Interspeech 2018
2-6 September 2018, Hyderabad

BUT system for low resource Indian language ASR

Bhargav Pulugundla[1,2], *Murali Karthick Baskar*[1], *Santosh Kesiraju*[1,3], *Ekaterina Egorova*[1], *Martin Karafiát*[1], *Lukáš Burget*[1], *Jan Černocký*[1]

[1]Brno University of Technology, Speech@FIT and IT4I Center of Excellence, Czechia
[2]Phonexia s.r.o., Czechia
[3]IIIT Hyderabad, India

{ipulugundla,baskar,kesiraju,iegorova,karafiat,burget,cernocky}@fit.vutbr.cz

Abstract

This paper describes the BUT 'Jilebi' team's speech recognition systems created for the 2018 low resource speech recognition challenge for Indian languages. We investigate modifications of multilingual time-delay neural network (TDNN) architectures with transfer learning and compare them to bi-directional residual memory networks (BRMN) and bi-directional LSTM. Our best submission based on system combination achieved word error rates of 13.92% (Tamil), 14.71% (Telugu) and 14.06% (Gujarati). We present the details of submitted systems and also the post-evaluation analysis done for lexicon discovery using unsupervised word segmentation.

Index Terms: Indian languages, low resource ASR, multilingual, LF-MMI

1. Introduction

Automatic speech recognition (ASR) requires a large amount of transcribed speech data to perform well which is a costly procedure and most of the languages in the world have limited or no-resource training data. Low resource ASR was one of the aspects of the IARPA Babel program [1] and was also the focus of the MGB-3 challenge [2]. Different approaches have been suggested in dealing with low resource training data: multilingual pre-training - using a multilingual framework to extract features which help in improving low resource ASR systems [3, 4], semi-supervised training - transfer learning to adapt an ASR trained on large dataset to a low resource language [5] and data augmentation [6]. We use all of these techniques in our systems.

India has 22 official languages and many of these can be considered low resourced for training an ASR system. The Low resource Indian language ASR challenge, organized by Microsoft India, involved building speech recognition systems on three Indian languages: Tamil, Telugu and Gujarati. The participants were provided with 40 hours of transcribed speech data for training and 5 hours for development. A 5 hour held-out blind set was later released for evaluation. The participants had access to two pronunciation dictionaries mapped to two different phone sets: a language specific *Indic* phone set and a common IPA phone set for the three languages. The challenge rules restricted us from using any external speech and text data for training the models. The speech data statistics for the three languages are shown in Table 1.

When analyzing the data we found two key issues.

- OOVs: Different transcriptions of same words in train and dev/eval sets. Although the vocabulary distributed for the challenge contains words from both dev and eval sets, this is not a realistic scenario — in post-evaluation analysis, we tried to discover the OOVs by an unsupervised approach.
- Many nouns and proper names occur less frequently. We try to explore various models to empirically see which performs best in this scenario. We also use multilingual approaches to overcome the issue of less data.

We trained monolingual acoustic models with different architectures; time delay neural network (TDNN) [7], bi-directional long short term memory (BLSTM) [8] and bi-directional residual memory networks (RMN) [9] using the *Indic* phone set and multilingual models using the common phone set. The results show that multilingual TDNN with transfer learning outperforms the monolingual TDNN by an WER of 0.5% absolute. We use Kneser-Ney n-gram and RNN language models for decoding and rescoring. Our best submission was a system combination of a low rank TDNN and a multilingual TDNN fine tuned to each language.

This paper is organized as follows: Section 2 describes our baseline system and other approaches to acoustic modeling. Section 3 describes the language modeling. In sections 4 and 5, we present the results with some discussions on the post-evaluation analysis and conclusions.

Table 1: *Speech data statistics of training, development and evaluation sets*

Lang.	# of utterances			Avg. duration of utterances(secs)		
	Train	Dev	Eval	Train	Dev	Eval
Tamil	39131	3081	2609	3.6	5.8	5.8
Telugu	44882	3040	2549	3.2	5.9	5.9
Gujarati	22807	3075	3419	6.3	5.8	5.2

2. Acoustic Modeling

2.1. Monolingual models

2.1.1. Baseline

Our baseline acoustic model was based on purely sequence-trained TDNN with the lattice-free maximum mutual information (LF-MMI) objective [7, 10] and was trained using the Kaldi toolkit [11]. This was similar to the approach followed by the organizers on the challenge website.

We followed the approach usual in Kaldi recipes for training Gaussian mixture models (GMM-HMM). A monophone GMM-HMM was initially trained on 16 dimensional PLP features with 3 Kaldi pitch features on the 10000 shortest utterances of the training data. We

10.21437/Interspeech.2018-1302

Table 2: *Comparison of baseline system's WER(%) on dev set: BUT and challenge organizers*

System	GMM			TDNN		
	Tamil	Telugu	Gujarati	Tamil	Telugu	Gujarati
Baseline(Organizers)	33.5	40.1	23.7	19.4	22.6	19.7
Baseline(BUT)	21.2	23.7	16.3	16.1	17.1	12.5
Low rank TDNN	-	-	-	15.6	16.5	12.2

applied a per-speaker mean normalization and a global variance normalization on all the input features. Then three GMM-HMM triphone models are trained. A regular GMM-HMM on the same input features along with their first and second derivatives is trained. Then we train a GMM-HMM on linear discriminant analysis (LDA) transformed features. Finally a speaker adaptive (SAT) GMM-HMM on the fMLLR adapted features. The fMLLR-SAT GMM system was built with 6000 tied triphone states. We use the alignments fMLLR-SAT GMM-HMM for TDNN supervision.

The input features to TDNN were 40 high resolution MFCCs with 100 dimensional i-vectors for speaker adaptation [12]. We also performed speed perturbation to augment $3\times$ the training data at speeds 0.9, 1.0 and 1.1, and then volume perturbation by a random factor between 0.1 and 2. The i-vector extractor was trained on the speed perturbed data. The baseline TDNN network had nine layers and eight million parameters. TDNN trained with LF-MMI uses lower frame rate and we prune the fMLLR-SAT GMM-HMM tree to produce one HMM state per phone. The resulting tree had 3500 tied states.

For decoding, we used a 3-gram modified Kneser-Ney language model trained using SRILM [13]. The challenge organizers used Kaldi LM for their 3-gram language model.

Our baseline system outperforms the organizers' one on the development set in all the three languages because of the difference in network architecture and number of parameters in the acoustic model. The results are shown in Table 2.

2.1.2. Low rank TDNN

Then, we explored an architecture of TDNN different from our baseline. We use the same LF-MMI objective function, input features, i-vectors and fMLLR-SAT GMM system as in our baseline system. The major difference is the addition of a) bottleneck linear transformation layer after every affine transformation of batchnorm ReLU layer and b) skip connections. This changes can be seen in recipes of Kaldi version 5.4[1]. We can assume this to be a low-rank factorization of the TDNN at every ReLU+linear layer pair. Table 2 shows the improvement over the baseline TDNN on the dev set and Table 3 gives the number of layers and parameters.

Table 3: *Architecture of BUT baseline and low rank TDNN*

System	Parameters	Layers
Baseline	8 mil	9
Low rank	18.5 mil	11

2.1.3. BRMN and BLSTM

In this section, we describes our experiments with bi-directional residual memory neural network (BRMN) architecture [9] as a way to model short-time dependencies using deep feed-forward layers having residual and time delayed connections. Here, the number of layers in BRMN signifies both the hierarchical processing depth and temporal depth. The computational complexity in training BRMN is significantly smaller than for deep recurrent networks due to its feed-forward design. The recognition experiments are performed with BRMN having 12 layers where each layer is a $[1024 \times 512]$ dimensional weight matrix with ReLU activation. The residual connections flow by skipping over every two layers. The model is trained using truncated BPTT with a minibatch size of 20 and a maximum of 40 parallel utterances in each minibatch. The initial learning rate is set to 0.0005 and reduced automatically for the next epoch by a factor of 0.5 if cross-entropy loss degrades. ℓ_2 regularization weight is fixed to 0.00001 and momentum is set to 0.9. We use the *Indic* phone set for these experiments. BRMN and BLSTM were both trained using CNTK [14] by extracting GMM-HMM alignments from Kaldi [11] toolkit.

A 3-layer BLSTM with 512 dimensional memory cells and 300 dimensional projection matrix are employed in this network. BLSTM is trained using truncated backpropagation through time (BPTT) with a minibatch size of 20. It also includes latency control technique with 22 past frames and 21 future frames as mentioned in [15] to limit future context size. The BLSTM training follows similar configuration as explained in BRMN.

Table 4 shows the speech recognition performance for all three languages on dev set using BRMN and BLSTM models. The BRMN shows consistent gain over BLSTM for all languages and shows absolute 0.2% gain with i-vectors. This shows that BLSTM suffers with less data compared to BRMN and we observed overfitting after a few epochs. The addition of 40-dimensional i-vectors did not help BLSTM, instead it degraded its performance consistently across all languages. Sequence-level training using sMBR (state minimum Bayesian risk) criterion gave absolute 0.1% gain for Tamil and 0.3-0.4% gain for Telugu and Gujarati.

2.2. Multilingual models

2.2.1. Transfer learning TDNN

A multilingual neural network was trained by pooling the three languages using the common IPA phone set. The acoustic model was based on the low rank TDNN architecture from Section 2.1.2 with 7000 tied states.

The hidden layers of a neural network learn the higher order representations of the input features. In transfer learning, a source model is trained on a large corpus and then the weights of the hidden layers are transfered to a smaller target dataset and re-trained for a similar or different task. In this approach, both models must be trained using the same objective function.

We use the transfer learning approach discussed in [16]. In our experiments the source and target models were the multilingual and monolingual TDNNs trained with LF-MMI objective. We transferred all layers of a pre-trained source

[1] `egs/swbd/s5c/local/chain/tuning/run_tdnn_7n.sh`

model and retrained with higher learning rate for the last layer for 2 epochs using target labels. The learning rate of the transferred layers (excluding last) is reduced by a factor of 0.25 of the initial learning rate. New alignments and lattices are generated using the multilingual TDNN model. In transfer learning the last layer is not transfered because source and target use different phone sets and trees but in our experiments we transfer all the layers since we use the same context-dependency tree of the source network. A new target lexicon with word pronunciation of source lexicon is created and target words not present in source lexicon are treated as OOVs. Phone LM used to create the denominator graph is generated by the weighted combination of alignments from source and target.

More experiments are required to analyze the effect of number of transferred layers in the scope of the challenge.

2.2.2. Multilingual BRMN and BLSTM

In case of multilingual experiments, a common phone set created using IPA rules is used to initially build a multilingual BRMN acoustic model with block-softmax layer with an initial learning rate of 0.005. This model is further adapted for each language using language dependent softmax layer in two steps, first, the pre-initialized multilingual hidden layers are frozen and only the last layer is retrained for 5 epochs with initial learning rate set to 0.00001. The resulting model is then retrained across all layers for 10 epochs with initial learning rate set to 0.0005. Multilingual BLSTM (Multi-BLSTM) is trained in a similar fashion as in Multi-BRMN.

The recognition performance of Multi-BLSTM and Multi-BRMN are denoted in Table 4. The results show that Multi-BLSTM showed slight gain compared to Mono-BLSTM with i-vectors model for Telugu and Gujarati, but degraded for Tamil. In case of Multi-BRMN, there was consistent and significant gain over its monolingual version across all three languages. It is interesting to notice that both multilingual models trained without i-vectors performed well over monolingual models trained with i-vectors.

Table 4: *Recognition performance on dev data with KN 3-gram LM. The mono-lingual system used Indic phone set lexicon and multi-lingual system used IPA phone set lexicon*

Dev set (languages)	**Tamil**	**% WER** **Telugu**	**Gujarati**
With BRMN - CE (sMBR)			
Mono	16.1 (16.0)	17.6 (17.4)	13.1 (13.0)
Mono (+ivec)	15.9 (15.8)	17.4 (17.0)	12.9 (12.8)
Multi	15.8 (15.7)	16.8 (16.5)	12.6 (12.5)
With BLSTM - CE (sMBR)			
Mono	16.3 (16.2)	18.0 (17.9)	13.6 (13.5)
Mono (+ivec)	16.6 (16.5)	18.4 (18.3)	13.8 (13.7)
Multi	16.7 (16.6)	18.1 (18.0)	13.5 (13.4)
With Low rank TDNN - LFMMI			
Mono (+ivec)	15.6	16.5	12.2
With Transfer Learning			
Multi (+ivec)	**15.2**	**16.0**	**11.9**

3. Language Modeling

A 3-gram language model is prepared with all train and dev text (Table 1) along with the evaluation vocabulary list. RNN language model (RNNLM) is also prepared by following the recipe in Kaldi, with a few modifications in fine-tuning. The RNNLM is built with 100 dimensional embedding (input) layer and a single LSTM layer containing 100-dimensional cell followed by output layer. RNNLM rescoring gives consistent improvement on both the low rank TDNN and transfer learning models. Table 5 shows the comparison of the languages models on the dev set for the three languages.

Table 5: *Comparing % WER of KN 3-gram and KN 3-gram with RNNLM rescoring on dev set*

LM	**Tamil**	**Telugu**	**Gujarati**
Low rank TDNN			
KN 3-gram	15.6	16.5	12.2
RNNLM	15.3	16.1	12.0
Transfer learning			
KN 3-gram	15.2	16.0	**11.9**
RNNLM	**15.0**	**15.7**	12.0

4. Results and Discussions

Table 6 shows the results of various acoustic models on eval set. Evaluation models were trained using the data from both train and dev sets. We used a Kneser-Ney 3-gram language model with a new lexicon including the evaluation vocabulary. The RNNLM rescoring provided good performance on dev set but failed on eval set with performance degradation (0.4% on Telugu).

Low rank TDNN with skip connections gave considerable absolute improvement of 0.6-1.1% over baseline TDNN. Transfer learning further improves by an additional 0.5-0.7% absolute over the low rank TDNN system. ROVER [17] was used to combine the CTM hypotheses of the systems. Our primary submission was a system combination with low rank TDNN and low rank TDNN with transfer learning. We also combined the primary submission with mono BRMN but it did not improve the performance.

As an post-evaluation experiment, we tried to improve multilingual BRMN (Multi BRMN) model as it showed considerable gains over multilingual BLSTM models. In this experiment, the i-vectors were also fed into the system after transforming them using a single feed-forward layer and adding the resulting output with the second layer of final Multi-BRMN model. This strategy is adopted from the multilingual model adaptation procedure in [18]. During adaptation, the final layer is retrained for 8 epochs with an initial learning rate of 0.0001 and further retrained across all layer for 15 epochs with initial learning rate of 0.005. The resulting Mult-BRMN model showed considerable gain over monolingual BRMN (BRMN) which can be devoted to inclusion of i-vectors and better initialization as denoted in Table 6. Multi BRMN gives absolute gain of 1.2% and 1.1% for Telugu and Gujarati over monolingual BRMN model with ivectors also seen in Table 6.

4.1. Unsupervised lexicon discovery

This section outlines the lexicon discovery strategy we explored post evaluation. This was not required for the evaluation because the organizers provided a full lexicon containing the words from evaluation set as well.

The motivation comes from the nature of orthography in Indian languages. Most of them share similar properties, where

Table 6: *% WER of post-evaluation models and submitted models on the evaluation set*

Eval set (languages)	**% WER** **Tamil**	**Telugu**	**Gujarati**
BRMN (+ivec)	14.6	15.7	15.3
Low rank TDNN (A)	14.5	15.5	15.1
Multi BRMN (+ivec)	14.1	14.5	14.2
Sys. comb. of A+B+Multi BRMN (+ivec)	**13.8**	**14.3**	**13.9**
Submitted systems			
Transfer learning TDNN (B)	14.0	14.9	14.4
Sys. comb. of A+B + BRMN (+ivec)	14.3	14.8	15.0
Sys. comb. of A+B	**13.9**	**14.7**	**14.1**

words (morphemes) can be combined to form a compound word, following certain morpho-phonological rules. This process is called *Sandhi*. Conversely, a word can be split into its constituents based on the same rules. This behavior is more prominent in *Dravidian* language family (Kannada, Malayalam, Tamil, Telugu, etc,.) and it makes them highly agglutinative. This gives rise to the possibility of several new words, including proper nouns. Moreover, it also depends on the choice of the person speaking and transcribing. Fig. 1 shows one such example from Telugu, where a compound proper noun is formed by the concatenation of two sub words (nouns). For such languages, ASR systems benefit by having a huge lexicon with all the compound words and sub words.

సుజనా చౌదరి → సుజనాచౌదరి
/sʊd͡ʒəna/ /t͡ʃauɖəri/ → /sʊd͡ʒənat͡ʃauɖəri/
sujana chowdary → *sujanachowdary*

Figure 1: *An example of discovered compound proper noun from two sub words (nouns) in Telugu.*

Most of the earlier work in discovering the words from Sandhi or sandhi splitting was based on linguistic rules, and recently seq2seq based neural networks were used to learn these mappings for Sanskrit [19]. However, it requires training data constituting of compound words and their constituent morphemes. We used existing models and tools to discover some of these compound and sub words. More specifically, we used unsupervised word segmentation based on nested Pitman-Yor language model [20]. In brief, given a sequence of characters without any word boundaries, the model discovers most likely word segments, that maximizes the likelihood of the character n-gram language model and the nested word level n-gram language model. This model and its extensions were used for discovering lexicon from acoustic input [21], and also from lattices [22, 23].

We make use of the open source implementation of the model[2] to obtain several segmentations of every utterance from training data. We chose various hyper-parameters (order of character and word level language models) that encourage over and under segmentation. Additionally, we set the average word length to be $\{4, 8\}$, which serves as the mean parameter for Poisson distribution. This resulted in several word segments (10 times the size of current vocabulary). For example, see Fig.1.

[2] https://github.com/fgnt/LatticeWordSegmentation

Table 7: *Statistics of discovered words with the corresponding improvements in word error rates (WER) on eval set.*

Lang.	**# of OOV words**	**# of words discovered**	**WER before**	**WER after**
Tamil	3935	501	29.0	27.7
Telugu	2349	358	25.4	24.5
Gujarati	2342	308	19.5	19.0

To test the viability of the segmentation approach, we considered only training and development data for segmentation, and later checked how many of the new words from segmentation, appear in the evaluation lexicon. Then, we decoded the eval set with low rank TDNN (Table 6) model and the existing language model but with updated lexicon containing the discovered words. These statistics are presented in Table 7, along with the corresponding improvement in word error rates (i.e., if we were not given evaluation lexicon, these will be the improvements in WER). It is important to note that this segmentation is a naïve approach, and will only discover a subset of words that are result of Sandhi and sandhi splitting, and it does not involve learning any morpho-phonological rules.

5. Conclusions

In this paper, we presented the details of the BUT team's submissions to the low resource speech recognition challenge for Indian languages. We investigated various monolingual and multilingual acoustic models for low resource training data in Tamil, Telugu and Gujarati. Results showed that feed-forward networks perform better than recurrent networks for low resource languages. Models with low rank TDNN architecture trained with LF-MMI objective outperforms the traditional TDNN. We also used a transfer learning approach to adapt a multilingual TDNN model which improves further. Our best submission was a ROVER combination of the two systems. During post evaluation, we explored automatic lexicon discovery using unsupervised word segmentation, exploiting the agglutinative nature of Indian languages. With this approach we were able to discover 12 - 15% of OOVs, at the expense of having a huge lexicon (almost 10 times the size of given lexicon).

In future, we will continue to explore lexicon discovery techniques as it was observed to be a promising direction, especially for Indian languages. We would also like to analyze the individual effects of linear bottleneck layers and skip connections in the low rank TDNN network. We will also look at substituting CE followed by sMBR objective with LF-MMI in BRMN.

6. Acknowledgements

This work was supported by the U.S. DARPA LORELEI contract No. HR0011-15-C-0115. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. The work was also supported by 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". Data provided by SpeechOcean.com and Microsoft.

7. References

[1] J. Cui, B. Kingsbury, B. Ramabhadran, A. Sethy, K. Audhkhasi, X. Cui, E. Kislal, L. Mangu, M. Nussbaum-Thom, M. Picheny, Z. Tske, P. Golik, R. Schlter, H. Ney, M. J. F. Gales, K. M. Knill, A. Ragni, H. Wang, and P. Woodland, "Multilingual representations for low resource speech recognition and keyword search," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Dec 2015, pp. 259–266.

[2] A. Ali, S. Vogel, and S. Renals, "Speech recognition challenge in the wild: Arabic MGB-3," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Dec 2017, pp. 316–322.

[3] E. Chuangsuwanich, "Multilingual techniques for low resource automatic speech recognition," Ph.D. dissertation, Massachusetts Institute of Technology, 2016.

[4] K. Vesel, M. Karafit, F. Grzl, M. Janda, and E. Egorova, "The language-independent bottleneck features," in *2012 IEEE Spoken Language Technology Workshop (SLT)*, Dec 2012, pp. 336–341.

[5] V. Manohar, D. Povey, and S. Khudanpur, "JHU Kaldi system for Arabic MGB-3 ASR challenge using diarization, audio-transcript alignment and transfer learning," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Dec 2017, pp. 346–352.

[6] A. Ragni, K. Knill, S. Rath, and M. Gales, "Data augmentation for low resource languages," pp. 810–814, 01 2014.

[7] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *INTERSPEECH*, 2015.

[8] M. Karafiát, M. K. Baskar, P. Matějka, K. Veselỳ, F. Grézl, and J. Černocky, "Multilingual BLSTM and speaker-specific vector adaptation in 2016 BUT Babel system," in *IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 637–643.

[9] M. K. Baskar, M. Karafiát, L. Burget, K. Veselỳ, F. Grézl, and J. Černockỳ, "Residual Memory Networks: Feed-forward approach to learn long-term temporal dependencies," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 4810–4814.

[10] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely Sequence-Trained Neural Networks for ASR Based on Lattice-Free MMI," in *INTERSPEECH*, 2016.

[11] 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.

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

[13] A. Stolcke, "SRILM – An extensible language modeling toolkit," in *7th International Conference on Spoken Language Processing (ICSLP)*, 2002, pp. 901–904.

[14] D. Yu, A. Eversole, M. Seltzer, K. Yao, O. Kuchaiev, Y. Zhang, F. Seide, Z. Huang, B. Guenter, H. Wang, J. Droppo, G. Zweig, C. Rossbach, J. Gao, A. Stolcke, J. Currey, M. Slaney, G. Chen, A. Agarwal, C. Basoglu, M. Padmilac, A. Kamenev, V. Ivanov, S. Cypher, H. Parthasarathi, B. Mitra, B. Peng, and X. Huang, "An introduction to computational networks and the computational network toolkit," Tech. Rep., October 2014. [Online]. Available: http://research.microsoft.com/apps/pubs/default.aspx?id=226641

[15] Y. Zhang, G. Chen, D. Yu, K. Yaco, S. Khudanpur, and J. Glass, "Highway long short-term memory RNNs for distant speech recognition."

[16] P. Ghahremani, V. Manohar, H. Hadian, D. Povey, and S. Khudanpur, "Investigation of transfer learning for ASR using LF-MMI trained neural networks," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Dec 2017, pp. 279–286.

[17] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER)," in *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, Dec 1997, pp. 347–354.

[18] M. Karafiát, M. K. Baskar, P. Matejka, K. Veselý, F. Grézl, L. Burget, and J. Cernocký, "2016 BUT babel system: Multilingual BLSTM acoustic model with i-vector based adaptation," in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24*, 2017, pp. 719–723.

[19] N. Gantayat, R. Aralikatte, N. Panwar, A. Sankaran, and S. Mani, "Sanskrit Sandhi Splitting using $seq2(seq)^2$," *ArXiv e-prints*, Jan. 2018.

[20] D. Mochihashi, T. Yamada, and N. Ueda, "Bayesian Unsupervised Word Segmentation with Nested Pitman-Yor Language Modeling," in *Proceedings of the 4th International Joint Conference on Natural Language Processing of the AFNLP, ACL*, August 2009, pp. 100–108.

[21] C.-y. Lee, T. J. O'Donnell, and J. Glass, "Unsupervised Lexicon Discovery from Acoustic Input," *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 389–403, 2015.

[22] G. Neubig, M. Mimura, S. Mori, and T. Kawahara, "Learning a language model from continuous speech," in *INTERSPEECH, ISCA*, 2010, pp. 1053–1056.

[23] J. Heymann, O. Walter, R. Haeb-Umbach, and B. Raj, "Iterative Bayesian word segmentation for unsupervised vocabulary discovery from phoneme lattices," in *IEEE ICASSP*, May 2014, pp. 4057–4061.