Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 171 (2020) 2476–2485

Procedia Computer Science

www.elsevier.com/locate/procedia

Third International Conference on Computing and Network Communications (CoCoNet'19)

Kaldi recipe in Hindi for word level recognition and phoneme level transcription

Karra Venkata Lakshmi Sri, Mayuka Srinivasan, Radhika Rajeev Nair, K. Jeeva Priya, Deepa Gupta*

Department of Electronics and Communication Engineering, Amrita School of Engineering, Bengaluru, Amrita Vishwa Vidyapeetham, India

Abstract

This paper discusses an automatic speech recognition (ASR) system in Hindi. The language models and acoustic models are built using the open source toolkit Kaldi. A significant portion of the corpus built for this work pertains to the medical domain, as our primary emphasis lies in the application of speech processing for medical transcription. The various acoustic models used for the comparison of word error rates (WER) in Kaldi include HMM-GMM (Hidden Markov Model-Gaussian Mixture Model) based Monophone, Triphone (tri1,tri2, tri3) and SGMM(Sub Space Gaussian Mixture Model). Comparing the WER for various acoustic models used, it was observed that tri3 model has the least WER over the other acoustic models. Also, the possible mappings of phonemes detected have been shown

Keywords: ASR; Kaldi; MFCC; Medical Transcription; triphone; WER; SGMM

* Mayuka Srinivasan. Tel.: +91 8096155339
E-mail address: mayuka.118@gmail.com

10.1016/j.procs.2020.04.268

1. Introduction

Over the past few years, researchers and engineers have been continuously working towards building an autonomous, intelligent artificial system that is capable of interacting with us in a human-like way. This has led to the development of many automatic speech recognition (ASR) systems. Automatic Speech Recognition is the technology that allows us to communicate with computers using human voice. It takes audios captured using a microphone, creates a wave file of the words spoken and breaks down the filtered wave files into phonemes. A highly efficient ASR system is trained on several hours of data for the most accurate results. It is used in various domains like healthcare, telecommunication, home automation, defence and aviation.

Within India, widespread research is ongoing for applications incorporating regional languages. Hindi, the official language of India, is the most commonly spoken language across the country by about 44% of the population [1]. With majority of the population speaking the language, the diversity in the vocabulary, slang and pronunciations are vast. The intricacies of the script written in Devanagari render the usage and pronunciations more complex a task for automatic speech recognition.

Presently there is humungous research being done on various languages, across the globe, using various toolkits like CMU Sphinx, HTK and Kaldi [2]. Over the past few years, Kaldi, an open source speech recognition tool kit is popular among researchers and quite a few works have been reported using this.

In this work, Hindi ASR is built using the open source Kaldi toolkit developed by Daniel Povey et al [3]. Kaldi, a toolkit developed in 2011, contains most of the algorithms required to build an ASR system. The recipes incorporated by Kaldi to train acoustic models (AMs) can be used on any speech corpus.. A huge advantage of this open-source toolkit is its supportive community [4] where researchers from various platforms share can communicate with the developers of Kaldi.

Kaldi is a collection of various tools that are used collectively to build a speech recognition system. It is programmed in C++, but also uses other script languages like Bash, Awk, Perl and Python. It also has vast linear algebra support that includes a matrix library. This library includes standard BLAS (Basic Linear Algebra Subprograms) and LAPACK (Linear Algebra Package) routines. Open FST (Finite State Transducer) is also used as a library in order attain better system accuracy. [5].

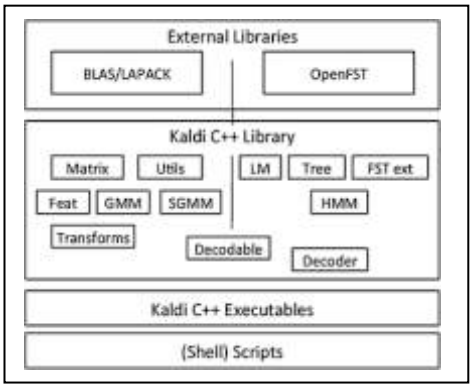


Fig. 1. Composition of KALDI [9]

This paper discusses the procedures involved in building the language and acoustic models to train and test an ASR system in Hindi, using the Kaldi toolkit and is organized as follows: Section 2 discusses similar works in speech recognition that involves the methods to build an ASR system in different languages using different tools. Section 3 discusses the different stages involved in creating the language models in detail. Section 4 provides

information on the training models used to train and test the data. In section 5, the performance of the system based on the computed WERs is discussed and also compare the accuracy of different acoustic models implemented.

2. Related Works

For the past few years extensive research has been performed on building speech recognition systems in various languages, including Indian regional languages. Some of these works are mentioned in this section. The steps involved in training and testing Arabic speech recognition system using Kaldi toolkit [6] is discussed by A. Ali et al. The system was trained using GALE data. Georgescu et al. discuss the improvement in the performance of a Romanian speech recognition system [7] developed by the Speech and Dialogue after increasing the size of the corpus and replacing the classic GMM-HMM approach with DNN based acoustic models.

An ASR system in Italian using the Kaldi toolkit [8] by Piero Cosi et al discusses using the DNN model, the WERs of children speech samples are recorded. A Malayalam ASR using KALDI [9] by Lavanya et al discusses the MFCC extraction procedure and performance using bi-gram LM(language model) of various models have been compared.

A Kannada ASR system with a dictionary size of 200 words for tourism application has been discussed in [10] by Jeeva Priya K et al. The system, built using HMM triphone acoustic modeling, using HTK toolkit, resulted with an accuracy of about 90.6% and 83.2% when used for recognition in offline and online mode respectively.

The different acoustic models for an ASR system in Kannada using the KALDI toolkit for two different data sets, one for digits another for different phrases is described in [11] by Jeeva Priya K et al. Similarly, the recognition of digits in Kannada language using the KALDI toolkit which is compared with the results obtained in a HTK based system[16] is discussed in [12] by Sundar Karthikeyan et al.

A large vocabulary for Tamil, consisting of 13,016 words and 4.5 hours of training data is discussed in [13] by A. Madhavaraj et al. Out of the two systems built, the system for phone recognition (PR) resulted with a 24.9% error rate for phones, the system for the continuous speech recognition (CSR) resulted with a word error rate of 3.5% .

In this work, the comparison of various acoustic models for determining the word error rates in Hindi is discussed, and the model with most accuracy of recognition is determined. The word level mappings of the various models and phoneme level representations are depicted in section 5.

3. Hindi ASR system using Kaldi toolkit

Building an ASR system using the Kaldi toolkit involves several pre-processing, data preparation and language modeling stages, along with creating various supporting files. Fig. 2 depicts the sequence of steps required to build an ASR system.

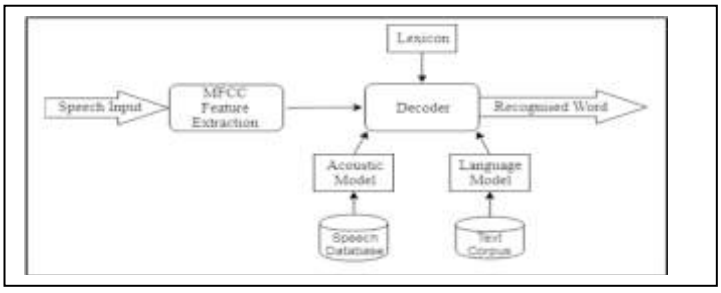


Fig2. Basic sequence of an ASR system

3.1. Data Preparation

A dataset of 158 Hindi medical terms was collected from 27 speakers (16 female and 11 male) contributing to about 3 hours of data, at 16KHz, mono channel. About 30% of this was used as the test set and the rest was used for training the system. The data was collected from a broad range of speakers, aged 18-50 years to incorporate the variations and intonations of their voices for a better recognition system.
Audacity software was used for reducing the background noise in the audio samples, segmenting them into words and exporting as individual .wav files. It is an open-source digital audio editor and audio recording application. The noise reduction function was implemented to improve the efficiency of the ASR system as the raw speech recordings comprised of background noise.

These segmented .wav files are divided into the test and train folders inside KALDI directory. Transcription files described in Table 1 were created in both these folders.

TABLE 1. FEW INTEGRAL FILES FOR KALDI ASR

Script file name	Description of File
spk2utt	consists of the mapping of speaker IDs to the various utterance ID (the set of audios for each speaker)
wav	provide the path to the recorded audio files provided as input to the toolkit
text	contains every utterance, recorded by the speakers, mapped to the text transcription
corpus	contains all the utterances that make up the database

3.2. Extraction of Mel-frequency Cepstral Coefficients (MFCC)

The next step after data preparation is feature extraction. The features to be extracted represent the phones within the words while other degrading factors in the signal such as the channel characteristics and the presence of noise in the background are suppressed [14]. The commonly used feature extraction methods are MFCC and Perceptual Linear Prediction Coefficients (PLP). In our work we have used the MFCC feature extraction method, as depicted in Fig. 3.

The feature is extracted by applying a 25 ms Hamming window, with a 10ms overlap .The speech signals are sampled at 16 KHz .Thus, in one window about 400 samples are reduced to 13 cepstral coefficients. Further, additional delta and delta-delta coefficients can be added to the feature.

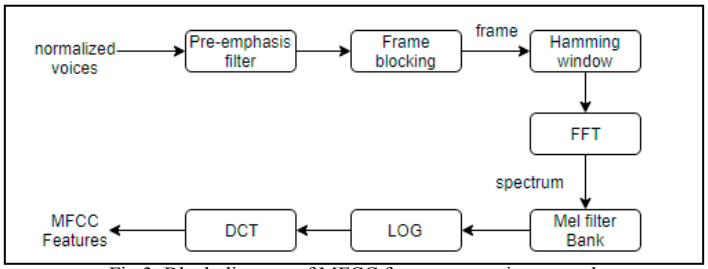
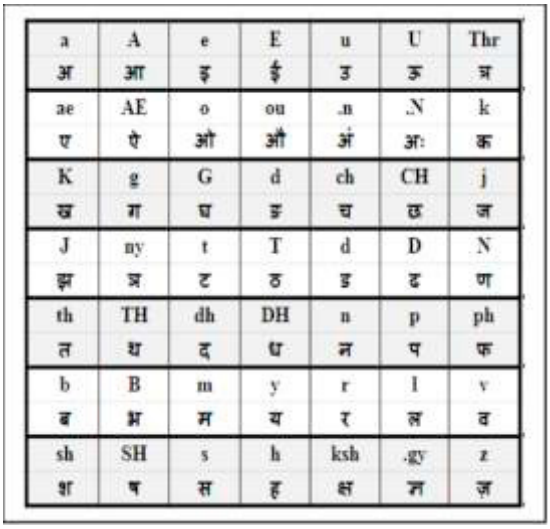


Fig 3. Block diagram of MFCC feature extraction procedure

3.3. Language Modelling (LM)

For the conversion of words to phoneme sequences, a dictionary has been created using the English script as symbol representations. The described ASR system is based on the 49 phonemes as depicted in Fig. 4, which represent the Devanagari script. Each phoneme in Hindi is represented by a unique combination of symbols and letters in the English script. Aside from the 49 phonemes, an additional silence phoneme, 'sil' is added to represent the silence between hyphenated words or the start or end of a sentence.



a	A	e	E	u	U	Thr
अ	आ	इ	ई	उ	ऊ	ऋ
ae	AE	o	ou	.n	.N	k
ए	ऐ	ओ	औ	अं	अः	क
K	g	G	d	ch	CH	j
ख	ग	घ	ङ	च	छ	ज
J	ny	t	T	d	D	N
झ	ञ	ट	ठ	ड	ढ	ण
th	TH	dh	DH	n	p	ph
त	थ	द	ध	न	प	फ
b	B	m	y	r	l	v
ब	भ	म	य	र	ल	व
sh	SH	s	h	ksh	.gy	z
श	ष	स	ह	क्ष	त्र	ज्ञ

Fig 4. Table representing Devanagari script and equivalent symbol used in the ASR

Using this dictionary, a lexicon file is created. This consists of the breakdown of phoneme sequences within each word and is depicted in Fig. 4. Multiple pronunciations for the same words are also incorporated using pronunciation probabilities [15], as indicated in Table 2.

TABLE 2. SNIPPET OF LEXICON FILE

Word	Phoneme Sequence
aankhonkadaktar	sil A n K o .n k a d a k t a r sil
aankhonkadaktar	sil A n K o .n k a d o k t a r sil
aankh	sil A n K sil
AIDS	sil ae d z sil
ausadhi	sil ou s a DH e sil
ausadhi	sil ou SH a DH e sil
beemar	sil b E m A r sil

3.4. File naming and delivery

For a given sequence, the probabilities of word(s) succeeding or preceding a particular word is found with the help of Language models (LMs), which in turn reduces the search in decoders. Using the tools in Kaldi , the ARPA (Advanced Research Project Agency) formatted LMs are converted into FSTs (Finite State Transducers) [3],[17].

For language modeling, the SRILM (Speech Technology and Research Laboratory) toolkit is used. Based on the number of phonemes considered at once, there is a wide range of language modeling types.Let us assume a sequence of length k. Let $P(w_1, w_2, w_3, \ldots, w_k)$ be the probabilities assigned by LM to the entire sequence. The probability of any word sequence $P(w_1, w_2, w_3, \ldots, w_k)$ [18] for a n-gram model is then given as:

$$P(w_1,w_2,\dots,w_k) = \prod_{i=1}^{k} P(w_i|w_1,\dots,w_{i-1}) \quad (1)$$

4. Training and decoding in Kaldi

Various acoustic models are used to validate the ASR system built, by computing the Word Error Rates based off our library. As mentioned earlier, our dictionary consists of 158 words and 49 phonemes. It is required to have time markings of the phoneme(s (alignment). Thus, during alignment, each audio file is divided into equal alignments by the system as shown in Fig. 5, where each division is mapped to the respective phoneme symbol in the sequence. Each model further refines the alignments, using different training techniques and passes them onto the next stage, which is then used for recognition.

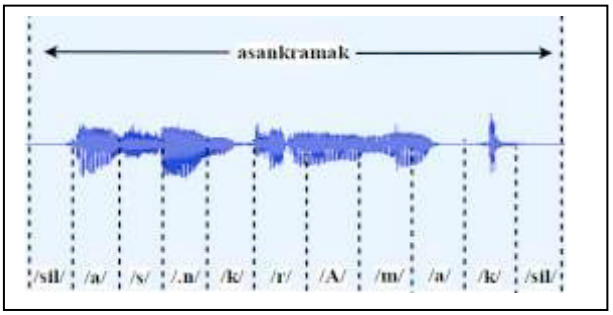


Fig. 5. Breakdown of phonemes in 'asankramak'

4.1. File naming and delivery

As opposed to the monophone model that compares each phoneme individually and assigns weights based on the probability of match, in triphone models the acoustic parameters are represented for a block of 3 consecutive phonemes. There are total of (49 x 3) possible HMM states that can be assigned, one for each of the triphones. However, the training dataset may not contain all the triphones. The triphones are grouped together into a smaller set of distinct units by creation of a decision tree.
The triphone model comprises of three training models which are delta+delta-delta (tri1), LDA+MLLT (tri2) and LDA+MLLT+SAT (tri3) models.

- **Delta+delta-delta training**: As a supplement to the MFCC features, this training method computes delta and double-delta features, also called 'dynamic coefficients'. They are the first and second order derivatives of the signal (features). These features are then used for recognition.
- **Linear Discriminant Analysis-Maximum Likelihood Linear Transform (LDA-MLLT):** LDA reduces the feature space and creates HMM states for the feature vectors [19]. This reduced feature space is used to build a transform specific to each speaker. MLLT thus incorporates speaker independency [20].

Speaker Adaptive Training (LDA+MLLT+SAT): Noise and speaker normalization is performed by using data transform for each speaker. This allows the model to use its parameters to compute the variance due to the phoneme, as opposed to the background environment. This result hence provides the most reduced word error rate as seen in Table 3.

4.2. SGMM (Subspace Gaussian Mixture Model)

As opposed to the conventional HMM-GMM models discussed above, in SGMM all HMM states, or the leaves on the decision tree share a common structure i.e. have the same number of Gaussians. The mixture weights and means in each HMM state are allowed to vary keeping equal number of Gaussians in each state. This model was

found to give better results when compared to conventional models [21], which is depicted in Table 3.

4.3. Alignment Algorithms

fMLLR (Feature Space Maximum Likelihood Linear Regression) and speaker independent alignments are briefed under this section. After attaining the speaker-normalized features for SAT training, the AM is no longer trained with the new normalized featured. In the alignment process, these partially speaker independent models are used [22].

4.4. Decoding

Kaldi's decoding algorithm uses Weighted Finite State Transducers (WFSTs). The WFSTs provide graph operations, used in acoustic modeling. These decoding graphs are assigned numerical values corresponding to context-dependent states, called pdf-ids. As different phones can have the same pdf-ids, "transition-ids" are used which encode the pdf-ids of phone member and use arc (transition) within the topology specified for that phone. Thus, decoding is performed on these decoding graphs (HCLG) which is constructed from simple FST graphs [5] [23].

$$HCLG = H \circ C \circ L \circ G. \quad (2)$$

The symbol ◦ represents an associative binary operation of composition on FST. HLGC is based on H- for HMM definitions, L for Language Models, G is the acceptor which is used to encode the grammar and C, the context dependency [24].

5. Experimental Results

An ASR system was built in Hindi and its performance was evaluated on the hour long test set. A single order (unigram) LM has been used for the speech set. For building the LM, the SRILM toolkit is used for computation of word probabilities [25]. The WERs for the AMs discussed above are mentioned in Table 3. It can be seen that as the complexity of the AM increases, the error rate gradually decreases. The best results are obtained with the TRI3 model (LDA+MLLT+SAT).

The performance of the system is evaluated using Word error rates determined by equation (2). This provides the accuracy of mapping of words from the test data set to the train data set.

$$WER(\%) = \frac{\text{Deletions+Substitutions+Insertions}}{\text{Total number of words}} \times 100 \quad (3)$$

As depicted in Table 3, the WER reduces as the complexity of the AM increases from monophone to triphone to SGMM. Monophone model takes a singular phoneme into consideration for recognition, whereas the triphone model takes 3 phonemes and creates a decision tree based on probabilities of preceding and succeeding phones.

About 30% of the data is taken as test data and the remaining as the training data. As the complexity of the models increase, the error rates significantly reduce. The accuracy of recognition of the TRI3 model over the other triphone models is due to the normalization of features which makes the audios independent of speakers and noise, which is provided by the SAT algorithm.

In the TRI 3 model, a first set of results are obtaining from the first-pass fMLLR. This resulted in a higher error rate of 11.87%. Then, the first-pass fMLLR transforms are obtained and a main lattice generation phase is done by estimating the fMLLR transforms a second time. On doing a final pass of acoustic rescoring, a second set of

WER results generated under TRI3. In Table 4 we have mentioned the TRI3 model WER after running the adaptive training algorithm. This resulted in a decreased word error rate to 4.91%. This decrease is due to the second adaptation of the fMLLR algorithm which takes the pre-computed lattices and reruns the algorithm on it.
The SGMM WERs are better than conventional HMM-GMM monophone training model as it incorporates more complex phoneme mapping and extraction methods than singular monophone model which compares on a single phoneme basis.

TABLE 3. COMPARISON OF WERS FOR DIFFERENT ACOUSTIC MODELS

Acoustic model type	***WER (%)***
MONOPHONE TRAINING	10.13
TRI1 TRAINING	8.23
TRI2 TRAINING	6.33
TRI3 TRAINING	1.9
SGMM	3.16

Using the command 'show-lattice', defined within Kaldi tools [5], we have attained the mapping for various words. Fig. 6--8 represents the mapping for the work "ek" for various model and the accuracies of mapping for various training models.

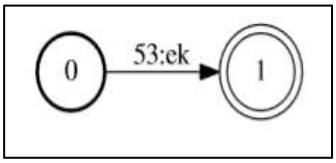


Fig. 6. Mapping for monophone – "ek"

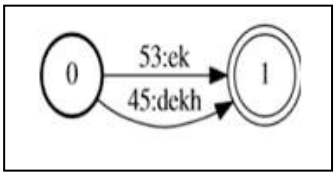


Fig. 7. Mapping for tri1 model- "ek"

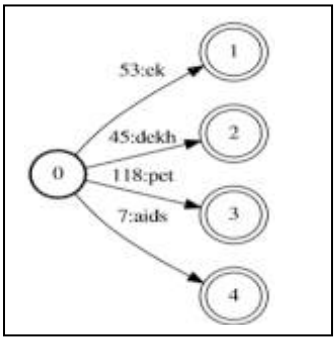


Fig. 8. Mapping for tri3 – "ek"

Beyond this, we have further attained the phoneme sequence order for the words in our data set, where each word can be resolved into the phonemes that make up the entire word , Using the command 'lattice-to-phone lattice'. A snippet of this output can be seen in fig. 9. Here, words like "Aankh" have been resolved into their phones (symbols from our dictionary) with 100% accuracy. However, there are also words like "ek" that have been resolved into the wrong phones, whose mapping is seen in Fig. 6 and Fig. 7, due to variance in the audio segments and unavoidable background noise.

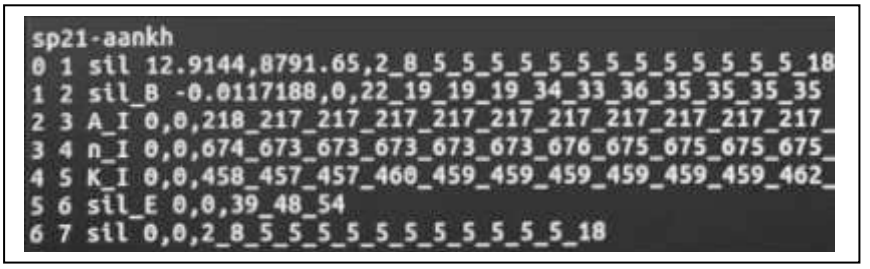


Fig. 9. Phoneme sequence obtained for the word "aankh"

6. Conclusion

In this work, an ASR system in Hindi is developed using Kaldi toolkit using the MFCC features . The acoustic models under consideration for WER determination are monophone, triphone (delta+delta-delta, LDA-MLLT, LDA-MLLT+SAT) and SGMM. As seen in Table 3, the WERs decrease with increasing complexity of the models. The best WER is obtained for the TRI3 model that uses SAT's normalization algorithm. Further, expansion of vocabulary, with more hours of training data would achieve a better recognition rate. This however, can be broadened to contain all the words in the language, to create a high level functioning ASR system.

As mentioned earlier, adaptation techniques can improve the WER (tri3 model). Using DNN (Deep Neural Networks) model, the need of providing the adaptation data in advance can be avoided and introduce the idea of unsupervised training or adaptation. Hence we can further improve our system by using DNN to model the acoustic units instead of using GMMs. Beyond attaining word level recognition, the end system can also contain translation files that are mappings from one language to another.

References

[1] Times of India, "Hindi mother tongue of 44% in India, Bangla second most spoken." [Online]. Available: https://timesofindia.indiatimes.com/india/hindi-mother-tongue-of-44-in-india-bangla-second-most-spoken/articleshow/64755458.cms.

[2] L. R. Rabiner, "Applications of speech recognition in the area of telecommunications," *IEEE Work. Autom. Speech Recognit. Underst. Proceedings, St. Barbar. CA, USA, 1997, pp. 501-510.*, 1997.

[3] D. Povey *et al.*, "The Kaldi Speech Recognition Toolkit," *IEEE 2011 Work. Autom. Speech Recognit. Underst.*, 2011.

[4] Google Groups, "Kaldi-help Google Forum." [Online]. Available: https://groups.google.com/forum/#!forum/kaldi-help.

[5] httpp://kaldi-asr.org, "Kaldi ASR Org." [Online]. Available: https://kaldi-asr.org/doc.

[6] A. Ali, Y. Zhang, P. Cardinal, N. Dahak, S. Vogel, and J. Glass, "A complete KALDI recipe for building Arabic speech recognition systems," *2014 IEEE Spok. Lang. Technol. Work.*, pp. 525–529, 2014.

[7] H. C. and C. B. A. Georgescu, "SpeeD's DNN approach to Romanian speech recognition," *2017 Int. Conf. Speech Technol. Human-Computer Dialogue (SpeD), Bucharest, 2017, pp. 1-8.*, 2017.

[8] P. Cosi, "A KALDI-DNN-based ASR system for Italian," *2015 Int. Jt. Conf. Neural Networks (IJCNN), Kill. 2015, pp. 1-5.*, 2015.

[9] K. R. S. and L. M. L. B. Babu, A. George, "Continuous Speech Recognition System for Malayalam Language Using Kaldi," *2018 Int. Conf. Emerg. Trends Innov. Eng. Technol. Res. (ICETIETR), Ernakulam, 2018, pp. 1-4.*, 2018.

[10] K. Jeeva Priya, S. S. Sree,Navya,Deepa Gupta "Implementation of Phonetic Level Speech Recognition in Kannada Using HTK," *2018 Int. Conf. Commun. Signal Process. (ICCSP), Chennai, 2018, pp. 0082-0085.*, 2018.

[11] . T. Sahana, N. Srilasya, S.Vinay,K. Jeeva Priya, Deepa Gupta, "Comparison of different Acoustic Models for Kannada language using

Kaldi Toolkit," *2018 Int. Conf. Adv. Comput. Commun. Informatics (ICACCI), Bangalore, 2018, pp. 2415-2420*, 2018.
[12] K. Sundar Karthikeyan, K. Jeeva Priya,Deepa Gupta , "Analysis of Digit Recognition in Kannada Using Kaldi Toolkit," *Emerg. Res. Electron. Comput. Sci. Technol. Proc. Int. Conf. ICERECT 2018*, p. Pages 813-821, 2018.
[13] A. M. and A. G. Ramakrishnan, "Design and development of a large vocabulary, continuous speech recognition system for Tamil," *2017 14th IEEE India Counc. Int. Conf. (INDICON), Roorkee, 2017, pp. 1-5.*, 2017.
[14] P. Nair, "The dummy's guide to MFCC." [Online]. Available: https://medium.com/prathena/the-dummys-guide-to-mfcc-aceab2450fd.
[15] R. S. Sharma, S. H. Paladugu, K. J. Priya and D. Gupta, "Speech Recognition in Kannada using HTK and Julius: A Comparative Study," *2019 International Conference on Communication and Signal Processing (ICCSP)*, Chennai, India, 2019, pp. 0068-0072.
[16] Sneha V., Hardhika G., Jeeva Priya K., Gupta D. (2018) Isolated Kannada Speech Recognition Using HTK—A Detailed Approach. In: Advances in Intelligent Systems and Computing, vol 564. Springer, Singapore
[17] http://kaldi-asr.org, "Kaldi Data preparation." [Online]. Available: http://kaldi-asr.org/doc/data_prep.html.
[18] https://web.stanford.edu, "Language Modeling: Probabilistic Language Models," *Stanford Univ. Nat. Lang. Process.*
[19] M. J. F. Gales, "Semi-tied covariance matrices for hidden Markov models."
[20] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Comput. Speech Lang. vol. 12, no. 2, pp. 75-98, 1998, ISSN 0885-2308*, 1998.
[21] P. A. D. Povey, Mohit Agarwal, "The subspace Gaussian mixture model- a structured model for speech recognition," *2010 IEEE Int. Conf. Acoust. Speech Signal Process.*, 2010.
[22] www.eleanorchodroff.com, "Kaldi Tutorial." [Online]. Available: https://www.eleanorchodroff.com/tutorial/kaldi/training-overview.html.
[23] A. Stolcke, S. R. I. International, and M. Park, "SRILM — AN EXTENSIBLE LANGUAGE MODELING TOOLKIT."
[24] http://kaldi-asr.org, "Decoding graph construction in Kaldi." [Online]. Available: http://kaldi-asr.org/doc/graph.html.
[25] D. Povey *et al.*, "GENERATING EXACT LATTICES IN THE WFST FRAMEWORK," vol. 213850, no. 102, pp. 3–6.