ADAPTING MULTILINGUAL NEURAL NETWORK HIERARCHY TO A NEW LANGUAGE

Frantisek Grézl and Martin Karafiát *

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

ABSTRACT

The neural network based features became an inseparable part of state-of-the-art LVCSR systems. With the increasing accent on fast development of ASR system on limited resources, there is an effort to alleviate the need of large amount of transcribed in-domain data. One successful way is to use data from other languages. We present extensive evaluation of several strategies to adapt hierarchical neural network in search for the most effective one. To avoid the bias towards one target language, our strategies were evaluated on five languages. Also, several multilingual neural network hierarchies were trained on two sets of languages. Thus the results provide solid insight into the problem of adapting hierarchical system.

Index Terms— feature extraction, Bottle-Neck features, neural network adaptation, multilingual neural networks, Stacked Bottle-Neck structure

1. INTRODUCTION

Quick delivery of ASR system for a new language is one of the challenges in the community. Hand in hand with the quick delivery comes limitation of available resources. Such scenario calls 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. This is particularly important for features obtained from neural networks (NNs). The pressure on this part comes from two sides: First, the neural networks are sensitive to amount of training data. In order to perform well, they need to be trained on large amount transcribed data. Unfortunately, the data collection and annotation is the most time- and money-consuming procedure. Second, since feature extraction is the first step in the whole speech to text (STT) system, the time one can spend on training the feature extractor is limited. On the other hand, feature extraction is the crucial step and quality of the features determines the performance of whole STT system.

This naturally raises the question whether features of sufficient quality can by obtained from different sources. The first study of portability of NN-based features was done in [1] where NNs trained on English data were applied to Mandarin and Levantine Arabic to produce probabilistic features. Consistent word error rate (WER) reduction was observed for both languages. In both cases however, the amount of training data would be itself sufficient for training good neural networks (100 and 70 hours respectively).

Our work [2] studied the possibility to train a multilingual NN to be used to derive features for a new language. Several approaches to create the target phoneme set for the multilingual training were explored. We have shown that concatenation of phoneme sets is a safe and simple approach (further denoted as *one softmax*). However, performing merging on phoneme sets of individual languages can be beneficial depending on the language set and desired features.

The integral way of obtaining multilingual (or lang.-independent) NN based (bottle-neck) features is presented in [3, 4]. Here, the NN is trained on several languages which makes the main body of the NN language-independent while the last – output – layer is divided into language-dependent parts. Only one part of the output (N^{th}) layer corresponding to the language of a particular input-output training pair is active. Thus the outputs of the $(N-1)^{th}$ layer provide information which should be equally useful for classification of any of the language-specific targets used in the training. This leads to truly multilingually trained weights in NN except for the language-specific parts of the output layer. This approach will be further denoted as *block softmax*. This technique was modified by Heigold et al. in [5] and tested in multilingual DNN hybrid system.

When comparing the two approaches, we should note that one output layer (*one softmax*) for all language-specific targets performs, together with classification of the input vectors, indirectly also language identification as it has to distinguish between similar (or the same) targets from different languages.

All the above techniques assume no data for the target language, which is somewhat unrealistic scenario as there has to be some transcribed data to train the acoustic model on. And since there is the data, forced alignment can be done on them and the input-output pairs can be used to adapt the neural network for feature extraction.

It should be also noted, that none of the techniques above led to significant improvement over the monolingual NN trained on the target language data only. On the other hand, adding target language data to multilingual training brought consistent improvement. This shows how important it is to present the target acoustic space during the NN training.

The adaptation to target language brings issues with language-specific phonemes. Vu et al. [6, 7] suggest to solve this problem by approximation of such phonemes by several phonemes from the source languages that, in combination, have the characteristic of the target phoneme. Then, NN is retrained on target language using only the outputs (phonemes) belonging to it.

The adaptation of NN trained on large amount of data from one language to target domain with little data by final fine-tuning was proposed in [8] and extended to multilingual NN in [9]. This approach eliminates the necessity of identification and approximation of new phonemes.

Our latest work [10] presents several strategies of adaptation of Stacked Bottle-Neck (SBN) (originally called "Universal Context")

*This work 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 the IT4Innovations Centre of Excellence CZ.1.05/1.1.00/02.0070. M. Karafiat was supported by Grant Agency of the Czech Republic post-doctoral project No. P202/12/P604.

Table 1. *Characteristics of used languages*

Language	# dialects	tonal?	Phonology	Morphology
Cantonese	5	yes (7)	19 consonants, 8 vowels, 11 diphthongs	analytic, very limited affixation
Pashto	4	no	30 consonants, 7 vowels	affixes on nouns, verbs, and adjectives; some stem allomorphy
Tagalog	3	no	19 consonants, 15 vowels, 11 diphthongs	verbs take prefixes, suffixes, infixes, reduplication for focus, aspect, mode, voice
Turkish	7	no	25 consonants, 1 semivowel, 16 vowels	agglutinating, vowel harmony, inflectional
Vietnamese	4	yes (6)	25 consonants and 45 vowels, 12 monophthongs, 25 diphthongs, 8 triphthongs	analytic, very limited affixation
Bengali (BE)	3	no	33 consonants, 2 semivowels, 10 vowels (8 monophthongs and 2 diphthongs) and 9 nasal vowels (7 monophthongs and 2 diphthongs)	Fusional, 4 noun cases, some noun classifiers; 3 politeness levels in 2nd person: intimate, neutral, formal
Assamese (AS)	3	no	30 consonants, 9 vowels (7 monophthongs and 2 diphthongs), 9 nasal vowels (7 monophthongs and 2 diphthongs)	Fusional, fairly extensive noun classifier system, 6 noun cases; three politeness levels as in Bengali
Haitian Creole (HA)	3	no	20 consonants, and 12 vowels (11 monophthongs and 1 diphthong)	analytic, minimal derivational morphology, no inflectional morphology
Lao (LA)	1	yes (6)	19 consonants, 2 semivowels, 22 vowels (18 monophthongs, 9 long, 4 diphthongs)	analytic, no inflectional morphology, 4 levels of politeness
Zulu (ZU)	1	yes (3)	28 consonants, 9 clicks, 2 semi-vowels, 7 vowels	agglutinative, with extensive inflection

NN hierarchy [11]. The SBN structure achieves significantly better performance than single NN and it is widely used these days.The question which naturally raised in case of NN hierarchy is which NN should be adapted, or if one of them can be trained on target data only. We have observed that different adaptation strategies can lead to 2% absolute WER difference. Unfortunately, the obtained results were not consistent which might be assigned to the property of jackknifing experiments - different training set and different test set.

To obtain stronger evidence, we have designed a new set of experiments. We have defined two sets of training languages and we evaluate the techniques on five languages. Thus the drawn concussions stay on solid ground.

2. EXPERIMENTAL SETUP

2.1. Data

The IARPA Babel Program data[1] simulate a case of what one could collect in limited time from a completely new language: it consists of two parts: scripted (speakers read text through telephone channel) and conversational (spontaneous telephone conversations). The *dev* data contains conversational speech only. Two training scenarios are defined for each language – Full Language Pack (FLP), where all collected data are available for training; and Limited Language Pack (LLP) which consist only of one tenth of FLP. Vocabulary and language model (LM) training data are also defined with respect to the Language Pack. They basically consists of transcripts of the given data pack.

For multilingual training, the FLP data from the first year of the program are used. Those are Cantonese language collection release IARPA-babel101-v0.4c (CA), Pashto IARPA-babel104b-v0.4aY (PA), Turkish IARPA-babel105-v0.6 (TU), Tagalog IARPA-babel106-v0.2g (TA) and Vietnamese IARPA-babel107b-v0.7 (VI). These languages will be further referred as source languages.

Table 2. *Evaluation data statistics. The LM and dictionary statistics are taken from LLP which is used to train HMM system. The OOV rate is reported with respect to LLP.*

Language	AS	BE	HA	LA	ZU
FLP speakers	726	793	752	789	743
FLP hours	69.5	74.1	72.3	71.6	57.4
LLP speakers	120	120	120	120	120
LLP hours	7.8	8.9	7.9	8.1	8.4
LM sentences	11814	11763	9861	11577	10644
LM words	75610	84334	93131	93328	60832
dictionary	8729	9497	5333	3856	14962
# tied states	1179	1310	1257	1453	1379
dev speakers	120	121	120	119	119
dev hours	6.4	6.9	7.4	6.6	7.4
# words	51931	56221	81087	81661	50053
OOV rate [%]	8.3	8.5	4.1	1.8	22.4

The evaluation (target) languages are the ones delivered in the second year: 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) and Zulu IARPA-babel206b-v0.1e (ZU). The LLP is used as adaptation data.

The characteristics of the languages are given in Tab 1 [12]. More detailed statistics for evaluation languages are given in Tab. 2. The reported amounts of data for FLP and LLP refer to the speech segments after dropping the silence.

2.2. NNs for feature extraction

The features obtained using Neural Networks are the Bottle-Neck (BN) features. A structure of two 6-layer NNs is employed according to [11]. It is depicted in Fig. 1.

The NN input features are composed of critical band energy features and fundamental frequency features. As critical band en-

[1] Collected by Appen `http://www.appenbutlerhill.com`

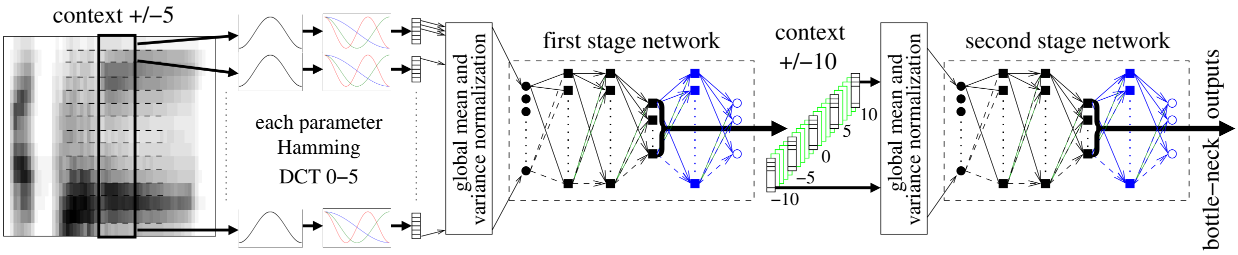

Fig. 1. Block diagram of Bottle-Neck feature extraction. The blue parts of NNs are used only during the training. The green frames in context gathering between the NNs are skipped. Only frames with shift -10, -5, 0, 5, 10 form the input to the second stage NN.

ergy features are used logarithmized outputs of 24 Mel-scaled filters applied on squared FFT magnitudes. The fundamental frequency features consists of F0 and probability of voicing estimates computed according to [13] and smoothed by dynamic programming, F0 estimates obtained by Snack tool [2] function *getf0* and seven coefficients of Fundamental Frequency Variations spectrum according to [14, 15]. This feature set provided consistent improvement over previously used set of features (15 critical bands augmented with F0 and probability of voicing) for all languages.

The conversation-side based mean subtraction is applied on the whole feature vector. 11 frames are stacked together. Hamming window followed by DCT consisting of 0th to 5th base are applied on the time trajectory of each parameter resulting in 204 coefficients on the first stage NN input.

The first stage NN has four hidden layers with 1500 units each except the BN layer. The BN layer is the third hidden layer and its size is 80 neurons. Its outputs are stacked over 21 frames and downsampled before entering the second stage NN; every fifth frame is taken. This NN has the same structure and sizes of hidden layers as the first one. The size of BN layer is 30 neurons and its outputs are the final outputs forming the BN features for GMM-HMM recognition system.

Neurons in both BN layers have linear activation functions as they were reported to provide better performance [16]. Before the features enter each NNs' input layer, global mean and variance normalization is performed.

Tied triphone states are used as NN targets. Features obtained from NNs trained towards these targets provide consistently slightly better performance then context-independent phone states targets.

The forced alignments were generated with provided segmentations, however it was found that they still contain large portion of silence (50%–60%). Therefore, new segmentation, which reduced the amount of silence to 15%-20%, was generated. We also cut out the parts of segments which were labeled as “unknown” (generally unintelligible speech).

2.3. Simplified recognition system

This system is used to evaluate all approaches. It is based on MLLT-BN features only and thus directly reflects the changes in neural networks we made.

First, for each language, PLP-based system was trained. This system was used for forced alignment of the data and as a initialization for system based on Bottle-Neck features. It is trained on FLP for source languages and on LLP for evaluation languages.

PLP coefficients consist of 13 parameters and their first, second and third order derivatives. HLDA is estimated with Gaussian components as classes to reduce the dimensionality to 39. Then the conversation-side based mean and variance normalization is applied. Based on these HLDA-PLP features, baseline recognition system system is trained. It is an HMM-based cross-word tied-states triphone system. The number of tied states is around 1300 for LLP training (evaluation languages - see Tab. 2 for precise numbers) and 4000-8000 for FLP training (source languages, see Tab 3 for precise numbers). Each state consists of 18 Gaussian mixture components. It is trained from scratch using mix-up maximum likelihood training. Performance of this system is poor and we do not report results for it.

To train the system on Bottle-Neck features, the BN outputs are transformed by Maximum Likelihood Linear Transform (MLLT), which considers HMM states as classes. Then, new models are trained by single-pass retraining from HLDA-PLP baseline system. 12 Gaussian components per state were found to be sufficient for MLLT-BN features. Next, 12 maximum likelihood iterations follow to better settle new HMMs in the new feature space.

Final word transcriptions are decoded using 3gram LM trained only on the transcriptions of LLP training data[3].

2.4. Full recognition system

The best performing SBN systems are evaluated with the full recognition system. This system is based on feature level fusion by Region Dependent Transform (RDT) [17]. Three feature streams: PLP-HLDA (39 dimensions), MLLT-BN features (30 dim.) and F0 with delta and acceleration coefficients (3 dim.), are concatenated and adapted using speaker-based CMLLR. This feature stream is fed to Region Dependent transform (RDT) performing dimensionality reduction to 69 dimension.

In RDT framework, an ensemble of linear transformations is trained with the discriminative Minimum Phone Error (MPE) criterion. The feature space is partitioned by global GMM and one transform is assigned to a each multidimensional Gaussian. Resulting vector is computed as weighted sum of input vector transformed by all transforms. Weights are given by Gaussian probabilities. Thresholding is applied to reduce the computation.

the RDT GMM consists of 125 components, which was found optimal in our previous experiments. The contextual information was incorporated to improve performance of the system. For detailed information on RDT configuration, see [18].

[2] `www.speech.kth.se/snack/`

[3] This is coherent to BABEL rules, where *the provided data only* can be used for system training.

Table 3. *Data for multilingual training*

Language	CA	PA	TU	TA	VI
FLP speakers	952	1189	980	1096	1096
FLP hours	65.0	64.7	56.6	44.1	73.9
monophone states	471	216	126	252	303
triphone states	4718	5541	3805	3475	7731

In our setup, two sets of RDTs were trained, both performed dimensionality reduction from 72 to 69 dimensions: RDT_{nonSAT} on top of the original 72-dimensional features for 1^{st} pass decoding and RDT_{SAT} on top of CMLLR-rotated features. The final GMM system was trained using MPE [19] on top of SAT RDT features.

3. EXPERIMENTS

3.1. Full and Limited language packs

To obtain the baselines, the NNs were first trained in monolingual manner on both, LLP and FLP. Training on LLP is our starting point and it serves as the lower bound. Training on FLP shows what performance it is possible to achieve with more training data and provides us with upper bound. The closer the results will be to this value, the more the technique benefits from other resources. The results obtained with monolingual NNs are given on the firsts lines in Tab. 4.

We can see a dramatic drop in recognition performance when the NNs are trained just on LLP data. Note, that HMM systems are always trained on the LLP data. The FLP data are used for training the SBN NNs only.

3.2. Multilingual NN

The next set of experiments is focused on the performance of BN features obtained from multilingual NNs. This case provides the starting point for the adaptation. The NNs were trained on FLPs of the source languages. The amounts of data available for NN training together with number of classes are specified in Tab 3.

To be able to evaluate the effect of the number of source languages, we decided to generate two sets of them:

- Source language set 1 (SLs1) contains three language: CA, PA and TU
- Source language set 2 (SLs2) contains all five source languages.

Two approaches to train multilingual NNs are evaluated:

The first one – *one softmax* – discriminates between all targets of all languages. No mapping or clustering of phonemes was done. This simple approach turned out to perform the best in [2]. Thus the resulting NN has quite a large output layer containing all phonemes 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. This approach was successfully used in [4].

In our former work [10], we have also experimented with the *Convolutive Bottleneck Network* [16] technique which would allow to retrain the whole structure in one step. But the computation expenses to train the multilingual NN and also to adapt this network were unacceptable for our scenario where fast adaptation is desired.

Table 4. WER [%] of simplified recognition system on target languages LLP based on MLLT-BN features obtained from monolingual and multilingual SBN systems

		WER				
target language		AS	BE	HA	LA	ZU
FLP monolingual SBN		61.5	62.9	57.2	55.1	68.9
LLP monolingual SBN		68.5	69.7	65.9	63.6	74.2
SLs1	one softmax	70.0	70.4	69.2	65.8	74.6
	block softmax	69.0	69.6	66.8	64.7	74.1
SLs2	one softmax	68.7	69.6	66.4	62.9	73.7
	block softmax	**66.8**	**68.2**	**64.9**	**60.7**	**72.6**

Context-independent phoneme states were used as targets for multilingual NNs. We intended to use the states of tied context-dependent phonemes, but so far it turned out to be unfeasible. We were only able to train the first stage NN for SLs1.

The WER obtained with BN features from multilingual SBN systems trained on two defined source languages sets by the two multilingual approaches are given in Tab. 4.

The first and most striking difference is in the performance of SBN CMLLR-BN features obtained from the two source languages sets. Whereas the features from SLs1 systems reach about the performance of a single language system, the features from SLs2 systems have much lower WER. This observation suggests that the multilingual system is trained trained on many languages, it can perform well on "unseen language" even without any adaptation.

The second observation is that the *block softmax* version of SBN system performs better than the *one softmax* one. This is consistent over both SLs's and can be assigned to the way the NNs are trained.

3.3. Adaptation of Multilingual NN

The adaptation of NN is done through retraining of already trained NN on target language data. The trained multilingual NN guarantees a good starting point which already produces good features as can be seen in Tab. 4. Retraining allows to shift the weights towards the acoustic space of target data.

Our approach to NN adaptation has two phases:

1. Training of the last layer. Since our initial NN is multilingual, the output layer has large number of units. We need to initialize the output layer randomly with the proper number of outputs matching the target language phoneme set. If the whole NN was retrained now, the error caused by the random weights in the last layer could be propagated deeper in the NN and the training could drift apart from the optimum. This is why the rest of NN is fixed and only the last layer is trained.
2. Retraining of the whole NN. Here, the other layers are released and the whole NN is retrained once more. Since this retraining starts from an already trained network, the learning rate for this phase is set to one tenth of its original value[4].

Even though the adaptation consists of two phases, the total number of NN training epochs is about the same as when the monolingual NN is trained. In case of monolingual training, the number of training epochs is set to 12. The first phase of the adaptation takes usually 5 to 7 epochs (we did not observe any training shorter or longer) and

[4] Learning rate of 0.004 is used for training from random weights, and 0.0004 for the retraining.

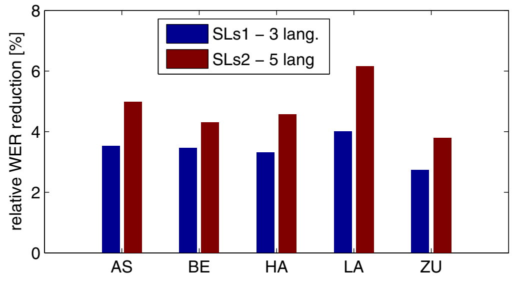

Fig. 2. Average relative WER [%] reduction depending on Source Language set for all evaluation languages

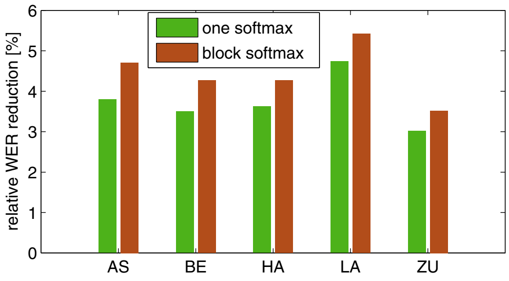

Fig. 3. Average relative WER [%] reduction depending softmax type in multilingual SBN.

the second phase is set to take 6 epochs. This means that the adaptation does not cost any extra time.

The SBN system is a hierarchy of two NNs, so the adaptation can take several forms. We can for example keep the first NN multilingual and train the second one on the LLP data only. Thus we consider the following four scenarios for adaptation of multilingual system:

1. Keep the first NN multilingual, train the second one on LLP data only - *multi-LLP* scenario.
2. Adapt the first NN, train the second one on the LLP data - *adapt-LLP* scenario.
3. Keep the first NN multilingual, adapt the second one - *multi-adapt* scenario.
4. Adapt the first and also the second NN - *adapt-adapt* scenario.

The last *adapt-adapt* scenario might be regarded as not very good idea since the inputs to the second NN are changed but, surprisingly, out initial experiments discovered that this scenario is not useless.

Since every multilingual SBN system was adapted according to all four scenarios for each of the evaluation languages, showing all results would not provide an easy survey. Rather, we will focus on one aspect at a time[5]. To be able to present the results together, each one is converted to relative WER reduction with respect to the its LLP baseline.

Number of languages for multilingual training

In Sec. 3.2, it was shown, that the number of training languages plays an important role when the features are generated directly from the multilingual system. Here we show the role it plays when the system is adapted. The results from all scenarios originated from the same SLs were averaged for each evaluation language. The relative improvements over the LLP monolingual system are shown in Fig. 2. It can be seen that even for the adaptation the number of source languages is important - the average WER reduction is more than 1% relative for all languages.

One softmax vs. block softmax

The *block softmax* performed slightly better in case of multilingual SBN systems. To see this effect in the adaptation, we again average results from all scenarios originating from multilingual SBN system with given softmax within each language. As can be seen from Fig. 3, the relative improvements achieved by adapting multilingual NNs with block softmax are higher than the ones from one softmax.

Adaptation scenario performance

Fig. 4 breaks the results to the level of individual adaptation scenarios. The results are averaged over the evaluation languages. For better readability, the results are split into two plots based on the Source Language set used to train the multilingual SBN systems.

First, note that the performance of *multi-LLP* scenario leads to the same performance regardless the softmax type in multilingual SBN. This suggests that this scenario cannot take advantage from the differently trained SBN. When the first NN is adapted before the training of the second one on LLP in *adapt-LLP* scenario, the WER can be further reduced. Note the increased difference between different types of softmax in multilingual SBN - the adaptation process can benefit more from the SBNs trained with block softmax.

In case of training on SLs1, the WER is reduced the least when only the second NN is adapted in *multi-adapt* scenario. The last scenario – *adapt-adapt* – when both NNs in SBN are adapted actually performs about the same as *multi-LLP* scenario. This shows that also the second NNs in the hierarchy can be adapted despite its changed inputs by former adaptation of the first NN.

When the multilingual NNs are trained on SLs2, the picture changes. The adaptation of the second NN only in*multi-adapt* scenario outperforms the *adapt-LLP* one and adapting both NNs - *adapt-adapt* - leads to systems with the best performance. It also seems that in case of adapting from SLs2 NNs, the difference between *block softmax* and *one softmax* NNs decreases.

4. FULL SYSTEM

Systems which provided the best performance - SLs2 *adapt-LLP* (this scenario provides slightly better performance for all languages except LA than *multi-adapt* scenario. LA is 2% absolute better in case of *multi-adapt*, which affected the overall results) and *adapt-adapt* scenario - were taken and the full system was trained. We also compared the multilingual approach with the semi-supervised training presented previously in [20]. The overview of these approaches from the point of view of required data (taking LLP as a unit) and time is the following:

[5] The full set of results can be found at `www.fit.vutbr.cz/~grezl/SLTU_recognition_results/`

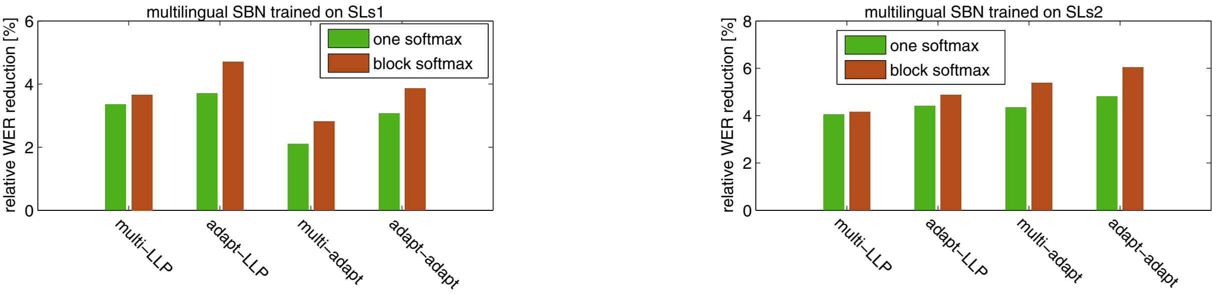

Fig. 4. Average relative WER [%] reduction depending on the adaptation scheme and softmax type in multilingual SBN.

Table 5. WER [%] of selected adaptation scenarios and semi-supervised technique with full system

	WER				
target language	AS	BE	HA	LA	ZU
monolingual LLP	63.0	64.4	59.1	56.4	71.0
semi-supervised	61.2	64.2	57.0	53.4	71.0
adapt-LLP	60.6	62.4	56.7	54.3	69.6
adapt-adapt	**59.9**	**62.0**	**56.4**	**53.0**	**69.1**

	Multilingual	**semi-supervised**
Data	5 × 9 × LLP data - the amount of LLP data is about nine times smaller then the FLP one and the multilingual NNs were trained on five languages; data are from other languages	9 × LLP data - System trained on LLP data is used to automatically transcribe the FLP data; data from the same source
Time	One training of SBN system on LLP data - adaptation time is the same as training time, multilingual NNs can be trained beforehand	Training of full LLP system, decoding of FLP data, training of NNs on about 9 times more data (9 times longer)

From the above it can be seen that the multilingual approach is more suitable for fast adaptation. The comparison of performance of the full systems for both techniques is given in Tab. 5. It can be seen that both techniques - multilingual and semi-supervised training - improve nicely (with exception of Zulu) over the LLP baseline. It is surprising to see that multilingual technique leads to better performance than semi-supervised training.

In case of Zulu, where the system suffers from high OOV rate, the semi-supervised technique was not able to improve the system performance. This suggests that the multilingual system would be specially beneficial in cases where the performance of purely monolingual system is too poor to allow for semi-supervised training.

5. CONCLUSIONS

This work addresses a thorough evaluation of multilingual techniques for adapting feature extraction neural network hierarchy - Stack Bottle-Neck system. We defined two sets of source languages used for training of multilingual Stacked Bottle-Neck system. Two SBNs have been trained on each source language set. The networks in them differ by the type of the last layer - the softmax nonlinearity. One type is the normal *one softmax*, which computes probabilities for all outputs. The other type is *block softmax* which splits the output targets into groups (one group is created by targets from one language) and computes the probability for each group. Such type of softmax does not force the NN to make the decision about the language together with the classification of given target and its weights are language independent.

Thus we have trained four multilingual systems. We have evaluated their performance on five evaluation languages prior to any adaptation. It was observed that SBN systems trained on more source languages performs much better than the ones trained on less data.

Each SBN system was adapted according four different scenarios: Keep the first NN multilingual, train the second one on LLP data; adapt the first NN, train the second one on the LLP data; keep the first NN multilingual, adapt the second one; adapt both NNs.

From the results, we made the following observations:

1. The use of more languages for multilingual training is definitely beneficial.
2. Training multilingual NNs with *block softmax* brings improvement over the *one softmax* systems.
3. Using the first NN without adaptation and training the second one on evaluation language is safe and leads to a good performance of adapted system.
4. When the multilingual system is “good enough”, the adaptation of both its NNs leads to the best performance.

Two best performing systems were evaluated further with the full recognition system and compared to the semi-supervised approach. The multilingual approach outperforms the semi-supervised one on all languages. The improvement is specially noticeable on Zulu, where the semi-supervised technique failed as Zulu has a poor monolingual system performance and suffers from large number of OOVs.

These observations suggest that adaptation of multilingual system is beneficial especially for cases with close-to-zero acoustic data. The time needed to obtain the final system is the same as training the monolingual one, leaving enough of it to work on top of the resulting MLLT-BN features.

The comparison of multilingual and semi-supervised approach also gives direction to our further development - the multilingual system can provide better automatic transcription for semi-supervised training. Cascading these systems can lead to further improvement with no additional cost over the current semi-supervised training.

It would be also interesting to observe, if the performance will further increase with adding more languages. Another goal, is to train multilingual neural networks with context-dependent tied states as targets. Having such network might be useful since these units are used as the training targets for evaluation languages.

6. REFERENCES

[1] A. Stolcke, F. Grézl, M.Y. Hwang, X. Lei, N. Morgan, and D. Vergyri, "Cross-domain and cross-language portability of acoustic features estimated by multilayer perceptrons," in *Proceedings of ICASSP 2006*, Toulouse, FR, 2006, pp. 321–324.

[2] F. Grézl, M. Karafiát, and M Janda, "Study of probabilistic and bottle-neck features in multilingual environment," in *Proceedings of ASRU 2011*, 2011, pp. 359–364.

[3] Stefano Scanzio, Pietro Laface, Luciano Fissore, Roberto Gemello, and Franco Mana, "On the use of a multilingual neural network front-end," in *Proceedings of INTERSPEECH-2008*, 2008, pp. 2711–2714.

[4] 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.

[5] Georg Heigold, Vincent Vanhoucke, Andrew W. Senior, Patrick Nguyen, Marc'Aurelio Ranzato, Matthieu Devin, and J. Dean, "Multilingual acoustic models using distributed deep neural networks," in *ICASSP*, 2013, pp. 8619–8623.

[6] Ngoc Thang Vu, Florian Metze, and Tanja Schultz, "Multilingual bottleneck features and its application for under-resourced languages," in *Proc. of The third International Workshop on Spoken Languages Technologies for Under-resourced Languages (SLTU'12)*, Cape Town, South Africa, 2012.

[7] Ngoc Thang Vu, Wojtek Breiter, Florian Metze, and Tanja Schultz, "An investigation on initialization schemes for multilayer perceptron training using multilingual data and their effect on ASR performance," in *Proc. Interspeech*, 2012.

[8] Samuel Thomas, Sriram Ganapathy, and Hynek Hermansky, "Multilingual MLP features for low-resource LVCSR systems," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4269–4272.

[9] Samuel Thomas, Michael Seltzer, Kenneth Church, and Hynek Hermansky, "Deep neural network features and semi-supervised training for low resource speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, Vancouver, Canada, may 2013, IEEE.

[10] F. Grézl, M. Karafiát, and K. Veselý, "Adaptation of multilingual stacked bottle-neck neural network structure for new language," in *accepted for ICASSP 2014*, 2014.

[11] F. Grézl, M. Karafiát, and L. Burget, "Investigation into bottleneck features for meeting speech recognition," in *Proc. Interspeech 2009*, Sep 2009, pp. 294–2950.

[12] M. Harper, "The babel program and low resource speech technology," in *Proc. of ASRU 2013*, Dec 2013.

[13] D. Talkin, "A robust algorithm for pitch tracking (RAPT)," in *Speech Coding and Synthesis*, W. B. Kleijn and K. Paliwal, Eds., New York, 1995, Elseviever.

[14] Kornel Laskowski, Mattias Heldner, and Jens Edlund, "The fundamental frequency variation spectrum," *Proceedings of FONETIK 2008*, pp. 29–32, 2008.

[15] Kornel Laskowski and Jens 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)*, Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, Eds., Valletta, Malta, may 2010, European Language Resources Association (ELRA).

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

[17] Bing Zhang, Spyros Matsoukas, and Richard Schwartz, "Recent progress on the discriminative region-dependent transform for speech feature extraction," in *Proc. of Interspeech2006*, Pittsburgh, PA, USA, Sep 2006, pp. 2977–2980.

[18] Martin Karafiát, František Grézl, Mirko Hannemann, Karel Veselý, and Jan "Honza" Černocký, "BUT BABEL System for Spontaneous Cantonese," in *Proceedings of Interspeech 2013*. 2013, number 8, pp. 2589–2593, International Speech Communication Association.

[19] Daniel Povey, *Discriminative Training for Large Vocabulary Speech RecognitionK*, Ph.D. thesis, Cambridge University Engineering Department, Mar. 2003.

[20] František Grézl and Martin Karafiát, "Semi-supervised bootstrapping approach for neural network feature extractor training," in *Proceedings of ASRU 2013*. 2013, pp. 470–475, IEEE Signal Processing Society.