Interspeech 2018
2-6 September 2018, Hyderabad

Development of Large Vocabulary Speech Recognition System with Keyword Search for Manipuri

Tanvina Patel, Krishna DN, Noor Fathima, Nisar Shah, Mahima C, Deepak Kumar, Anuroop Iyengar

Cogknit Semantics, Bangalore, India

{tanvina, krishna, noorfathima, nisar, mahima, deepak, anuroop}@cogknit.com

Abstract

Research in Automatic Speech Recognition (ASR) has witnessed a steep improvement in the past decade (especially for English language) where the variety and amount of training data available is huge. In this work, we develop an ASR and Keyword Search (KWS) system for Manipuri, a low-resource Indian Language. Manipuri (also known as Meitei), is a Tibeto-Burman language spoken predominantly in Manipur (a northeastern state of India). We collect and transcribe telephonic read speech data of 90+ hours from 300+ speakers for the ASR task. Both state-of-the-art Gaussian Mixture-Hidden Markov Model (GMM-HMM) and Deep Neural Network-Hidden Markov Model (DNN-HMM) based architectures are developed as a baseline. Using the collected data, we achieve better performance using DNN-HMM systems, i.e., 13.57% WER for ASR and 7.64% EER for KWS. The KALDI speech recognition tool-kit is used for developing the systems. The Manipuri ASR system along with KWS is integrated as a visual interface for demonstration purpose. Future systems will be improved with more amount of training data and advanced forms of acoustic models and language models.

Index Terms: Manipuri, ASR, KWS, GMM-HMM, DNN-HMM

1. Introduction

India is a diverse and multilingual country. It has vast linguistic variations spoken across its billion plus population. Languages spoken in India belong to several language families, mainly the Indo-Aryan languages spoken by 75% of Indians, followed by Dravidian languages spoken by 20% of Indians and other language groups belonging to the Austroasiatic, Sino-Tibetan, Tai-Kadai, and a few other minor language families and isolates [1]. Speech and language technologies are now designed to provide interfaces for digital content that can reach the public and facilitate the exchange of information across people speaking different languages. Therefore, it is a sound area of research to develop speech technologies in multilingual societies such as India that has about 22 major languages, i.e., Assamese, Bengali (Bangla), Bodo, Dogri, Gujarati, Hindi, Kannada, Kashmiri, Konkani, Maithili, Malayalam, Manipuri (Meitei), Marathi, Nepali, Odia, Punjabi, Sanskrit, Santali, Sindhi, Tamil, Telugu and Urdu. These languages are written in 13 different scripts, with over 1600 languages/dialects [2]. Over the years, several efforts have been made to develop resources/data for various speech technology related applications in Indian languages [3, 4, 5]. With the growing use of Internet and with the idea of digitalization, speech technology in Indian languages will play a crucial role in the development of applications in various domains like agriculture, health-care and services for common people, etc. [6].

On the similar lines, in this work, we present our efforts in building a speech corpus for Manipuri language and using it for Speech-to-Text (STT) and Keyword Search (KWS) application. Manipuri (also known as Meitei) is one of the official languages spoken widely in the northeastern part of India. It belongs to the Sino-Tibetan family of languages and is mostly spoken by the inhabitants of Manipur and people at Indo-Myanmar border [7]. According to the 2001 census, there are around 1.46M native Manipuri speakers [8]. Manipuri has three main dialects: Meitei proper, Loi and Pangal. It is tonal in nature and the tones are of two types: level and falling [9]. It is mostly written using the Bengali and Meitei Mayek script (in the 18th century the Bengali script was adopted over Meitei script). Currently, a majority of the Manipuri documents are written with Bengali script. The Bengali script has 55 symbols to represent 38 Manipuri phonemes. It's phonological system consists of three major systems of sound, i.e., vowels, consonants, and tones [10]. It is the official language in government offices and is a medium of instruction up to the undergraduate level in Manipur.

Data collection and speech technology research are much explored for languages like Hindi, Tamil, Telugu, etc., having larger speaking population [5, 11]. Speech technology applications for Manipuri language includes Text-to-Speech (TTS) synthesis developed from the Consortium efforts for Indian languages [12, 13]. A few studies have been carried out demonstrating speech recognition application in Manipuri language. This includes developing a phonetic engine for language identification task using a very small dataset of 5 hours [14]. Further, on this data, phoneme-based KWS has also been demonstrated stating an accuracy of 65.24% [15]. In [16], the effect of phonetic modeling on Manipuri digit recognition had also been studied. A brief description of various other technologies related to Manipuri language processing is given in [17]. One of the main reason of Manipuri being less researched is the lack of speech data for tasks like Automatic Speech Recognition (ASR) as compared to the data available for other northeastern languages like, Assamese and Bengali [18, 19]. In addition, for Manipuri, less digitized data is available from Internet sources (especially in UTF-8 format), i.e., it is a less computerized language and hence, requires more text processing. Due to the landscape, the language is known to only a few people. Several other challenges with respect to Manipuri language processing are mentioned in [20]. Thus, developing an ASR system for Manipuri language has been a challenge.

In this paper, we present our efforts in developing an ASR system for Manipuri language. The rigorous process of data collection, speech data transcription, etc. have been described. In addition, the performance of the KWS jointly with the ASR system is highlighted. The KALDI speech recognition toolkit [21] has been used to develop a baseline system for Manipuri language. The models have been ported and a visual interface has been created for demo purpose.

10.21437/Interspeech.2018-2133

2. Building the Manipuri Speech Corpora

2.1. Text Data Collection

While collecting the text for applications like ASR, it is important to have a huge text corpus source from which the optimal sub-set has to be extracted as the reliability of the language model depends on it. The corpus should be unbiased to a specific domain and large enough to convey the syntactic behavior of the language. Text for Manipuri is collected from commonly available and mostly error-free sources on the Internet [22]. A total of ∼57000+ prompts were collected from online Manipuri websites and articles from news sites. Some part of the corpora has been corrected with the help of Manipuri linguists.

It is required that the text is in a representation that can be easily processed. Normally, the text on web-sources will be available in a specific font. So, font converters are needed to convert the corpus into the desired electronic character code. Thus, all the text is made available in the UTF-8 format and it is ensured that there are no mapping and conversion errors.

2.2. Speech Data Recording

Recordings corresponding to ∼100 hours of telephonic and read speech from 300+ native Manipuri speakers is collected. A dedicated portal was built where the user registers and logs in. The user is asked to enter details such as name, age, email, phone number, etc. and is then asked to dial a toll-free number. The text appears on the portal screen and the speakers speak over a mobile device a set of assigned sentences.

After reading the displayed text, the speaker presses the # key and the next text appears. Each user receives a set of 100-150 utterances, corresponding to around 30 minutes of speech data. Recordings are done in normal office environments, hostels, colleges, i.e., quiet as well as environments with babble and other noises. Once the recordings are done, the user receives coupons as a token of appreciation for his/her contribution towards the data collection process as shown in Figure 1.

2.3. Speech Data Annotation

2.3.1. Data Cleaning

The speech data collected may have empty audio files or corrupt files. There may be instances when the speaker does not speak, difficulty in reading, technical issues, etc. Thus, the recordings are scanned to identify such cases and remove the audio files.

2.3.2. Transcription

In the next task, the speech data is transcribed. The speech data is validated to check for any mismatch between the text and audio, etc. The data is also tagged for non-speech parts and other fillers such as: *no-speech*: when there is no speech/silence in the recording, *hes*: if the speaker hesitates while speaking, *cough*: if the speaker coughs while speaking, *laugh*: if the speaker laughs while speaking, *sta*: when there is background noise in the recordings, e.g., fan noise, *int*: when there is foreground noise in the recordings, *babble*: when other speakers are speaking in the background, *ring*: when some sort of ringing is there in the recordings, e.g., telephone ringing. During transcription, the unwanted very large silences at the start and end, etc. are removed, hence, the amount of data will be less after transcription. The annotation/transcription was carried out by 5 trained Manipuri linguists in a period of 6 months using wave-surfer as a tool for transcription [23] (as shown in Figure 2).

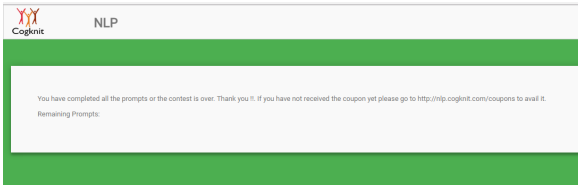

Figure 1: *View of the Manipuri speech data recording portal*

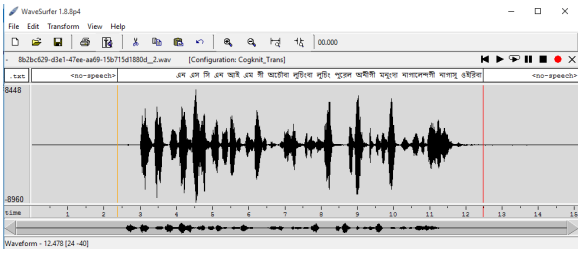

Figure 2: *Snapshot of the wavesurfer tool used for labeling*

2.4. Grapheme-to-Phoneme (G2P)

A Grapheme-to-Phoneme module converts words to pronunciations. In Indic languages, the letters and sounds have a close correspondence; there is an almost one-to-one relation between the written form of the words and their pronunciations. The words are first broken down into their syllables. A syllable is a unit of spoken language that is spoken without interruption. Hence, breaking an input word into syllables works like a delimiter that leads to phoneme mapping. The phonemes are then mapped to a standard form of representing phonemes. Manipuri lexicon is obtained through a rule-based parser developed as part of TBT-Toolkit [13]. It uses the Common Label Set (CLS) that provides a standardized representation of phonemes across different Indian languages [24, 25]. Occasional issues where identified and where manually rectified by the linguists. A part of the lexicon generated for Manipuri language is shown below.

অ	a
অং	a q
অংগো	a q g o
অংব্রাসুদা	a q b r aa s u d aa
অংব্রুসুদা	a q b r u s u d aa
.....	

3. Speech Recognition System

The overall architecture for the Manipuri speech recognition system includes the data collection and training pipeline for ASR and the testing pipeline for ASR as well as KWS as shown in Figure 3. In this section, we discuss the basic building blocks of the system, i.e., the language modeling, and acoustic modeling including the feature extraction process and decoding stage.

3.1. Language Model (LM)

The language model estimates the probability of a hypothesized word sequence, or LM score, by learning the correlation between words from a training text corpora. The LM score often can be estimated more accurately if the prior knowledge about the domain or task is known [27]. The CMU-Cambridge Language Model (LM) toolkit is used to build the LM [28].

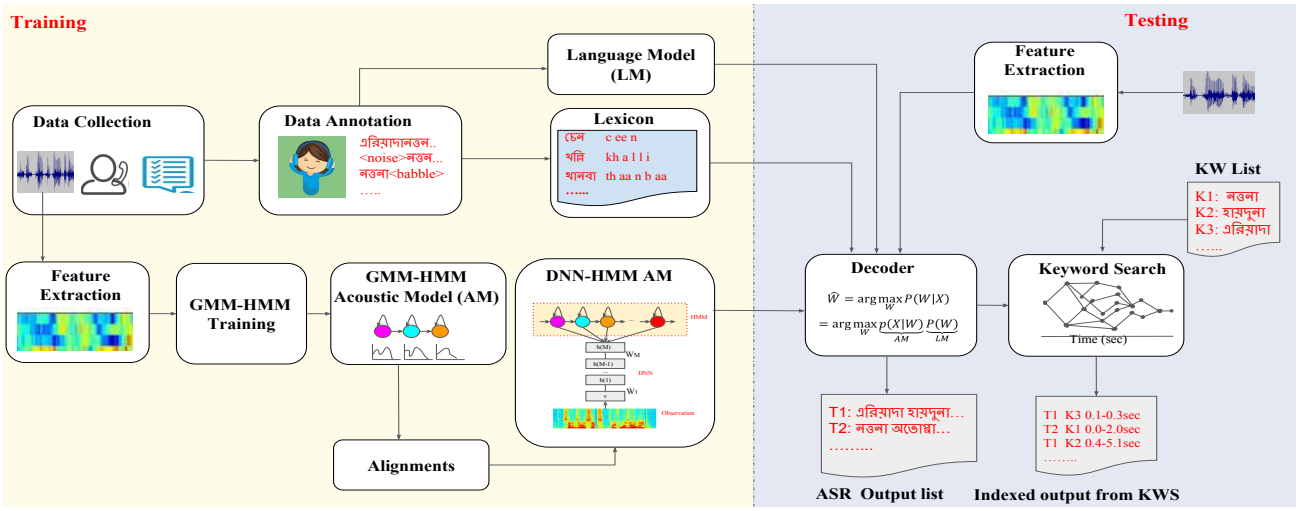

Figure 3: *Overall architecture of Manipuri ASR and KWS system. Adopted from [26]*

3.2. Acoustic Models (AMs)

The Manipuri ASR systems were trained using two different Acoustic Models (AMs), i.e., a Gaussian Mixture Model (GMM)-Hidden Markov Model (HMM) and a hybrid Deep Neural Network (DNN)-HMM for building the ASR system.

3.2.1. GMM-HMM systems

In the 1980s, state-of-the-art ASR systems used Mel-Frequency Cepstral Coefficient (MFCC) or RelAtive Spectral Transform-Perceptual Linear Prediction (RASTA-PLP) [29, 30] as feature vectors along with GMM-HMM. These GMM-HMM AMs were trained using the Maximum Likelihood (ML) training criterion. Later, in the 2000s, sequence discriminative training algorithms such as Minimum Classification Error (MCE) and Minimum Phone Error (MPE) were proposed that further improved the ASR accuracy [31]. In this work, MFCC features are extracted with the Δ and $\Delta\Delta$ features for initial speaker independent GMM-HMM training. The speaker dependent GMM-HMM model is build using Feature space Maximum Likelihood Linear Regression (FMLLR) features [32].

3.2.2. DNN-HMM systems

Over the last few years, efficient methods for training DNNs for ASR have been witnessed [33] showing that DNNs are better classifiers than GMMs [34]. The output layer accommodates the number of HMM states that arise when each phone is modeled by a number of different triphone HMMs taking into account the phones on either side [35]. The GMM-HMM model is used to obtain alignments for training data and finally a DNN is trained by feeding the FMLLR features as the input and senone probabilities as the output. The DNN training uses p-norm activation function [36] with cross-entropy as the loss function using natural Stochastic Gradient Descent (SGD) [37].

3.3. Decoding

The decoding step generates the most probable hypothesis for a given speech and generates multiple hypothesis based on the AM and LM weights. The total cost in generating hypothesis is based on AM and the weighted LM cost. The DNN gives posterior probability for every speech frame which is combined with LM cost and embedded into a FST graph. An FST graph is composed of lexicon, LM, context and HMM states [38]. Senone posteriors are generated from DNN to search the graph for the 1st best output. On decoding a test audio signal, corresponding lattices are generated which are used for KWS. In the statistical framework, the fundamental equation of speech recognition states that, if $\boldsymbol{X}$ is the sequence of acoustic feature vectors (observations) and $\boldsymbol{W}$ denotes a word sequence, the most likely word sequence $\boldsymbol{W}^*$ is given by,

$$\boldsymbol{W}^* = arg\max_{W} P(\boldsymbol{W}|\boldsymbol{X}) \quad (1)$$

Applying Bayes Theorem on (1), we get,

$$\boldsymbol{W}^* \approx arg\max_{W} P(\boldsymbol{X}|\boldsymbol{W})P(\boldsymbol{W}) \quad (2)$$

where in (2) likelihood $P(\boldsymbol{X}|\boldsymbol{W})$ is generally called the AM as it estimates the probability of a sequence of acoustic observations, conditioned on the word string and $P(\boldsymbol{W})$ is the LM.

4. Experimental Setup and Results

4.1. Details of the Setup

4.1.1. Database

For training the ∼90 hours dataset is split in training sets of 30, 40, 50 and 70 hours. The end system built with 70 hours of training set includes ∼65,000 words and ∼36000 sentences. The test data is a fixed ∼11 hours set corresponding to ∼6500 words from about 60 speakers. The additional 10 hours amounts to non-speech tags in the dataset.

4.1.2. Tools and Performance Measures

The KALDI toolkit is used for ASR system building [21]. As the collected data is read in nature and the speaker information is known, we use the LibriSpeech recipe [39] that uses speaker adaptive training. The CMU-LM toolkit is used to build a 2-gram LM [28]. The LM was built on around 37213 sentences, with an average of 10-15 words per sentence. As the text available for training the LM is less, we build a 2-gram model as all the possible combinations needed for 3-gram pairs would not be available. The performance of ASR and KWS is evaluated using Word Error Rate (WER) and Equal Error Rate (EER) [40].

4.1.3. Parameters Settings

We use 3 hidden layer DNN for model training. Each hidden layer has 2000 dimensional hidden units with p-norm activation. Here, p=2 and group size = 5 which leads to input p-norm dimension 2000 and output p-norm dimension 400. The DNN has an input layer which takes 360-dimensional input and the output of the DNN is 2365 context dependent phonemes states. The input to the neural network is obtained by concatenating 4 left and 4 right FMLLR features each of dimension 40. The outputs are obtained by GMM-HMM alignment. We minimize the cross-entropy loss function using back-propagation with an initial learning rate of 0.01 and final learning rate of 0.001.

4.1.4. KWS Setup

The KW set includes 100 unique unigram words randomly selected from the test set. There are a total of 4068 instances of the KWs in the test set as the KWs may have many occurrences. The KWS module indexes the lattices and given a keyword/phrase searches through the indexed lattices to obtain a list of occurrences of the desired KWs [41].

4.1.5. Infrastructure Details

Training is done with a minibatch size of 512 which fits on single GPU card. Training of the AM is done on 2 Telsa K40 GPUs (each 12GB RAM, Cuda cores: 2880). It is observed that the use of GPUs significantly reduces training time [26]. For the final set of training with more amount of data, we used GeForce GTX 1080 Ti (with 11GB RAM and Cuda cores = 3584).

4.2. Experimental Results

The results of the Manipuri ASR and KWS system is shown in Table 1. It is observed that, on increasing the training data, the WER and EER for DNN-HMM systems is less than that of the GMM-HMM systems. It is observed that the KWS system gives better performance as it works on n-best lattices as compared to ASR that uses only the 1st best lattice. For the 50 hours of training data, we achieve the best performance of 13.57% WER (ASR) and 7.64% EER (KWS). For KWS, as compared to the 4068 instances of KWs in the test set, the detected KWs were 7688 and 7548 for GMM-HMM and DNN-HMM systems, respectively. Therefore, for KWS, false alarms occur which is reflected in the EER.

The results for the final experiment with larger training dataset of 70 hours show that the WER and EER increase by 2%. It may be due to the reason that the network parameters (e.g., the layer sizes) needs to be tuned considering the smaller amount of data. Further, a modern architecture must be used that gives performance gain both in WER and decoding speed.

Table 1: *Results for Manipuri ASR systems and KWS system*

Training (∼Hours)	No. of Speakers	Systems	ASR (%WER)	KWS (%EER)
30	117	GMM-HMM	19.41	11.41
		DNN-HMM	15.26	7.94
40	172	GMM-HMM	19.57	11.38
		DNN-HMM	14.50	7.66
50	232	GMM-HMM	19.28	11.58
		DNN-HMM	**13.57**	**7.64**
70	261	GMM-HMM	19.94	11.8
		DNN-HMM	15.02	9.24

4.3. Demo for Manipuri ASR and KWS

Figure 4 shows the visual interface that is available to display the Manipuri ASR and KWS system. The interface has options to play the selected wavefile. As the audio is played, the decoded ASR transcript is displayed in the second panel. If the words in the transcript match with the KW list, then the KW is highlighted. It is possible that the ASR transcript does not exactly match to any of the KWs, however, the search algorithm may identify the presence of the KWs in the audio file and give a possible hit. These cases are displayed in the third panel. As the KWs are identified with a start and end duration (shown in Figure 3), in the visual interface, a feature is provided to click the transcript and the corresponding location in the audio panel is highlighted. Details regarding the integration of the models for demonstration are given in [42].

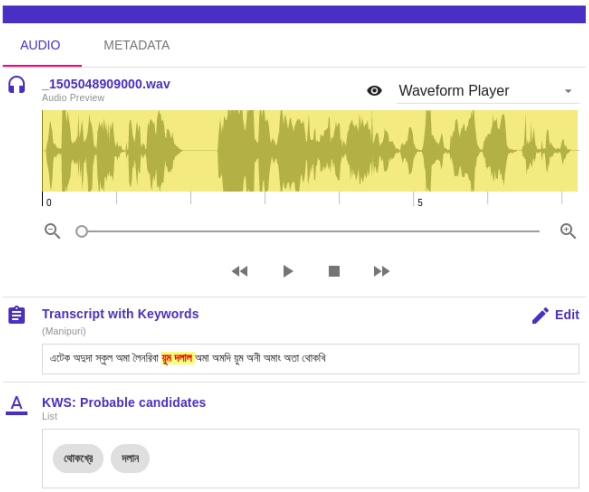

Figure 4: *Demonstration of Manipuri ASR and KWS system*

5. Summary and Conclusions

In this paper, we demonstrate our efforts in building an initial Manipuri ASR and KWS system. On an average, the DNN-HMM system shows improvement of 1% WER for every 10 hours of data (excluding the final experiment). Overall, the KWS improves by 34% from GMM-HMM to DNN-HMM. Additional data is being collected, multiple pronunciations are being added to the lexicon and the use of advanced forms of acoustic models and language models are currently in process. As a part of the future work, the collected data is being explored for language identification task with regionally similar languages like Assamese and Bengali. Currently, the systems are built for read speech and extending it to conversational speech data would assist in building robust applications.

6. Acknowledgements

The authors would like to thank the Manipuri speakers for giving their valuable time in recording. In addition, we acknowledge the contribution of the Manipuri linguists, Bijaya Takhellambam, Aheibam Linthoi Chanu, Sarita Wahengbam, Biyanati Haobam, who carried out the speech annotation task and Dinesh Wangkhem for managing the recording and transcription. We thank Rajashree Jayabalan and the product engineering team at Cogknit for their efforts in integrating and developing the visual interface for the Manipuri ASR and KWS system.

7. References

[1] Wikipedia. (2018) Languages of India. [Online]. Available: https://en.wikipedia.org/wiki/Languages_of_India

[2] New world encyclopedia, languages of India (2018). [Online]. Available: http://www.newworldencyclopedia.org/entry/Languages_of_India

[3] A. Baby, A. L. Thomas, N. N. L, and TTS Consortium, "Resources for Indian languages," in *Int. Conf. on Text, Speech, and Dialogue (TSD), Springer*, 2016, pp. 37–43.

[4] INTERSPEECH-2018. Special Session: Low resource speech recognition challenge for Indian languages. [Online]. Available: https://www.microsoft.com/en-us/research/event/interspeech-2018-special-session-low-resource-speech-recognition-challenge-indian-languages/

[5] G. A. Numanchipalli *et al.*, "Development of Indian language speech databases for large vocabulary speech recognition systems," in *SPECOM*, Patras, Greece, 2005, pp. 1–5.

[6] A. Mohan, R. Rose, S. H. Ghalehjegh, and S. Umesh, "Acoustic modelling for speech recognition in Indian languages in an agricultural commodities task domain," *Speech Comm.*, no. 56, pp. 167–180, Jan. 2014.

[7] Wikipedia. (2018) Meitei Language. [Online]. Available: https://en.wikipedia.org/wiki/Meitei_language

[8] G. F. Simons and C. D. Fennig. (2018) Ethnologue: Languages of the world. Dallas, Texas. [Online]. Available: https://www.ethnologue.com/language/mni

[9] U. Shastri and A. U. Kumar, "Production and perception of lexical tones in Manipuri language," *Jour. of Advanced Linguistic Studies*, vol. 3, no. 1-2, pp. 216–231, 2014.

[10] L. G. Singh, L. Laitonjam, and S. R. Singh, "Automatic syllabification for Manipuri language," in *Int. Conf. on Comput. Linguistics (COLING)*, Osaka, Japan, 2016, pp. 349–357.

[11] M. Kumar, N. Rajput, and A. Verma, "A large-vocabulary continuous speech recognition system for Hindi," *IBM Journal of Research and Development*, vol. 48, no. 5-6, pp. 703–715, 2004.

[12] H. A. Patil *et al.*, "A syllable-based framework for unit selection synthesis in 13 Indian languages," in *Int. Conf. Oriental COCOSDA jointly with Conf. on Asian Spoken Lang. Research and Evaluation (O-COCOSDA/CASLRE)*, Gurgaon, India, 2013.

[13] A. S. Ghone *et al.*, "TBT (toolkit to build TTS): A high performance framework to build multiple language HTS voice," in *INTERSPEECH*, Stockholm, Sweden, 2017, pp. 3427–3428.

[14] S. K. Dutta, S. Nandakishor, and L. J. Singh, "Development of Manipuri phonetic engine and its application in language identification," *Int. Jour. of Engg and Tech. Research (IJETR)*, vol. 3, no. 8, pp. 2454–4698, 2015.

[15] L. Rahul, S. Nandakishor, L. J. Singh, and S. K. Dutta, "Design of Manipuri keywords spotting system using HMM," in *Conf. on Computer Vision, Pattern Recogn., Image Process. and Graphics (NCVPRIPG)*, Jodhpur, India, 2014, pp. 1–4.

[16] S. I. Choudhury *et al.*, "Effect of phonetic modeling on Manipuri digit recognition systems using CDHMMs," in *Applied Computing*, S. Manandhar, J. Austin, U. Desai, Y. Oyanagi, and A. K. Talukder, Eds. Berlin, Heidelberg: Springer, 2004, pp. 153–160.

[17] N. Ningthoujam and P. V. R, "Designing of a feature extraction model for Manipuri language," *Int. Jour. for Scientific Research and Develop.*, vol. 4, no. 2, pp. 1471–1473, 2016.

[18] LDC. (2016) IARPA Babel Assamese Language Pack. [Online]. Available: https://catalog.ldc.upenn.edu/ldc2016s06

[19] LDC. (2016) IARPA Babel Bengali Language Pack. [Online]. Available: https://catalog.ldc.upenn.edu/LDC2016S08

[20] S. Singh, S. Gunasekaran, A. Kumar, and K. P. Soman, "A short review about Manipuri language processing," *Research Jour. of Recent Science*, vol. 3, no. 3, pp. 99–103, 2014.

[21] D. Povey *et al.*, "The kaldi speech recognition toolkit," in *Workshop on Auto. Speech Recogn. and Understanding (ASRU)*, Hawaii, US, 2011, pp. 1–4.

[22] Poknapham. (2016) Manipuri News. [Online]. Available: http://www.poknapham.in/

[23] K. Sjlander and J. Beskow, "Wavesurfer-An open source speech tool," in *Int. Conf. on Spoken Lang. Process. (ICSLP)*, Beijing, 2000, pp. 1–4.

[24] B. Ramani *et al.*, "A common attribute based unified HTS framework for speech synthesis in Indian languages." in *ISCA Speech Synthesis Workshop (SSW)*, 2013, pp. 291–296.

[25] Indian Language TTS Consortium and ASR Consortium. (2016) Indian Language Speech sound Label set (ILSL12). [Online]. Available: https://www.iitm.ac.in/donlab/tts/downloads/cls/cls_v2.1.6.pdf

[26] D. N. Krishna *et al.*, "Automatic speech recognition for low-resource Manipuri language," in *GPU Technology Conf. (GTC)*, SanJose, California, 26-29 March 2018.

[27] D. Yu and L. Deng, *Automatic Speech Recognition: A Deep Learning Approach*, ser. Signals and Communication Technology. Springer, 2015.

[28] P. Clarkson and R. Rosenfeld, "Statistical language modeling using the CMU-Cambridge toolkit," in *EUROSPEECH*, 1997, pp. 2707–2710.

[29] S. B. Davis and P. Mermelstein, "Comparison of parametric representation for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. on Acous., Speech and Sig. Process.*, vol. 28, no. 4, pp. 357–366, 1980.

[30] H. Hermansky, N. Morgan, A. Bayya, and P. Kohn, "RASTA-PLP speech analysis technique," in *Int. Conf. on Acous., Speech and Sig. Process. (ICASSP)*, Washington, DC, 1992, pp. 121–124.

[31] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *INTERSPEECH*, Lyon, France, 2013, pp. 2345–2349.

[32] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Comp. Speech and Lang. (CSL)*, vol. 12, no. 2, pp. 75–98, 1998.

[33] G. Hinton *et al.*, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Sig. Process. Magazine*, vol. 29, no. 6, pp. 82–97, Nov 2012.

[34] A. rahman Mohamed, G. Dahl, and G. Hinton, "Deep belief networks for phone recognition," *NIPS Workshop on Deep Learning for Speech Recognition and Related Applications*, pp. 1–9, 2009.

[35] K.-F. Lee and H.-W. Hon, "Speaker-independent phone recognition using hidden Markov models," *IEEE Trans. on Acous., Speech and Sig. Process.*, vol. 37, no. 11, pp. 1641–1648, 1989.

[36] X. Zhang, J. Trmal, D. Povey, and S. Khudanpur, "Improving deep neural network acoustic models using generalized maxout networks," in *ICASSP*, Florence, Italy, 2014, pp. 215–219.

[37] D. Povey, X. Zhang, and S. Khudanpur, "Parallel training of deep neural networks with natural gradient and parameter averaging," 2014. [Online]. Available: http://arxiv.org/abs/1410.7455

[38] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition," *Comp. Speech and Lang. (CSL)*, vol. 16, no. 1, pp. 69–88, 2002.

[39] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *Int. Conf. on Acous., Speech and Sig. Process. (ICASSP)*, Brisbane, Australia, pp. 5206–5210.

[40] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The DET curve in assessment of detection task performance," in *EUROSPEECH*, 1997, pp. 1895–1898.

[41] D. Can and M. Saraclar, "Lattice indexing for spoken term detection," *IEEE Trans. on Audio, Speech, and Lang. Process.*, vol. 19, no. 8, pp. 2338–2347, Nov 2011.

[42] T. B. Patel *et al.*, "An automatic speech transcription system for Manipuri language," *accepted* for Show and Tell Session in *INTERSPEECH*, Hyderabad, Sept. 2018.