An Investigation of Hybrid architectures for Low Resource Multilingual Speech Recognition system in Indian context

Ganesh S Mirishkar **Aditya Yadavalli** **Anil Kumar Vuppala**
Speech Processing Laboratory,
Language Technologies Research Centre,
Kohli Center on Intelligent Systems,
International Institute of Information Technology, Hyderabad
India
{mirishkar.ganesh, aditya.yadavalli}@research.iiit.ac.in,
anil.vuppala@iiit.ac.in

Abstract

India is a land of language diversity. There are approximately 2000 languages spoken around, and among which officially registered are 23. In those, there are very few with Automatic Speech Recognition (ASR) capability. The reason for this is the fact that building an ASR system requires thousands of hours of annotated speech data, a vast amount of text, and a lexicon that can span all the words in the languages. At the same time, it is observed that Indian languages share a common phonetic base. In this work, we build a multilingual speech recognition system for low-resource languages by leveraging the shared phonetic space. Deep Neural architectures play a vital role in improving the performance of low-resource ASR systems. The typical strategy used to train the multilingual acoustic model is merging various languages as a unified group. In this paper, the speech recognition system is built using six Indian languages, namely Gujarati, Hindi, Marathi, Odia, Tamil, and Telugu. Various state-of-the-art experiments were performed using different acoustic modeling and language modeling techniques.

1 Introduction

According to the 2011 census, India has 23 constitutionally recognized official languages and 1600 other languages (Wikipedia, 2021). In this era of digitization, speech technologies for Indian languages play a pivotal role across various business domains. Building automatic speech recognition (ASR) (Reddy, 1976) and text-to-speech (TTS) (Dutoit, 1997) based interfaces for Indian markets is big challenge as majority of languages are "low resourced" (Miao et al., 2013). In general, a language is referred to as low resource when there is: (i) lack of availability of speech, text, transcribed data, (ii) lack of linguistic expertise in a particular language, or (iii) lack of pronunciation dictionary (Lu et al., 2013). In order to build state-of-the-art ASR systems for Indian languages, tons of training data is required for achieving human parity. In the training ASR system, the model expects audio data and corresponding transcripts as an input. Though there is a lot of raw speech freely available for Indian languages, it is a complex and costly process to get the corresponding transcription. Hence, very few efforts were made in these directions over the past decade.

As Indian languages are syllabic, efforts were put in generating the pronunciation dictionary from a simple rule-based parser (Prahallad et al., 2012; Baby et al., 2016; Ramani et al., 2013; Pandey et al., 2017). Indian languages have a peculiar attribute of sharing the same phonetic space. However, they differ phonotactically[1] (Prahallad et al., 2012). So these attributes could be exploited to build an ASR system for achieving better performance. With the introduction of Digital India Mission in 2015, there have been efforts towards handling these low resource languages for building speech recognition technologies.

Different approaches have been proposed in acoustic modeling over the recent years to address the low resource speech recognition problem. In (Swietojanski et al., 2012), the attempt has been made to use cross-lingual acoustic data to initialize deep neural network (DNN) based acoustic models through unsupervised restricted Boltzmann machine (RBM) pre-training. It showed that unsupervised pre-training remains vital for the hybrid setups, especially with limited amounts of transcribed training data. The idea of transfer learning approach was introduced for handling low resource languages in (Imankulova et al., 2019; Cho et al., 2018; Das and Hasegawa-Johnson, 2015). Data augmentation approaches proposed in (Liu et al.,

[1] In general phonotactic is defined as the study of the rules governing the possible phoneme sequences in a language.

Proceedings of the 18th International Conference on Natural Language Processing, pages 205–212
Silchar, India. December 16 - 19, 2021.

2019; Thomas et al., 2020; Cui et al., 2015) were effective in low resource settings.

A different line of work (Chuangsuwanich, 2016; Thomas et al., 2012; Rahimi et al., 2019; Xu et al., 2015), where attempts have been made to extract multilingual features that help in improving low resource ASR systems started gaining prominence. In practice, a language identification (LID) block is used a front-end wherein it tries to predict the language first, and later it is mapped to the corresponding monolingual system or the best possible monolingual system. In this type of approach, the LID block, which acts as a front-end, should be as accurate as possible and robust enough to operate in a multi-thread environment. So to circumvent this issue, acoustic models where it handles multiple languages without any prior knowledge of the language have been proposed (Vydana et al., 2018). In (Shetty and Umesh, 2021), authors have explored the benefits of phonetic sound principles and treated each character unit across the languages as a separate entities with the help of Common Label Set (CLS) approach.

In this paper, the authors compare and analyze different Time Delay Neural Networks (TDNN) variants (Sugiyama et al., 1991) as they seem to be best suited for such kind of low resource speech recognition tasks. This paper investigates the effectiveness of different acoustic models for low resource multilingual speech recognition for six Indian languages (namely Hindi (Hi), Marathi (Mr), Odia (Od), Tamil (Ta), Telugu (Te), and Gujarati (Gu)). Among these languages, Hindi, Marathi, Odia and Gujarati are Indo-Aryan languages, while Tamil and Telugu fall under the category of Dravidian languages. As mentioned above, the authors considered six Indian languages ie., Hi, Mr, Od, Ta, Te and Gu, among these languages except Hi and Mr all others have different orthography (grapheme style). Moreover, every languages has its own training set they are (X_{Hi},L_{Hi}), (X_{Mr},L_{Mr}), (X_{Od},L_{Od}), (X_{Ta},L_{Ta}), (X_{Te},L_{Te}), and (X_{Gu},L_{Gu}), where X corresponds to the input acoustic sequence and L represents the label for target acoustic sequence. Initially, the monolingual systems are built for each language and trained with corresponding pairs which is acoustic sequence and its labels. Later, the authors attempt to show that the developed multilingual systems offer improved performance. As a part of multilingual system a joint acoustic model is trained, for which training dataset is constructed by pooling data from all the six languages, i.e., $(X_{all}, L_{all}) = (X_{Hi}, L_{Hi}) \cup (X_{Mr}, L_{Mr}) \cup (X_{Od}, L_{Od}) \cup (X_{Ta}, L_{Ta}) \cup (X_{Te}, L_{Te}) \cup (X_{Gu}, L_{Gu})$. The joint acoustic model is a single acoustic where the parameters are shared across all the six languages. This keeps the authors on similar lines with other research done on the development of ASR systems for low resource languages. In this work, a joint acoustic model-based ASR has been developed using different acoustic models. A joint acoustic model (JAM), which recognizes multiple languages with a single acoustic model, is widely appreciated. Many research groups have been actively working on this for the past few years (Chen et al., 2014). The usage of the joint acoustic model leverages the cross-lingual knowledge transfer. It reduces the complexity in the ASR pipeline significantly compared with the monolingual model as it has to maintain one model per language. The results show that the multilingual TDNN system result in lower word error rates (WER). We use KenLM (Heafield, 2011) and Recurrent Neural Network (RNN) based language models (Mikolov et al., 2010a) for decoding and rescoring respectively.

The organization of the remainder of the paper is as follows: Section 2 details the proposed lexicon along with the rationale behind the chosen mappings and database analysis. Section 3 explains the experimental setup. Section 4 discuss experimental results and few of the analysis which we have carried on. Finally, the study is concluded in Section 5, with possible future research directions to explore.

2 Dataset & Lexicon Details

INTERSPEECH 2018 has organized ASR Challenge as a special session (Srivastava et al., 2018). As part of the challenge, 40 hours of speech data has been released for three languages of Gujarati, Telugu, and Tamil. In continuation, INTERSPEECH 2021 has extended the challenge to six languages, i.e., Hindi (Hin), Marathi (Mar), Odia (Odi), Tamil (Tam), Telugu (Tel) and Gujarati (Guj) respectively (Diwan et al., 2021). The speech data statistics for the six languages are tabulated in the Table 1. To maintain a uniform sampling rate across all the languages, the authors have down-sampled all the wave files to 8kHz while building the multilingual system.

Table 1: Database statistics for training and test sets (where Trn and Tst indicates Training and Testing sets respectively)

	Hindi		Marathi		Odia		Tamil		Telugu		Gujarati	
	Trn	**Tst**	**Trn**	**Tst**	**Trn**	**Tst**	**Trn**	**Tst**	**Trn**	**Tst**	**Trn**	**Tst**
fs (KHz)	8	8	8	8	8	8	16	16	16	16	16	16
# Hours	95.05	5.55	93.89	5.0	94.54	5.49	40	5	40	5	40	5
# Utt	99926	3843	79432	4675	59782	3471	39131	3081	44882	3040	22807	3075
Avg dur	5.2	6.0	4.5	5.8	3.9	5.2	3.6	5.8	3.2	5.9	6.3	5.8

2.1 Database Analysis

From Table 1, it is observed that Gujarati has the least number of utterances when compared to other languages in the database. Hindi has the maximum number of utterances when compared to others. Among the six languages, Hindi and Marathi follow the same orthography. So, there is a chance that either of the languages might get benefited while training together. The text data statistics for all the six languages are shown in Table 2.

The last row of the table corresponds to number of graphemes. Among the total number of graphemes mentioned, for each language, the number of diacritic marks[2] are 16,16,16,7,17,17 respectively for Hindi, Marathi, Odia, Tamil, Telugu, and Gujarati. In analysis of the text data the authors found that, for a given word, the transcriptions are different in some cases. There are very few proper nouns in the database. In some utterances it is found that even English words are present but in the transliterated form. For example, *copy* → कॉपी, *book*→ बुक. So this paper tries to capture such variations by exploring different models to empirically see which performs better in the present scenario. Throughout this paper, the authors use multilingual approaches to solve the data scarcity problem.

2.2 Lexicon

In building a low-resource speech recognition system, the lexicon plays a vital role in providing a phonetic representation for a given word sequence. In this section, the authors will be discussing about the unified parser which was introduced by (Baby et al., 2016). The proposed parser was primarily used for speech synthesis task across all the languages. The parser has two-folds:

- In the first fold, for a given UTF-8 word sequence it converts into corresponding syllables. Recently this type of approach has been explored by (Shetty and Umesh, 2021) for building end-end speech recognition system to improve the performance of low-resource Indian Languages.
- Later, in the second fold, letter-to-sound rules are applied and the corresponding phone sequence is generated. This type of parser is popular in building Indic TTS. The authors hypothesize this will work in their ASR pipeline. Therefore, in this paper, a similar approach is followed to build low-resource speech recognition system for Indian languages.

We have come up with a unique parser to generate the pronunciation sequence for all the words as shown in Figure 1.

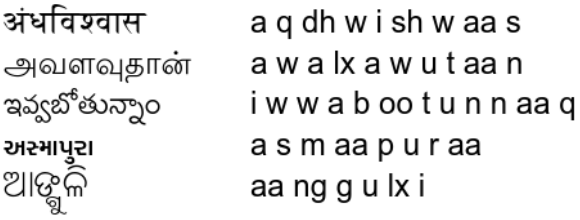


Figure 1: A example of Common Label Set based lexicon generated for Indian Languages

3 Speech Recognition Experimental Setup

In this section, the authors describe the experimental setup for both the monolingual and multilingual ASR systems. The word error rate (WER) is the metric used to evaluate the performance of ASR systems throughout this paper.

3.1 Language Modeling

The language model (LM) tries to estimate the probability of a hypothesized word sequence by learning the words from the text corpora. A Kneser-Ney (Chelba et al., 2010) trigram LM is built using SRILM toolkit (Stolcke, 2002) by normalizing training corpus. Recurrent neural networks

[2]A diacritical mark is a symbol that tells a reader how to pronounce a letter.

Table 2: Lexical analysis for training and test sets (where Trn and Tst indicates Training and Testing sets respectively)

	Hindi		Marathi		Odia		Tamil		Telugu		Gujarati	
	Trn	**Tst**	**Trn**	**Tst**	**Trn**	**Tst**	**Trn**	**Tst**	**Trn**	**Tst**	**Trn**	**Tst**
Uniq sent	4506	386	2543	200	820	65	30329	3060	34176	2997	20257	3069
Uniq words	6092	1681	3245	547	1584	334	50124	12279	43270	10859	39428	10482
OOV (%)	26.17		25.59		17.91		33.19		28.82		17.02	
# graphemes	69		61		68		50		64		65	

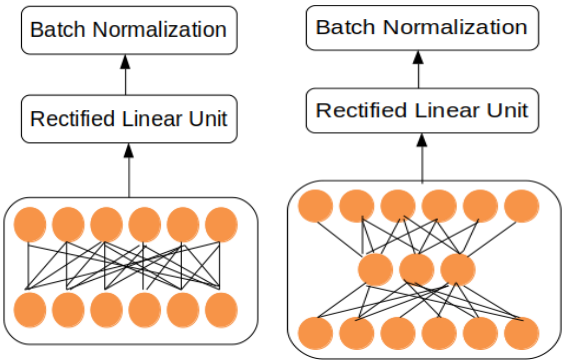
(RNN) based LM helps in preserving the information due to its feedback mechanism in its architecture (Mikolov et al., 2010b). It tries to take the previous information, $h_i = w_{i-1}, ..., w_1$ to predict the current word in sequence w_i. RNNLM has an input layer consisting of history vector h_i, previous word vector w_{i-1} and v_{i-2} is the context vector. The activation function used in this RNNLM is softmax. The input and output layer calculate the RNNLM probabilities $P_{RNNLM}\ (w_i|w_{i-1}, v_{i-2})$ using this activation function. Similarly, this process is repeated for calculating the probability of the next word.

In this paper, we use a TDNN-LSTM system for bulding RNNLM. The TDNN-LSTM has three LSTM layers with 1024 cells, 256 dimension projection and 9 layers of 1024 neurons. The L^2-regularization for hidden layers is 0.01 and output softmax is 0.004.

3.2 Acoustic Modeling

Sequence-trained TDNN architecture (Peddinti et al., 2015) is explored for building the baseline acoustic model using Kaldi toolkit (Povey et al., 2011). For training the baseline, the alignments from Gaussian Mixture Model-Hidden Markov Model (GMM-HMM) are considered. 13-dimensional Mel-Frequency Cepstral Coefficients (MFCC) features are used for mono-phone training. Next, Δ and $\Delta\Delta$ features are used for tri-phone modeling. The alignments generated from speaker adaptive training (SAT) based tri-phone models are used in the training of baseline TDNN. Feature space Maximum Likelihood Linear Regression (FMLLR)(Yao et al., 2012) is opted for SAT with 6000 tied states.

The input features to TDNN were 40 dimensional high resolution MFCCs with 100 dimensional iVectors for speaker adaptation (Madikeri et al., 2016). Initially, a three way speed perturbation is performed on the training data with 0.9, 1.0 and 1.1. Later, a volume perturbation has been adapted with a random factor between 0.9 to 2. All these perturbations are extracted for training iVectors.



(a) Feed-forward layer in conventional TDNN system. (b) Factorized layer in low rank TDNN system

Figure 2: Difference between a normal TDNN and low rank TDNN

Low Rank TDNN based architecture is explored which is different from the conventional TDNN. In this low rank TDNN, a bottleneck linear layer after every affine transformation of batch normalised ReLU is applied with skip connections. As it is factorized at every linear layer pair of each ReLU unit, it is also referred as low rank TDNN. The low rank TDNN based recipe can be seen in Kaldi [3]. The difference between a normal TDNN layer and a low rank TDNN layer is shown in Figures 2a and 2b.

In the low rank TDNN, the dimension for the linear bottleneck is 256. In general, the skip connection takes the inputs and outputs of the previous layer and selects other prior layers that are appended to the previous ones. In this experimental setup, apart from the output, it receives three non-consecutive layers as a skip connection. For example, consider 1280 x (256 x 2) as a dimension of conventional TDNN; after considering three skip

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

Table 3: The WER(%) results of monolingual (mono), multilingual (multi) models. The results showing the impact of the language model (LM) with and without external text data, and different acoustic models are also reported.

Type	Model	Hindi	Marathi	Odia	Tamil	Telugu	Gujarati	Average
Mono	GMM-HMM	69.03	33.22	55.78	48.81	47.27	28.33	**46.88**
	SGMM	61.01	26.41	51.36	39.68	28.08	28.61	**40.85**
	TDNN	30.16	19.65	35.58	21.89	21.67	17.35	**24.80**
	Low Rank TDNN_11L + External text data	16.30	12.35	13.48	15.36	11.36	13.65	**13.73**
Multi	SGMM	38.69	30.6	39.68	36.96	35.68	33.65	**35.87**
	TDNN_8L	35.16	31.58	37.6	35.98	34.89	31.62	**34.48**
	TDNN_LSTM	26.58	15.62	29.68	20.12	20.02	17.89	**21.65**
	TDNN_BLSTM	36.47	15.14	28.89	19.63	20.1	17.02	**22.87**
	TDNN_13L	16.36	12.65	23.22	19.04	19.34	16.88	**17.91**
	Low Rank TDNN_11L	17.27	10.69	21.65	18.34	18.68	15.08	**16.95**
	Low Rank TDNN_11L +External Telugu text data	17.27	10.69	21.65	18.34	9.86	15.08	**15.48**
	Low Rank TDNN_11L +External 6 language text data	8.37	6.35	10.78	9.15	9.86	7.25	**8.62**

connections, the dimension would be 1280 x (256 x 5).

4 Experimental Results & Analysis

The experiment results for the six languages are reported in the Table 3. The monolingual results show that TDNN system with 8 layers and six million parameters performs better than SGMM and GMM-HMM based models for monolingual systems. The TDNN system consistently improves the WER performance for all languages. The best WER is obtained for Gujarati which is 17.35%. We hypothesize that is due to low OOV%.

A multilingual neural network was trained by pooling the six languages using common label set. The JAM was trained with a similar configuration as the low rank TDNN system described in the Section 4. The results are reported for the SGMM, 7-layer TDNN, TDNN-LSTM, TDNN-BLSTM, 8-layer TDNN, 13-layer TDNN and low rank TDNN respectively. The best performance is observed for the case of low rank TDNN. The performance of models trained using SGMM is poorer than the performance of TDNN based models.

The triphones which are modeled by GMM-HMM do not share the common distribution among the six languages. This has led to the poor performance when the JAM is trained using SGMM. JAM trained using low rank TDNN has yielded better performance than TDNN and SGMM based systems. Unlike SGMMs, low rank TDNN and TDNN acoustic models are trained to model long-term temporal variations. The variabilities caused due to the presence of six languages have been effectively handled using low rank TDNNs due to the attributes of the architecture like skip connection, bottleneck linear transformation layer and batch normalization. From the last row of the Table 3, it is evident that due to inclusion of skip connections in low rank TDNN (with and without external text data), the performance of multilingual ASR system across all the languages has improved. Initially external text is collected for Telugu language and RNNLM is built using this. The authors evaluate this on low rank TDNN acoustic model. The collected external text and lexicon created using this can be found here [4]. The multilingual system

[4]External crawled data can be found here https://github.com/mirishkarganesh/icon_submission

with the same configurations as before was built with added external Telugu text. Due to which, as mentioned in the Table 3, the performance for Telugu language has improved. Motivated by the improvement in performance of ASR for Telugu, this was extended to all languages. The external text for other five languages is taken from AI4Bharat[5]. Similar improvement is observed for all languages with added external text.

5 Conclusion & Future Work

In this paper, a TDNN based multilingual ASR system for six Indian languages, i.e., Hindi, Marathi, Odia, Tamil, Telugu, and Gujarati was explored. Our experiment results show that the multilingual models achieve comparable results to the monolingual models when the parameters are in a comparable range. In few cases, ASR performance improved by including external text data while building languages model. CLS is investigated for studying the effectiveness of JAM for building a multilingual ASR system for six Indian languages. It is observed that low rank TDNN has shown superior performance over conventional TDNNs. Since most Indian languages are syllabic in nature and share a common phonetic space, the authors believe that the CLS approach can be further extended to more Indian languages in future. The feasibility of adapting LM along with the AM can also be explored for improving the performance of low resource multilingual ASR system. As India is a multilingual society, it is common occurrence for code-switching to be observed. In general, the authors would like to focus on two types of code-switching; (i) intra-sentential and (ii) inter-sentential. In the regular conversation, people try to switch, that is, the language exchange takes place at the sentence boundaries, and in the latter case, the languages switch into sentences, thus, creating a more complex problem. The six language multilingual model which we have built can deal with the inter-sentential cases. However, in Hindi and Marathi both intra and inter-sentential cases works as these two languages share same grapheme structure. Therefore, our future work will focus on recognizing intra-sentential code-switching utterances by exploring different architecture and utilizing monolingual data. The experimental findings from this paper will benefit researchers planning to build multilingual ASR systems for syllabic languages. We hope our work will encourage future research that leverages the findings.

[5]AI4Bharat text data can be found here https://indicnlp.ai4bharat.org/corpora/

References

Arun Baby, NL Nishanthi, Anju Leela Thomas, and Hema A Murthy. 2016. A unified parser for developing indian language text to speech synthesizers. In *International Conference on Text, Speech, and Dialogue*, pages 514–521. Springer.

Ciprian Chelba, Thorsten Brants, Will Neveitt, and Peng Xu. 2010. Study on interaction between entropy pruning and kneser-ney smoothing. In *Eleventh Annual Conference of the International Speech Communication Association*.

Dongpeng Chen, Brian Mak, Cheung-Chi Leung, and Sunil Sivadas. 2014. Joint acoustic modeling of triphones and trigraphemes by multi-task learning deep neural networks for low-resource speech recognition. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5592–5596. IEEE.

Jaejin Cho, Murali Karthick Baskar, Ruizhi Li, Matthew Wiesner, Sri Harish Mallidi, Nelson Yalta, Martin Karafiat, Shinji Watanabe, and Takaaki Hori. 2018. Multilingual sequence-to-sequence speech recognition: architecture, transfer learning, and language modeling. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 521–527. IEEE.

Ekapol Chuangsuwanich. 2016. Multilingual techniques for low resource automatic speech recognition. Technical report, Massachusetts Institute of Technology Cambridge United States.

Jia Cui, Brian Kingsbury, Bhuvana Ramabhadran, Abhinav Sethy, Kartik Audhkhasi, Xiaodong Cui, Ellen Kislal, Lidia Mangu, Markus Nussbaum-Thom, Michael Picheny, et al. 2015. Multilingual representations for low resource speech recognition and keyword search. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 259–266. IEEE.

Amit Das and Mark Hasegawa-Johnson. 2015. Cross-lingual transfer learning during supervised training in low resource scenarios. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Anuj Diwan, Rakesh Vaideeswaran, Sanket Shah, Ankita Singh, Srinivasa Raghavan, Shreya Khare, Vinit Unni, Saurabh Vyas, Akash Rajpuria, Chiranjeevi Yarra, Ashish Mittal, Prasanta Kumar Ghosh, Preethi Jyothi, Kalika Bali, Vivek Seshadri, Sunayana Sitaram, Samarth Bharadwaj, Jai Nanavati, Raoul Nanavati, Karthik Sankaranarayanan, Tejaswi Seeram, and Basil Abraham. 2021. Multilingual and code-switching asr challenges for

low resource indian languages. *arXiv preprint arXiv:2104.00235*.

Thierry Dutoit. 1997. *An introduction to text-to-speech synthesis*, volume 3. Springer Science & Business Media.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*, pages 187–197.

Aizhan Imankulova, Raj Dabre, Atsushi Fujita, and Kenji Imamura. 2019. Exploiting out-of-domain parallel data through multilingual transfer learning for low-resource neural machine translation. *arXiv preprint arXiv:1907.03060*.

Chunxi Liu, Qiaochu Zhang, Xiaohui Zhang, Kritika Singh, Yatharth Saraf, and Geoffrey Zweig. 2019. Multilingual graphemic hybrid asr with massive data augmentation. *arXiv preprint arXiv:1909.06522*.

Liang Lu, Arnab Ghoshal, and Steve Renals. 2013. Acoustic data-driven pronunciation lexicon for large vocabulary speech recognition. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 374–379. IEEE.

Srikanth Madikeri, Subhadeep Dey, Petr Motlicek, and Marc Ferras. 2016. Implementation of the standard i-vector system for the kaldi speech recognition toolkit. Technical report, Idiap.

Yajie Miao, Florian Metze, and Shourabh Rawat. 2013. Deep maxout networks for low-resource speech recognition. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 398–403. IEEE.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010a. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010b. Recurrent neural network based language model. volume 2, pages 1045–1048.

Ayushi Pandey, Brij Mohan Lai Srivastava, and Suryakanth V Gangashetty. 2017. Adapting monolingual resources for code-mixed hindi-english speech recognition. In *2017 International Conference on Asian Language Processing (IALP)*, pages 218–221. IEEE.

Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. 2015. A time delay neural network architecture for efficient modeling of long temporal contexts. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. 2011. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, CONF. IEEE Signal Processing Society.

Kishore Prahallad, E Naresh Kumar, Venkatesh Keri, S Rajendran, and Alan W Black. 2012. The iiit-h indic speech databases. In *Thirteenth annual conference of the international speech communication association*.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Multilingual ner transfer for low-resource languages.

B Ramani, S Lilly Christina, G Anushiya Rachel, V Sherlin Solomi, Mahesh Kumar Nandwana, Anusha Prakash, S Aswin Shanmugam, Raghava Krishnan, S Kishore Prahalad, K Samudravijaya, et al. 2013. A common attribute based unified hts framework for speech synthesis in indian languages. In *Eighth ISCA Workshop on Speech Synthesis*.

D Raj Reddy. 1976. Speech recognition by machine: A review. *Proceedings of the IEEE*, 64(4):501–531.

Vishwas M. Shetty and S. Umesh. 2021. Exploring the use of Common Label Set to Improve Speech Recognition of Low Resource Indian Languages. pages 7228–7232.

Brij Mohan Lal Srivastava, Sunayana Sitaram, Rupesh Kumar Mehta, Krishna Doss Mohan, Pallavi Matani, Sandeepkumar Satpal, Kalika Bali, Radhakrishnan Srikanth, and Niranjan Nayak. 2018. Interspeech 2018 low resource automatic speech recognition challenge for indian languages. In *SLTU*, pages 11–14.

Andreas Stolcke. 2002. Srilm-an extensible language modeling toolkit. In *Seventh international conference on spoken language processing*.

Masahide Sugiyama, Hidehumi Sawai, and Alexander H Waibel. 1991. Review of tdnn (time delay neural network) architectures for speech recognition. In *1991., IEEE International Sympoisum on Circuits and Systems*, pages 582–585. IEEE.

Pawel Swietojanski, Arnab Ghoshal, and Steve Renals. 2012. Unsupervised cross-lingual knowledge transfer in dnn-based lvcsr. In *2012 IEEE Spoken Language Technology Workshop (SLT)*, pages 246–251. IEEE.

Samuel Thomas, Kartik Audhkhasi, and Brian Kingsbury. 2020. Transliteration based data augmentation for training multilingual asr acoustic models in low resource settings. *Proc. Interspeech 2020*, pages 4736–4740.

Samuel Thomas, Sriram Ganapathy, and Hynek Hermansky. 2012. Multilingual mlp features for low-resource lvcsr systems. In *2012 IEEE International*

Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 4269–4272. IEEE.

Hari Krishna Vydana, Krishna Gurugubelli, Vishnu Vidyadhara Raju Vegesna, and Anil Kumar Vuppala. 2018. An exploration towards joint acoustic modeling for indian languages: Iiit-h submission for low resource speech recognition challenge for indian languages, interspeech 2018. In *INTERSPEECH*, pages 3192–3196.

Wikipedia. 2021. List of languages by number of native speakers in India. https://en.wikipedia.org/w/index.php?title=List_of_languages_by_number_of_native_speakers_in_India&oldid=1012063187. [Online; accessed 22-March-2021].

Haihua Xu, Van Hai Do, Xiong Xiao, and Eng Siong Chng. 2015. A comparative study of bnf and dnn multilingual training on cross-lingual low-resource speech recognition. In *Sixteenth annual conference of the international speech communication association*.

Kaisheng Yao, Dong Yu, Frank Seide, Hang Su, Li Deng, and Yifan Gong. 2012. Adaptation of context-dependent deep neural networks for automatic speech recognition. In *2012 IEEE Spoken Language Technology Workshop (SLT)*, pages 366–369. IEEE.