MULTILINGUAL BLSTM AND SPEAKER-SPECIFIC VECTOR ADAPTATION IN 2016 BUT BABEL SYSTEM

Martin Karafiát, Murali Karthick Baskar, Pavel Matějka, Karel Veselý, František Grézl 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

This paper provides an extensive summary of BUT 2016 system for the last IARPA Babel evaluations. It concentrates on multi-lingual training of both deep neural network (DNN)-based feature extraction and acoustic models including multilingual training of bidirectional Long Short Term memory networks. Next, two low-dimensional vector approaches to speaker adaptation are investigated: i-vectors and sequence-summarizing neural networks (SSNN). The results provided on three Babel Year 4 languages show clear advantage of both approaches in case limited amount of training data is available. The time necessary for the development of a new system is addressed too, as some of the investigated techniques do not require extensive re-training of the whole system.

***Index Terms*—** Automatic speech recognition, Multilingual neural networks, Bidirectional Long Short Term Memory, i-vectors, Sequence Summarizing Neural Networks.

1. INTRODUCTION

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" so far, but also for effective use of available resources, as, without any question, the data collection and annotation are the most time- and money-consuming procedures.

The IARPA Babel program that is nearing its end aims at fast development of ASR systems with decreasing amounts of target language data. This paper describes our system built for the final, 2016, Babel evaluations and concentrates on two main issues:

Multi-lingual experiments for feature extraction and acoustic modeling. For humans, borrowing the information from other sources when trying to learn 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 low-level components (feature extraction and partially also acoustic models) built and trained on various sources of data. In past, we have verified [1] that multilingual pre-training for feature extraction is an important technique especially if not-enough training data is available. We have also performed an analysis of combining semi-supervised training and multilingual approaches in feature domain [2], and a recent work on multilingual DNN acoustics models also shows significant gains with adding more languages into acoustic model training [3]. We are extending this work into currently very popular Long-Short Term Memory Recurrent Neural Networks (LSTM-RNN) including their bi-directional variant (BLSTM). Our work includes also multi-lingual training of feature transformations, namely Region Dependent Transforms (RDT) [4].

New approaches to environment and speaker adaptation. The environment adaptation addressing the mismatch between telephone and far-field data was done by Weighted Prediction Error (WPE) [5, 6] that has been recently shown to greatly improve ASR in reverberant conditions for several tasks [7, 8]. As far as we know, this is the first successful use of this technique in Babel framework. In speaker adaptation, we are not neglecting classical approaches, such as Constrained Maximum Likelihood Linear Regression (CMLLR) [9] adapted for the NN-based features, but our focus is on speaker adaptation based on low-dimensional vectors. We have investigated into i-vectors [10] popular in the speaker recognition community and compared them with recently introduced sequence-summarizing NNs [11].

The paper is structured as follows: section 2 provides an overview of the data. Section 3 and 4 discusses the existing methods for multilingual feature extraction and domain or speaker adaptation. In section 5, ASR systems based on GMM, DNN, LSTM and BLSTM are described, Section 6 experimentally analyses the multi-lingual approaches to feature extraction and acoustic model training. In section 7.2, we present the results of domain and speaker adaptation techniques and conclude in section 8.

2. DATA

The IARPA Babel program data simulates a case of what one could collect in limited time from a completely new language. The data consists mainly of conversational telephone speech (CTS) but some scripted recordings and far field recordings are present too. Table 1 presents the details of languages used in this work sorted by years of BABEL program. The amounts of data can be found in table 2. Note, that the data sizes are summarized after limiting the silence in all audio files to 150 ms on the edges of voice segments by forced

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" and Czech Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project "IT4Innovations excellence in science - LQ1602"

Year 1 (Y1)		
Cantonese	IARPA-babel101-v0.4c	CA
Pashto	IARPA-babel104b-v0.4aY	PA
Turkish	IARPA-babel105-v0.6	TU
Tagalog	IARPA-babel106-v0.2g	TA
Vietnamese	IARPA-babel107b-v0.7	VI
Year 2 (Y2)		
Assamese	IARPA-babel102b-v0.5a	AS
Bengali	IARPA-babel103b-v0.4b	BE
Haitian Creole	IARPA-babel201b-v0.2b	HA
Lao	IARPA-babel203b-v3.1a	LA
Zulu	IARPA-babel206b-v0.1e	ZU
Tamil	IARPA-babel204b-v1.1b	Tam
Year 3 (Y3)		
Kurdish	IARPA-babel205b-v1.0a	KU
Cebuano	IARPA-babel301b-v2.0b	CE
Kazach	IARPA-babel302b-v1.0a	KA
Telugu	IARPA-babel303b-v1.0a	TE
Lithuanian	IARPA-babel304b-v1.0b	LI
TokPisin	IARPA-babel207b-v1.0c	TP
Swahili	IARPA-babel202b-v1.0d	SW
Year 4 (Y4)		
Pashto	see Year 1 – progress set	PA2
Javanese	IARPA-babel402b-v1.0b	JA
Igbo	IARPA-babel306b-v2.0c	IG
Mongolian	IARPA-babel401b-v2.0b	MO
Dholuo	IARPA-babel403b-v1.0b	DH
Guarani	IARPA-babel305b-v1.0b	GU
Amharic	IARPA-babel307b-v1.0b	AM
Non-Babel		
Levantine Arabic QT training data set 5		LEV
Fisher English training speech part 1,2 limited to 250 hours		FSH
Mandarin HKUST + Mandarin CallHome/CallFriend		MAN
Spanish Fisher + Spanish CallHome/CallFriend		SPA

Table 1. Languages used.

alignment. More details about Year 1–3 languages can be found in [1].

On contrary to the previous Babel evaluations, where all development for given language was restricted to the language pack of that particular language (for full language pack condition). In Year 4, multilingual training and web text data collection were allowed. We have not worked on the language models (LM) and limited their training data to the respective language packs. Pronunciation dictionaries were not provided and participants had to rely on graphemes in all conditions. However, for multilingual acoustic models and feature extractor training, several data-sets based on packs from table 2 were generated, simulating a real situation with the data "growing" over time. As the target languages, Year's 4 Javanese, Pashto and Amharic were chosen.

3. MULTILINGUAL FEATURE EXTRACTION

3.1. Stacked Bottle-Neck feature extraction

The original idea of Stacked Bottle-Neck feature extraction is described in [12]. The scheme (see Fig. 1) consists of two NN stages: The first one is reading short temporal context, its output is stacked, down-sampled, and fed into the second NN reading longer temporal information.

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

Table 2. Amount of data used for the training.

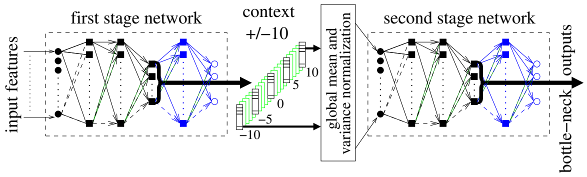

Fig. 1. Stacked Bottle-Neck Neural Network feature extraction.

The first stage bottle-neck NN input features are 24 log Mel filter bank outputs concatenated with different fundamental frequency features: "BUT F0" has 2 coefficients (F0 and probability of voicing), "snack F0" is just a single F0 estimate and "Kaldi F0" has 3 coefficients (Normalized F0 across sliding window, probability of voicing and delta). Fundamental frequency variation (FFV) produces a 7 dimensional vector. Therefore, the whole feature vector has 24+2+1+3+7=37 coefficients (see [12] for details on fundamental frequency features).

Conversation-side based mean subtraction is applied and 11 consecutive frames are stacked. Hamming window followed by discrete cosine transform (DCT) retaining 0^{th} to 5^{th} coefficients are applied on the time trajectory of each parameter resulting in 37×6=222 coefficients at the first-stage NN input. These features are later also used independently for DNN systems (section 6.3), and will be called "11FBank_F0".

In this work, the first-stage NN has 4 hidden layers with 1500 units in each except the bottle-neck (BN) one. The BN layer has 80 neurons. The neurons in the BN layer have linear activations as found optimal in [13]. 21 consecutive frames from the first-stage NN are stacked, down-sampled (each 5 frame is taken) and fed into the second-stage NN with an architecture similar to the first-stage NN, except of BN layer with only 30 neurons. Both neural networks were trained jointly as suggested in [13] in CNTK toolkit [14].

3.2. Multi-lingual Region-Dependent Transforms (RDT)

RDT [15] is a popular transform allowing for non-linear warping of the acoustic space to suit better GMM-HMM acoustic models. In our recent work [4], we investigated ways to train RDT in multi-lingual fashion — the statistics necessary to update the RDT model are collected on all target languages, averaged, and a single RDT transform can then serve to transform features for all languages, including unseen ones.

Here, multilingual RDT was used to fuse PLP features with multi-lingual BN features trained on 17 languages (see the next section). The multilingual RDT was trained on Y1-Y3 data (containing 17 languages excluding Pashto due to duplicity with Y4 data). These will be later referred as “MultRDT” features.

4. DOMAIN AND SPEAKER ADAPTATIONS

4.1. WPE-based de-reverberation

Most of Babel data is CTS, but about 10% in Year 3-4 was acquired by far-field microphones. For these, reverberation is responsible for non-stationary distortions that are correlated with the speech signal. Reverberation cannot be suppressed using conventional noise reduction approaches. Therefore, we used the weighted-prediction error (WPE) de-reverberation method [5, 6] that was shown to greatly improve ASR in reverberant conditions for several tasks [7, 8].

WPE is based on long-term multi-channel linear prediction (LP), but introduces modifications to conventional LP to make it effective for de-reverberation: speech is modeled with a short-term Gaussian distribution with time-varying variance [16], and a short time delay is introduced in the LP filters to prevent the equalization of the speech production [17]. Note that the WPE algorithm does not require a pre-trained model of speech. The ASR system (feature extraction and acoustic modeling) does not need to be re-trained.

4.2. Speaker adaptation techniques

4.2.1. i-vectors

The i-vectors provide an elegant way of reducing large-dimensional sequential input data to a small- and fixed-dimensional feature vector while retaining most of the information relevant for speaker recognition [10].

We used 19 Mel-frequency cepstral coefficients (MFCC) + energy augmented with their delta and double delta coefficients, resulting in 60-dimensional feature vectors. The silence frames were removed according to VAD, after which we applied short-time (300 frames) 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 diagonal-covariance 512-component GMM and it was trained on target language data. The variance flooring was used in each iteration of EM algorithm during the UBM training.

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 [18]. The results are reported with 100-dimensional i-vectors.

4.2.2. Sequence summarizing neural network

SSNN is a new DNN adaptation technique [11], producing a fixed-length ‘summary vector’ per speaker or per-utterance. The “summary vector” is obtained by enclosing a sequence-averaging operation into the last component of the SSNN. The “summary vector” is then appended to the input of the main network (acoustic model), and both networks are trained together, while the gradients for SSNN are calculated by back-propagating through the main network, see figure 2. Both networks are trained to optimize a single loss function.

The SSNN can be either initialized randomly (as we did in [11]), or it can be pre-trained as per-frame classifier of speakers (as was done in [19, 20]). After the network learns to classify speakers, we can extract the ‘summary vectors’ by averaging the signals from the last hidden layer over time.

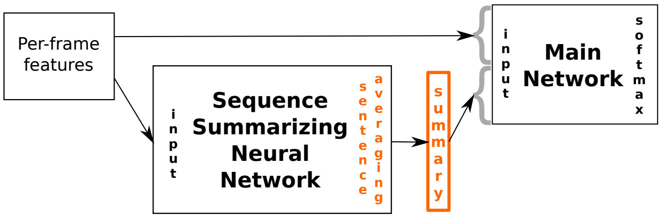

Fig. 2. *Topology of the main-network with “sequence summary” input. The summary is computed by SSNN with sentence-averaging on the output.*

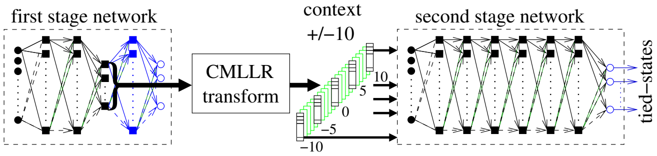

Fig. 3. *DNN model with speaker adaptation.*

5. ASR SYSTEMS

Our systems were built with a variety of software and tools: STK/HTK [21] toolkit[1] was used for feature extraction and CMLLR adaptation. Kaldi [22] was used for maximum likelihood (ML) Gaussian mixture model (GMM) training and baseline DNN acoustic model training. CNTK [14] served for most of the advanced NN training (SBN, LSTM and BLSTM).

5.1. GMM system

The GMM acoustic models based on cross-word tied-states were trained from scratch using standard ML algorithm. Initial baseline models were trained on 13-dimensional PLP (including C0) extended with 3-dimensional Kaldi pitch features [23]. We applied per-speaker mean/variance normalization, spliced vectors by +/- 4 frames and then projected down to 40 dimensions using Linear Discriminant Analysis (LDA). Finally, the features were rotated by single feature-space maximum likelihood linear regression (fMLLR) [25] transform estimated per speaker. The baseline GMM system with 4000 cross-word triphone tied states and 7 Gaussians per state was used to prepare baseline HLDA+fMLLR features. The state-alignments obtained with GMM systems were further used for DNN training.

5.2. DNN system

The baseline DNN system is described in detail in [12]. The first stage is identical to the SBN feature extraction described in section 3.1, with the second stage NN replaced directly with the DNN acoustic model, see figure 3. Note, in the past we also experimented with using second stage NN features but not gain was observed.

The SBN feature extractor was trained first and the features only from the first stage NN were generated. For such architecture, we have shown in [9] that CMLLR adaptation improves the system performance. These features will be further called “BN-CMLLR”

The BN-CMLLR features are spliced in sequence (-10,-5:5,10) and mean normalized. For the experiments, we used DNNs with 6 hidden layers each containing 2048 sigmoidal neurons. The

[1] STK is BUT’s variant of HTK, however not properly documented, see `http://speech.fit.vutbr.cz/software/hmm-toolkit-stk` with care.

DNN system is pre-trained using restricted Boltzmann machine (RBM) [26]. This is followed by frame classification training (cross-entropy) using 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, each of size 256.

5.3. LSTM and BLSTM systems

"Highway" Long-Short Term Memory system is built according to recipes presented in [28]. We did not observe any significant gain by using highway architecture in contrast to the original one, but we found this architecture more stable to train.

The LSTM architecture contains 3 layers with 1024 memory units and a projection layer with 512 neurons as suggested in [29]. The training employs the truncated back-propagation through time (BPTT) to update the model parameters [30]. We use a fixed time step T_{bptt} (e.g. 20) to forward-propagate the activations and backward-propagate the gradients.

The latency-controlled BLSTM [28] architecture contains 3 hidden layers in both directions each with 512 memory units and 300 neurons in the projection layer.

5.4. Results of mono-lingual baselines

The results of purely mono-lingual GMM and DNN systems are in the left columns of Tables 3 and 4. All results are given as word error rates (WER) in %. In the DNN systems, both networks were trained on the target language only.

6. MULTI-LINGUAL EXPERIMENTS

6.1. Analysis of multilingual RDT features

This section showcases the experiments on GMM and DNN systems using MultRDT (see section 3.2) features. Table 3 shows about 10% absolute gain by training simple GMM systems on MultRDT features — note that these performances almost reach the ones obtained by DNNs. Although looking similar, table 4 with results of DNN systems presents very different results: here, both neural networks are still trained on the target language only with standard architecture (11FBANK_F0→SBN→CMLLR→DNN); the only thing that changes is the initial alignment of frames obtained with a GMM system with either plain PLPs (left column) or MultRDT features (right column). Although DNN systems are not particularly sensitive to this initial alignment, there is still about 1% absolute improvement we can obtain by providing the best initial alignment. Note, that we also experimented with training of DNN on top of MultRDT features but no gain over multilingual SBN was observed.

Language	PLP	MultRDT
Javanese	66.4	**55.9**
Amharic	56.2	**46.2**
Pashto	61.1	**51.2**

Table 3. GMM systems - PLP vs. MultRDT features.

6.2. Analysis of multilingual SBN features

All multilingual architectures in this work were trained with the final softmax layer – – split into several blocks. Each block accommodates training targets from one language [31]. Context-independent phoneme states were used as the training targets for the feature-extraction NN, otherwise the size of the final layer would be prohibitive.

Language Initial GMM alignment	Mono-ling. SBN PLP	Mono-ling. SBN MultRDT
Javanese	56.1	**54.9**
Amharic	45.0	**43.7**
Pashto	50.7	**49.5**

Table 4. Comparison of % word error rate (WER) for Mono-lingual DNNs trained using GMM alignment: obtained with PLP vs. MultRDT features. Note that on contrary to table 3, the features are mono-lingual SBN, only the initial alignment differs.

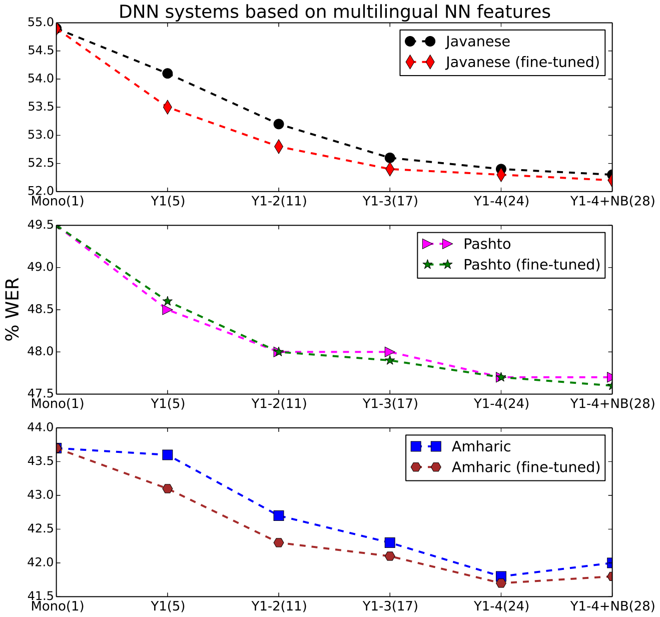

Fig. 4. DNN systems based on various multilingual NN features.

The feature-extraction NNs were trained on data from various languages: in figure 4, 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 data is available for training of the feature extraction. On 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 NN was trained in the standard mono-lingual fashion and its last layer produced posterior probabilities of tied-states for HMM models.

Figure 4 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. The CMLLR (see Fig. 3) is used. The gains after adding more than 11 languages are minimal; probably the language variety is already sufficient. Adding of non-Babel data almost does not help although the amount of data is almost twice compared to Y1-4. We have made a similar observation in our previous work [32] where we found the language diversity was more important than the amount of data.

Figure 4 also presents the results in case of fine-tuning of feature-extraction towards the particular target language. First, the last layer of multi-lingual NN is removed, initialized randomly and trained to discriminate the phonemes of the target language (the rest of the NN is fixed). Then, the whole NN is re-trained with a small learning rate (0.1 of the original one). Such fine-tuning brings only small gain for DNN systems although in GMM systems, we found it crucial [1, 32]. It seems than a DNN acoustic model can cope with this language mismatch as the feature extraction is also done by NN. CMLLR was also employed here. The final features obtained on all 28 languages — "Mult28_CMLLR" — were performing the best (in figure 4), therefore they were used for experiments with other architectures.

6.2.1. LSTM and BLSTM

Table 5 presents the results for all architectures built on top of "Mult28_CMLLR" features. The acoustic models were still trained on the target language only. The BLSTMs provide the best results as expected. In all experiments, we also tested the features with CMLLR off (as it is often claimed that more powerful acoustic models can work with simpler and less adapted features), however, CMLLR was found useful even for the most powerful BLSTM, providing 1% absolute improvement.

Language	CMLLR	DNN	LSTM	BLSTM
Javanese	no	53.6	53.1	51.4
Javanese	yes	52.2	52.1	**50.5**
Amharic	no	43.4	43.8	41.8
Amharic	yes	41.8	42.1	**40.4**
Pashto	no	49.0	49.3	47.5
Pashto	yes	47.6	47.7	**46.5**

Table 5. Comparison of %WER of monolingual DNN, LSTM and BLSTM based system on top of multilingual features.

Language	Features	DNN	LSTM	BLSTM
Javanese	11FBANK_F0	60.1	55.1	**54.4**
Javanese	SBN_Mono	57.4	56.9	55.4
Amharic	11FBANK_F0	48.4	44.9	**44.0**
Amharic	SBN_Mono	46.5	46.1	45.2
Pashto	11FBANK_F0	53.7	50.7	**49.3**
Pashto	SBN_Mono	52.0	52.4	51.3

Table 6. Comparison of %WER of monolingual DNN, LSTM and BLSTM based system on top of monolingual non-adapted features.

6.3. Analysis of multilingual acoustic models

Next, we were interested in training not only the feature extraction, but the whole architecture in multilingual fashion. The input of all architectures were the 11FBank_F0 features defined in section 3.1 — direct filter bank outputs, normalized, stacked and post-processed by DCT plus f0 features. The output layer of acoustic model NN was based on multilingual softmax the same way as for feature extractor training. The following architectures were built and tested:

- *DNN* is similar to feature extractor + acoustic model: the first NN has 3 layers with 1500 neurons followed by a bottle-neck layer with 80 neurons. The BN features are stacked in context [-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.

Language	DNN	LSTM	BLSTM
Javanese	53.6	51.8	**49.2**
Amharic	43.4	42.1	**39.8**
Pashto	49.3	47.8	**46.0**

Table 7. DNN systems: Multilingual architectures %WER.

- *LSTM* is the same as above (3 LSTM layers with 1024 memory units) but the system is trained directly on 11FBank_F0 features as the BN features were not found advantageous, see table 6.
- *BLSTM* - is the same as above (3 BLSTM layers with 512 memory units) but the system is trained directly on 11FBank_F0 features as simmilarly to LSTM, BN features did not help.

The multilingual acoustic models were built on all 28 languages, except of BLSTM where only 24 Babel languages (Y1-4) were used in order to save computation time with expected tiny loss of performance (predicted from figure 4). To train the target language system, the procedure was similar to crafting the multilingual features (section 6.2):

1. the final multilingual layer (context-independent phones for all languages) was stripped and replaced with target-language specific layer (tied-state triphones) with random initialization.
2. This new layer was trained with standard learning rate by 8 epochs while the rest of the NN was fixed.
3. Finally, 10 epochs of fine-tuning the whole NN to the target language were run, with 0.1 of the original learning rate (resp. 0.5 for BLSTM) used as the starting point for learning rate scheduler.

Table 7 shows 0.4-0.7% absolute improvement of multilingual BLSTM systems over the same architecture trained on CMLLR-adapted multilingual BN features (table 5). The outcome is quite interesting and in our opinion, it is related to the impossibility to pre-train complex LSTM/BLSTM systems. When trained on only 50 h of training data, their performances are not as good as when they are initialized on huge amount of data from many languages. Another advantage is the simplicity of such systems and the speed of training - only fine-tuning needs to be done into the target language, with standard feature extraction.

7. EXPERIMENTS ON ADAPTATION TECHNIQUES

7.1. Analysis of WPE speech enhancement

We have investigated the impact of WPE on far-field data in Javanese and Amharic with the best performing BLSTM system. Pashto was not included as it did not contain far-field data. Table 8 presents over 2.5% absolute improvement by application of WPE on this data. As this type covers only small portion of the development set (about 10%), the overall improvement is only 0.4%. Application of WPE also on training far-field data helps only on Amharic but in general we found it useful on other languages not reported in this paper, therefore we stick with using this technique in further experiments.

Language	WPE	Overall WER[%]	Far-field WER[%]
Javanese	no	49.2	66.9
Javanese	dev	**48.8**	**64.5**
Javanese	train+dev	49.1	66.0
Amharic	no	39.8	57.1
Amharic	dev	39.4	53.6
Amharic	train+dev	**39.3**	**53.1**

Table 8. DNN systems: Effect of WPE de-reverberation on far-field channels %WER.

	No adapt		Adapt 2L		
Language	1L	2L	ivec	d-vec	ivec+dvec
Javanese	49.1	48.8	**48.3**	48.6	48.3
Amharic	39.3	39.3	**39.0**	39.3	39.1
Pashto	46.0	45.7	45.5	**45.2**	45.3

Table 9. Multilingual BLSTM systems: adaptation by speaker-specific vectors %WER.

7.2. Analysis of speaker adaptation for Multilingual BLSTM

Until now, our best system has been based on Multilingual BLSTM without any speaker-adaptation. As classical speaker adaptation approaches such as CMLLR are difficult with BLSTMs, we were interested injecting this architecture with speaker-specific vectors. Two techniques were investigated: i-vectors well known from speaker recognition, and newly introduced sequence-summarizing neural networks (SSNN).

Typically, the low-dimensional vector-based adaptation involves concatenating input feature vectors with speaker-specific vector that is constant across whole utterance, see [33] for an example of i-vector-adapted GMM system and [34] for DNN one. This approach was however not feasible in fine-tuning of multilingual NNs to target languages, as re-training the whole multi-lingual structure would be too prohibitive. Therefore, only the input of the final, target language-specific layer, was extended by the speaker-specific vector. In further experiments, we found that adding another layer before the final single softmax one and extending the input to this next-to-last layer by the speaker-specific vector provides even better performance. The adapted results in table 9 were obtained with this architecture. For comparison, collumns ”No Adapt - 1L/2L” compare results given by adding another layer before the final single softmax into non-adapted system.

In our work, we trained SSNN on top of “Mult28_CMLLR” features. The SSNN output was appended to input layer of the main DNN. Next, the SSNN was cut out and used for generation of speaker specific vectors - *‘d-vectors’* (an acronym introduced in [19]).

Table 9 shows the results of BLSTM system adapted with i-vectors only, d-vectors only or both. Additional layer was inserted before the final single softmax one and the adaptation took place in this next-to-last layer. It is evident that both methods work and provide 0.3-0.5% absolute improvement. Their combination usually reaches an improvement close to the better of the two techniques.

8. CONCLUSION

This paper provides an extensive summary of BUT 2016 system for the last Babel evaluations. It concentrates on multi-lingual training of both DNN-based features and acoustic models and on the low-dimensional vector approaches to speaker adaptation.

We have shown clear advantage of multi-lingual training both for feature-extraction and for acoustic models for 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 “raw” features at the input.

WPE is a “cheap” technique to improve results on far-field speech and it does not deteriorate results on CTS data. Speaker-specific vector adaptations have shown a great potential and capability of integration with complex DNN and RNN architectures. Here, we will investigate into a scheme that would be directly trainable with the most powerful acoustic model - the BLSTM.

9. REFERENCES

[1] František Grézl and Martin 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*. 2014, pp. 39–45, International Speech Communication Association.

[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] Arnab Ghoshal, Pawel Swietojanski, and Steve Renals, “Multilingual training of deep neural networks.,” in *ICASSP*. 2013, pp. 7319–7323, IEEE.

[4] Martin Karafiát, Lukáš Burget, František Grézl, Karel Veselý, and Jan Černocký, “Multilingual region-dependent transforms,” in *Proceedings of the 41th IEEE International Conference on Acoustics, Speech and Signal Processing*. 2016, pp. 5430–5434, IEEE Signal Processing Society.

[5] T. Yoshioka, T. Nakatani, M. Miyoshi, and H. G. Okuno, “Blind separation and dereverberation of speech mixtures by joint optimization,” *IEEE Trans. Audio, Speech, Language Process.*, vol. 19, no. 1, pp. 69–84, 2011.

[6] T. Yoshioka and T. Nakatani, “Generalization of multi-channel linear prediction methods for blind MIMO impulse response shortening,” *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 10, pp. 2707–2720, 2012.

[7] Marc Delcroix, Takuya Yoshioka, Atsunori Ogawa, Yotaro Kubo, Masakiyo Fujimoto, Nobutaka Ito, Keisuke Kinoshita, Miquel Espi, Takaaki Hori, Tomohiro Nakatani, and Atsushi Nakamura, “Linear prediction-based dereverberation with advanced speech enhancement and recognition technologies for the REVERB challenge,” in *Proc. of REVERB’14*, 2014.

[8] T. Yoshioka, X. Chen, and M. J. F. Gales, “Impact of single-microphone dereverberation on DNN-based meeting transcription systems,” in *Proc. ICASSP’14*, 2014.

[9] Martin Karafiát, František Grézl, Mirko Hannemann, and Jan ”Honza” Černocký, “BUT neural network features for spontaneous vietnamese in BABEL,” in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, Florence, Italy, May 2014, IEEE.

[10] Najim Dehak, Patrick Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, “Front-end factor analysis for speaker verification,” *IEEE Transactions on Audio, Speech & Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[11] Karel Veselý, Shinji Watanabe, Kateřina Žmolíková, Martin Karafiát, Lukáš Burget, and Jan Černocký, "Sequence summarizing neural network for speaker adaptation," in *Proceedings of the 41th IEEE International Conference on Acoustics, Speech and Signal Processing*. 2016, pp. 5315–5319, IEEE Signal Processing Society.

[12] Martin Karafiát, František Grézl, Mirko Hannemann, Karel Veselý, Igor Szoke, and Jan "Honza" Černocký, "BUT 2014 Babel system: Analysis of adaptation in NN based systems," in *Proceedings of Interspeech 2014*, Singapure, September 2014, IEEE.

[13] Karel Veselý, Martin Karafiát, and František Grézl, "Convolutive bottleneck network features for LVCSR," in *Proceedings of ASRU 2011*, 2011, pp. 42–47.

[14] Amit Agarwal, Eldar Akchurin, Chris Basoglu, Guoguo Chen, Scott Cyphers, Jasha Droppo, Adam Eversole, Brian Guenter, Mark Hillebrand, Ryan Hoens, Xuedong Huang, Zhiheng Huang, Vladimir Ivanov, Alexey Kamenev, Philipp Kranen, Oleksii Kuchaiev, Wolfgang Manousek, Avner May, Bhaskar Mitra, Olivier Nano, Gaizka Navarro, Alexey Orlov, Marko Padmilac, Hari Parthasarathi, Baolin Peng, Alexey Reznichenko, Frank Seide, Michael L. Seltzer, Malcolm Slaney, Andreas Stolcke, Yongqiang Wang, Huaming Wang, Kaisheng Yao, Dong Yu, Yu Zhang, and Geoffrey Zweig, "An introduction to computational networks and the computational network toolkit," Tech. Rep. MSR-TR-2014-112, August 2014.

[15] Bing Zhang, Spyros Matsoukas, and Richard Schwartz, "Recent progress on the discriminative region-dependent transform for speech feature extraction," in *Proc. of Interspeech 2006*, Pittsburgh, PA, USA, Sep 2006, pp. 2977–2980.

[16] T. Nakatani, T. Yoshioka, K. Kinoshita, M. Miyoshi, and B.-H. Juang, "Blind speech dereverberation with multi-channel linear prediction based on short time Fourier transform representation," in *Proc. of ICASSP'08*, 2008, pp. 85–88.

[17] K. Kinoshita, M. Delcroix, T. Nakatani, and M. Miyoshi, "Suppression of late reverberation effect on speech signal using long-term multiple-step linear prediction," *IEEE Trans. Audio, Speech, Language Process.*, vol. 17, no. 4, pp. 534–545, 2009.

[18] Pavel Matějka, Ondřej Glembek, Ondřej Novotný, Oldřich Plchot, František Grézl, Lukáš Burget, and Jan Černocký, "Analysis of dnn approaches to speaker identification," in *Proceedings of the 41th IEEE International Conference on Acoustics, Speech and Signal Processing*. 2016, pp. 5100–5104, IEEE Signal Processing Society.

[19] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Proceedings of ICASSP 2014*.

[20] Souvik Kundu, Gautam Mantena, Yanmin Qian, Tian Tan, Marc Delcroix, and Khe Chai Sim, "Joint acoustic factor learning for robust deep neural network based automatic speech recognition," in *Proceedings ICASSP 2016*.

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

[22] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. Dec. 2011, IEEE Signal Processing Society, IEEE Catalog No.: CFP11SRW-USB.

[23] Pegah Ghahremani, Bagher BabaAli, Daniel Povey, Korbinian Riedhammer, Jan Trmal, and Sanjeev Khudanpur, "A pitch extraction algorithm tuned for automatic speech recognition," in *Proceedings of ICASSP*, 2014.

[24] N. Kumar and A. G. Andreou, "Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition," *Speech Communication*, pp. 283–297, 1998.

[25] Daniel Povey and George Saon, "Feature and model space speaker adaptation with full covariance Gaussians," in *Proc. of INTERSPEECH*, 2006.

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

[27] Karel Veselý, Arnab Ghoshal, Lukáš Burget, and Daniel Povey, "Sequence-discriminative training of deep neural networks," in *Proceedings of Interspeech 2013*. 2013, pp. 2345–2349, International Speech Communication Association.

[28] Yu Zhang, Guoguo Chen, Dong Yu, Kaisheng Yao, Sanjeev Khudanpur, and James 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.

[29] Hasim Sak, Andrew W. Senior, and Françoise 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.

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

[31] Karel Veselý, Martin Karafiát, František Grézl, Miloš Janda, and Ekaterina Egorova, "The language-independent bottleneck features," in *Proceedings of IEEE 2012 Workshop on Spoken Language Technology*. 2012, pp. 336–341, IEEE Signal Processing Society.

[32] František Grézl, Ekaterina Egorova, and Martin Karafiát, "Further investigation into multilingual training and adaptation of stacked bottle-neck neural network structure," in *Proceedings of 2014 Spoken Language Technology Workshop*. 2014, pp. 48–53, IEEE Signal Processing Society.

[33] Martin Karafiát, Lukáš Burget, Pavel Matějka, Ondřej Glembek, and Jan "Honza" Černocký, "ivector-based discriminative adaptation for automatic speech recognition," in *Proc. ASRU 2011*, dec 2011.

[34] George Saon, Hagen Soltau, David Nahamoo, and Michael 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.