Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 81 (2016) 144 – 151

Procedia Computer Science

5th Workshop on Spoken Language Technology for Under-resourced Languages, SLTU 2016, 9-12 May 2016, Yogyakarta, Indonesia

Bottle-Neck Feature Extraction Structures for Multilingual Training and Porting

František Grézl*, Martin Karafiát

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

Abstract

Stacked-Bottle-Neck (SBN) feature extraction is a crucial part of modern automatic speech recognition (ASR) systems. The SBN network traditionally contains a hidden layer between the BN and output layers. Recently, we have observed that an SBN architecture without this hidden layer (i.e. direct BN-layer – output-layer connection) performs better for a single language but fails in scenarios where a network pre-trained in multilingual fashion is ported to a target language. In this paper, we describe two strategies allowing the direct-connection SBN network to indeed benefit from pre-training with a multilingual net: (1) pre-training multilingual net with the hidden layer which is discarded before porting to the target language and (2) using only the the direct-connection SBN with triphone targets both in multilingual pre-training and porting to the target language. The results are reported on IARPA-BABEL limited language pack (LLP) data.

Peer-review under responsibility of the Organizing Committee of SLTU 2016

Keywords: DNN topology; Stacked Bottle-Neck; feature extraction; multilingual training; system porting

1. Introduction

One of the recent challenges in speech recognition community is to build an ASR system with limited in-domain data. The data hungry algorithms for training ASR system components have to be modified to be effective with less data. This applies mainly to neural networks (NNs) which are part of essentially any state-of-the-art ASR system today and can be placed in any of the main ASR parts: feature extraction (e.g. [1]), acoustic model (e.g. [2]) and language model (e.g. [3]).

NNs usually have to be trained on a large amount of in-domain data in order to perform well. The need for large training data sets can be alleviated by layer-wise training [4] or unsupervised pre-training [5]. Another techniques such as dropout [6] and maxout [7] effectively reduce the number of parameters in the neural network during the training.

* Corresponding author. Tel.: +420-541-141-280 ; fax: +420-541-141-270.
E-mail address: grezl@fit.vutbr.cz

Peer-review under responsibility of the Organizing Committee of SLTU 2016
doi:10.1016/j.procs.2016.04.042

To improve the performance of a neural network, its size can be increased. The above mentioned dropout and maxout techniques are employed to prevent over-training. The over-training can be also prevented by introducing a regularization term into the objective function [8,9].

Another way to improve NN performance is to impose a certain structure on the NN or compose more NNs together. The typical example of the first method are Convolutive Neural Networks [10,11]. The NN compositions typically consist of two NNs, where the outputs of one NN form inputs to the other one. Those composed NNs are mostly used in place of feature extractors and the most typical compositions today are Stacked Bottle-Neck (SBN) [1], Hierarchical MRASTA [12] and Shifting Deep Bottle-Neck [13] which is very similar to [1] and its one-network version [14]. It became evident that there are two factors important for the success of these compositions:

- compression of the features through a Bottle-Neck (BN) layer [15]
- putting larger contexts of the first NN outputs into the input of the second NN

Another advantage of using a Bottle-Neck layer in a NN, at least to our experience, is, that it serves as some form of regularization and other regularization techniques are not necessary.

The IARPA BABEL program with its goal to quickly train a keyword spotting system for new language with minimum in-domain transcribed speech data encouraged research in training multilingual NN and porting such multilingual NN to new language [16,17,18]. Thus the effort to improve the NN performance has to be evaluated also in the context of multilingual training and porting of trained NN to target language.

2. Experimental setup

The setup is adopted from [16] and all results are directly comparable.

2.1. Data

The IARPA Babel Program requires the use of a limited amount of training data which simulates the case of what one could collect in limited time from a completely new language. It consists mainly of telephone conversation speech, but scripted recordings as well as far field recordings are present too. Two training scenarios are defined for each language – Full Language Pack (FLP), where all collected data are available for training – about 100 hours of speech; and Limited Language Pack (LLP) consisting only of one tenth of FLP. As training data, we consider only the transcribed speech. Vocabulary and language model (LM) training data are defined with respect to the Language Pack. They consist of speech word transcriptions of the given data pack.

The following data releases were used in this work: 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), 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 characteristics of the languages can be found in [19]. The FLP data of IARPA-babel10* (CA, PA, TU, TA, VI, AS, BE) languages are used for multilingual NN training. The rest of the languages (HA, LA, ZU) are considered as target languages. LLP data are used for NN porting and for training of GMM-HMM system. Statistics for LLP training set of target languages are given in Tab. 1 together with the development set used for system evaluation. The amounts of data refer to the speech segments after dropping the long portions of silence.

2.2. SBN DNN hierarchy for feature extraction

The NN input features are composed of logarithmized outputs of 24 Mel-scaled filters applied on squared FFT magnitudes (critical band energies, CRBE) and 10 F0-related coefficients. The filter-bank spans frequencies from 64Hz to 3800Hz. The F0-related coefficients consist of F0 and probability of voicing estimated according to [20] and

Table 1. Statistics of test languages data. Training (LLP) and development set

Language	HA	LA	ZU
LLP hours	7.9	8.1	8.4
LM sentences	9861	11577	10644
LM words	93131	93328	60832
dictionary	5333	3856	14962
# tied states	1257	1453	1379
dev hours	7.4	6.6	7.4
# words	81087	81661	50053
OOV rate [%]	4.1	1.8	22.4

smoothed by dynamic programming, F0 estimates obtained by Snack tool[1] function *getf0* and seven coefficients of Fundamental Frequency Variations spectrum[21].

Conversation-side based mean subtraction is applied on the whole feature vector. 11 frames of *CRBE+F0s* are stacked together. A Hamming window is used, followed by DCT. 0th to 5th cosine base are applied on the time trajectory of each parameter resulting in $34 \times 6 = 204$ coefficients on the first-stage NN input. Such an input vector is mean and variance normalized by norms computed over the whole training set.

A structure of two 6-layer DNNs is employed according to[1]. The first stage DNN in the Stacked Bottle-Neck (SBN) hierarchy has four hidden layers. The 1^{st}, 2^{nd} and 4^{th} layers have 1500 units with sigmoid activation function. The 3^{rd} is the BN layer having 80 units with linear activation function. The BN layer outputs are stacked (hence Stacked Bottle-Neck) over 21 frames and downsampled by factor of five before entering the second stage DNN. The second stage DNN is the same as the first one with exception of the BN layer size. In this DNN, it has 30 units. Outputs of the second stage DNN BN layer are the final outputs forming the BN features for GMM-HMM recognition system.

Forced alignments were generated with the provided segmentations. Re-segmentation stripping off long silence parts was done afterwards. Tied triphone states are used as NN targets.

2.3. Recognition system

The evaluation system is based on BN features only and thus directly reflects the changes in neural networks we made. The BN features are BN outputs transformed by Maximum Likelihood Linear Transform (MLLT), which considers HMM states as classes. The models are trained by single-pass retraining from an HLDA-PLP initial system. 12 Gaussian components per state were found to be sufficient for MLLT-BN features. 12 maximum likelihood iterations are done to settle HMMs in the BN feature space.

The final word transcriptions are decoded using 3gram LM trained only on the transcriptions of LLP training data – this is consistent with BABEL rules, where *the LLP data only* can be used for system training.

2.4. Multilingual SBN training and porting

The multilingual DNNs in SBN system are trained with the last layer – softmax – split into several blocks. Each block accommodates training targets from one language. This was found superior to having NNs with one-softmax representing either full or compacted target set[22]. Context-independent phoneme states were used as targets for multilingual NN training.

The trained multilingual DNN is ported to the target language in two steps:

1. **Training of the last layer**. The last layer of multilingual NN is dropped and a new one is initialized randomly with number of outputs given by the number of tied states in the target language. Only this layer is trained keep-

[1] www.speech.kth.se/snack/

Table 2. Performance of SBN hierarchies employing DNNs with different topologies. DNNs are trained on LLP data of individual languages. WER[%]

DNN structure	HA	LA	ZU
IN-**2xHL**-BN-**HL**-OUT	65.9	63.6	74.2
IN-**2xHL**-BN-OUT	65.1	63.3	**73.7**
IN-**3xHL**-BN-OUT	**64.8**	**62.6**	73.9

ing the rest of the NN fixed.

2. **Retraining of the whole NN**. The remaining layers are released and the whole NN is retrained. The starting learning rate for this phase is set to one tenth of the usual value.

The best performing scenario from our previous work[23] in which both NNs from SBN hierarchy undergo the same porting process is used here. Though porting the first NN basically changes the inputs to the second one, so that problems with adaptation could be expected. The experiments revealed that while retraining the NN with small learning rate (fine-tuning), the NN in able to adapt also to slight changes in input features.

3. Experiments

3.1. Changing the DNN topology

Experiments with topology of NN with BN were done shortly after introduction of BN features in[24]. Three hidden layer NNs with constant number of trainable parameters were used with Bottle-Neck layer being the middle one. The experiments with changing the ratio of neurons in layer before and after the BN layer show that the layer before should be bigger than the layer after BN. However, the results were not very consistent as a further increase of the size in the first hidden layer led to the degradation of ASR performance.

Another set of experiments compared the three hidden layer NN version with NN having only two hidden layers, where the bottle-neck layer directly precedes the output one. Again, the number of parameters in both versions was fixed, so the number of neurons in the first hidden layer of the two hidden layer NN was higher than in the three hidden layer version. The results showed that using three hidden layers – i.e. having large hidden layer between BN and output layer – is preferable.

During the time between the work[24] and today we enlarged the NN – increased the number of hidden layer as well as increased the number of neurons in the hidden layers – and used finer target units. However, there were still the big hidden layers between the Bottle-Neck and output layers.

Our experiments tested the necessity of hidden layers (HLs) after the BN again. Two kinds of SBN hierarchies were trained. The first one followed the description in section 2.2, i.e. two NNs with topology *IN-HL-HL-BN-HL-OUT* ~ *IN-**2xHL**-BN-**HL**-OUT*. In the second case, the hidden layer after the bottle-neck was omitted and the NNs have the following topology: *IN-HL-HL-BN-OUT* ~ *IN-**2xHL**-BN-OUT* with a direct BN-layer – output-layer connection. Note that the total number of trainable parameters is not fixed, the hidden layers have always 1500 neurons, NN with topology *IN-**2xHL**-BN-OUT* has about 65% of parameters compare to the NN with structure *IN-**2xHL**-BN-**HL**-OUT*.

The recognition results using these two variations of DNNs are shown in the first two lines of Table 2. To our surprise, the second version of DNN provided better results than the original structure. Encouraged by these results, the third version of SBNs was trained. It had one more hidden layer before BN, thus having topology *IN-HL-HL-HL-BN-OUT* ~ *IN-**3xHL**-BN-OUT*. This version has similar number of parameters as the original structure (the number of parameters increases by about 15%). Results using this structure are on the third line of Table 2. For Haitian and Lao, further improvement was achieved, a slight degradation is observed for Zulu.

Table 3. Summary of training sets used for multilingual SBN training.

# languages	***5***	**7**
amount of data [hours]	283	405
monophone state targets	1368	1656
tied-triphone state targets	25270	31768

Table 4. Performance of SBN hierarchies ported from multilingual ones. The multilingual DNNs have different topologies and training targets. WER[%]

multilingual training set			5 languages			7 languages		
multilingual DNN structure	targets	porting	HA	LA	ZU	HA	LA	ZU
IN-**2xHL**-BN-**HL**-OUT	phoneme states	original	62.0	58.3	71.2	61.0	57.7	70.5
IN-**3xHL**-BN-OUT	phoneme states	original	62.4	58.8	72.1	61.8	58.5	71.9
IN-**3xHL**-BN-OUT	tied triphones	original	60.8	57.1	70.8	59.9	56.7	70.4
IN-**3xHL**-BN-**HL**-OUT	phoneme states	modified	61.4	57.6	71.0	60.7	57.3	70.8

3.2. Multilingual SBN porting

Since the topology of DNNs in target language SBN feature extraction is inherited from the multilingual one, the next step was to train the multilingual DNNs with the best topology (*IN-**3xHL**-BN-OUT*) and evaluate the ported system. Our previous work[25,16] has shown that training multilingual DNN with block-softmax output layer, where each block accommodates one language, is preferable to one-softmax for all languages. Therefore, here we report results obtained by porting multilingual NNs having the block-softmax output layer.

Two sets of training languages were created to strengthen significance of the results. The smaller one contained 5 languages – CA, PA, TU, TA, VI. The bigger one contained all 7 training languages. Table 3 summarizes these two training sets.

The multilingual SBN hierarchies were ported to target languages according to sec 2.4.

The first two lines of Table 4 show the results obtained with ported multilingual network together with the results obtained with the original SBN hierarchy (topology: *IN-**2xHL**-BN-**HL**-OUT*; output non-linearity: block-softmax). We can see that after porting the multilingual SBN hierarchy to the target language, the old topology performs better than the new one.

3.3. Tied-triphone states targets

Attempts to use the tied-triphone states as targets for multilingual DNN training in our previous work[16] were not successful due to a parameter explosion. The number of weights between the large hidden layer and the even larger output layer was dominating the DNN size. Since the modified DNN topology has a small BN layer before the output one, the size of weight matrix will be reduced significantly making the use of tied-triphones as targets feasible.

Multilingual SBN DNN hierarchies were trained on both training sets using tied-triphone states as targets.

The results are shown on the third line of Table 4. It can be seen that using the modified DNN topology together with tied-triphone states as targets leads to an improvement over the original SBN architecture.

3.4. Modifications in multilingual SBN porting

The modified DNN topology using monophone state targets for multilingual training does not perform as well as expected after the porting described in section 3.2. The performance of such a SBN hierarchy is lower than the original topology. But the advantages coming from the modified DNN topology as seen in the section 3.1 should appear in the subsequent steps in the processing chain such as semi-supervised training and speaker adaptive training[26] where the DNNs are again retrained on larger amount of target language data.

To be able to make the most of both positive aspects – having an output layer right after bottle-neck one for monolingual NNs, and having a (large) hidden layer between Bottle-Neck and output layers – we need to change the

Table 5. Performance of systems where multilingual SBN hierarchy with DNN topology *IN-**2xHL**-BN-**HL**-OUT* trained on 5 languages was ported by the original and modified way. The topology of ported DNNs is either *IN-**2xHL**-BN-**HL**-OUT* or *IN-**2xHL**-BN-OUT*. WER[%]

structure of ported DNN	HA	LA	ZU
IN-**2xHL**-BN-**HL**-OUT	62.7	58.6	71.6
IN-**2xHL**-BN-OUT	61.9	58.2	71.4

porting procedure. The multilingual DNN will be trained with hidden layer between bottle-neck and output layer. Then:

1. All layers after the bottle-neck will be cut off. A new BN-to-output layer will be initialized randomly and trained, keeping the rest of the NN fixed.

2. The whole network will be retrained as in previous cases.

Before running extensive training of multilingual NNs, an evaluation of this idea was done on an already trained multilingual NNs trained on 5 languages. They have the original topology *IN-**2xHL**-BN-**HL**-OUT* and the block-softmax output non-linearity accommodating phoneme states targets. The topologies of ported DNNs are either *IN-**2xHL**-BN-**HL**-OUT* when the original porting approach is used, or *IN-**2xHL**-BN-OUT* when the proposed changes are applied.

From Table 5 it can be seen that the proposed changes in porting strategy have a positive effect on WER of the ported SBN hierarchy. Note, the improvement is achieved despite the reduction of trainable parameters in ported DNNs.

Next, a SBN hierarchy was trained on each training set. The DNNs have topology *IN-HL-HL-HL-BN-HL-OUT* ~ *IN-**3xHL**-BN-**HL**-OUT*. The DNNs with tied-triphone state targets are not trained as it was shown that a parameter explosion prevents efficient DNN training [16].

The results after porting the multilingual DNNs to the target language with the altered porting procedure are given on the forth line of Table 4. It can be seen that *IN-**3xHL**-BN-**HL**-OUT* topology together with the altered porting procedure outperforms the original strategy and brings additional improvement over the results shown in Table 5. It is also clear that the modified DNN topology (*IN-**3xHL**-BN-OUT*) with tied-triphone state targets provides further improvement. However, the difference between these competing modifications is not big.

3.5. Performance of multilingual BN features

Since the difference between the performance of ported systems obtained from either (*i*) modified DNN topology with tied-triphone state targets by the original porting procedure or (*ii*) the original DNN topology with phoneme state targets with altered porting procedure is not big, the decision whether to use one or the other may depend on behavior of these systems in different conditions. In our case, the performance of purely multilingual BN features on the target language is also important. With such multilingual features (after multilingual RDT), the audio data for new language are aligned and automatically transcribed when the reference transcription is missing.

Table 6 presents the performance of multilingual BN features processed the same way as the target language specific features to allow straight and fair comparison with them (sec. 3.1). The first line gives the performance of BN features trained only on target data with the original DNN topology *IN-**2xHL**-BN-**HL**-OUT* – the first line in Table 2. The following lines show performance of BN features obtained from discussed multilingual SBN DNN hierarchies – (*i*) DNN with topology *IN-**3xHL**-BN-OUT* and triphone state targets and (*ii*) *IN-**3xHL**-BN-**HL**-OUT* DNN topology with phoneme state targets.

Both variants of the multilingual features outperform the language-specific ones. Comparing between different DNN topologies, we see that the results are very similar. Slightly better performance is provided by the (*ii*) – the original DNN topology with phoneme state targets. Thus if higher quality initial forced alignment and mainly automatic transcription is preferred, the DNNs with this topology would be chosen to generate the bottle-neck features.

Table 6. Performance of multilingual BN features on target languages. WER[%]

training set	DNN structure	HA	LA	ZU
original		65.9	63.6	74.2
5 languages	(*i*)	64.8	60.6	72.5
	(*ii*)	64.7	61.1	72.4
7 languages	(*i*)	63.8	60.1	72.1
	(*ii*)	63.9	60.4	71.7

4. Conclusions

We show the effect of modified DNN topology in the Stacked Bottle-Neck hierarchy feature extractor. It was shown that the conclusions made shortly after introduction of Bottle-Neck features are not valid in the current settings. Namely, we have contradicted the necessity of large hidden layer between bottle-neck and output layer. We showed improved performance when this layer was omitted and a direct BN-layer – output-layer connection is introduced. The improvement is achieved despite dramatic one-third reduction of trainable parameters in DNNs. By moving the previously omitted layer before the Bottle-Neck one (which leads to similar number of trainable parameters) further improvement can be achieved.

We continued our effort by introducing such modified DNN topology to multilingual training because the DNN topology for the target language is inherited from the multilingual one by a porting procedure. It was shown, that the modified DNN topology is not suitable for multilingual training and subsequent porting.

Therefore, two alternation are investigated. The first one we have applied was the replacement of phoneme states targets by tied-triphone states. Thanks to the small Bottle-Neck layer, a parameter explosion and thus large computational demands are avoided. Porting the SBN hierarchy with modified DNN topology and tied-triphone state targets brings improvement over the original method.

The second evaluated alternation took place in the porting process. Here, the multilingual DNN still has a large hidden layer between the bottle-neck and output layers during the training. But it is dropped in the first phase of porting, when all layers after BN are removed, and single BN-to-output layer is initialized. This led to an improvement over the original training and porting procedure too.

In both cases the monolingual SBN hierarchy with desired DNN topology is obtained. Lower WER was achieved by the first variant which uses the tied-triphone states as targets and a direct BN to output layer connection during the multilingual training. Since the differences in results are not so large, other criteria may drive the decision which method to use. In our case it is the performance of multilingual BN features themselves, prior to porting to the target language. We show that the multilingual bottle-neck features obtained by the second variant achieves slightly better results.

Acknowledgements

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. This work was also supported by the European Union's Horizon 2020 project No. 645523 BISON, and by Technology Agency of the Czech Republic project No. TA04011311 "MINT".

References

1. Grézl, F., Karafiát, M., Burget, L.. Investigation into Bottle-Neck features for meeting speech recognition. In: *Proc. Interspeech 2009*. 2009, p. 294–2950.

2. Miao, Y., Metze, F.. Improving low-resource CD-DNN-HMM using dropout and multilingual DNN training. In: *Proceedings of Interspeech 2013*; 8. 2013, p. 2237–2241.
3. Mikolov, T., Kombrink, S., Burget, L., Černocký, J., Khudanpur, S.. Extensions of recurrent neural network language model. In: *Proceedings of the 2011 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2011*. IEEE Signal Processing Society. ISBN 978-1-4577-0537-3; 2011, p. 5528–5531.
4. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H.. Greedy layer-wise training of deep networks. In: *Advances in Neural Information Processing Systems 19 (NIPS'06)*. 2007, p. 153–160.
5. Erhan, D., Bengio, Y., Courville, A., Manzagol, P.A., Vincent, P., Bengio, S.. Why does unsupervised pre-training help deep learning? *J Mach Learn Res* 2010;**11**:625–660. URL: `http://dl.acm.org/citation.cfm?id=1756006.1756025`.
6. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR* 2012;**abs/1207.0580**.
7. Goodfellow, I.J., Warde-Farley, D., Mirza, M., Courville, A., Bengio, Y.. Maxout networks. In: *ICML*. 2013.
8. Yu, D., Yao, K., Su, H., Li, G., Seide, F.. KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition. In: *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. 2013, p. 7893–7897. doi:10.1109/ICASSP.2013.6639201.
9. Tomar, V.S., Rose, R.C.. Manifold regularized deep neural networks. *Proceedings on Interspeech - on line* 2014;**2014**(9):348–352.
10. Abdel-Hamid, O., Mohamed, A., Jiang, H., Penn, G.. Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition. In: *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. 2012, p. 4277–4280. doi:10.1109/ICASSP.2012.6288864.
11. Sainath, T., Mohamed, A.R., Kingsbury, B., Ramabhadran, B.. Deep convolutional neural networks for LVCSR. In: *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. 2013, p. 8614–8618. doi:10.1109/ICASSP.2013.6639347.
12. Valente, F., Hermansky, H.. Hierarchical and parallel processing of modulation spectrum for ASR applications. In: *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. 2008, p. 4165–4168. doi:10.1109/ICASSP.2008.4518572.
13. Gehring, J., Lee, W., Kilgour, K., Lane, I.R., Miao, Y., Waibel, A., et al. Modular combination of deep neural networks for acoustic modeling. In: *Proceedings of Interspeech 2013*; 8. 2013, p. 94–98.
14. Veselý, K., Karafiát, M., Grézl, F.. Convolutive bottleneck network features for LVCSR. In: *Proceedings of ASRU 2011*. ISBN 978-1-4673-0366-8; 2011, p. 42–47.
15. Grézl, F., Karafiát, M., Kontár, S., Černocký, J.. Probabilistic and Bottle-Neck features for LVCSR of meetings. In: *Proc. ICASSP 2007*. Honolulu, Hawaii, USA. ISBN 1-4244-0728-1; 2007, p. 757–760.
16. Grézl, F., Egorova, E., Karafiát, M.. 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. ISBN 978-1-4799-7129-9; 2014, p. 48–53.
17. Tuske, Z., Nolden, D., Schluter, R., Ney, H.. Multilingual MRASTA features for low-resource keyword search and speech recognition systems. In: *Proc. of Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. Florence, Italy: IEEE; 2014, p. 5607–5611.
18. Nguyen, Q.B., Gehring, J., Muller, M., Stuker, S., Waibel, A.. Multilingual shifting deep bottleneck features for low-resource ASR. In: *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. Florence, Italy: IEEE; 2014, p. 5607–5611.
19. Harper, M.. The BABEL program and low resource speech technology. In: *Proc. of ASRU 2013*. 2013.
20. Talkin, D.. A robust algorithm for pitch tracking (RAPT). In: Kleijn, W.B., Paliwal, K., editors. *Speech Coding and Synthesis*. New York: Elseviever; 1995.
21. Laskowski, K., Edlund, J.. A Snack implementation and Tcl/Tk interface to the fundamental frequency variation spectrum algorithm. In: *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. Valletta, Malta. ISBN 2-9517408-6-7; 2010.
22. Grézl, F., Karafiát, M., Janda, M.. Study of probabilistic and Bottle-Neck features in multilingual environment. In: *Proceedings of ASRU 2011*. ISBN 978-1-4673-0366-8; 2011, p. 359–364.
23. Grézl, F., Karafiát, M., Veselý, K.. Adaptation of multilingual stacked Bottle-Neck neural network structure for new language. In: *Proc. of Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. Florence, Italy: IEEE; 2014.
24. Grézl, F., Fousek, P.. Optimizing Bottle-Neck features for LVCSR. In: *2008 IEEE International Conference on Acoustics, Speech, and Signal Processing*. ISBN 1-4244-1484-9; 2008, p. 4729–4732.
25. Grézl, F., Karafiát, M.. Adapting multilingual neural network hierarchy to a new language. In: *Proc. of The 4th International Workshop on Spoken Language Technologies for Under-resourced Languages (SLTU'14)*. St. Petersburg, Russia; 2014.
26. Karafiát, M., Grézl, F., Hannemann, M., Černocký, J.H.. BUT neural network features for spontaneous Vietnamese in BABEL. In: *Proc. of Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. Florence, Italy: IEEE; 2014.