The 6th Intl. Workshop on Spoken Language Technologies for Under-Resourced Languages
29-31 August 2018, Gurugram, India

Interspeech 2018 Low Resource Automatic Speech Recognition Challenge for Indian Languages

Brij Mohan Lal Srivastava, Sunayana Sitaram, Rupesh Kumar Mehta, Krishna Doss Mohan, Pallavi Matani, Sandeepkumar Satpal, Kalika Bali, Radhakrishnan Srikanth, Niranjan Nayak

Microsoft Corporation, India

v-bmlals, sunayana.sitaram, rupesh.mehta, krishna.doss, pamatani, ssatpal, kalikab, rsrikan, niranjan@microsoft.com

Abstract

India has more than 1500[1] languages, with 30 of them spoken by more than one million native speakers. Most of them are low-resource and could greatly benefit from speech and language technologies. Building speech recognition support for these low-resource languages requires innovation in handling constraints on data size, while also exploiting the unique properties and similarities among Indian languages. With this goal, we organized a low-resource Automatic Speech Recognition challenge for Indian languages as part of Interspeech 2018. We released 50 hours of speech data with transcriptions for Tamil, Telugu and Gujarati, amounting to a total of 150 hours. Participants were required to only use the data we released for the challenge to preserve the low-resource setting, however, they were not restricted to work on any particular aspect of the speech recognizer. We received 109 submissions from 18 research groups and evaluated the systems in terms of Word Error Rate on a blind test set. In this paper we summarize the data, approaches and results of the challenge.

Index Terms: speech recognition, low-resource, Indian languages

1. Introduction

Although speech technologies, particularly Automatic Speech Recognition (ASR) has made tremendous progress in the last few years due to advances in deep learning and the availability of large speech corpora, most languages in the world still do not have accurate speech recognizers. Many of these languages are spoken by more than one million native speakers, some of whom have low levels of literacy and can greatly benefit from speech technologies for access to information. India has over 1500 languages, however, only a handful of languages have speech recognition support, and in most cases the accuracies of such systems are far behind those of higher resource languages such as English and Mandarin. One of the greatest challenges in building ASR systems for Indian languages is the lack of transcribed speech data. The Linguistic Data Consortium has speech databases spanning up to a couple of hundred hours in some Indian languages [1, 2, 3, 4, 5], however, this data can be out of reach to some academic institutions that do not have the required subscription.

To address this, we conducted a Low Resource ASR challenge for Indian Languages [2] as part of Interspeech 2018 for which we released 50 hours of transcribed speech in three Indian languages - Tamil, Telugu and Gujarati, amounting to a total of 150 hours of data. Participants were required to use the data released as part of the challenge to build ASR systems which would be evaluated on a blind test set. Participants were restricted from using any external data for building models to preserve the focus on the low-resource setting, however, they were free to use the released data to build cross-lingual or multilingual models. They were also free to innovate in any aspect of the speech recognition pipeline and use any framework or models of their choice. Participants can use this data in the future for research purposes, after providing the following attribution when they publish their findings "Data provided by SpeechOcean.com and Microsoft", which we believe will foster innovation in building systems for low-resource languages.

The rest of the paper is organized as follows. Section 2 describes the challenge organization, and section 3 describes the data we released and the baseline systems we built using standard Kaldi recipes. We summarize results from the evaluation in Section 5. Section 6 concludes.

2. Challenge Organization

Registration for the Low Resource ASR challenge for Indian languages began in January 2018. When participants registered, they were sent a link to download the data sets in all three languages. We released around 40 hours of training data and around 5 hours of test data for each language. We also released baseline Word Error Rate (WER) numbers of systems built using standard Kaldi recipes (described in Section 4) evaluated on the 5 hour test set. In addition, we released lexicons using two phone sets for each language. More details about the data can be found in Section 3.

We created a LinkedIn discussion group to facilitate discussions among participants and between participants and organizers. In March 2018, we opened up the evaluation portal using which we tested all submitted systems over a period of 3 days. Participants were allowed to use all the 45 hours of released data (train+test) in each language to build their final systems. They were sent links to 5 hours of blind test audio data in each language on which they ran their systems and submitted a hypothesis file via email to an automated scoring system which calculated the Word Error Rate and sent it back to them as a reply to their email. Each participating team was allowed 3 attempts per language, so participants could submit up to 3 models or combinations of hyperparameters for each language. The automated scoring system was created using Microsoft Flow [3], and it blocked participants from submitting more

[1] http://www.censusindia.gov.in/Census_Data_2001/Census_Data_Online/Language/gen_note.html

[2] https://www.microsoft.com/en-us/research/event/interspeech-2018-special-session-low-resource-speech-recognition-challenge-indian-languages/

[3] https://flow.microsoft.com

10.21437/SLTU.2018-3

than three attempts per language. The system also alerted users if their hypothesis file or email was in an incorrect format. It ran the Kaldi Word Error Rate script to calculate the WER by comparing the submitted hypothesis file to a ground truth transcription file. We considered the highest scoring (lowest WER) attempt while ranking teams for the final evaluation.

Each team was also required to submit a description of their system with the models to us so that we could independently verify the scores after the automatic evaluation period ended. We also released the blind test data with the transcriptions so that participants could do additional experiments and evaluation after the official challenge evaluation period was over.

3. Data and baselines

3.1. Data

The data released for the challenge was provided by SpeechOcean.com and Microsoft. It consisted of phrasal (recorded as read-out phrases) and conversational speech in Tamil, Telugu and Gujarati. The speech was split into utterances and was transcribed using the native script of each language. Table 1 describes some statistics about the data.

Table 1: *Description of the train, test and blind test data*

Language	Train hrs	Test hrs	Blind hrs
Tamil	40	5	4.2
Telugu	40	5	4.2
Gujarati	40	5	5

3.2. Baseline systems

We build baseline systems using the data released for the challenge using the Kaldi toolkit [6]. We used standard Kaldi recipes without any modification so that participants could replicate the numbers easily. We provided a README file with details about the replicating the baselines along with the Word Error Rates of the baseline systems on the released test data.

3.2.1. Acoustic Models

We build three Acoustic Models for the baselines: GMM-HMM, Karel's DNN and TDNN.

- GMM-HMM: This system was built using LDA+MLLT+SAT training over MFCC features and is commonly referred to as tri3b.
- Karel's DNN [7]: This is commonly referred to as nnet. We trained the fully-connected DNN with 7 hidden layers and 2048 neurons per layer. We used Feature-space Maximum Likelihood Linear Regression (fMLLR) as input to this network. We provided alignments generated by the tri3b model. We decoded the test set using this model with an acoustic scale parameter of 0.08.
- TDNN: Time-delay Neural Networks [8] have shown a consistent drop in WER and were our best-performing models. They are also referred to as chain models. We used the $run\_tdnn.sh$ script for TDNN modeling. We provided tri3b as the GMM for the target data during modeling. We gave 128 chunks per minibatch for 4 epochs.

3.2.2. Language Models

Participants were required to use only the transcripts provided to build Language Models, and were not allowed to use external data for challenge systems. We built the baseline trigram LMs using the kaldi-LM tool. Participants were free to use any tool or model of their choice to build LMs. They were also allowed to use the training transcripts to generate more data, for example, using a Recurrent Neural Network Language Model.

3.2.3. Pronunciation lexicons

For each language, we released two pronunciation lexicons created using the Festvox Indic frontend [9], which used a phoneset similar to SAMPA and IIT Madras' Common Label Set [10] which used an IPA-based phoneset. Both these phonesets were created for Indian languages and facilitated phoneme sharing between the three languages. Participants were free to use either of the lexicons to train their models and to learn grapheme to phoneme systems. The baseline Word Error Rates reported in Table 2 are for models that used the Indic frontend lexicon. The pronunciation lexicon had complete coverage of all words in the train and test sets to alleviate the problem of Out of Vocabulary words.

3.2.4. Baseline WERs

Table 2 shows the Word Error Rates of GMM-HMM, Karel's DNN and TDNN (chain) models built using the training data, tested on the released test data. As expected, the TDNN models perform significantly better than GMM-HMM and DNN models on all three languages. In case of Gujarati, the GMM-HMM system performs better than DNN. The Gujarati data contains a higher proportion of phrasal or read speech, which may have made it easier to model compared to the other languages.

Table 2: *Kaldi baseline Word Error Rates*

Language	GMM-HMM	DNN	TDNN
Tamil	33.55	25.47	19.45
Telugu	40.12	34.97	22.61
Gujarati	23.78	27.79	19.76

4. Systems and results

We received over a 100 registrations from academic institutions, industry labs and startups from all over India and across the world. 40 models from 18 teams were submitted for Gujarati, 36 models from 14 teams for Tamil and 33 models from 18 teams were submitted for Telugu during the automatic evaluation period. All systems were evaluated using WER on the blind test set for each language.

4.1. Participating teams

Participants were asked to choose a team name for the evaluation. The following teams participated in the evaluation: Jilebi, SpeechCDACK, AAA, ISI-Billa, Cogknit, DAIICT+IIIT Vadodara, SMTIITM, MILE, Genesys, IIT Guwahati, IITH-SIPLAB, SPIRE, IITM Speech Lab, CMU, CSALT-LEAP, IIIT Bangalore, IIIT Hyderabad and SLPG. Participants spanned academic institutions, industry labs and startups in India and abroad.

4.2. Results

Table 3 shows the Word Error Rates obtained by the best models for each language. In all cases, the best systems were able to significantly outperform the TDNN baseline.

Table 3: *Word Error Rates of top performing models*

Language	Team	WER
Tamil	Jilebi	13.92%, 14.08%, 14.27%
Tamil	Cogknit	16.07%
Tamil	CSALT-LEAP	16.32%
Telugu	Jilebi	14.71%, 14.86%, 15.07%
Telugu	Cogknit	17.14%
Telugu	CSALT-LEAP	17.59%
Gujarati	Jilebi	14.06%, 14.70%, 15.04%
Gujarati	Cogknit	17.69%
Gujarati	ISI-Billa	19.31%

The Jilebi system [11] which performed the best across all languages was a multilingual TDNN based system with transfer learning. The system used an n-gram based LM for decoding and an RNN-based LM for rescoring. The multilingual TDNN was compared with monolingual TDNN, bi-directional long short term memory (BLSTM) and bi-directional residual memory networks (RMN) and was found to outperform all the monolingual models. The best performing model was a ROVER combination of the multilingual TDNN and a low rank TDNN architecture trained with the LF-MMI objective.

The Cogknit system [12] was also a multilingual TDNN system. Smoothed n-gram Language Models were used during decoding. The authors experimented with training with acoustic and lexical data from two and three languages at a time and found that the Gujarati system improved with three language training, while the best Tamil and Telugu systems were trained with two languages and did not benefit from the inclusion of Gujarati training data.

The ISI-Billa system [13] was an EESEN [14] based end-to-end multilingual LSTM network trained using the CTC training criterion. Both monolingual and multilingual systems trained using this model outperformed the baselines in all three languages significantly. Multilingual training was performed by pooling all the data together and by using a language-specific Language Model during decoding.

The CSALT-LEAP system uses unsupervised acoustic units by clustering multilingual acoustic data using an HMM followed by a bottleneck layer. The Bottleneck features are used in a DNN-based acoustic model along with mel-frequency based features in a TDNN-based acoustic model which provides the best performance.

Figures 1, 2 and 3 show the WERs obtained by all submitted models and the TDNN baseline, shown in red for all three languages. A significant number of submitted systems did not beat the TDNN baseline, however, many participants chose to use models such as GMM-HMM, which were expected to perform worse, but tried innovative data sharing and multilingual approaches. Next, we describe some of the other approaches taken by submitted systems.

The DAIICT-IIITV Gujarati system [15] used a combination of TDNN and TDNN-LSTM Acoustic Models with various acoustic features. RNN-based Language Models for rescoring were found to outperform n-gram models. The best performing system had a WER of 19.67% and was a combination of five

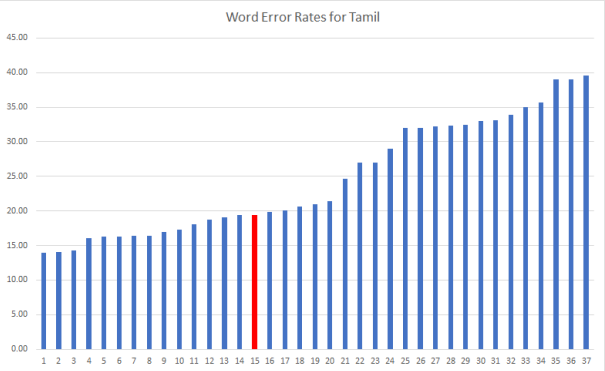

Figure 1: *Word Error Rates of all submitted models for Tamil. The best baseline is in red.*

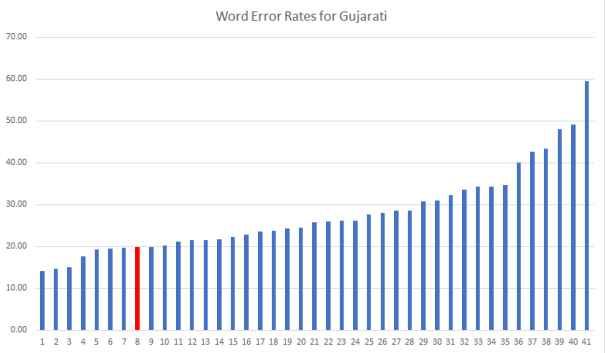

Figure 2: *Word Error Rates of all submitted models for Gujarati. The best baseline is in red.*

TDNN and TDNN-LSTM systems.

The Genesys system achieved Word Error Rates of 18.7% for Telugu, 17.2% for Tamil, and 21.6% for Gujarati using a TDNN-based system with sMBR. During error analysis it was found that for Telugu, many of the errors were due to confusions between compound words and split words. Mapping the compound word and split word pairs led to improvements in all three languages. They did not find a significant reduction in WER on using RNNLM.

The IIIT Hyderabad system [16] was also an end-to-end RNN-based system trained using CTC which facilitated data sharing between the languages. The joint RNN-CTC acoustic model was found to perform better than an HMM-SGMM model with the best systems achieving Word Error Rates of 21.42%, 20.66% and 21.97% for Telugu, Tamil and Gujarati respectively.

The IITM Speech Lab systems [17] used articulatory and stacked bottlneck (SBN) features in the Acoustic Model by creating an bidirectional long short term memory (BLSTM) articulatory feature classifier using the pooled data. The best system with Word Error Rates of 24.29%, 30.33%, 17.9% for Gujarati, Telugu, Tamil respectively was a TDNN-based system with articulatory and SBN features.

5. Discussion

From the system descriptions, it can be seen that TDNN-based systems performed best even in the low-resource setting. Multilingual transfer learning as well as data sharing approaches were

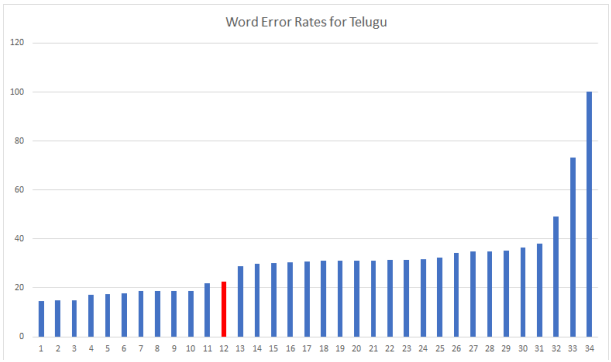

Figure 3: *Word Error Rates of all submitted models for Telugu. The best baseline is in red.*

promising, and are a fruitful direction to explore for building systems with low resource, related languages. RNNLM-based rescoring was found to help in most cases. However, recent advances in end-to-end ASR were also shown to be promising, with some of the top systems using CTC-based multilingual training, which has not been shown earlier for low resource languages. Other interesting approaches tried by teams were unsupervised acoustic unit discovery and articulatory feature extraction using the pooled data.

The speech data released will be made publicly available to the community for research purposes, which we hope will help push the boundaries of low resource speech recognition for Indian languages and make it easier to build systems rapidly for languages with limited data.

6. Acknowledgements

The authors would like to thank Basil Abraham, Satarupa Guha, Shambo Chatterjee and Swapnajeet Padhi for help with data preparation and evaluation and the speech group at IIT Madras for providing lexicons using the Common Label Set. The authors would also like to thank all teams that participated in the challenge.

7. References

[1] "IARPA Babel Assamese Language Pack IARPA-babel102b-v0.5a," https://catalog.ldc.upenn.edu/LDC2016S06.

[2] "IARPA Babel Bengali Language Pack IARPA-babel103b-v0.4b," https://catalog.ldc.upenn.edu/LDC2016S08.

[3] "IARPA Babel Tamil Language Pack IARPA-babel204b-v1.1b," https://catalog.ldc.upenn.edu/LDC2017S13.

[4] "CSLU: 22 Languages Corpus," https://catalog.ldc.upenn.edu/LDC2005S26.

[5] "ARL Urdu Speech Database, Training Data," https://catalog.ldc.upenn.edu/LDC2007S03.

[6] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[7] K. Veselỳ, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks." in *Interspeech*, 2013, pp. 2345–2349.

[8] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[9] A. Parlikar, S. Sitaram, A. Wilkinson, and A. W. Black, "The festvox indic frontend for grapheme to phoneme conversion," in *WILDRE: Workshop on Indian Language Data-Resources and Evaluation*, 2016.

[10] A. Baby, N. Nishanthi, A. L. Thomas, and H. A. Murthy, "A unified parser for developing indian language text to speech synthesizers," in *International Conference on Text, Speech, and Dialogue*. Springer, 2016, pp. 514–521.

[11] B. Pulugundla, M. K. Baskar, S. Kesiraju, E. Egorova, M. Karaat, L. Burget, and J. Cernocky, "But system for low resource indian language asr," in *Interspeech*, 2018.

[12] N. Fathima, T. Patel, M. C, and A. Iyengar, "TDNN-based Multilingual Speech Recognition System for Low Resource Indian Languages," in *Interspeech*, 2018.

[13] J. Billa, "ISI ASR System for the Low Resource Speech Recognition Challenge for Indian Languages," in *Interspeech*, 2018.

[14] Y. Miao, M. Gowayyed, and F. Metze, "Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding," in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. IEEE, 2015, pp. 167–174.

[15] H. Sailor, M. V. S. Krishna, D. Chhabra, A. Patil, M. Kamble, and H. Patil, "Da-iict/iiitv system for low resource speech recognition challenge 2018," in *Interspeech*, 2018.

[16] H. K. Vydana, K. Gurugubelli, V. V. V. Raju, and A. K. Vuppala, "An Exploration Towards Joint Acoustic Modeling for Indian Languages: IIIT-H submission for Low Resource Speech Recognition Challenge for Indian languages,INTERSPEECH2018," in *Interspeech*, 2018.

[17] V. M. Shetty, A. R. Sharon, B. Abraham, T. Seeram, A. Prakash, N. Ravi, and S. Umesh, "Articulatory and Stacked Bottleneck Features for Low Resource Speech Recognition," in *Interspeech*, 2018.