Interspeech 2018
2-6 September 2018, Hyderabad

BUT OpenSAT 2017 speech recognition system

Martin Karafiát, Murali Karthick Baskar, Igor Szoke, Vladimír Malenovský, Karel Veselý, František Grézl, Lukáš Burget, and Jan "Honza" Černocký

Brno University of Technology Speech@FIT and IT4I Center of Exellence
Brno, Czechia

{karafiat,baskar,szoke,malenov,grezl,iveselyk,cernocky}@fit.vutbr.cz

Abstract

The paper describes BUT Automatic Speech Recognition (ASR) systems for two domains in OpenSAT evaluations: Low Resourced Languages and Public Safety Communications. The first was challenging due to lack of training data, therefore multilingual approaches for BLSTM training were employed and recently published Residual Memory Networks requiring less training data were used. Combination of both approaches led to superior performance. The second domain was challenging due to recording in extreme conditions: specific channel, speaker under stress, high levels of noise. A data augmentation process was very important to get reasonably good performance.

Index Terms: speech recognition, multilingual training, BLSTM, data augmentation, robustness

1. Introduction

The NIST Speech Analytic Technologies (OpenSAT) pilot evaluation running in June 2017 focused on three tasks: Automatic Speech Recognition (ASR), Speech Activity Detection (SAD) and Keyword Search (KWS) and on three domains: Low Resourced Languages (Babel), Speech from Video - (VAST, with SAD task only) and Public Safety Communications (PSC).

Our team decided to work on the ASR task for both possible domains: Low Resource Languages and Public Safety Communications. While the first one is challenging by limited resources, the second one is addressing English but contains a lot of speech under stress conditions recorded through specific channel. To be successful in both challenges, several acoustic models were built (section 2) on top of various feature extraction schemes (section 3). The detailed system descriptions are given in sections 4 and 5 and results are presented in sections 4.2 and 5.3.

1.1. Low Resource Languages

Pashto language was selected for this challenge from the recently completed IARPA Babel program [1], therefore this challenge is later noted as Babel. In Babel, data from 24 low-resource languages were collected, which allowed us to focus on multilingual experiments for feature extraction and acoustic modeling. Moreover, substantial improvement of generic monolingual acoustic models was also achieved. BUT worked on Babel as part of "Babelon" team (led by BBN).

ASR systems tend to produce system dependent errors, therefore system complementarity and system fusion are crucial for good performance. Therefore, the fusion of multilingual and monolingual systems trained on different features was important for our final system.

1.2. Public Safety Communications

The target data for this challenge is taken from dispatcher logs from the Sofa Super Store Fire (SSSF) that occurred on June 18, 2007 in Charleston and claimed the lives of nine U.S. firefighters (the challenge will be later called SSSF). The data is real operational data, therefore it cannot be duplicated through controlled scientific collection. It contains sensitive and disturbing content (e.g. pleas from trapped fire fighters) therefore it has to be treated respectfully. The data is really challenging due to:

- Various transmission effects between land, mobile and radio systems.
- Speech under cognitive and physical stress.
- Varying background noise types and levels.

Here, our system was built on a simulated channel (see section 3.4 attempting to close the gap between real and clean data.

2. Acoustic modeling approaches

2.1. DNN and BLSTM

Common hybrid Deep Neural Network (DNN) acoustic models have already been replaced by more accurate Recurrent Neural Network (RNN) architectures (such as Long-Short Term Memory – LSTM) and their bi-directional variant (BLSTM) [2] in state-of-the-art speech recognition systems. Although DNNs are less accurate than RNN based architectures, they are significantly faster due to lower complexity and also usually better generalize to unseen data. Therefore, we used DNNs in our SSSF system due to expected data mismatch. DNN architecture in this work has 7 layers and 2048 sigmoid neurons in each layer. Restricted Boltzmann Machine (RBM) pre-training was used to get better starting point for cross-entropy (XE) training.

The BLSTMs are based on latency-controlled BLSTM architecture [3] with 3 bi-directional layers. For each direction, there were 512 memory units and 300 dimensional projection layer as suggested in [4].

2.2. Multilingual BLSTM system

Its very natural for humans to borrow information from other sources when trying to learn a new language. Humans share the same vocal tract architecture and phonetic systems of languages overlap, therefore automatic systems should be able to have the low-level components (feature extraction and partially also acoustic models) built and trained on various data sources.

During Babel, we verified [5] that multilingual pre-training for feature extraction is an important technique especially if not-enough training data is available. Recently, we also extended multilingual DNN acoustics models [6] to BLSTM [7] and presented significant gains with adding more languages into acoustic model training. Moreover, low-dimensional i-vector based adaptation was also investigated and detailed in Section 2.3.

Compared to standard BLSTM described in the previous section, Multilingual BLSTM architecture has the last output layer divided into parts according to individual languages. During the training, only the part of the output layer correspond-

10.21437/Interspeech.2018-2457

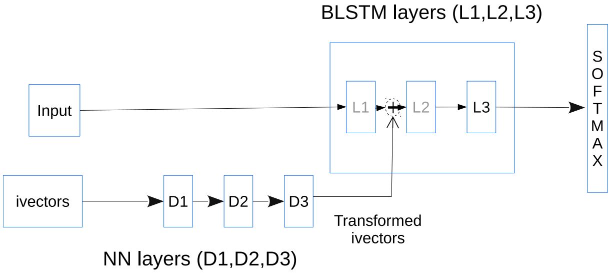

Figure 1: *Adding i-vector to multilingually pre-trained NN*

ing to the target language is activated. Originally [8], this block-softmax layer, was successfully trained with context-independent phoneme states as targets. We intended to use the states of tied context-dependent phonemes, but so far it turned out to be unfeasible.

The procedure of porting the multilingual NN to a new language can be described in the following steps: (1) the final multilingual layer (context-independent phoneme states for all languages) is stripped and replaced with a layer specific to target-language (tied-state triphones) with random initialization. (2) This new layer is trained for 8 epochs with a standard learning rate, while the rest of the NN is fixed. (3) Finally, the whole NN is fine-tuned with 10 epochs, the initial value of learning-rate schedule is set to 0.5 of the original value.

2.3. Integration of i-vector

Typically, the low-dimensional speaker vector adaptation involves concatenating input feature vectors with speaker-specific vector that is constant across the whole utterance [9]. This approach is however not feasible with multilingually pre-trained NNs as a change of architecture is not practical. Therefore, the speaker specific i-vector (see section 3.3) is added into input of the first or the second layer in a similar way as presented in [10]. The vector is transformed by “ivector NN” in such way that makes it additive to input layer of the main NN and leds to improving training criteria (cross-entropy of classified states). The details are shown in Figure 1. Note, that this system was naturally used in the Babel part of the challenge.

2.4. Residual memory network

Recently, we have presented Residual Memory Networks (RMN) [11], which are trying to overcome drawbacks of common state-of-the-art acoustic models: 1) *Recurrent Neural Network* architectures are difficult to train [4] when extended to deeper structures, which are essential for learning more abstract information. 2) *Feed-forward neural networks* can be made much deeper, lead to better generalization to unseen data and are less prone to over-fitting. But they fail to perform well for tasks requiring long-term information.

RMN is a variant of DNN where the number of layers denotes both the temporal length to learn and the structural depth as shown in Figure 2. The contributions of RMN are:

- RMNs use a feed-forward architecture inspired by RNN . Different RMN variants were originally proposed in [11]. Figure 2. shows the variant used in this work, where n-th layer processes its input as follows: The input vector is transformed by layer specific affine transformation W_n. The resulting vector is element-wise summed with the same vector delayed by $N-n+1$ frames and further transformed by ”memory component” affine transformation. Finally, non-linearity is applied to produce the layer's output. Note that the ”memory component” transformation is shared by all layers.

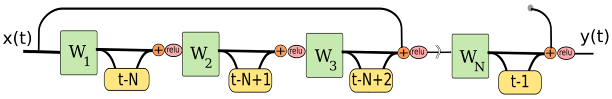

Figure 2: *RMN: showing memory component and residual connection*

- RMNs use residual connections after every few layers, which allows us to increase the network depth, make training faster and improves the recognition performance. During back-propagation, the residual lines allows for unimpeded flow of gradients.

The combination of these two components allows RMN to learn long-term dependencies and higher level abstractions simultaneously in a much simpler and efficient way. Bi-directional RMN (BRMN) is a simple extension to RMN, where extra ”memory components” are added to make predictions not only from past frames, but (symmetrically) also from future frames.

Our BRMN network contains 20 layers each having 1024 hidden units and residual connections bypassing over every 5 layers. The training is done with minibatch size 256, learning rate of 1×10^{-3} and $l2$ regularization constant 1×10^{-4}.

3. Front-End processing

3.1. Features

Our features are 24 log Mel filter bank energies and various estimates of fundamental frequency (F0). Four F0 estimators are used:

- BUT F0 - F0 and probability of voicing (2 coefficients) obtained by our tool implemented according to [12]).
- GetF0 - 1 coefficient obtained using snack library[1].
- Kaldi F0 - 3 coefficients (F0 normalized over sliding window, probability of voicing and F0 delta) [13].
- Fundamental Frequency Variations (FFV) - continuous 7-dimensional vector representation of F0 variation, obtained by comparing the harmonic structure of the frequency magnitude spectra of the left and right half of an analysis frame [14].

The whole feature vector has 37 coefficients, these features are later called FBANK_F0.

Conversation-side based mean subtraction is applied on the speaker basis and 11 frames are stacked. Hamming window followed by DCT consisting of 0th to 5th base are applied on the time trajectory of each parameter (37×6) resulting in 222 coefficients at the first stage NN input. These temporal trajectories (TRAPs) were first investigated in [15, 16] and recently used also in multilingual BLSTM systems [17]. These features are later called FBANK_F0_TRAP.

3.2. Stacked Bottle-Neck Neural Network

Stacked BNeck Neural Network (SBN) architecture [18] is a hierarchical composition of two Neural Networks - “Context NN” and “Merger”. The first-stage NN is trained on top of FBANK_F0_TRAP features and has five hidden layers with 1500 units each except the BN layer. The BN layer is the fourth hidden layer and its size is 80 neurons. Its BN outputs (SBN_1stage) are stacked over 21 frames and down-sampled before entering the second-stage NN. 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 later called SBN

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

features. Neurons in both BN layers have linear activation functions as they were reported to provide better performance [19].

3.3. i-vector extraction

We used 19 MFCC coefficients + energy and their delta and double delta coefficients, resulting in in 60-dimensional feature vectors. The silence frames were removed according to Voice Activity Detection (VAD), after which we applied short-time (300 frame window) cepstral mean and variance normalization. The MFCC features were augmented with SBN features trained trained on Babel languages.

A gender-independent UBM was represented as GMM with 512 diagonal-covariance components. It was trained on the target language data. Finally, gender-independent i-vector extractor was trained (in 10 iterations of a joint Expectation Maximization and Minimum Divergence steps) on the same data-set as the UBM. More details on i-vector extraction can be found in [20]. The results are reported with 100-dimensional i-vectors.

3.4. SSSF training data enhancement

Uniqueness of the SSSF target data lead to special processing of clean speech training data in two main parts:

Adding noise samples: Various noise sounds were downloaded from freesound.org, about 7 minutes of running engines were used at the end. Furthermore, we downloaded about 91 minutes of fire engines passing, parades, etc. from YouTube. Finally, we used SSSF development data segmentation and extracted 87 minutes of non-speech audio. All the data (185 minutes in total) was split into 63 segments (no longer than 4 minutes).

Next, each audio file from the training dataset was corrupted by a single, randomly selected, noise segment. The starting position of the noise segment was randomly selected. If end of file was reached before filling the requested length, the noise file was re-started. The target SNR was chosen randomly from the interval -25dB to -7dB.

Passing data through the "SSSF channel": We have analyzed the SSSF channel by listening and basic spectral analyses. First, we tried to simulate the HW (radio station) and ambiance in the following way:

1. Normalization of the training audio to 0 dB gain.
2. Increasing gain from 0dB to 20dB to introduce clippings.
3. Application of high-pass filter at 300Hz, 600Hz, 1000Hz, and 1500Hz randomly.
4. Application of the phaser effect with sox tool to simulate phase distortions.

The fire departments in the U.S. use the proprietary AMBE codec in their Digital Mobile Radio (DMR). The source code of this codec is not available. As a replacement we used the EU version of the TETRA codec [21]. to simulate the effects of signal coding. We assume that the TETRA codec has similar characteristics as the AMBE codec. Therefore, after adding noise and simulating the HW and the ambiance, we passed the data through the TETRA codec.

4. BABEL system

Our systems were built with several toolkits: We used STK/HTK [22] toolkit[2] for feature extraction. Kaldi [23] was used for maximum likelihood (ML) Gaussian mixture model (GMM) training. Finally, DNN, BLSTM and RMN networks are trained using CNTK [24].

GMM based acoustic models were trained to produce phoneme alignments as the labels for the following NN training. These models based on cross-word tied-states were trained from scratch using standard ML algorithm. The baseline GMM systems have approximately 4000 cross-word triphone tied states for BABEL and 9100 for SSSF.

Our BABEL system is based on standard ROVER fusion of two complementary systems:

- *Multilingual BLSTM* was pre-trained on 11 Babel languages using BANK_F0_TRAP features as explained in Section 4.1.
- *Bi-directional RMN acoustic model* was trained on target language only and on 80 dimensional SBN_1stage features concatenated with 100-dimensional ivectors as mentioned in sections 3.3 and 3.2.

Both systems were trained using XE criteria followed by state-Minimum Bayes Risk (sMBR) discriminative criteria.

Voice Activity Detection was trained on languages from section 4.1. It was based on feed-forward Neural Network with 2 outputs and it was used for cepstral mean and variance normalization and for the definition of decoding segments.

Standard 3-gram back-off ARPA model was used for decoding. The acoustic model training data transcriptions were used for the training. In addition, WEB data generated during Babel program by BBN was added to LM training [25]

4.1. Data

Multilingual BLSTM acoustics models were pre-trained on 11 Babel Languages (all languages available on LDC at that time): Cantonese, Pashto, Turkish, Tagalog,Vietnamese, Assamese, Bengali, Hait. Creole, Lao, Swahili, Georgian. The final BABEL system was trained only on given Pashto data-set (cca 99 hours).

Table 1: *WER on dev data.*

System	WER[%]
RMN	42.2
RMN+ivec	41.8
RMN+webLM	42.0
RMN+ivec+webLM (ctm1)	41.4
BLSTM	42.4
MultBLSTM	42.0
MultBLSTM+ivec	40.6
MultBLSTM+ivec+webLM (ctm2)	40.3
Rover (Primary) (ctm1+ctm2)	38.7

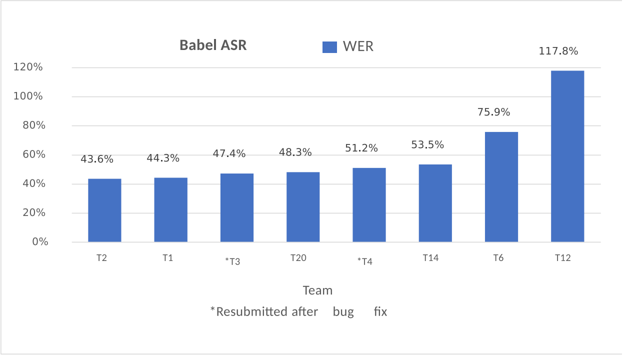

Figure 3: *Pashto evaluation results. T2 is BUT Primary system.*

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

4.2. Results

Table 1 presents performances on development (dev) set. We observed tiny (0.4%) gain from using multilingual BLSTM (MultBLSTM) over the monolingual one. It is probably due to the sufficient amount of training data (92 hours). RMN provides performance similar to MultBLSTM as expected and the ROVER combination of both systems provide an impressive gain of 1.6% absolute. It is a proof of our assumption that the systems are highly complementary due to the different architectures and features. This system also performed favorably on the eval data (43.6%) as compared to the other evaluation participants (see figure 3).

5. SSSF system

The system was build on top of the GMM based alignments generated by mix-up training in similar way to the Babel case (section 4). It contains 9100 cross-word triphone tied states. The alignments were generated using GMM-HMM system trained on the clean data and were reused for the systems trained on the noised data (enhancement described in section 3.4).

We experimented with two types of acoustic models:

- *DNN* trained on top of SBN_1stage features generated for the "*Full*" data set (see later in sec. 5.2). Note, that the NN based feature extractor was taken from our ASR system already pretrained for telephone speech to save training time. No sMBR training was used.
- *BLSTM* was trained with FBANK_F0 features as input. Due to the time limitations, the training run only on "*10persspeaker*" data set (see later in sec. 5.2). The final models were trained in 11 epochs with XE criterion followed by one epoch with sequence state-Minimum Bayes Risk (sMBR) objective function.

5.1. Voice Activity Detection

Voice activity detection was re-used from RATS project, as the channel is similar to SSSF data. It was performed by NN with an input of block of log Mel filter bank outputs with 300ms context width. The NN has 18 outputs: 9 for speech and 9 for non-speech, each corresponding to one of the channels (source plus 8 re-transmitted) from RATS data. HMM with Viterbi decoding was used to smooth out and merge the outputs to speech and non-speech regions. This NN was trained on RATS data defined for the speech activity detection (SAD) task [26].

5.2. Data

Various English corpora were used for **acoustic model** training: English Fisher1+2, Switchboard 1 Release2, Call Home English, AMI and ICSI-meetings. This "*Full*" set had 2240 hours. Next, we created a 230 hour subset,"*10perspeaker*", where only 10 sentences were selected per speaker to limit the training time. **Language Model** was 3-gram trained on various corpora with different weights:

Data	# Vocab	weight
Fisher (Part 1 + Part 2)	21.2M words	0.03
CNN transcripts	64.5M words	0.02
AMI meetings	0.8M words	0.01
ICSI meetings	0.8M words	0.01
Switchboard + CallHome	3.5M words	0.01
Open Subtitles	61.1 M words	0.24
SSSF dev	2.6k words	0.65

We split SSSF dev set into two parts and tuned the weights independently. Then we averaged them and did some small hand corrections (rounding). We used CMU pronunciation dictionary and limited their vocabulary to 48k words (original vocabulary size was about 220k words) by dropping the infrequent words. Next, we applied perplexity pruning ($p = 1 \times 10^{-8}$) to limit the size of the language model.

Table 2: *WER on dev data.*

System	VAD	WER[%]
DNN	STM	63.6
DNN	RATS	65.3
BLSTM	STM	63.2
BLSTM	RATS	66.1

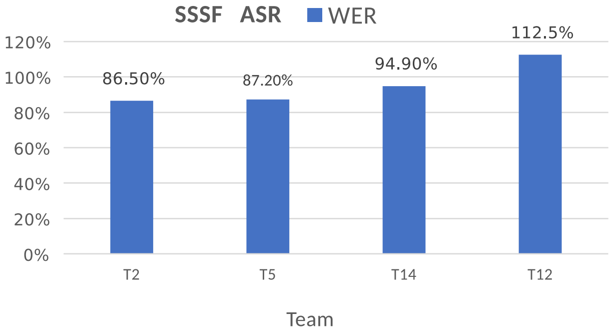

Figure 4: *WER on eval data. T2 marks BUT system.*

5.3. Results

Table 2 presents the performances on the development data set. The significant difficulty of the data is reflected in the word error rates. The lowest WER in the table is 63.2% obtained with BLSTM with reference segmentation. Note, that the dev set transcriptions were used for the language model training as well. In more detailed analyses, we found that dispatcher logs are transcribed reasonably and most of the errors are coming from the firefighters. It can be attributed to the Lombard effect caused by the specific environment and the different speaking style (not following any grammar). The automatic segmentation results in performance 1.7-2.9% worse than the reference one. Finally, we decided to submit DNN system as it performs slightly better than BLSTM one for the automatic segmentation. Figure 4 presents evaluation results for all the participating teams. Our team performed the best although the word error rates are still very poor.

6. Conclusion

The paper presented our effort for OpenSAT pilot evaluation 2017. We participated in two domains: Low Resource Languages and Public Safety Communications. In the first one, we confirmed the importance of creating highly complementary systems as well as using multilingual approaches. In the second one, we presented specific training data enhancement approach, which led to the best performing system in the evaluations. Nevertheless, the performance of this system is still very poor, which leaves us space for experimenting with more advanced adaptation techniques in the future.

7. Acknowledgements

The work was supported by Czech Ministry of Interior project No. VI20152020025 "DRAPAK", Technology Agency of the Czech Republic project No. TJ01000208 "NOSICI", and by Czech Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project "IT4Innovations excellence in science - LQ1602".

8. References

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

[2] M. Schuster and K. Paliwal, "Bidirectional recurrent neural networks," *Trans. Sig. Proc.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1109/78.650093

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

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

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

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

[7] M. Karafiát, K. M. Baskar, P. Matějka, K. Veselý, F. Grézl, L. Burget, and J. Černocký, "2016 BUT babel system: Multilingual BLSTM acoustic model with i-vector based adaptation," in *Proceedings of Interspeech 2017*, 2017.

[8] K. Veselý, M. Karafiát, F. Grézl, M. Janda, and E. Egorova, "The language-independent bottleneck features," in *Proceedings of IEEE 2012 Workshop on Spoken Language Technology*. IEEE Signal Processing Society, 2012, pp. 336–341.

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

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

[11] K. M. Baskar, M. Karafiát, L. Burget, K. Veselý, F. Grézl, and J. Černocký, "Residual memory networks: Feed-forward approach to learn long-term temporal dependencies," in *Proceedings of ICASSP 2017*. IEEE Signal Processing Society, 2017, pp. 4810–4814.

[12] 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.

[13] P. Ghahremani, B. BabaAli, D. Povey, K. Riedhammer, J. Trmal, and S. Khudanpur, "A pitch extraction algorithm tuned for automatic speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. Florence, Italy: IEEE, May 2014.

[14] K. Laskowski, M. Heldner, and J. Edlund, "The fundamental frequency variation spectrum," 2008.

[15] S. Sharma and H. Hermansky, "Temporal patterns (TRAPS) in ASR of noisy speech," in *Proc. IEEE Int. Conf. on Acoustic, Speech and Signal processing (ICASSP)*, Phoenix, Arizona, 1999, pp. 289–292.

[16] M. Karafiát, F. Grézl, and J. Černocký, "Trap based features for lvcsr of meeting data," in *Proc. 8th International Conference on Spoken Language Processing*, vol. 2004, no. 10. Sunjin Printing Co,, 2004, pp. 437–440. [Online]. Available: http://www.fit.vutbr.cz/research/view_pub.php?id=7485

[17] M. Karafiát, M. K. Baskar, K. Veselý, F. Grézl, L. Burget, and J. Černocký, "Analysis of multilingual BLSTM acoustic model on low and high resource languages," in *Proceedings of ICASSP 2018*, 2018.

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

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

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

[21] D. Stein, T. Winkler, J. Schwenninger, and R. Bardeli, "Tetra channel simulation for automatic speech recognition," *European Signal Processing Conference*, pp. 1653–1657, 2012.

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

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

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

[25] L. Zhang, D. Karakos, W. Hartmann, R. Hsiao, R. M. Schwartz, and S. Tsakalidis, "Enhancing low resource keyword spotting with automatically retrieved web documents," in *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*, 2015, pp. 839–843.

[26] T. Ng, B. Zhang, L. Nguyen, S. Matsoukas, X. Zhou, N. Mesgarani, K. Veselý, and P. Matějka, "Developing a speech activity detection system for the darpa rats program," in *Proc. Interspeech*, 2012.