INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

2016 BUT Babel system: Multilingual BLSTM acoustic model with i-vector based adaptation

Martin Karafiát, Murali Karthick Baskar, Pavel Matějka, Karel Veselý, František Grézl, Lukáš Burget and Jan "Honza" Černocký

Brno University of Technology, Speech@FIT and IT4I Center of Excellence, Brno, Czech Republic
{karafiat,baskar,grezl,iveselyk,matejkap,cernocky}@fit.vutbr.cz

Abstract

The paper provides an analysis of BUT automatic speech recognition systems (ASR) built for the 2016 IARPA Babel evaluation. The IARPA Babel program concentrates on building ASR system for many low resource languages, where only a limited amount of transcribed speech is available for each language. In such scenario, we found essential to train the ASR systems in a multilingual fashion. In this work, we report superior results obtained with pre-trained multilingual BLSTM acoustic models, where we used multi-task training with separate classification layer for each language. The results reported on three Babel Year 4 languages show over 3% absolute WER reductions obtained from such multilingual pre-training. Experiments with different input features show that the multilingual BLSTM performs the best with simple log-Mel-filter-bank outputs, which makes our previously successful multilingual stack bottleneck features with CMLLR adaptation obsolete. Finally, we experiment with different configurations of i-vector based speaker adaptation in the mono- and multi-lingual BLSTM architectures. This results in additional WER reductions over 1% absolute.

Index Terms: Automatic speech recognition, Multilingual neural networks, Bidirectional Long Short Term Memory, i-vector,

1. Introduction and prior work

Quick delivery of an automatic speech recognition (ASR) system for a new language is one of the challenges in the community. Such scenarios call not only for automated construction of systems, that have been carefully designed and crafted "by hand", but also for effective use of available resources. Without any question, the data collection and annotation are the most time- and money-consuming processes.

The recently finished IARPA Babel program focused on fast development of ASR systems, while the amount of per-language data was decreasing from year to year. The data from 24 low-resource languages were collected, which led to numerous multilingual experiments.

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 Technology Agency of the Czech Republic project No. TA04011311 "MINT", European Union's Horizon 2020 project No. 645523 BISON and Czech Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project "IT4Innovations excellence in science" - LQ1602.

For humans, borrowing the information from other sources when learning a new language is very natural. We all share the same vocal tract architecture and phonetic systems of languages overlap, therefore automatic systems should be able to have the universal and language-independent low-level components (feature extraction and partially also acoustic models), that would be built with various sources of data. In the past, we have verified that the multilingual pre-training for feature extraction [1] is an important technique, especially if limited amount of training data is available. We have also performed an analysis of combining semi-supervised and multi-lingual training of NN-based bottleneck feature extractors [2]. Also the hybrid DNN-HMM systems benefit from the multi-lingual training [3]. Recently in [4], we extended this idea to Bi-directional Long-Short Term Memory Recurrent Neural Networks (BLSTM-RNN).

For the adaptation of feed-forward DNN systems, we have witnessed an increased popularity of speaker-specific vectors. The most prominent are i-vectors, originally developed for speaker verification [5], which provide an elegant way of encoding a sequential input with arbitrary length to a single vector with fixed-dimension. The i-vector retains most of the speaker information, so the i-vectors found its way to the ASR field: At first as additional input features to discriminatively trained Region Dependent Linear Transform for GMMs [6], later as additional input features of a DNN [7, 8, 9], while they were also successfully used in robust ASR [10, 11, 12]. An alternative way of using i-vectors is to train a small adaptation network, which converts the i-vectors into offsets of input-features of the main DNN network [13]. However, it is not clear what is the best way to integrate the i-vectors in the BLSTM model, because all the previous works were with feed-forward DNNs.

Yet another approach to extract a fixed length speaker representation is to use the Sequence summarizing neural network (SSNN) trained together with the main DNN acoustic model [14].

In this paper, we focus on multi-lingual training. We experiment both with the multi-lingual feature extraction and multi-lingual acoustic modeling. Then, we also focus on the i-vector based speaker adaptation of multilingual BLSTM acoustic model.

2. Data

The IARPA Babel program data simulate a situation, in which the data for a new language are collected in a limited time. The data consists mainly of conversational telephone speech (CTS) but some scripted recordings and far field recordings are present too. During the 4-year project, extensive collection of 24 languages was created: **Year 1**: Cantonese (CA), Pashto (PA), Turkish (TU), Tagalog (TA), Vietnamese(VI). **Year 2**: Assamese (AS), Bengali (BE), Haitian Creole (HA), Lao (LA),

http://dx.doi.org/10.21437/Interspeech.2017-1775

Table 1: *Amounts of data used for the training.*

Y1 Langs.	CA	PA	TU	TA	VI		
Hours	65	65	57	44	53		
Y2 Langs.	AS	BE	HA	LA	ZU	Tam	
Hours	47	54	55	57	58	56	
Y3 Langs.	KU	CE	KA	TE	LI	TP	SW
Hours	37	38	40	38	41	26	34
Y4 Langs.	PA2	JA	IG	MO	DH	GU	AM
Hours	32	40	39	39	38	39	39
Non-Babel	LEV	FSH	MAN	SPA			
Hours	136	239	153	199			

Zulu (ZU), Tamil (Tam). **Year 3**: Kurdish (KU), Cebuano (CE), Kazakh (KA), Telugu (TE), Lithuanian (LI), TokPisin (TP), Swahili (SW). **Year 4**: Pashto progress set (about 40h subset of Year 1) (PA2), Javanese (JA), Igbo (IG), Mongolian (MO), Dholuo (DH), Guarani (GU), Amharic (AM), Georgian (not used in this work) (GE). In addition, the **Non-Babel** data were allowed for multilingual pre-training in the 4th year of the program: Levantine Arabic QT training data set 5 (LEV), Fisher English training speech parts 1,2 (limited to 250 hours) (FSH), Mandarin HKUST + Mandarin CallHome/CallFriend (MAN), Spanish Fisher + Spanish CallHome/CallFriend (SPA). The amounts of data can be found in table 1. Note, that the data sizes are summarized after trimming the silence to 150 ms on the edges of speech segments, according to a forced alignment. More details about Year 1–3 languages can be found in [1].

We limited the language model training corpus to the transcriptions of the training audio we received from the Babel program. Pronunciation dictionaries were not provided, so we relied on graphemic lexicons. Several data-sets based on packs from table 1 were generated, for the multi-lingual acoustic model and feature extractor training. We simulated a real situation with the data "growing" over time (see section 5). All the experiments were evaluated with Javanese (JA), Pashto (PA) and Amharic (AM), the languages are from the 4th year of the program.

3. System description

Our systems were built with a myriad of tool-kits: We used STK/HTK [15] toolkit[1] for feature extraction and CMLLR adaptation [16], Kaldi [17] was used for maximum likelihood (ML) Gaussian mixture model (GMM) training and baseline DNN acoustic model training. Finally, with CNTK [18] we trained our Stacked Bottle-Neck (SBN) networks [19] and BLSTM networks.

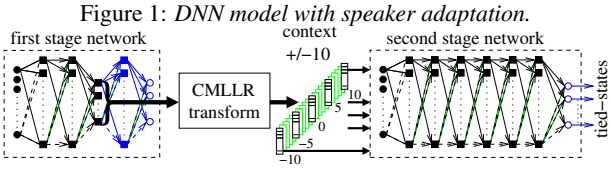
3.1. DNN system

The baseline DNN system is described in detail in [20]. The features are generated from Bottle-Neck (BN) feature extractor described later in section 3.3 and fed into DNN acoustic model, as shown in figure 1. For such architecture, we have shown in [21] that CMLLR adaptation of the bottleneck features improves the system performance. These CMLLR features will be further called as "BN-CMLLR"

The BN-CMLLR features are spliced in with the partially consecutive frame offsets (-10,-5:5,10) and mean normalized. For the experiments, we used DNNs with 6 hidden layers each containing 2048 sigmoidal neurons. The DNN system is pre-trained using restricted Boltzmann machine (RBM) [22]. This

[1]STK is BUT's variant of HTK: `http://speech.fit.vutbr.cz/software/hmm-toolkit-stk`

Figure 1: *DNN model with speaker adaptation.*



is followed by frame classification training (cross-entropy) with mini-batch stochastic gradient descent algorithm. The learning rate scheduling is based on relative improvement of the training objective (frame cross-entropy) on 10% held-out set. The input frames are randomized and grouped into mini-batches of 256 frames.

3.2. BLSTM systems

The latency-controlled BLSTM architecture [23] contains 3 bi-directional layers, for each direction there are 512 memory units and 300 dimensional projection layer as suggested in [24]. The training is done with truncated back-propagation through time (BPTT) algorithm [25]. Each update is based on $T_{bptt} = 20$ time-steps of recurrent forward-propagations and back-propagations.

3.3. Bottle-Neck feature extraction

We also used the Stacked Bottle-Neck (SBN) feature extraction [20]. It consists of two NN stages: The first one is reading short temporal context, then the bottleneck frames are spliced with offsets (-10,-5,0,5,10) and fed into the second NN reading the longer temporal information.

The first-stage bottle-neck NN input are 24 log-Mel-filterbank features concatenated with different pitch features: "BUT F0" has 2 coefficients (F0 and probability of voicing), "snack F0" is a single F0 estimate and "Kaldi F0" which has 3 coefficients (F0 normalized with a sliding window, probability of voicing and delta). Fundamental frequency variation (FFV) produces a 7 dimensional vector. The whole feature vector has 24+2+1+3+7=37 coefficients (see [20] for details on pitch features).

After a conversation-side mean subtraction, we apply a Hamming window and Discrete cosine transform to the feature trajectories spanning 11 frames. We retain 0^{th} to 5^{th} DCT coefficients for each of the original 37 features resulting in 37×6=222 coefficients at the first-stage NN input. These features are later also used independently for DNN systems, and will be called "11FBank_F0".

In this work, the first-stage NN has 4 hidden layers, each of 1500 sigmoid neurons except the 80-dimensional linear bottleneck [26] (3rd hidden layer). Then, after splicing of bottleneck features we have a second-stage NN with an architecture similar to the first-stage NN, except of BN layer with only 30 linear neurons. Both neural networks were trained jointly as suggested in [26] with the CNTK toolkit [18].

We extract the features from the 80 dimensional bottleneck of the first-stage network. These features are used in the DNN systems. In case of their mono-lingual training, they will be later called "BN1_Mono", if multilingually trained - "BN1_Multi" (see section 5.1.1). The features already contain CMLLR speaker adaptation.

4. Analysis of feature extraction

First, we were interested in optimal feature extraction for our DNN and BLSTM architectures. No speaker adaptation was

Table 2: *Comparison of %WER of monolingual feed-forward DNN and recurrent BLSTM system on top of monolingual non-adapted features.*

Language	Features	DNN	BLSTM
Javanese	11FBANK_F0	60.1	**54.0**
Javanese	BN1_Mono	57.4	55.4
Amharic	11FBANK_F0	48.4	**44.0**
Amharic	BN1_Mono	46.5	45.2
Pashto	11FBANK_F0	53.7	**48.7**
Pashto	BN1_Mono	52.0	51.3

used and the feature extractor was trained monolingually on the target language only. According to table 2, the BN feature extractor is beneficial for the feed-forward DNN based acoustic models. On contrary, for the recurrent BLSTM models, the BN features are malicious and the basic 11FBANK_F0 are giving better performance.

5. Multi-lingual experiments

5.1. Multilingual architectures

All multilingual architectures in this work were trained with a 'block-softmax' output layer, which consists of per-language softmaxes [27]. The training targets are the 'context-independent phoneme states', otherwise the size of the final layer would be excessively large.

We were interested in comparison of multilingual feature extraction with monolingual acoustic model (*Fea:BN1_Multi,AM:MonoL*) and training of whole system (joint training of feature extraction + acoustic model) in multilingual way followed by porting to target language (*AM:MultiL*). Such procedure can be described in the following steps: (1) the final multilingual layer (context-independent phoneme states for all languages) was stripped and replaced with a layer specific to target-language (tied-state triphones) with random initialization. (2) This new layer was trained for 8 epochs with a standard learning rate, while the rest of the NN was fixed. (3) Finally, the whole NN was fine-tuned with 10 epochs, the initial value of learning-rate schedule was set to 0.1 of the original value (resp. 0.5 for BLSTM).

The following architectures were built and tested:

- *Fea:BN1_Multi,AM:MonoL*: multi-lingually trained SBN architecture defined in section 3.3, where output from 1st-stage NN was used as a feature extraction for monolingual DNN system.
- *Fea:11FBank_F0,AM:MultiL,DNN*: joint architecture of feature extractor and DNN acoustic model (section 3.1). The first NN has 3 layers with 1500 neurons followed by a bottle-neck layer with 80 neurons. The BN features are spliced with offsets (-10, -5:5, 10) and followed by 6 layers with 2048 neurons each. The first NN (BN part) was initialized from Y1-Y3 feature extraction, the rest was RBM initialized.
- *Fea:11FBank_F0,AM:MultiL,BLSTM*: here the composite architecture of bottleneck network and DNN is replaced by 3 BLSTM layers (section 3.2) with 512 memory units trained directly on 11FBank_F0 features. The BN features were not found to be a suitable BLSTM input (see table 2).

5.1.1. Multilingual feature extractor - MultFE

Figure 2 presents the results[2] with multilingual feature extractor trained on data from various languages. The sets are denoted as "Mono" (target language only), "Y1" (all languages from Year 1), "Y1-2" (languages from Years 1 and 2) and so one. Note, that we excluded Pashto from Y1, Y1-2 and Y1-3 in order to simulate a scenario where no target language data is available for training of the feature extraction. On the contrary, Y1-4 contains all Pashto, Amharic and Javanese. In addition, Y1-4+nonBabel set contains also large non-Babel resources (Levantine Arabic, US English, Mandarin and Spanish). The acoustic-model DNNs were trained in the standard monolingual fashion and its last layer produced posterior probabilities of tied-states for HMM models.

[2] Already published in [4] but added here for sake of completeness

Figure 2: *DNN systems based on various multilingual NN features.*

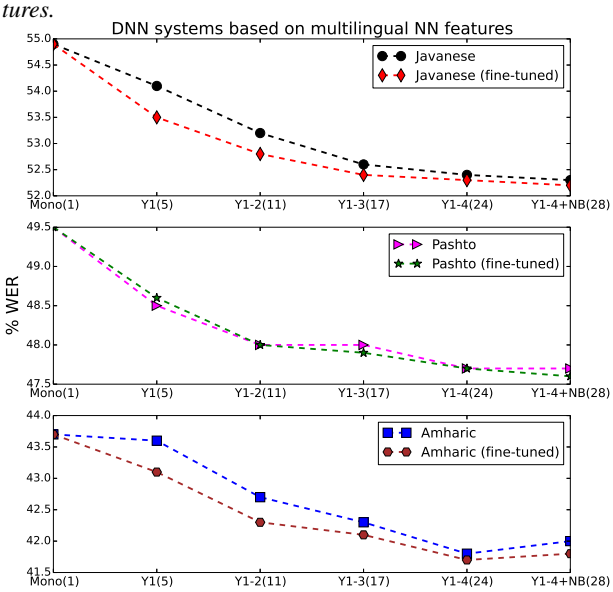


Figure 2 shows the important effect of number of languages for multilingual feature extraction. Here, the feature extraction was not tuned towards a particular target language and all Pashto, Javanese and Amharic systems use exactly the same feature extraction network, while the features were rotated by per-speaker CMLLR (see Fig. 1). The gains after adding more than 11 languages are minimal; probably the language diversity is already sufficient. Adding the non-Babel data does not help much although the amount of data is almost doubled compared to Y1-4. We have made a similar observation in our previous work [28], where we found that the language diversity was more important than the amount of data.

Figure 2 also presents the results of porting+fine-tuning of feature-extraction NN towards the specific target language. It brings only a small gain for DNN systems although with GMM systems, we found it crucial [1, 28]. It seems that the DNN acoustic model can better compensate this language mismatch, as the feature extraction is also done by NN. CMLLR was also employed here. The final features were obtained from all 28 languages, they were performing the best, and we will denote them as "BN1_Multi".

5.1.2. Multilingual acoustic model - AM:MultiL

In table 3 we compare feature extractions and multi-lingual pre-training of acoustic models 'MultiL'. The outcome is interesting and in our opinion, this shows that BLSTMs are better when pre-trained with large amounts of data. Recall that for the target language we have only 50 hours of training data. Another advantage is the simplicity of such systems and the speed of train-

Table 3: *Comparison of %WER of multilingual features (monolingual DNN) and multilingual acoustic models.*

Lang.	Feats.	CMLLR	AM	DNN	BLSTM
JA	BN1_Multi	no	MonoL	53.6	51.4
JA	BN1_Multi	yes	MonoL	52.2	50.5
JA	11FBank_F0	no	MultiL	53.6	**49.2**
AM	BN1_Multi	no	MonoL	43.4	41.8
AM	BN1_Multi	yes	MonoL	41.8	40.4
AM	11FBank_F0	no	MultiL	43.4	**39.8**
PA	BN1_Multi	no	MonoL	49.0	47.5
PA	BN1_Multi	yes	MonoL	47.6	46.5
PA	11FBank_F0	no	MultiL	49.3	**46.0**

ing; only fine-tuning needs to be done for the target language, and the feature extraction is without bottleneck network.

6. BLSTM adaptation experiments

At this point, our best system is based on Multilingual BLSTM without any speaker-adaptation. The classical speaker adaptation approaches such as CMLLR are not suitable for FBANK features, therefore we are interested in "injecting" the i-vectors into BLSTM models.

6.1. i-vector extraction

We used 19 MFCC coefficients + energy and their their delta and double delta coefficients which results in 60-dimensional feature vectors. The silence frames were removed according to VAD, after which we applied short-time (300 frame window) cepstral mean and variance normalization. The MFCC features were augmented with SBN features trained on Y1+Y2 languages. A gender-independent UBM was represented as GMM with 512 diagonal-covariance components. It was trained on the target language data. Finally, gender-independent i-vector extractor was trained (in 10 iterations of a joint Expectation Maximization and Minimum Divergence steps) on the same data set as the UBM. More details on i-vector extraction can be found in [29]. The results are reported with 100-dimensional i-vectors.

6.2. Analysis of speaker adaptation for Multilingual BLSTM

Typically, the low-dimensional vector-based adaptation involves concatenating input feature vectors with speaker-specific vector that is constant across the whole utterance [7].This approach is however not feasible with multilingually trained NNs, as re-training of the whole multi-lingual structure is not practical. Therefore, we experimented with two approaches to make the speaker adaptation feasible for pre-trained BLSTM:

- *Augmentation* - only the output of the last hidden layer (3th) is extended by speaker specific vector because the following language-specific output layer is anyway newly trained from random initialization.
- *SAT-DNN* - the i-vector is transformed by a small NN (ivec-NN) and added to the input of the main DNN acoustic model [30]. As the original architecture is "un-touched", the integration of ivectors with the pre-trained BLSTM is straightforward. In addition, we experimented with adding the i-vector to the outputs of others BLSTM layers.

According to results in table 4, the SAT-DNN approach is the best performing one. Interestingly, adding i-vectors to the output of the first hidden layer (1L) outperforms adding them to the input features - known as the best approach from DNN. It inspired us to repeat the analysis with monolingual BLSTM.

Table 4: *Multilingual BLSTM systems: adaptation by i-vectors %WER.*

	No	Augm.	SAT-DNN			
Language	adapt.	L3	Input	1L	2L	3L
JA	49.2	49.1	48.9	**48.6**	49.2	49.1
AM	39.8	39.8	39.8	**39.4**	**39.4**	39.7
PA	46.0	45.6	45.1	**44.7**	45.1	45.5

6.3. Analysis of BLSTM+ivec on monolingual systems

A detailed comparison of i-vector augmentation and addition by SAT-DNN on all layers could run only in monolingual systems due to the need to train from random initialization.

i-vector augmentation: according to table 5, the output of the first layer is the most optimal for i-vector augmentation in BLSTM systems. The 0.9-1.6% absolute improvement was reached on all tested languages.

SAT-DNN: both NNs (ivec-NN and BLSTM) were trained from random initialization. We also experimented with adding ivec-NN to already trained speaker-independent BLSTM as suggested in [30] but we did not observe any improvement over speaker independent BLSTM.

Table 6 shows that SAT-DNN outperforms the classical i-vector augmentation (table 5), but it is not possible to clearly determine the optimal connection layer.

Table 5: *Monolingual BLSTM systems: augmentation of i-vectors %WER.*

Language	No-Adapt	Input	1L	2L	3L
Javanese	54.0	53.3	**52.5**	53.1	54.1
Amharic	44.0	44.8	**42.4**	43.4	44.5
Pashto	48.7	47.6	**47.8**	49.2	48.7

Table 6: *Monolingual BLSTM systems: SAT-DNN %WER.*

Language	No-Adapt	Input	1L	2L	3L
Javanese	54.0	**52.4**	53.1	53.0	53.2
Amharic	44.0	43.1	**42.2**	42.9	42.9
Pashto	48.7	47.6	47.4	**47.1**	48.4

7. Conclusion

This paper concentrates on multi-lingual training of both DNN-based features and acoustic models as well as adaptation of BLSTMs with i-vectors.

We have shown clear advantage of multi-lingual training of acoustic models and features in low-resource scenarios. SBN feature extraction trained in multi-lingual way is an elegant way to produce high-quality features and obtain a good system trained on target data only. However, BLSTM acoustic models trained in multi-lingual way and fine-tuned towards the target language provide better performance with simpler "FBANK+pitch" features at the input.

We experimented with i-vector based BLSTM adaptation and found that BLSTM's middle layers are more suitable for such adaptation than input features which are used traditionally.

8. References

[1] F. Grézl and M. Karafiát, "Adapting multilingual neural network hierarchy to a new language," in *Proceedings of the 4th International Workshop on Spoken Language Technologies for Under-resourced Languages SLTU-2014. St. Petersburg, Russia, 2014*. International Speech Communication Association, 2014, pp. 39–45.

[2] F. Grezl and M. Karafiat, "Combination of multilingual and semi-supervised training for under-resourced languages," in *Proceedings of Interspeech 2014*, Singapore, 2014, pp. 820–824.

[3] A. Ghoshal, P. Swietojanski, and S. Renals, "Multilingual training of deep neural networks." in *ICASSP*. IEEE, 2013, pp. 7319–7323. [Online]. Available: http://dblp.uni-trier.de/db/conf/icassp/icassp2013.html#GhoshalSR13

[4] M. Karafiát, K. M. Baskar, P. Matějka, K. Veselý, F. Grézl, and J. Černocký, "Multilingual blstm and speaker-specific vector adaptation in 2016 but babel system," in *Proceedings of SLT 2016*. IEEE Signal Processing Society, 2016, pp. 637–643.

[5] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 19, no. 4, pp. 788–798, 2011. [Online]. Available: http://dx.doi.org/10.1109/TASL.2010.2064307

[6] M. Karafiát, L. Burget, P. Matějka, O. Glembek, and J. Černocký, "ivector-based discriminative adaptation for automatic speech recognition," in *Proceedings of ASRU 2011*. IEEE Signal Processing Society, 2011, pp. 152–157. [Online]. Available: http://www.fit.vutbr.cz/research/view_pub.php?id=9762

[7] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 55–59.

[8] M. Rouvier and B. Favre, "Speaker adaptation of DNN-based ASR with i-vectors: does it actually adapt models to speakers?" in *Proceedings of Interspeech*, 2014.

[9] V. Peddinti, G. Chen, D. Povey, and S. Khudanpur, "Reverberation robust acoustic modeling using i-vectors with time delay neural networks," in *Proceedings of Interspeech*, 2015.

[10] P. Karanasou, Y. Wang, M. J. Gales, and P. C. Woodland, "Adaptation of deep neural network acoustic models using factorised i-vectors," in *Proceedings of Interspeech*, 2014.

[11] C. Yu, A. Ogawa, M. Delcroix, T. Yoshioka, T. Nakatani, and J. H. Hansen, "Robust i-vector extraction for neural network adaptation in noisy environment," in *Proceefings of Interspeech*, 2015.

[12] S. Ganapathy, S. Thomas, D. Dimitriadis, and S. Rennie, "Investigating factor analysis features for deep neural networks in noisy speech recognition," in *Proceedings of Interspeech*, 2015.

[13] Y. Miao, L. Jiang, H. Zhang, and F. Metze, "Improvements to speaker adaptive training of deep neural networks," in *Proceedings of SLT*, 2014.

[14] K. Veselý, S. Watanabe, K. Žmolíková, M. Karafiát, L. Burget, and J. Černocký, "Sequence summarizing neural network for speaker adaptation," in *Proceedings of the 41th IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE Signal Processing Society, 2016, pp. 5315–5319. [Online]. Available: http://www.fit.vutbr.cz/research/view_pub.php?id=11145

[15] S. Young, J. Jansen, J. Odell, D. Ollason, and P. Woodland, *The HTK book*. Cambridge, UK: Entropics Cambridge Research Lab., 2002.

[16] M. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," 1997. [Online]. Available: citeseer.ist.psu.edu/gales97maximum.html

[17] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, Dec. 2011, iEEE Catalog No.: CFP11SRW-USB.

[18] A. Agarwal, E. Akchurin, C. Basoglu, G. Chen, S. Cyphers, J. Droppo, A. Eversole, B. Guenter, M. Hillebrand, R. Hoens, X. Huang, Z. Huang, V. Ivanov, A. Kamenev, P. Kranen, O. Kuchaiev, W. Manousek, A. May, B. Mitra, O. Nano, G. Navarro, A. Orlov, M. Padmilac, H. Parthasarathi, B. Peng, A. Reznichenko, F. Seide, M. L. Seltzer, M. Slaney, A. Stolcke, Y. Wang, H. Wang, K. Yao, D. Yu, Y. Zhang, and G. Zweig, "An introduction to computational networks and the computational network toolkit," Tech. Rep. MSR-TR-2014-112, August 2014. [Online]. Available: http://research.microsoft.com/apps/pubs/default.aspx?id=226641

[19] F. Grezl, M. Karafiat, and L. Burget, "Investigation into bottleneck features for meeting speech recognition," in *Proc. Interspeech 2009*, 2009, pp. 2947–2950.

[20] M. Karafiát, F. Grézl, M. Hannemann, K. Veselý, I. Szoke, and J. H. Černocký, "BUT 2014 Babel system: Analysis of adaptation in NN based systems," in *Proceedings of Interspeech 2014*. Singapure: IEEE, September 2014.

[21] M. Karafiát, F. Grézl, M. Hannemann, and J. H. Černocký, "BUT neural network features for spontaneous vietnamese in BABEL," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. Florence, Italy: IEEE, May 2014.

[22] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, pp. 1527–1554, 2006.

[23] Y. Zhang, G. Chen, D. Yu, K. Yao, S. Khudanpur, and J. R. Glass, "Highway long short-term memory RNNS for distant speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*, 2016, pp. 5755–5759.

[24] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14-18, 2014*, 2014, pp. 338–342. [Online]. Available: http://www.isca-speech.org/archive/interspeech_2014/i14_0338.html

[25] R. J. Williams and J. Peng, "An efficient gradient-based algorithm for on-line training of recurrent network trajectories," *Neural Computation*, vol. 2, pp. 490–501, 1990.

[26] K. Veselý, M. Karafiát, and F. Grézl, "Convolutive bottleneck network features for LVCSR," in *Proceedings of ASRU 2011*, 2011, pp. 42–47.

[27] 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*. IEEE Signal Processing Society, 2012, pp. 336–341.

[28] F. Grézl, E. Egorova, and M. Karafiát, "Further investigation into multilingual training and adaptation of stacked bottle-neck neural network structure," in *Proceedings of 2014 Spoken Language Technology Workshop*. IEEE Signal Processing Society, 2014, pp. 48–53. [Online]. Available: http://www.fit.vutbr.cz/research/view_pub.php?id=10798

[29] P. Matějka, O. Glembek, O. Novotný, O. Plchot, F. Grézl, L. Burget, and J. Černocký, "Analysis of dnn approaches to speaker identification," in *Proceedings of the 41th IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE Signal Processing Society, 2016, pp. 5100–5104. [Online]. Available: http://www.fit.vutbr.cz/research/view_pub.php?id=11140

[30] Y. Miao, H. Zhang, and F. Metze, "Speaker adaptive training of deep neural network acoustic models using i-vectors," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 23, no. 11, pp. 1938–1949, Nov. 2015.