Comparison of Different Speech Features for Connected Number Recognition of Indian Vernacular Languages

Mayurakshi Mukherji
Research and Development Centre
Hitachi India Private Limited
Bengaluru, India
Email: mayurakshi.mukherji@hitachi.co.in

Shreyas Kulkarni
Research and Development Centre
Hitachi India Private Limited
Bengaluru, India
Email: shreyas.kulkarni@hitachi.co.in

Vivek Kumar
Research and Development Centre
Hitachi India Private Limited
Bengaluru, India
Email: vivek.kumar@hitachi.co.in

Senthil Raja G, Senior Member, IEEE
Research and Development Centre
Hitachi India Private Limited
Bengaluru, India
Email: senthil.raja@hitachi.co.in

Thiruvengadam Samon
Research and Development Centre
Hitachi India Private Limited
Bengaluru, India
Email: thiruvengadam.s@hitachi.co.in

Kingshuk Banerjee
Research and Development Centre
Hitachi India Private Limited
Bengaluru, India
Email: kingshuk.banerjee@hitachi.co.in

Yuichi Nonaka
Research and Development Centre
Hitachi India Private Limited
Bengaluru, India
Email: ynonaka@hitachi.co.in

***Abstract*— This paper presents the experimental results and comparative analysis of Connected Number Speech Recognition (CNR) models trained using four feature combinations: Mel Frequency Cepstral Coefficient (MFCC), MFCC+Pitch, Perceptual Linear Prediction (PLP), and PLP+Pitch. The set of experiments is conducted for five Indian Native Languages- Bengali, Hindi, Tamil, Kannada, and Marathi. We have collected connected number speech datasets for all five languages and have trained speech recognition models. The Kaldi speech recognition toolkit was used to train acoustic model and the SRILM toolkit was used to build an N-gram language model to prepare a speech recognition system. The model performances were compared and analyzed using Word Error Rate (WER) and Sentence Error Rate (SER) as accuracy metrics. Although, above mentioned Indian languages are atonal in nature, our experiments show that adding pitch features along with MFCC features show overall improvements in WER and SER Values for connected number speech recognition. Moreover, all the speech recognition models are trained under identical conditions but show significantly different WER and SER for different languages.**

Keywords-MFCC; PLP; speech features; pitch; Indian Langauge.

I. INTRODUCTION

Speech is a natural and effective way of communication between human beings, which can be used to communicate with machines since it can be captured by a microphone as a vibration signal with respect to time. This signal can be processed, and the speech content of the signal can be recognized and understood to perform further downstream tasks. Automatic Speech Recognition (ASR) system has two types of architectures broadly classified as (a) conventional acoustic model plus language model-based ASR and (b) End to End ASR. The conventional architecture makes use of statistical, neural network based, or hybrid (of both) models to develop ASR systems [1]. These models are typically trained on speech features extracted from speech data. Even though diverse types of representations are available in terms of extracted features, extensive robustness is still being investigated. In ASR systems, every speech feature vector extracted from the audio is classified as a particular phoneme (smallest unit of sound). This step is carried out by the acoustic model, which learns the characteristics of each phoneme using the speech features extracted from the audios in the training set. The acoustic model is often built using a hybrid method consisting of Hidden Markov Models (HMM), Gaussian Mixture Model (GMM), and Neural Networks. Several data augmentation techniques are employed at this stage to generate variations in the pronunciation of phonemes, such as speed perturbation, volume perturbation, frequency perturbations, etc. Phonemes, as recognized by the acoustic model are grouped together to form words, which are then grouped to form sentences. This is achieved by the language model, which can be a simple N-gram language model, or a Neural Network based language model, trained on some text corpus to learn the grammar. The acoustic model and the language model are used in combination to build a decoding graph which is used for model inference. The second architecture, End-to-End speech recognition [2] has gained significant attention in recent years. The End-to-End neural networks are trained to learn directly from raw audio data, without the need for feature-engineering or complex modeling and show state of the art performance on a wide range of speech recognition tasks. However, End-to-End models have limited interpretability as they operate as black boxes. One of the objectives of the presented work is to gain comparative insights over multiple Indian language speech recognition models when trained under identical conditions. Therefore, we decided to experiment with the conventional architecture over End-to-End architecture.

In this paper, we present our experiments on Connected Number Recognition (CNR) – a domain-specific task in ASR – we have primarily used Hitachi Dataset–I consisting of connected number samples from 1000 speakers for each of the five languages (Hindi, Bengali, Marathi, Tamil, Kannada), with speakers contributing approximately 56 connected number samples each. ASR models for CNR using different speech feature combinations, namely – MFCC, PLP, MFCC+Pitch, and PLP+Pitch, have been built and tested using the Kaldi Speech Recognition Toolkit. The business use-case and market significance of CNR are detailed in Section II. Data collection process for our work is included in Section III. The discussion on different speech features for ASR is elaborated in Section IV. Further experimental details are mentioned in Section IV. Experimental evaluation results for Connected Number recognition and following discussion are Section VI and VII respectively. Finally, we have concluded the best features for CNR, as well as future directions, in Section VIII.

II. USE CASE

Connected numbers are at the heart of financial transactions of every kind, be it withdrawing cash, transferring money, conducting offline transaction with merchants, etc. In a country like India with a large rural population with limited literacy, a large Section of the society is excluded from many financial services, creating an ecosystem where people often have to depend on others to access the respective services. In order to truly democratize financial services, there is an immense need of voice-aided financial applications either over smartphones or feature phones, ideally combined with multi-modal user interface (possible with smartphone) and available in a variety of Indian languages. To realize such systems, ASR capability for several Indian languages, particularly in this domain, needs to be built. Our experiments with connected number recognition on five Indian languages aims to identify the best type of speech feature to build such a system. This investigation will be useful for any use case surrounding connected number recognition, for instance, a voice-aided number based electronic device, use-cases involving vehicle numbers, ticket booking etc.

III. DATA COLLECTION

Since our objective is to build highly accurate speech recognition systems for various Indian vernacular languages in the finance domain, we collected data from various parts of the country following a systematic and planned approach. Most vernacular Indian languages can be considered resource-poor languages, which is why data collection is especially necessary for this project.

Speech data was collected mainly in the form of connected numeral samples between zero to one million. Ten Indian languages were targeted and most major states in the country were covered along with seventy percent coverage being given to rural areas, i.e., the places where we believe our vernacular language speech recognition system will have the most amount of positive impact. Data from five out of these ten languages have been used for the presented work. The data collection was performed in natural setting hence contains native environmental noise and background sounds. Collection was done in a way that number of samples for connected numerals most likely to be used during financial transactions are maximized and have diverse representations in the dataset. Volunteers within the age group of eighteen to fifty willing to record audio samples were taken through a guided data collection process which resulted in approximately 56 samples per volunteer. Therefore, per language we have collected 56000 samples. The data was collected from thousand such volunteers for each of the ten languages. The volunteers' data privacy and security measures were taken during this entire exercise.

IV. SPEECH FEATURES

In ASR, every speech audio is processed to extract speech features, which are then used for training and testing purposes. The popular speech features are- Linear Prediction Cepstral Coefficients (LPCC), Mel-Frequency Cepstral Coefficients (MFCC), and Perceptual Linear Prediction (PLP).

The human speech generation system can be broadly represented as shown in Figure 1. The vocal folds generate periodic excitation input which passes through vocal tracts that convert it into speech. The MFCC/PLP features aim to model the vocal tracts and pitch features aim to learn about the excitation signal. We aim to present a comparative study of the performance achieved by four speech feature combinations- MFCC, PLP, MFCC+Pitch and PLP+Pitch when tested on five different Indian vernacular languages- Hindi, Bengali, Marathi, Tamil, and Kannada. The performance of various speech features has been tested in past studies, for instance a study published in 2019 concluded the superior performance of MFCC compared to PLP in case of Spanish language [5]. However, a study specifically focused on Indian languages is yet to be seen. In 2014, a study on pitch features introduced Kaldi Pitch tracker, a modified version of a previously existing pitch extractor, and claimed an improvement for both tonal and atonal languages, the former showing a larger reduction in WER [6].

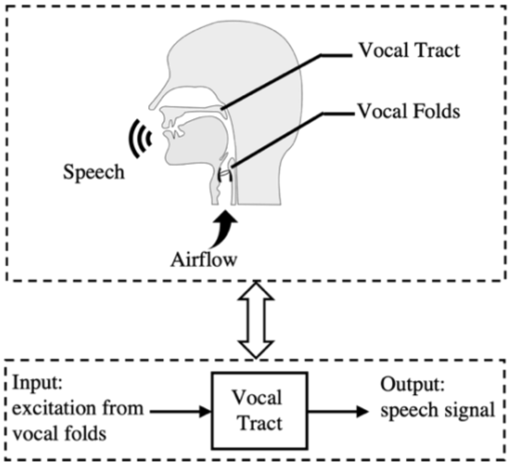

Figure. 1 Human Speech Generation System

We hope to provide further insights on the performance of pitch features when combined with both MFCC and PLP features for a small vocabulary domain-specific ASR task for Indian languages. A short overview of some common speech features and the pitch features is included in this Section.

A. Mel-frequency Cepstral Coefficients

Calculating MFCC of short audio segments in frequency domain is one of the most popular methods of extracting speech features for speech processing. It utilizes the concept of Mel-scale, a non-linear frequency scale which is based on human auditory perception. Mel-filter bank maps the actual frequency, f to Mel-scale frequency, f^* , i.e., the perceived frequency [7] as in (1). Audio segments of 25 ms with an overlap of 10 ms are windowed (Hamming window of length N with coefficients $W(n)$, (2)) and represented in frequency domain using FFT, and subsequently the Mel-filter bank is applied to the log of the amplitude spectrum to represent the frequency measurements on the Mel-scale.

$$f^* = 2595 \log\left(\frac{1+f}{700}\right) \quad (1)$$

$$W(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right), 0 \le n \le N-1 \quad (2)$$

Discrete Cosine Transform (DCT) of the output from the previous step is calculated to obtain MFCC coefficients [8]. Typically, along with the first 12 coefficients and the energy of the segment, the first order derivatives and the second order derivatives of these 13 features are also included to form a 39-feaures MFCC feature vector of a short audio segment. The first 12 MFCC features are phonetically significant features which are critical for analysis of speech signal.

B. Predictive Coding

Linear Predictive Coding (LPC) is a method commonly used for estimating speech parameters such as spectra and pitch formants [9]. It is used to faithfully encode speech for low bit-rate transmission. It is based on the principal that the value of a sample $\tilde{s}(n)$ can be estimated by a linear combination of all p previous samples as displayed in (3). The first method to calculate LPC coefficients is by minimizing the estimation error and solving a system of linear equations by autocorrelation method, covariance method, or lattice method. The LPC coefficients α_k define the formants of the signal, i.e., the frequencies at which there is an occurrence of resonant peaks, same as the peaks in the spectrum of the linear prediction filter resulting from the transfer function (5) [7][8][10].

The coefficients are calculated over the entire speech signal by using sliding time window with overlap of 10 ms and multiplying the frame with the Hamming window. The set of LPC coefficients of each frame constitute the feature vector for the respective audio segment.

$$\tilde{s}(n) = \sum_{k=1}^{p} a_k s(n-k) \quad (3)$$

$$e(n) = s(n) - \tilde{s}(n) \quad (4)$$

$$\frac{S(z)}{E(z)} = \frac{1}{1 - \sum_{k=1}^{p} \alpha_k z^{-k}} \quad (5)$$

C. Perceptual Linear Prediction

PLP is similar to LPC, but it takes into account the human auditory perception. It uses critical bands, intensity-to-loudness compression, and equal loudness pre-emphasis to remove irrelevant information and extract feature vectors. It utilizes the non-linear frequency scale called Bark scale to map frequency in Hertz, f, to frequency in Bark scale, f^b (6).

$$f^b = 7 \log\left(\frac{f}{650} + \sqrt{1 + \left(\frac{f}{650}\right)^2}\right) \quad (6)$$

The speech signal segment is windowed using the Hamming window and the power spectrum is calculated, post which the Bark filter bank is applied. The Bark filter bank incorporates the process of frequency warping to the Bark scale, smooths the spectrum using the simulated critical-band masking curve, and down-samples the smoothened spectrum to ~ 1 bark intervals. It essentially compresses the higher frequencies into a narrow band. The filter bank outputs are weighted using equal loudness pre-emphasis weights to reflect human sensitivity of hearing. Linear prediction is applied to this warped spectrum to obtain predictor coefficients. From these coefficients, the cepstral coefficients are calculated by performing an inverse Fourier transform over the log of linear prediction model spectrum [7][8][11].

D. Pitch Features

MFCC+Pitch features and PLP+Pitch are combinations of the regular MFCC/PLP coefficients with pitch features. There are various pitch feature extraction methods such as Yin [12], Getf0 (get fundamental frequency) [13], SAcC [14],

Wu [15], SWIPE [16], and YAAPT [17] to extract pitch features from speech signal, however all of them process voiced and unvoiced audio frames separately [18]. All pitch trackers aim to get an estimate of the fundamental frequency (F0) of a signal, which is a property of all periodic signals and is a good indicator of perceived pitch. Estimating F0 requires the classification frames as voiced or unvoiced. This estimation has 3 steps- pre-processing, generation of estimate candidates for the true period, and post-processing to select the best estimate [13].
Pre-processing is carried out to perform high-pass and low-pass filtering, and to remove the DC-offset, noise, vocal-tract filter influences, etc. Subsequently to generate period candidates, various methods such as auto-correlation, cross-correlation, and cestrum can be used, although the best approach is using normalized cross-correlation function (NCCF). It overcomes the issues of the other methods but with the caveat of higher computational complexity.

$$\emptyset_{i,\ k} = \frac{\sum_{j=m}^{m+n-1} s_j s_{j+k}}{\sqrt{e_m e_{m+k}}} \tag{7}$$

$$e_j = \sum_{l=j}^{j+n-1} s_l^2 \tag{8}$$

NCCF (7) of voiced samples tend towards 1.0 (maxima) for lags corresponding to integer multiples of ‘true period’ whereas NCCF of unvoiced samples has maximum values at zero lag. In post-processing, dynamic programming is used for voicing decision and selection of the ‘true period’, consequently determining F0 [13].

The Kaldi pitch tracker used in our experiments is based on the Getf0 pitch extractor. While the original Getf0 makes hard decisions on whether a frame is voiced or unvoiced, the Kaldi pitch extractor treats all frames as voiced and uses Viterbi search to interpolate over unvoiced frames naturally. It is based on finding lag values that maximize the Normalized Cross Correlation Function (NCCF). Instead of just the local maxima, the search is conducted over a fine grid. A 'ballast' term is added to the NCCF formula such that it tends to zero for quieter regions of the signal. NCCF in combination with the raw pitch feature is used to compute the three default output features of the Kaldi pitch tracker, namely the Probability Of Voicing (POV) feature, mean-subtracted-log-pitch, and delta-of-raw-pitch. c being the NCCF of an audio frame, $a = abs(c)$ and $l = -5.2 + 5.4\exp\big(7.5(a-1)\big) + 4.8a - 2\exp(-10a) + 4.2\exp\big(20(a-1)\big)$. POV, p, is given by (9). The POV feature, f, is described by (10).

$$p = \frac{1}{1 - \exp(-l)} \tag{9}$$

$$f = 2((1.0001 - c)^{015} - 1) \tag{10}$$

It gives a gaussian distribution to the feature. For the mean-subtracted-log-pitch, at time t, the average of pitch value over a window of width 151 frames, centered at t and weighted by POV, is subtracted from log pitch value to normalize it. The third default feature, delta-of-raw-pitch, is calculated from the unnormalized log pitch in the standard way using ±2 frames of context [6]. The three extra pitch features are added to the standard MFCC/PLP coefficients, and the first and second derivatives are calculated for the MFCC/PLP coefficients plus the three additional features as per the standard procedure, to form the speech feature vector for MFCC+Pitch and PLP+Pitch.

V. MODEL TRAINING

All experiments on the comparison of speech features for spoken CNR were performed using the Kaldi Speech Recognition Toolkit. ASR model training for all five languages was performed using 80% connected number dataset and 20% general dataset. The connected number data collection was outsourced to an external party by Hitachi India Pvt. Ltd. and resulted in the Hitachi Dataset-I. The general data is obtained from various opensource datasets such as OpenSLR [19][20], CommonVoice, and Shrutilipi [21]. For a given language, all training conditions except speech feature type was ensured to be identical. We used a distribution of 70% train, 20% dev, and 10% eval for all model training and testing. Additionally, mutual exclusivity with regards to speakers was maintained for train, dev, and eval datasets to avoid bias towards any speaker. The details of model training are explained in following Subsections.

A. Data Pre-processing

To train the acoustic model, the initial pre-processing of audio training data included volume pre-normalization and volume perturbation. An ASR model should be robust against volume/amplitude variations of audio signals, thus requiring a dataset with a variety of amplitudes. The range of volume levels selected for our experiments was 0.125 to 1.

Before the extraction of speech features, a dataset of speed-perturbed audios was created, such that the model could learn diverse representations of each phoneme. Speed perturbation simply involves resampling the signals to change the tempo and pitch. Typically, speed-perturbation-based data augmentation improves the ASR performance [22]. We selected 0.9, 1, and 1.1 as the three speeds to create the speed-perturbed dataset, thus increasing the effective data size for feature extraction to three times its previous size.

To prepare the language dictionary with phoneme representations, a transliteration tool based on ILSL 2.0 was used. Indian languages, in general, have one-to-one grapheme to phoneme mapping, unlike the English language. ILSL 2.0 consists of unified transliteration standards specifically designed for Indian languages, which define common phoneme representations for corresponding graphemes in respective Indian languages. This is not necessarily an IPA-like representation of phonemes but a way to represent the most similar sounding phonemes in multiple Indian languages by a common representation. This step is critical to perform the comparison of ASR models in multiple languages.

B. Speech Features Extraction

For our experiments, we tested two primary speech features: MFCC and PLP. Furthermore, we combined both MFCC and PLP with pitch features to test for MFCC+Pitch and PLP+Pitch, respectively. The speech feature extraction scripts are a part of Kaldi. For CNR experiments, a 39-dimensional feature vector was generated for each frame of audio, which had a duration of 25ms and was shifted by 10ms. The first 13 dimensions of the feature vector consisted of 12 speech features along with the energy of the spectrum, while the remaining 26 dimensions were constituted by the first and second order derivatives of the same. This architecture was maintained for both MFCC and PLP feature vectors.

Pitch feature extraction and combining them with MFCC and PLP features were done using the scripts for Kaldi pitch-tracker, which is based on the concepts introduced in "A pitch extraction algorithm tuned for automatic speech recognition" [6]. However, to maintain the 39-dimensional structure of the feature vectors, we considered 11 speech features and the energy of the spectrum, along with their first and second order derivatives for the first 36 dimensions of the feature vector. The last 3 dimensions were dedicated to the three pitch features extracted using the Kaldi pitch tracker. After extracting the features, Cepstral Mean and Variance Normalization (CMVN) was applied to reduce differences in feature representation of different speakers and enhance the noise robustness of speech recognition [23].

C. Phoneme Alignment Training

As a part of conventional speech recognition model training, the next step involved alignment of phonemes. We implemented Montreal Force Alignment (MFA) [24] for phoneme alignment. MFA trains HMM-GMM model in consecutive steps, i) Monophone training, ii) Triphone training (tri-1), iii) Triphone + LDA + MLLT training (tri-2), iv) Triphone + LDA + MLLT + SAT training (tri-3). The final alignments generated from tri-3 are passed to the TDNN-LSTM network to train the acoustic model. The hyperparameters tuned during the training process are listed Table 1. The ‘numleaves’ and ‘totgauss’ decide the number of triphones which can be modelled and their fine-grained nature. The hyper parameter values were obtained as per Kaldi community guidelines. Further, we performed Bayesian optimization to tune hyper parameters, but it showed insignificant improvements. We fixed identical values of these hyper parameters for all experiments to compare the model performances based on speech feature type and language.

D. TDNN-LSTM Model training

Once phoneme alignments are learnt, next step is to learn the temporal sequences of speech signals. Recurrent Neural Network (RNN) is a popular approach to learn sequential information; however, it suffers from the vanishing gradient problem. Hence, Long Short-Term Memory (LSTM) neural networks were invented. Furthermore, Time Delay Neural Networks (TDNN) can learn the localized temporal patterns better than traditional Deep Neural Networks (DNN). We integrated TDNN with LTSM, instead of integrating DNN, thus making it a TDNN-LSTM network. Moreover, our objective is to compare the speech recognition performance for multiple speech features over multiple languages. Therefore, we did not necessarily investigate the network with the best performance and the best tuning parameters. Rather, we opted for a standard network architecture and trained multiple models to derive comparative insights. The hyperparameters used in the training are listed in Table 1. These hyperparameters were set as per IIT Madras Speech Lab ASR Challenge demo Kaldi recipe [25] and further modified as per Kaldi community guidelines. Again, these hyperparameters were set to be identical for all the experiments to compare the model performances.

E. Language Model training

The connected number language models were built for all the languages separately using the SRILM toolkit. The corpus for training the language model consisted of the text representation of connected numbers from 1 to 100,000 in the respective language. 3-gram language models showed the best perplexity scores and were thus selected for decoding.

The model inference of test data is achieved by combining the results of the acoustic model and the language model.

VI. Experimental Evaluation

The investigation of speech features with and without pitch for five Indian language for CNR has been presented with the standard metrics, i.e., Word Error Rate (WER) and Sentence Error Rate (SER). The connected number samples in the eval set are used to display the results of our investigation with the mentioned metrics in Figure 2 and Figure 3. The training, dev, and eval set are identical for all experiments pertaining to a specific language. The eval set connected number audio samples have been collected in a natural environment, therefore contain various types of background noises native to the environment.

On average, MFCC+Pitch yields the least WER and SER among all the speech features. Inclusion of pitch features with MFCC features reduces WER for all languages except Bengali, where it shows a slight increase by 0.58%. MFCC+Pitch shows 0.68% average reduction of WER and 1.27% average reduction of SER when compared with MFCC. The highest improvement was observed in Hindi language in this case. However, adding pitch features with PLP shows a slight increase in WER (~0.8%) across three languages.

TABLE I. RANGE OF HYPERPARAMETER AND TRAINING CONDITION USED AT DIFFERENT STAGES OF TRAINING

Training Stage	Hyperparameters	Range
Monophone	iterations	40
Triphone (tri-1)	iterations	35
	numleaves	2750
	totgauss	50000
Triphone + LDA + MLLT (tri-2)	iterations	35
	numleaves	2750
	totgauss	50000
Triphone + LDA + MLLT + SAT (tri-3)	iterations	35
	numleaves	2750
	totgauss	50000
TDNN-LSTM	Epochs	6
	Hidden layers	13
	Dimension of layers	1024
	Non- linearity	ReLU
	Initial learning rate	0.0001
	Final learning rate	0.00001

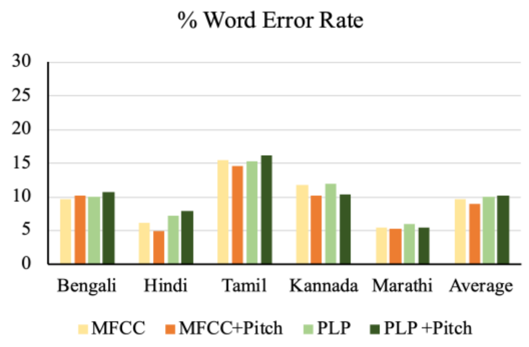

Figure. 2 Comparison of % WER for CNR models built for 5 Indian languages with 4 feature combinations

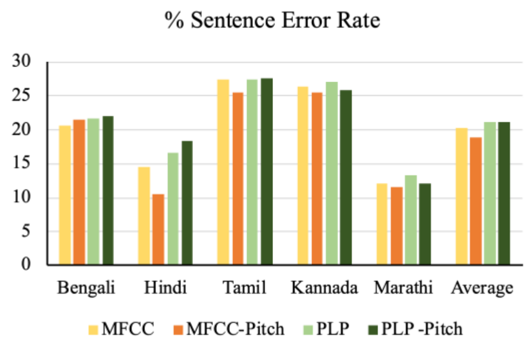

Figure. 3 Comparison of % SER for CNR models built for 5 Indian languages with 4 feature combinations.

The PLP and PLP+Pitch based models show comparable results overall. MFCC features with and without pitch show better performance than PLP features with and without pitch features.

VII. DISCUSSION

India is a diverse country with multiple languages spoken in different regions. The presented experimental work helps in building a single acoustic model for multiple languages. The aim is to gather insights regarding the tuning of a multilingual ASR model to meet different recognition criteria depending on the part of the country where the model is to be deployed. Therefore, the model should ultimately show higher accuracy for the region-specific language, while also supporting multiple other languages.

For example, if the model needs to be deployed in the West Bengal region, it should meet the high accuracy criteria for Bengali and Hindi languages and also support other languages like Marathi or Kannada. The experimental results show that for same amounts of training and testing data, and identical training conditions, the Hindi model shows better performance as compared to the Bengali model. Therefore, while training the multilingual ASR model, one can think of including more data for Bengali than for Hindi to get decent performance over both languages. Also, using MFCC features can be helpful since it shows the best performance in Bengali. The presented work provides such heuristics for multilingual ASR developments, without which we lose out on language-specific nuances of training conditions. This leads to higher resource requirements in terms of data, infrastructure, and time to get the desired performance.

This study also provides language-specific learnings. The five Indian languages in this experimentation are broadly classified into two language families; The Indo-Aryan language family to which Hindi, Bengali, and Marathi belong, and the Dravidian language family to which Tamil and Kannada belong. Amongst these languages, Marathi and Hindi are phonetically similar, moreover, they use the same 'Devanagari' script. As mentioned in Section V.A., we use an ILSL 2.0 transliteration tool for grapheme-to-phoneme representation. For both Hindi and Marathi, the grapheme-to-phoneme map is one-to-one, hence both languages show similar and relatively better performance as per Figure 2 and Figure 3. On the other hand, Bengali language has some elements in the grapheme-to-phoneme map which exhibit many-to-one mapping, leading to relatively poorer recognition performance. Tamil and Kannada languages belong to the same language family, and show similar and relatively poorer performance. Therefore, one needs to tune their training conditions differently to get better performance from the models.

VIII. CONCLUSION

In this paper, we have presented the experimental results for building Connected Number Speech Recognition (CNR) models in multiple Indian languages. The experiments were conducted for five Indian languages Bengali, Hindi, Tamil, Kannada, and Marathi. Different speech recognition models were trained for four feature combinations: MFCC, MFCC+Pitch, PLP, and PLP+Pitch features. The

MFCC+Pitch features offers the best result on average; however, results may vary from one language to another. Such comparative analysis can help select the best feature combination for any given training conditions and dataset. The presented work provides language specific insights and heuristics for building multilingual ASR models.

In future, we hope to build a single speech recognition model for multiple languages for Connected Number recognition (CNR), which can be more adaptive for language switching and have a smaller memory footprint.

REFERENCES

[1] M.A.Anusuya and S.K. Katti, “Speech Recognition by Machine: A Review”, International Journal of Computer Science and Information Security, vol. 6, no. 3, 2009

[2] Wang Dong, Xiaodong Wang, and Shaohe Lv,"An Overview of End-to-End Automatic Speech Recognition", Symmetry, vol.11, no.8, 1018, 2019

[3] H. Hirsh, P. Meyer, and H.W. Ruehl, “Improved speech recognition using high-pass filtering of sub-band envelopes”, Proc. EUROSPEECH 91, pp. s413-416, Genova, Italy, 1991

[4] A. Acero, “Acoustical and environmental robustness in automatic speech recognition”, PhD thesis, CMU, 1990.

[5] J. M. R. Sánchez, M. Bereau, and J. R. C. de Lara, “Feature selection for automatic speech recognition in noisy Scenarios,” In 2nd International Conference of Information Processing , vol. 40, pp. 51-71, Dec. 2019.

[6] P. Ghahremani et al., "A pitch extraction algorithm tuned for automatic speech recognition," 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2494-2498, 2014

[7] D. Namrata, “Feature extraction methods LPC, PLP and MFCC in speech recognition”, International Journal For Advance Research in Engineering And Technology (ISSN 2320-6802), vol. 1, pp. 1-6 , 2013

[8] S. A. Alim and N. K. A. Rashid, "Some commonly used speech feature extraction algorithms", in From Natural to Artificial Intelligence - Algorithms and Applications. London, United Kingdom: IntechOpen, 2018 [Online]. Available: https://www.intechopen.com/chapters/63970 doi: 10.5772/intechopen.80419

[9] O. Buza, G. Toderean, A. Nica, and A. Caruntu, “Voice signal processing for speech synthesis,” IEEE InternationalConference on Automation, Quality and Testing Robotics, vol. 2, pp. 360-364, 25-28, May 2006.

[10] H. Gupta and D. Gupta, “LPC and LPCC method of feature extraction in speech recognition system.” 2016 6th International Conference - Cloud System and Big Data Engineering (Confluence), pp. 498-502, 2016.

[11] L. Xie and Z. -q. Liu, "A comparative study of audio features for audio-to-visual conversion in Mpeg-4 compliant facial animation," 2006 International Conference on Machine Learning and Cybernetics, pp. 4359-4364, 2006, doi: 10.1109/ICMLC.2006.259085.

[12] A. de Cheveigné and H. Kawahara, Eds., “YIN, a fundamental frequency estimator for speech and music,” The Journal of the Acoustical Society of America, vol. 111 4, pp. 1917–30, Apr. 2002.

[13] D. Talkin and W. Bastiaan Kleijn. "A robust algorithm for pitch tracking (RAPT)." Speech coding and synthesis, pp. 497-518, 1995

[14] B. S. Lee, "Noise Robust Pitch Tracking by Subband Autocorrelation Classification.", Ph.D. dissertation, Columbia University, Ann Arbor, 2012.

[15] M. Wu, D. Wang, and G. J. Brown, "A multi-pitch tracking algorithm for noisy speech," 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. I-369-I-372, 2002, doi: 10.1109/ICASSP.2002.5743731.

[16] A. Camacho and J. G. Harris, “A sawtooth waveform inspired pitch estimator for speech and music,” The Journal of the Acoustical Society of America, vol. 124, no. 3, pp. 1638–1652, Sep. 2008, doi: 10.1121/1.2951592.

[17] K. Kasi and S. A. Zahorian, "Yet another algorithm for pitch tracking," 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. I-361-I-364, 2002, doi: 10.1109/ICASSP.2002.5743729.

[18] X. Lei. “Modeling lexical tones for mandarin large vocabulary continuous speech recognition.”, Ph.D. dissertation, University of Washington, 2006.

[19] O. Kjartansson, S. Sarin, S. Pipatsrisawat, M. Jansche, and L. Ha, “Crowd-sourced speech corpora for Javanese, Sundanese, Sinhala, Nepali, and Bangladeshi Bengali”, Proc. 6th Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU 2018), 2018, pp. 52-55, doi: 10.21437/SLTU.2018-11.

[20] F. He et al., ‘Open-source multi-speaker speech corpora for building Gujarati, Kannada, Malayalam, Marathi, Tamil and Telugu Speech synthesis systems’, Proc. The 12th Language Resources and Evaluation Conference (LREC), pp. 6494–6503, 2020.

[21] K. S. Bhogale et al., ‘Effectiveness of mining audio and text pairs from public data for improving ASR systems for low-resource languages’. arXiv, 2022.

[22] G. Chen, et al., "Data augmentation for children's speech recognition”, The "Ethiopian" System For The SLT 2021 Children Speech Recognition Challenge, arXiv preprint arXiv:2011.04547, 2020.

[23] N. V. Prasad and S. Umesh, "Improved cepstral mean and variance normalization using Bayesian framework", 2013 IEEE Workshop on ASRU, pp. 156–161, 2013

[24] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, ‘Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi’, Proc. Interspeech 2017, pp. 498–502, 2017.

[25] “IITM Hindi Speech Corpus: a corpus of native Hindi Speech Corpus” – Speech signal processing lab, IIT Madras, 2020.