Multilingual Stacked Bottleneck Features (SBN) in language recognition domain. These features are extracted using bottleneck neural networks trained on data from multiple languages. Previous results have shown benefits of multilingual training of SBN feature extractor for speech recognition. Here we focus on its impact on language recognition. We present results obtained with monolingual and multilingual networks, and their fusions. Using multilingual features, we obtain 16% relative improvement on 3 s condition of NIST LRE09 dataset with respect to features trained on a single language.

Index Terms: multilingual training, stacked bottleneck features, language identification

1. Introduction

The neural networks (NN) have become a widely used technique for state-of-the-art Large Vocabulary Continuous Speech Recognition (LVCSR) systems and are expanding very fast to other fields of speech recognition. This paper describes the usage of bottleneck (BN) features in the context of Language Identification (LID).

There are several works related to using neural networks as feature extractor for LID system. Ma et al. [1] used log of phoneme posteriors generated by neural network in conjunction with a block of PLP coefficients followed by HLDA dimensionality reduction as an input to standard i-vector system and reported dramatic gain on noisy data in RATS project¹. Diez et al. [2] used phone log-likelihood ratios (PLLR) as an input to an i-vector based system, and fused it with an MFCC-SDC system on the score level. Generally, both approaches share the same idea, only the fusion is done differently: feature- versus score-level.

Han and Pelecanos [3] applied Shifted Delta Cepstra [4] concept to capture timing information of frame-level log-likelihood ratios of phones produced by Arabic phone recognizer. The second approach they compared was to stack several frames of frame-level phone features and to apply PCA dimensionality reduction. They report nice gain on 120 s condition of RATS task.

This work is a follow up of our previous research, where BN features outperformed conventional MFCC-based features by 50% relative [5] on noisy RATS data. Similar work, but on clean NIST LRE 2009 data was done by Jiang et al. [6] with about the same relative gain.

Another approach to use deep NN in LID was proposed by Ignacio Moreno [7], where the NN is trained frame-by-frame to directly classify languages. The final decision is based on language log-posteriors averaged over frames. This approach

¹Robust Automatic Transcription of Speech, project by DARPA

works great for short utterances. However, from personal communication with Ignacio and from our experiments, we know that the performance for long utterances is still superior with the conventional i-vector approach.

This paper proposes multilingual BN features for LID. The term multilingual means that the NN is trained on several languages simultaneously. The motivation for our work is that a multilingual person has better abilities to discriminate between unknown phoneme inventories in contrast to a person speaking just one language. The same holds for multilingual bottleneck features which describe better the feature space for our target task – LID. In [8, 9, 10], similar BN features were used for the LVCSR task and the multilingual BN was found to be superior to the one trained on single language.

We evaluate this idea on clean NIST LRE 2009 database, comparing performance of monolingual (i.e. trained on single language) and multilingual systems. We include results obtained with SDC features for reference.

2. Stacked Bottleneck Features

Bottleneck NN refers to such topology of a NN, where one of hidden layers has significantly lower dimensionality than the surrounding layers. It is assumed that such layer – referred to as the bottleneck – compresses the information needed for mapping the NN input to the NN output, increasing the system robustness to noise and overfitting. A bottleneck feature vector is generally understood as a by-product of forwarding a primary input feature vector through the BN network and reading off the vector of values at the bottleneck layer. In other words, after a BN network is trained for its primary task (e.g. phone state classification), the bottleneck layer is declared to be the output layer and all succeeding layers are ignored. Such NN then maps the primary features to the bottleneck features.

We have used a cascade of two such NNs (see Figure 1). The outputs of the first network are *stacked* in time, defining broader context input features for the second NN, hence the term Stacked Bottleneck Features [11].

2.1. SBN input feature extraction

The NN input features are 24 log mel-scale filter bank outputs augmented with fundamental frequency features. The fundamental frequency features consist of f_0 and probability of voicing estimates computed according to [12], f_0 estimates obtained by Snack tool² function *getf0*, seven coefficients of Fundamental Frequency Variations spectrum according to [13] and f_0 computed using Kaldi³ with its delta coefficients and probability of voicing. Together we have 13 f_0 related features, see [14] for more details.

²www.speech.kth.se/snack/

³<http://kaldi.sourceforge.net>

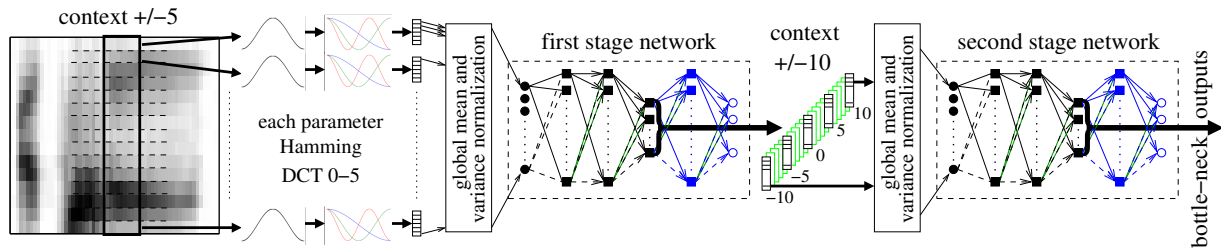


Figure 1: Block diagram of Stacked Bottle-Neck (SBN) feature extraction. The blue parts of NNs are used only during the training. The green frames in context gathering between the NNs are skipped. Only frames with shift -10, -5, 0, 5, 10 form the input to the second stage NN.

The conversation-side based mean subtraction is applied on the whole feature vector. 11 frames of log filter bank outputs and fundamental frequency features are stacked together. Hamming window followed by DCT consisting of 0^{th} to 5^{th} base are applied on the time trajectory of each parameter resulting in $(24 + 13) \times 6 = 222$ coefficients on the first stage NN input.

2.2. Neural network architecture

The cascade of neural networks is shown in Figure 1 and described in detail in [15, 11]. The configuration for the first NN is $222 \times 1500 \times 1500 \times 80 \times 1500 \times N$, where N is the number of targets. The 80 bottleneck outputs from the first NN are sampled at times $t-10$, $t-5$, t , $t+5$ and $t+10$, where t is the index of the current frame. As there are 11 acoustic frames on the input of the first stage NN, this 1:5 subsampling corresponds to one half context overlap and results in total context of 31 frames (325 ms). The resulting 400-dimensional features are input to the second stage NN with a configuration of $400 \times 1500 \times 1500 \times 80 \times 1500 \times N$. The bottleneck layers in both NNs have linear activation function which was shown to provide better performance [16]. All other hidden layers have sigmoids as non-linearities. The 80 bottleneck outputs from the second NN (referred as SBN) are taken as features for the conventional GMM/UBM i-vector based LID system. The targets for training both NNs are either context-independent phoneme states, phonemes or context-dependent triphones. These are taken from an alignment with a conventional HMM PLP-based LVCSR system. The number of targets varies depending on the language and on the type of targets (see Table 3).

3. Multilingual Bottleneck Features

The basic idea is to train the NN in the multilingual manner – on more languages, so that the final BN features describe the space for more than one language. These two approaches to train multilingual NNs worked best in our previous experiments:

The first one – **one softmax** – discriminates between all targets of all languages. No mapping or clustering of phonemes is done. The resulting NN has quite a large output layer containing all phonemes/triphones from all languages with one softmax activation function.

The second approach – **block softmax** – divides the output layer into parts according to individual languages. During the training, only the part of the output layer corresponding to the language the given target belongs to, is activated. Simplified structure for this schema is in Figure 2. Detailed description can be found in [9]. We have used the block softmax approach in our experiments because of better performance.

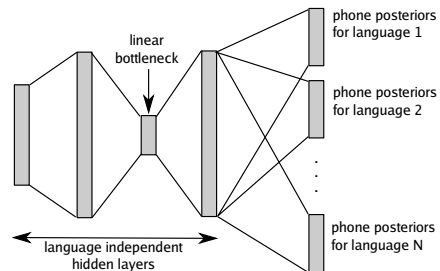


Figure 2: Multilingual bottleneck network.

3.1. Multilingual training data

For training the neural networks, the IARPA Babel Program data⁴ were mainly used. This data simulates a case of what one could collect in limited time from a completely new language. It consists mainly of telephone conversational speech, but scripted recordings as well as far field recordings are present.

The full language packs (all training data) from the collections shown in Table 1 were used in our experiments. More details about the characteristics of the languages can be found in [17]. The speech was force-aligned using our BABEL ASR system [18].

For SBN training, we used either one language, the first five languages denoted as **multi5**, or all 11 languages denoted as **multi11**.

For experiments where we needed a lot of training data for single language, we used Fisher English Training Part 1/2.

4. Experimental setup

4.1. LID training and evaluation corpora

We used NIST LRE 2009 database to evaluate our LID systems. To train our models (estimation of i-vectors and logistic regression training), we used the same data setup as in [19].

Our training data was taken from the following databases: Callfriend, Fisher English Part 1 and 2, Fisher Levantine Arabic, HKUST Mandarin, Mixer (data from NIST SRE 2004, 2005, 2006, 2008), Foreign Accented English, OGI-multilingual, OGI 22 languages, Voice of America radio broadcasts and development data for LRE 2005 and LRE 2007.

Three sets of datasets were defined for training. The first contains all the utterances in the databases for 54 languages and it is further denoted as *full54* (this set contains 79 thousand files and 2500 hours of speech). The second dataset is subset

⁴Collected by Appen, <http://www.appenbutlerhill.com>

Table 1: Training data used to train SBN networks (amounts of clean speech).

	Language	Dataset	# hours
CA	Cantonese	IARPA-babel101-v0.4c	65.0
PA	Pashto	IARPA-babel104b-v0.4aY	64.7
TU	Turkish	IARPA-babel105-v0.6	56.6
TA	Tagalog	IARPA-babel106-v0.2g	44.1
VI	Vietnamese	IARPA-babel107b-v0.7	53.2
AS	Assamese	IARPA-babel102b-v0.5a	46.7
BE	Bengali	IARPA-babel103b-v0.4b	53.6
HA	Haitian Creole	IARPA-babel201b-v0.2b	55.0
LA	Lao	IARPA-babel203b-v3.1a	71.6
TM	Tamil	IARPA-babel204b-v1.1b	72.7
ZU	Zulu	IARPA-babel206b-v0.1e	57.8
ML5	multi5	CA+PA+TU+TA+VI	283.6
ML11	multi11	all languages	641.0

of *full54* set and contains 23 target languages from NIST LRE 2009 and is denoted as *full23* (51 thousand files, 1550 hours). The third contains a maximum of 500 utterances for every language from *full23* set and it is further denoted *balanced* (9.8 thousand files, 360 hours). For training the UBM, the *balanced* dataset was used, for i-vector extractor, the *full54* dataset was used and for the multiclass logistic regression classifier we use *full23* set. The calibration and fusion was trained on the development dataset, which comprises data from all previous NIST LRE evaluations, OGI-multilingual, OGI 22 languages, Foreign Accented English, SpeechDat-East, Switch Board and Voice of America radio broadcasts.

4.2. LID system description

We based our experiments on the state-of-the-art acoustic i-vector system [20]. I-vectors provide an elegant way of reducing the large-dimensional variable-length input data (time sequence of features) to a small-fixed-dimensional feature vector while retaining most of the relevant information [21].

4.2.1. Feature extraction

For the multilingual SBN feature extraction, please refer to Section 2 and 3, where it is described in detail.

As the reference features, we use popular SDC features [4] with usual configuration 7-1-3-7, concatenated with 7 MFCC coefficients (including C0). The frame rate is 10 ms. Vocal Tract Length Normalization (VTLN) [22], Cepstral mean and variance normalization (CMVN) and RASTA filtering are applied before SDC.

4.2.2. Estimation of i-vectors, scoring and fusion

After feature extraction (either SDC or SBN), voice activity detection was performed by our Hungarian phoneme recognizer – we simply drop all frames labeled as silence or speaker noises.

Our i-vector extractor was trained in 5 iterations of jointly applying the Expectation Maximization (EM) algorithm and the Minimum Divergence (MD) step [23]. If not stated otherwise, sufficient statistics for both the i-vector extractor training and the i-vector estimation were collected using a 512 component GMM with diagonal covariance matrices and the i-vector dimensionality was set to 400.

We used regularized multiclass logistic regression (LR) [24] trained on our *full23* set to produce the scores. The i-vectors are transformed using within-class covariance normalization (WCCN) to make L2 regularization effective during logistic regression training.

Scores were calibrated and fused by another LR trained on the development set. For details on LR training, see [24].

5. Results

5.1. Multilinguality

We performed a set of experiments to see how multilingual training affects LID performance of SBN features. Intuitively, this should bring improvement, as more different speech units are used as targets during NN training, which should lead to the ability to better discriminate between languages. Moreover, it was already proven to be effective for ASR [9].

Firstly, we trained five monolingual systems for each language from multi5 set and one multilingual system using all data from this set, see top of Table 2. Note, that all monolingual systems for languages from multi5 set use less data (60h each) compared to multilingual system multi5 (280h) and multi11 (640h). We did not have more data for these languages, so to do fair comparison, we trained another monolingual network on random subset of Fisher English database of comparable length. We can see from Table 2, that the multi5 system outperforms by more than 10% relative this monolingual system (Fisher 250h). We also show a system trained on 60h subset of Fisher English, that is in turn comparable to other monolingual systems taken from multi5 set and is superior to them. Finally we show, that using whole Fisher (2000h) to train monolingual system still does not approach the performance of multilingual system.

Table 2: Comparison of systems based on monolingual and multilingual SBN features.

		$C_{avg}[\%]$		
Features		3 s	10 s	30 s
SDC	SDC56	17.20	6.72	3.34
1	Cantonese	12.67	4.52	2.45
2	Tagalog	12.43	4.66	2.41
3	Vietnamese	12.38	4.64	2.29
4	Turkish	11.46	4.14	2.17
5	Pashto	11.17	3.96	2.12
	English (Fisher 60h)	11.11	3.80	1.87
	English (Fisher 250h)	10.65	3.66	1.84
	English (Fisher 2000h)	10.22	3.42	1.72
ML5	multi5 (280h)	8.99	3.15	1.68
ML11	multi11 (640h)	8.58	2.93	1.58
F1	1+2+3+4+5	8.24	2.28	1.23
F2	1+2+3+4+5+ML5	7.81	2.07	1.14
F3	1+2+3+4+5+ML5+SDC	7.60	2.08	1.11

Table 2 also shows a score fusion of five monolingual systems from multi5 set (F1) and the same fusion with multi5 system included (F2). When comparing multilingual system and the fusion of all monolingual systems, we can see that the fusion is better mainly on long segments. On the other hand, the multi5 system is 5× smaller and faster than the fusion of 5 monolingual systems (even it uses the same amount of training data). The multi5 system is still complementary to the monolingual

systems and brings additional gain of 7% compared to the fusion F1.

We also tried adding baseline SDC system to this fusion to find out, how much of the complementary information still remains in the classical cepstral features. As we can see, this yields only minor improvement, mainly for 3 s condition.

5.2. Neural network training targets

We have explored what targets we should use for training the NN for SBN extraction. We have shown in [5] that the cross-word tied triphone states (shortly "triphones") are better targets than monophones. Table 3 shows similar comparison. We tried phoneme states (3 states per phoneme), monophones and triphones. For the monolingual Turkish network, we see that the triphones are again better targets compared to the monophones. Relative gain is similar to what we have seen in [5].

The second part of Table 3 shows results for multilingual SBN features. Overall the triphones as targets yield slightly better results, but training such NN is several times longer as the output layer has order of magnitude more output neurons. We decided to use phoneme states for our multilingual SBN features as it provides only slightly worse results with much smaller computation effort during the training.

Table 3: Experiments with different training targets.

Features	Targets	# Targ.	$C_{avg}[\%]$		
			3 s	10 s	30 s
Turkish	monophones	42	12.47	4.60	2.31
Turkish	phonestates	126	11.46	4.14	2.17
Turkish	triphones	3805	11.06	3.97	2.11
multi5	phonestates	1368	8.99	3.15	1.68
multi5	triphones	25270	9.22	3.02	1.61

5.3. Overall comparison

The overall results are shown in Table 4. Note that there are several systems of different sizes mixed together.

We are reporting also results for Phone Log-Likelihood Ratios (PLLR) features from [2] and Deep Bottleneck Features (DBF) from [6] for comparison. The authors of [6] used an approach similar to Parallel Phone Recognition followed by Language Modeling (PPRLM) [25] to obtain a multilingual system. They train several deep bottleneck networks for different languages and estimate i-vectors on top of bottleneck features (DBF43). The multilingual system is then obtained by combining these i-vectors, either by concatenating i-vectors (PDBF-TV2) or at score level.

Table 4 illustrates, how far we can get with the proposed approach. It is interesting to see the dependency of test condition on UBM covariance type (full or diagonal). For SDC features, we see improvement when using full covariance matrices across all conditions. This does not hold for multi11 features, where using full covariance helps only for short utterances. This can be also seen on fusions F1 (UBM with diagonal covariances) and F2 (UBM with full covariances).

From F1 and F2 system fusions, we see that combining our multilingual multi11 system with SDC-based system brings meaningful improvement. We obtain on average 20% relative improvement over multi11 system for both F1 and F2. This

Table 4: Final comparison of Multilingual SBN features. We show results also for bigger systems using 2048 component UBM with diagonal (D) or full (F) covariance matrices. We used 400 dimensional i-vectors in all cases.

Features	UBM	$C_{avg}[\%]$		
		3 s	10 s	30 s
1 SDC56	512D	17.20	6.72	3.34
2 SDC56	2048D	14.84	5.55	2.75
3 SDC56	2048F	14.35	5.26	2.51
4 SBN English (250h)	512D	10.65	3.66	1.84
5 SBN multi5	512D	8.99	3.15	1.68
6 SBN multi11	512D	8.58	2.93	1.58
7 SBN multi11	2048D	7.60	2.51	1.43
8 SBN multi11	2048F	6.75	2.34	1.43
F1 2+7	D	6.83	1.88	1.04
F2 3+8	F	5.99	1.81	1.04
PLLR+Proj.+PCA [2]	1024D	-	-	2.19
DBF43 Mandarin [6]	2048D	9.69	2.43	1.29
PDBF-TV2 [6]	2048D	7.87	2.05	1.16

does not conflict with our results from Section 5.1, where we have found, that adding SDC-based system to the fusion does not help. We think, that all the complementary information is already delivered by five monolingual systems in that case.

6. Conclusions

In this work, we applied multilingual training paradigm of SBN neural networks to extract linguistically rich features. This paper shows that features enriched in this way are more informative and better fitting for language identification task, which is demonstrated on the standard NIST LRE09 dataset. The results indicate that multilingual features give 16% relative gain for 3 s condition, 14% for 10 s and 8% for 30 s over the single best monolingual features. We also showed that such multilingual system is still complementary to the fusion of monolingual systems and brings additional 7% relative gain over all conditions. When we fuse SBN features with our MFCC-SDC baseline, we obtain one of the best reported results on this task.

7. Acknowledgements

This work was supported by the DARPA RATS Program under Contract No. HR0011-15-C-0038. The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense US Army Research Laboratory contract number W911NF-12-C-0013. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government.

The work was also supported by Czech Ministry of Interior project No. VG20132015129 "ZAOM" and IT4Innovations Centre of Excellence CZ.1.05/1.1.00/02.0070.

8. References

- [1] J. Ma *et al.*, “Improvements in language identification on the RATS noisy speech corpus,” in *Interspeech 2013*, Lyon, France, 2013.
- [2] 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,” *Signal Processing Letters, IEEE*, vol. 21, no. 9, pp. 1073–1077, Sept 2014.
- [3] K. Han and J. Pelecanos, “Frame-based phonotactic language identification,” in *Spoken Language Technology Workshop (SLT)*, Miami, Florida USA, 2012.
- [4] P. Torres-Carrasquillo, E. Singer, M. Kohler, R. Greene, D. Reynolds, and J. Deller, “Approaches to language identification using gaussian mixture models and shifted delta cepstral features,” in *ICSLP 2002*, Sep. 2002, pp. 89–92.
- [5] P. Matějka *et al.*, “Neural network bottleneck features for language identification,” in *IEEE Odyssey: The Speaker and Language Recognition Workshop*, Joensuu, Finland, 2014.
- [6] B. Jiang, Y. Song, S. Wei, J.-H. Liu, I. V. McLoughlin, and L.-R. Dai, “Deep bottleneck features for spoken language identification,” *PLoS ONE*, vol. 9, p. e100795, 7 2014. [Online]. Available: <http://dx.doi.org/10.1371/journal.pone.0100795>
- [7] I. Lopez-Moreno, J. Gonzalez-Dominguez, and O. Plchot, “Automatic language identification using deep neural networks,” in *ICASSP 2014*, Florence, Italy, 2014.
- [8] S. Scanzio, P. Laface, L. Fissore, R. Gemello, and F. Mana, “On the use of a multilingual neural network front-end,” in *Interspeech 2008*, 2008, pp. 2711–2714.
- [9] K. Veselý, M. Karafiát, F. Grézl, M. Janda, and E. Egorova, “The language-independent bottleneck features,” in *Proceedings of IEEE 2012 Workshop on Spoken Language Technology*, 2012, pp. 336–341.
- [10] F. Grézl, E. Egorova, and M. Karafiát, “Further investigation into multilingual training and adaptation of stacked bottle-neck neural network structure,” in *Spoken Language Technology Workshop (SLT)*, South Lake Tahoe, Nevada USA, 2014.
- [11] F. Grézl, M. Karafiát, and K. Veselý, “Adaptation of neural network feature extractor for new language,” in *Interspeech 2013*, Lyon, France, 2013.
- [12] D. Talkin, “A robust algorithm for pitch tracking (RAPT),” in *Speech Coding and Synthesis*, W. B. Kleijn and K. Paliwal, Eds. New York: Elsevier, 1995.
- [13] K. Laskowski and J. Edlund, “A Snack implementation and Tcl/Tk interface to the fundamental frequency variation spectrum algorithm,” in *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC’10)*, Valletta, Malta, may 2010.
- [14] M. Karafiát, F. Grézl, K. Veselý, M. Hannemann, I. Szóke, and J. Černocký, “BUT 2014 Babel system: Analysis of adaptation in NN based systems,” in *Interspeech 2014*, 2014, pp. 3002–3006.
- [15] F. Grézl, M. Karafiát, and L. Burget, “Investigation into bottleneck features for meeting speech recognition,” in *Interspeech 2009*, Brighton, GB, 2009.
- [16] K. Veselý, M. Karafiát, and F. Grézl, “Convolutional bottleneck network features for LVCSR,” in *ASRU 2011*, 2011, pp. 42–47.
- [17] M. Harper, “The BABEL program and low resource speech technology,” in *ASRU 2013*, Dec 2013.
- [18] M. Karafiát, F. Grézl, M. Hannemann, K. Veselý, and J. H. Černocký, “BUT BABEL System for Spontaneous Cantonese,” in *Interspeech 2013*, no. 8, Lyon, France, 2013, pp. 2589–2593.
- [19] Z. Jančík, O. Plchot, N. Brummer, L. Burget, O. Glembek, V. Hubeika, M. Karafiát, P. Matějka, T. Mikolov, A. Strasheim, and J. Černocký, “Data selection and calibration issues in automatic language recognition - investigation with BUT-AGNITIO NIST LRE 2009 system,” in *Proc. Odyssey 2010 - The Speaker and Language Recognition Workshop*, 2010, pp. 215–221.
- [20] D. G. Martínez, O. Plchot, L. Burget, O. Glembek, and P. Matějka, “Language recognition in ivectors space,” in *Interspeech 2011*, 2011, pp. 861–864.
- [21] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Front end factor analysis for speaker verification,” *IEEE Transactions on Audio, Speech and Language Processing*, 2010.
- [22] L. Welling, S. Kanthak, and H. Ney, “Improved methods for vocal tract normalization,” in *ICASSP*, vol. 2, Mar 1999, pp. 761–764 vol.2.
- [23] N. Brummer, “The EM algorithm and minimum divergence,” Agnitio Labs, Tech. Rep., Jan 2014. [Online]. Available: <http://niko.brummer.googlepages.com/EMandMINDIV.pdf>
- [24] O. Plchot, D. M. Sánchez, M. Soufifar, and L. Burget, “PLLR features in language recognition system for RATS,” in *Interspeech 2014*, 2014, pp. 3048–3051.
- [25] M. Zissman and E. Singer, “Automatic language identification of telephone speech messages using phoneme recognition and N-gram modeling,” in *ICASSP*, vol. i, Apr 1994, pp. I/305–I/308 vol.1.