Combination of Multilingual and Semi-Supervised Training for Under-Resourced Languages

František Grézl, Martin Karafiát

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

grezl,karafiat@fit.vutbr.cz

Abstract

Multilingual training of neural networks for ASR is widely studied these days. It has been shown that languages with little training data can benefit largely from the multilingual resources for training. The use of unlabeled data for the neural network training in semi-supervised manner has also improved the ASR system performance. Here, the combination of both methods is presented. First, multilingual training is performed to obtain an ASR system to automatically transcribe the unlabeled data. Then, the automatically transcribed data are added. Two neural networks are trained - one from random initialization and one adapted from multilingual network - to evaluate the effect of multilingual training under presence of larger amount of training data. Further, the CMLLR transform is applied in the middle of the stacked Bottle-Neck neural network structure. As the CMLLR rotates the features to better fit given model, we evaluated whether it is better to adapt the existing NN on the CMLLR features or if it is better to train it from random initialization. The last step in our training procedure is the fine-tuning on the original data.

Index Terms: feature extraction, neural networks, stacked bottle-neck, multilingual training, semi-supervised training

1. Introduction

One of today's interests in speech recognition community is the development of ASR system with only limited resources from the target domain. An example of such focus is the IARPA BABEL project with the goal of developing methods to build speech recognition technology for any spoken language with little training data that is also much noisier and more heterogeneous than the one used for training current state-of-the-art ASR systems. This requires innovations and techniques to rapidly model a novel language.

One of the key components in today's state-of-the-art systems are neural networks (NNs) in the role of either feature extractor [1] or acoustic model [2]. As the time passed, the computation power increased tremendously and allowed to train large neural networks on huge speech databases. The challenge of these days was to train networks on this huge data [3]. Today, the opposite problem is faced: how to make neural networks work well with little in-domain training data.

The clue to the solution is actually hidden in the problem definition: "*little in-domain training data*". Thus the focus turned to leveraging the out-of-domain and non-training data.

The out-of-domain data are mostly speech collections from other languages. The recent research in this area has generated several methods of multilingual training for neural networks. The performance of multilingual NN feature extractor is evaluated in [4]. The ways of compacting the multilingual phoneme set were also studied here. Using language-specific output layer, while keeping the main body of NN multilingual, was proposed in [5]. In both cases, no data from target language is used for NN training. The adaptation of multilingual network by fine-tuning on the target domain data can boost system performance [6]. On the other hand, the issue of language-specific phonemes arises. Vu et al. [7, 8] approximated such phoneme by several ones from the training languages so that, in combination, they had the characteristic of the target phoneme. Different architectures of NNs and reuse of trained monolingual NNs were also examined: [9] shows the performance of NN with the final part being language-specific (the last 2 layers). A modular multilingual system is studied in [10]. Our latest work [11] evaluated multilingual training and adaptation strategies of Stacked Bottle-Neck NN architecture.

The non-training data is untranscribed data not prepared for training. But this data can be transcribed automatically and used together with the training one – this leads to semi-supervised training techniques. These techniques are well studied in the GMM training framework. The untranscribed data can be labeled either on segment level by the full ASR system [12, 13] or on data-point level by assigning it the label of the closest labeled one internally during the training process [14]. In these cases, selecting reliable segments/data-points is necessary. It is also possible to assign them soft weights by some kind of confidence measure. Other semi-supervised training methods incorporate the uncertainty of unlabeled data into the objective functions and minimize its entropy [15, 16, 17], or are based on feature-space manifold assumption using a graph-based framework [18, 19], where the nearest supervised data-point suggests the label.

In the neural network training framework, these techniques have been studied only recently. In the work of Huang [20], automatic transcriptions are obtained by a ROVER fusion of three systems and confidences are re-calibrated by per-word degree of agreement. Thomas et al. [6] use combination of per-word C_{max} confidence and MLP posteriorgram phoneme occurrence confidence, which is used for sentence-level data selection. In our work [21], we rely on a single DNN system, and use per-

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.

frame confidences for frame-weighted training. In [22], the data segments for training a neural network feature extractor were selected based on their C_{max} measures.

The goal of this work is to show a way to quickly build a recognition system on little transcribed data while having resources from other languages as well as in-domain untranscribed data. It directly extends our past work in multilingual training [11] where we found the optimal approach to multilingually train the Stacked Bottle-Neck (SBN) (originally called "Universal Context") NN hierarchy [23] and to adapt it to the target language. Here, we describe semi-supervised training on top of the multilingual one. New forced alignment is obtained with recognition system based on adapted multilingual SBN NNs. Also, untranscribed portion of language-specific data is automatically transcribed and C_{max} confidence measures are obtained. The approach used in [22] is evaluated in two scenarios: Training of NNs from random initialization and adapting the multilingual SBN with manually and automatically transcribed data.

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. 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) consisting 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.

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 remaining data in FLP are regarded as untranscribed and will be used in semi-supervised manner.

The characteristics of the languages can be found in [24]. More detailed statistics for evaluation languages are given in Tab. 1. The reported amounts of data for FLP and LLP refer to the speech segments after dropping the long portions of 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 [23]. It is depicted in Fig. 1.

The NN input features are composed of critical band energy (CRBE) features and fundamental frequency ones. As critical band energy features, we use logarithmized outputs of 24 Mel-scaled filters applied on squared FFT magnitudes. The fundamental frequency features consist of F0 and probability of voicing estimated according to [25] and smoothed by dynamic programming, F0 estimates obtained by Snack tool[2] function *getf0* and seven coefficients of Fundamental Frequency Variations spectrum [26, 27]. Together, there are 10 F0 related coefficients. The resulting feature vector will be referred to as *CRBE+F0s*. It provides 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 of *CRBE+F0s* 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 $34 \times 6 = 204$ coefficients on the first stage NN input. The whole data set is mean and variance normalized.

The first stage NN in stacked bottle-neck 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, which improves recognition performance over the sigmoid activations [28]. The BN layer outputs are stacked (hence Stacked Bottle-Neck) over 21 frames and downsampled by factor of five before entering the second stage NN. The second stage NN is the same as the first one with exception of BN layer size. In this NN, it has 30 units. Outputs of the second stage NN BN layer are the final outputs forming the BN features for GMM-HMM recognition system.

Tied triphone states are used as NN targets. Features obtained from NNs trained towards these targets provide consistently slightly better performance than context-independent phone-state 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 by removing the long silences at the ends of the segments and splitting one segment into two when silence longer than 300ms was detected. We also cut out the parts of segments which were labeled as "unknown" (generally unintelligible speech).

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 HLDA-PLP initial system. 12 Gaussian components per state were found to be sufficient for MLLT-BN features. 12

Table 1: *Statistics of the data. The LM and dictionary statistics are taken from LLP which is used to train the 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

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

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

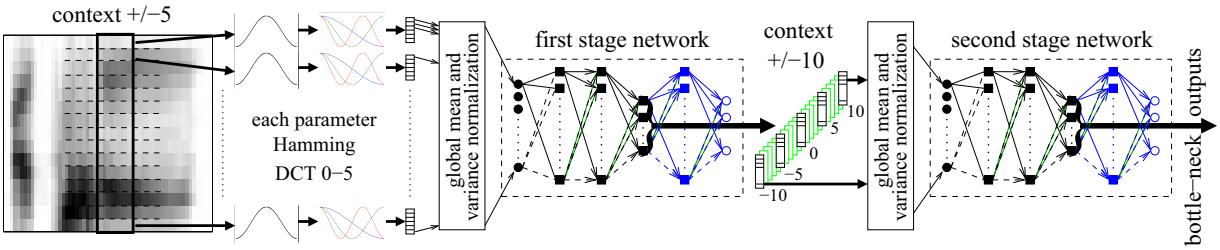

Figure 1: Block diagram of Stacked Bottle-Neck feature extraction. The blue parts of NNs are used only during the training. The green frames in context stacking between the NNs are skipped. Only frames with shift -10, -5, 0, 5, 10 form the input to the second stage NN.

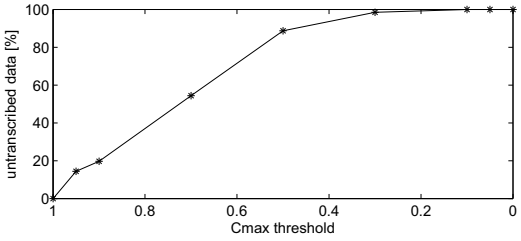

Figure 2: Percentage of untranscribed with C_{max} bigger than the given threshold. Average over five evaluation languages.

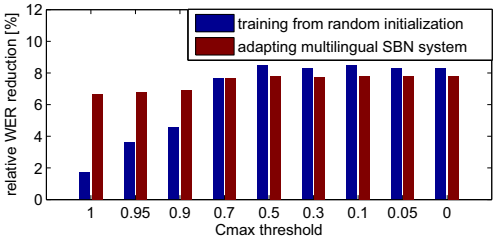

Figure 3: Percentage of untranscribed with C_{max} bigger than the given threshold. Average over five evaluation languages.

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[3].

2.4. Multilingual SBN training and adaptation

The NN 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 accommodating either full or compacted target set [4]. The context-independent phoneme states were used as training targets in multilingual NN training. The adaptation of trained NN to target language is done in two steps:

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

This process is the same for both NNs in SBN hierarchy and provided the best results in our former work although adapting the first NN basically changes the inputs to the second one so it could have problems with adaptation. But it appears that NN can adapt also to slight changes in input features.

The context-dependent phoneme states were used as training targets in the adaptation phase.

GMM-HMM system is trained on the MLLT-BN features from adapted multilingual SBN hierarchy. The automatic transcriptions and new forced alignments are obtained with this system.

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

3. Experiments

3.1. Semi-supervised training

In this step, we evaluate the effect of adding automatically transcribed data to the training set. The data are selected in the same way as in [22]. Fig. 2 shows the percentage of added data with respect to chosen C_{max} threshold. The total amount of untranscribed data is FLP size minus LLP size – see Tab. 1. The selected data are added to the LLP part and two systems are trained: One from random initialization and another adapted from multilingual NNs as described in Sec. 2.4. We would like to compare how efficiently the new data is used in both cases and at which level both systems would perform about the same – i.e. when the adaptation of multilingual NNs will not bring any advantage over training from random initialization. However, note that training from random initialization takes advantage of multilingual system in the form of better forced alignment.

The average relative improvements over the LLP baseline are shown in Fig. 3. It can be seen that adaptation of multilingual NNs achieves higher improvements for higher C_{max} values over the training from random initialization. The performances of both systems become equal for $C_{max} = 0.5$. This is also the optimal threshold and decreasing it (adding more, but less reliable, data) does not further improve the performance.

We can also evaluate the effect of new forced alignment on just LLP set. This alignment is generated after adaptation of multilingual NNs to a new language. The effect can be seen on the differences in bars “multiling adapt” and “LLP only” in Fig. 4. The “LLP only” bar is the same as bar for $C_{max} = 1$ in Fig. 3. Note that there is no multilingual adaptation for “training from random weights” track.

3.2. Speaker-adaptive neural network training

The optimal semi-supervised training scenario was taken to perform the next step in the system building – speaker adaptive training of neural networks.

In our recent work [29], we discussed several strategies

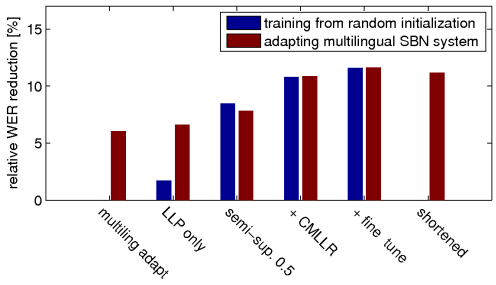

Figure 4: Overview of the WER reduction in individual steps. Average over five evaluation languages.

for NN adaptation and compared two CMLLR adaptation approaches in combination with neural network training:

1. Adaptation of SBN input features: CRBEs were decorrelated using DCT, adapted by CMLLR and projected back into original space by inverse DCT.
2. Adaptation of the SBN inner product – output of the 1^{st}stage NN. BN outputs are known to be far less correlated, so the transform could be applied directly.

The second approach was found more effective. Moreover, according to our analysis, the first-stage NN is doing mainly the acoustic feature extraction and only the second-stage NN is processing acoustic clues in wider context. Therefore, it is more convenient to place speaker-specific block between the two NNs and thus follow the classical speech recognition scenario – feature extraction, speaker adaptation, acoustic modeling.

The 1^{st}stage BN-CMLLR features are used in the following ways:

- New 2^{nd} stage NN is trained from random initialization.
- The existing 2^{nd} stage NN is retrained. We considered not only the fully trained NN for retraining, but also the NNs several iterations before the training ended to allow the retraining to find a possibly better optimum.

Two sets of 1^{st}stage BN-CMLLR features were generated - one originated from semi-supervised training from random initialization and another from multilingual SBN adaptation.

We have found that training new 2^{nd} stage NN from random initialization gives slightly lower word error rate ($\sim 0.2\%$) for all languages than the adaptation of existing NN. There is no difference when adaptation of NN starts from different epochs of the original training. The relative improvement over the LLP baseline after adding the speaker adaptive training is shown in Fig. 4, see the "+ CMLLR" bars.

3.3. Fine tuning

We proceed with the speaker-dependent SBN system where the 2^{nd} stage NN is trained from random initialization. The goal of this step is to fine tune the system with reliable data only.

At the beginning, we have done (on one language – Assamese) a thorough evaluation looking for the optimal starting point – the initial neural network for the tuning – and portion of data to use for the fine tuning. For this step, only the LLP data were considered and the oracle error rate measured as the probability of transcription in the recognition lattice was used to filter out less reliable segments.

Our analysis shows only marginal differences in the results. Thus we proceed with using the whole LLP data and final NN from previous step. The results are shown in Fig. 4, the "+ fine tune" bars.

Table 2: *WER [%] obtained on Bengali – multilingual track*

LLP baseline	multiling adapt	+semi--sup 0.5	+CMLLR	+fine tune	shortened
69.7	65.9	64.9	62.7	62.2	62.5

3.4. Shortening the pipeline

As you can see, our optimal pipeline ended up with several trainings – we train the SBN system in semi-supervised manner and then the 2^{nd} stage NN is trained in speaker adaptive training on the 1^{st}stage BN-CMLLR features. For quick build-up of a system, it would be good to eliminate some of these trainings. It is obvious that training the 2^{nd} stage NN in the semi-supervised training step is not necessary as the CMLLR transforms are trained on the outputs of 1^{st} stage NN.

But there is one more training which may be skipped – the 1^{st} stage NN need not be trained in the semi-supervised step. As mentioned above, it performs the acoustic feature extraction. When it stays the same (i.e. adapted from multilingual NN), the second stage NN should compensate this slight difference.

In our shortened pipeline, the CMLLR transforms are computed on the 1^{st} stage NN from the multilingual system adapted to the target language. Then, the new 2^{nd} stage NN is trained on 1^{st}stage BN-CMLLR features of LLP plus automatically transcribed data. This NN is fine-tuned with the LLP data. In this way, not only the time needed for neural network training is saved, but also the CMLLR training and generation of automatic transcription can be done at the same time.

The results are shown in Fig 4, bars "shortened". As can be seen, the degradation of performance compare to the full pipeline is very small. Note, that the pipeline cannot be shortened without having the multilingual NNs.

4. Conclusions

We show further improvements on top of the multilingual SBN systems adapted to target language. These systems reduce the WER by 4-8% relative to the LLP baseline. The semi-supervised training reduces the WER by additional 1-3% relative. The semi-supervised training was performed in two ways: starting from random initialization and by adapting the multilingual NNs. In this phase, the training from random initialization performs slightly better.

In the next step, the speaker-adaptive neural network training was done by the means of CMLLR transform applied on the BN outputs from the first stage NN in SBN hierarchy. The additional improvement is between 2.5 and 5% relative. Fine tuning on LLP data adds up to 1.5%. Overall, the improvement was almost doubled by our effort. The averaged results from individual steps are shown in Fig. 4, all results with per-languages figures can be found at `www.fit.vutbr.cz/~grezl/IS2014`. The largest WER reduction was achieved for Haitian (5.9 by multilingual system; 14.2 with full pipeline), the smallest for Zulu (4.0; 7.4). The results for the multilingual track of Bengali (chosen as "average" language) are given in Tab. 2.

Significantly simplified pipeline was also evaluated. It consists of only one NN training on top of the 1^{st}stage BN-CMLLR features (computed directly from the initial adapted multilingual SBN) and fine tuning. The drop in relative WER reduction is between 0.1 and 1.3%. This is only small price for simplification and speed-up we gain.

5. References

[1] S. Sivadas, "Tandem feature extraction for automatic speech recognition," Ph.D. dissertation, OGI School of Science & Engineering Oregon Health & Science University, Nov. 2004.

[2] N. Morgan and H. Bourlard, "Continuous speech recognition: An introduction to the hybrid HMM/connectionist approach," *IEEE Signal Processing Magazine*, vol. 12, no. 3, pp. 25–42, 1995.

[3] Q. Zhu, A. Stolcke, B. Chen, and N. Morgan, "Using MLP features in SRI's conversational speech recognition system," in *Proc. INTERSPEECH 2005*, Lisbon, Portugal, Sep. 2005.

[4] 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.

[5] S. Scanzio, P. Laface, L. Fissore, R. Gemello, and F. Mana, "On the use of a multilingual neural network front-end," in *Proceedings of INTERSPEECH-2008*, 2008, pp. 2711–2714.

[6] S. Thomas, M. Seltzer, K. Church, and H. 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.

[7] N. T. Vu, F. Metze, and T. 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.

[8] N. T. Vu, W. Breiter, F. Metze, and T. Schultz, "An investigation on initialization schemes for multilayer perceptron training using multilingual data and their effect on ASR performance," in *Proc. Interspeech*, 2012.

[9] J. Gehring, Q. B. Nguyen, F. Metze, and A. Waibel, "Multilingual acoustic models using distributed deep neural networks," in *ICASSP*, 2013, pp. 8619–8623.

[10] G. Heigold, V. Vanhoucke, A. W. Senior, P. Nguyen, M. Ranzato, M. Devin, and J. Dean, "DNN acoustic modeling with modular multi-lingual feature extraction networks," in *Proc. of ASRU 2013*, Dec 2013.

[11] F. Grézl and M. Karafiát, "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, May 2014.

[12] L. Lamel, J.-L. Gauvain, and G. Adda, "Lightly supervised and unsupervised acoustic model training," *Computer Speech & Language*, vol. 16, no. 1, pp. 115–129, 2002.

[13] L. Wang, M. Gales, and P. Woodland, "Unsupervised training for Mandarin broadcast news and conversation transcription," in *Proc. ICASSP*, vol. 4. IEEE, Apr 2007.

[14] A. Subramanya and J. Bilmes, "The semi-supervised Switchboard transcription project," in *Proc. INTERSPEECH 2009*, Sep 2009.

[15] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *Proc. of NIPS*, 2004.

[16] J.-T. Huang and M. Hasegawa-Johnson, "Semi-supervised training of Gaussian mixture models by conditional entropy minimization," *Optimization*, vol. 4, p. 5, 2010.

[17] D. Yu, B. Varadarajan, L. Deng, and A. Acero, "Active learning and semi-supervised learning for speech recognition: A unified framework using the global entropy reduction maximization criterion," *Computer Speech & Language*, vol. 24, no. 3, pp. 433–444, 2010.

[18] T. Joachims, "Transductive inference for text classification using support vector machines," in *machine learning-international workshop then conference*. Morgan Kaufmann Publishers, INC., 1999, pp. 200–209.

[19] J. Malkinn, A. Subramanya, and jeff Bilmes, "On the semi-supervised learning of multi-layered perceptrons," in *Proc. INTERSPEECH 2009*, Sep 2009.

[20] Y. Huang, D. Yu, Y. Gong, and C. liu, "Semi-supervised GMM and DNN acoustic model training with multi-system combination and confidence re-calibration," in *Proc. of INTERSPEECH 2013*, 2013, pp. 2360–2364.

[21] K. Veselý, M. Hannemann, and L. Burget, "Semi-supervised training of deep neural networks," in *Proc. of ASRU 2013*, Dec 2013.

[22] F. Grézl and M. Karafiát, "Semi-supervised bootstrapping approach for neural network feature extractor training," in *Proceedings of ASRU 2013*, 2013, pp. 470–475.

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

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

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

[26] K. Laskowski, M. Heldner, and J. Edlund, "The fundamental frequency variation spectrum," *in Proceedings of FONETIK 2008*, pp. 29–32, 2008.

[27] K. Laskowski and J. Edlund, "A Snack implementation and Tcl/Tk interface to the fundamental frequency variation spectrum algorithm," in *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may 2010.

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

[29] M. Karafiát, F. Grézl, M. Hannemann, and J. H. Černocký, "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, May 2014.