



Continuous Kannada Speech Recognition System Under Degraded Condition

P. S. Praveen Kumar¹ · G. Thimmaraja Yadava¹ · H. S. Jayanna²

Received: 18 October 2018 / Revised: 27 June 2019 / Accepted: 27 June 2019 / Published online: 15 July 2019
© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

In this paper, a continuous Kannada speech recognition system is developed under different noisy conditions. The continuous Kannada speech sentences are collected from 2400 speakers across different dialect regions of Karnataka state (a state in the southwestern region of India where Kannada is the principal language). The word-level transcription and validation of speech data are done by using Indic transliteration tool (IT3:UTF-8). The Kaldi toolkit is used for the development of automatic speech recognition (ASR) models at different phoneme levels. The lexicon and phoneme set are created afresh for continuous Kannada speech sentences. The 80% and 20% of validated speech data are used for system training and testing using Kaldi. The performance of the system is verified by the parameter called word error rate (WER). The acoustic models were built using the techniques such as monophone, triphone1, triphone2, triphone3, subspace Gaussian mixture models (SGMM), combination of deep neural network (DNN) and hidden Markov model (HMM), combination of DNN and SGMM and combination of SGMM and maximum mutual information. The experiment is conducted to determine the WER using different modeling techniques. The results show that the recognition rate obtained through the combination of DNN and HMM outperforms over conventional-based ASR modeling techniques. An interactive voice response system is developed to build an end-to-end ASR system to recognize continuous Kannada speech sentences. The developed ASR system is tested by 300 speakers of Karnataka state under uncontrolled environment.

Keywords Automatic speech recognition (ASR) · Continuous speech data · Kannada language · Kaldi tool kit · Word error rate (WER)

✉ G. Thimmaraja Yadava
thimrajyadav@gmail.com

Extended author information available on the last page of the article

1 Introduction

Human beings are the only creatures on this planet whose communication is based on the speech. There are so many people around the world who speak different languages, and we can recognize someone by hearing their speech provided if we are able to understand the language. From the past few decades, analysis, synthesis and recognition of human voice by the machine have moved toward becoming a baited subject in the field of research. There are billions of individuals around the globe talking distinctive dialects, but then, we can recognize somebody by tuning in to somebody's discussion or discourse as long as one can comprehend the dialect [18]. ASR system enables a client to communicate with PCs with no interface; say the console. The rule of inspiration driving speech recognition (SR) is to remove the progression of speech sounds and the messages which best matches the data speech flag. With the advancement of information technology (IT), the human and machine communication utilizing common human speech has expanded its request both in scholastics and additionally in business. For speech researchers, it has been an objective for long to build up a framework which can comprehend the common dialect. To build up man–machine association, the correspondence through common dialect is thought to be as one of its fundamental points. In case, because of a characteristic dialect as an information, the framework makes a typically amend move, and at that point, it can be stated that the framework can comprehend a characteristic dialect.

The procedure of speech perceiving and understanding is an information escalated process, which must consider the diverse parts of speech. ASR is one of the testing research zones on account of the varieties, for example, environment, speaker and context [19]. The SR systems comprehensively characterized as two sorts: speaker-dependent framework and speaker-independent framework. The speaker-dependent system infers that the framework is depending upon the speaker's unique voice characteristics. For this sort of frameworks, it is required to prepare the framework with the speech samples of the same speaker. Moreover, the restriction of such a framework is: it is speaker-dependent and is profitable just to an obliged course of action of individuals for which the framework has prepared. In spite of the fact that speaker-independent SR frameworks are more open to various sorts of voices, these frameworks can be utilized by various clients without training their speech information to perceive. In any case, for this kind of frameworks, exactness is less. Speaker versatile SR frameworks are in like manner there in the literature. There are many popular SR toolkits available such as HMM Tool Kit (HTK), CMU Sphinx and Kaldi.

The author in [33] depicts a large-vocabulary, speaker-independent, continuous SR framework that depends on HMM of phoneme-sized acoustic units utilizing continuous blend Gaussian densities. The pressure technique is produced to build the word transcription lexicon from the acoustic–phonetic marks of sentence articulations. The Viterbi beam search is utilized for deciphering, and the segmental K-means calculation is actualized as a benchmark for assessing the base up consolidating procedure. For the experimental purpose, the TIMIT database with the dictionary size of 855 words has been considered. For test sets of size 25, 105 and 855, the translating word correctnesses are 91%, 87.0% and 63%, respectively. The efficiency obtained by utilizing the combining algorithm is 4.2% which is more than that of utilizing the segmental

K-means (23% reduction in error), and the decoding precision utilizing the compacted lexicon is 3.2% more than that utilizing a standard lexicon (18% reduction in error). In [7], the authors researched the blend of complementary acoustic feature streams in large-vocabulary continuous SR (LVCSR). They investigated the utilization of acoustic features acquired utilizing a pitch-synchronous analysis with regular features by using Mel-frequency cepstral coefficients (MFCC). The features obtained using the pitch-synchronous analysis are exceptionally compelling when combined with the length of the vocal tract. They had consolidated these ghastrly portrayals specifically at the acoustic feature level utilizing heteroscedastic linear discriminant analysis (HLDA) and at the framework level. The outcomes demonstrate that consolidating the traditional and the pitch-synchronous acoustic feature sets utilizing HLDA brings about a predictable, noteworthy reduction in WER.

In [31], the authors attempted to overcome the drawbacks of HMMs with the help of template matching. For this purpose, they made use of dynamic time warping (DTW) algorithm. The experiments were done on the Resource Management (RM) benchmark where four test sets were used. HMM alone yields the WER of 3.54% compared to that of the proposed system where the combination of DTW and HMM yields the WER of 2.41%. The study made in [25] demonstrated the concept of Missing Feature Theory (MFT) for improving the overall robustness of SR. The system which they proposed operates on both binary mask (uses a hard decision criterion) and fuzzy mask (uses a soft decision criterion). Since the maximum likelihood (ML) estimation technique is used in the back end of the MFT system, the system has developed robustness to convolutional noise and additive noise. The Aurora 4 large-vocabulary database is used for experimental purpose. Seven sets of data were created by corrupting with seven different varieties of noises. The training set consists of total 7138 test utterances from 83 speakers that are nearly equal to 14 h of data. The test dataset consists of 340 test utterances from eight individuals. The baseline dataset (the dataset that did not use any noise removal algorithm) has obtained the WER of 28.96%. The WER for MFT using a priori (AP) and vector quantization (VQ) masks are 12.21% and 19.67%, respectively.

In search of further progress in the field of large-vocabulary continuous SR (LVCSR), the authors in [30] carried out the research to find the role of Reservoir Computing (RC) for acoustic modeling. The implementations were made on the WSJ0 5k benchmark comprising of 7240 utterances from 84 speakers nearly equal to 15 h of data. The proposed RC-HMM hybrid model yields impressive results compared to all other AMs of ASR. The RC-HMM model has obtained the WER of 6.2% (bigram language model) and 3.9% (trigram). The researchers in [29] felt that the automatic model complexity control still remained as one of the challenging problems in statistical modeling for many practical applications especially in the case of speech and language processing systems. To overcome this, the authors introduced a novel model complexity control method for generalized variable parameter HMMs (GVP-HMM). They proved that the proposed method increases the computational efficiency of the traditional GVP-HMMs. The research was carried out on the Aurora 2 database which consists of 420 utterances. The results showed that the proposed methodology increases the efficiency of GVP-HMM-based AMs by minimizing the WER by 20–28%.

The work in [5] reported that micro-modulation components will capture the significant characteristics of spoken speech which in turn results in the improvement of the efficiency of the ASR system. The experiment is conducted on the Aurora 4 database. The training set comprised of 7137 utterances from 83 speakers, and test set contained 330 utterances from 8 speakers. The results show that the efficiency of the proposed system has increased up to 8% compared to the single-stream system. Sailor and Patil [22] have proposed a new concept of unsupervised learning model which is based on the convolutional restricted Boltzmann machine (RBM). They have done experiments on three types of databases, namely small-vocabulary speech database (TIMIT database which consists of 462 utterances for training and 50 utterances for testing from 24 speakers), large-vocabulary speech databases (Wall Street Journal database which consists of 7138 utterances for training which approximately equals to 14h of speech data from 84 speakers and 5K-word and 20K-word vocabulary is used for testing) and noisy speech database (Aurora 4 database with different types of additive noises which consists of 7138 utterances for training and 330 utterances for testing). The authors have experimented with both cepstral and filter-bank features. For Wall Street Journal database, they obtained improvement in WER in the range of 7.21–17.8% compared to that of MFCC features. For Aurora 4 database, the improvement of 4.8–13.65% in WER was achieved using DNN-HMM model.

Lu and Renals [11] proposed the application of highway deep neural network (HDNN). The experimental evidence shows that the recognition accuracy of HDNNs is more compared to regular DNNs. Also, the authors claim that HDNNs are more controllable and adaptable than DNNs. The experiments were conducted on Augmented Multi-party Interaction (AMI) with 80h of training data. HDNNs have achieved the WER of 24% compared to that of DNN which has got WER of 24.6%. This shows that the efficiency of HDNN AM is marginally improved over conventional DNN AM. More recently, Ganapathy [6] highlighted that the efficiency of the ASR system drops down drastically in the presence of noise and reverberation. The advantages of autoregressive (AR) modeling such as preserving high-energy regions of the signal are efficiently implemented. A new method for extracting the features of speech data which combines the advantages AR approach and multivariate AR modeling (MAR) was proposed. The experiment was performed on WSJ Aurora 4 corpus whose training set has 7138 utterances from 84 speakers and testing set has 330 recordings from 8 speakers. The Kaldi toolkit is used for training the ASR system. The performance is evaluated for clean speech as well as noisy speech data. The WER obtained for the proposed algorithm for clean speech and noisy speech is 3.1% and 12.8%, respectively, which is far better than that of WER obtained using other feature extraction algorithms.

Of late, Pulkit et al. stressed the need for multi-level decomposition [26] for SR by quoting that the single-level decomposition is inappropriate for the speech signal. It was proved that the speech features obtained using sparse representation yield more convincing results in SR tasks. Based on the concept of sparse representation, the authors have introduced the concept of deep sparse representation (DSR) which is based on multi-level decomposition. The efficiency of the presented algorithm has experimented on three databases: TIMIT, Wall Street Journal and Aurora 4. For all the three databases, the features obtained using DSR have the best recognition rate com-

pared to that of features obtained using MFCC. Thus, the proposed DSR algorithm outperforms the existing feature extraction algorithms. In [23], the authors implemented the continuous SR for the Persian language. For this purpose, they created their own lexicon for the Persian language. To bring robustness to the SR system, they implemented a new robustness technique through the PC-PMC algorithm. This algorithm is a combination of principal component analysis (PCA), cepstral mean subtraction (CMS) and parallel model combination (PMC). The principal advantage of this algorithm is that the additive noise compensation ability of PMC and convolutional noise removal capability of both PCA and CMS methods are implemented together. The database has 6090 Persian sentences from 310 speakers. Out of this, 150 sentences from 7 speakers are used as the test set and the remaining utterances belong to the training set. The proposed algorithm has achieved the WER of 5.21%.

In [27], Siniscalchi et al. exhibited that the word recognition exactness can be altogether upgraded by masterminding DNNs in a hierarchical structure. The suggested arrangement is assessed on the 5000-word WSJ task, bringing about predictable and noteworthy upgrades in both telephone and word recognition precision rates. They have additionally broken down the impacts of different displaying decisions on the framework execution. It was shown that the two-stage configurations take into account better frame-classification accuracies. Framework execution can likewise be enhanced by supplanting MLP with a solitary nonlinear concealed layer with profound neural models that has five hidden layers. The optimum WER of 6.2% is accounted for WSJ0 utilizing an ANN-HMM hybrid framework.

Late improvements in the dynamic Bayesian networks (DBN) permit their utilization in genuine world applications is the primary effective utilization of DBNs to a large-scale SR issue. Regardless of whether their advance is gigantic, those models do not have a prejudicial capacity particularly on SR, for example, the HMM. Zarrouk et al. [32] have talked about the execution of the hybridization of SVMs with DBN for Arabic triphones-based continuous speech. The best outcomes were obtained with the presented framework SVM/DBN where the authors had accomplished the optimum recognition rate of 79.86% for a test speaker. The speech recognizer was assessed with Arabic database and obtained 8.15% WER when compared with 11.12% for triphones blend Gaussian DBN framework, 11.63% for hybrid model demonstrate SVM/HMM and 13.14% for HMM guidelines. The literature in [2] portrays the usage of an SR framework in Assamese dialect. The database used for this exploration work comprises a dictionary of ten Assamese words. The models for speech recognition are prepared to utilize HMM, VQ and I-vector strategy. The two new combination strategies are presented in this exploration by joining the above-mentioned three techniques. In the first strategy, the recognition results of HMM, VQ and I-vector technique combined together to improve the recognition rate. The authors named this technique as fusion-1 technique. They proceed further in the quest to improve the recognition rate, and they combined the outputs of HMM, VQ, I-vector and the fusion-1 technique; they called it as fusion-2 technique. The demonstrated results show that the fusion-2 technique outperforms all other existing techniques including fusion-1 technique.

The work in [14] demonstrated that convolutional neural networks (CNNs) can model telephone classes from the original acoustic speech signal, achieving execution keeping pace with other existing feature-based methodologies. The experiments

proved that the CNN-based approach accomplishes superior execution compared to the traditional ANN-based approach. They additionally demonstrated that the features gained from the original speech by the CNN-based approach could sum up crosswise over various standard databases available. In [20], Suman Ravuri proposed DNN Latent Structured SVM (LSSVM) AMs as an alternate for DNN-HMM hybrid acoustic models. Contrasted with existing strategies, approaches in view of margin maximization, as is considered in this work, appreciate better hypothetical legitimization. The author additionally stretched out the structured SVM model to incorporate latent factors in the model to represent the vulnerability in state arrangements. Presenting the latent structure reduces the complexity, regularly requiring 34% to 67% fewer utterances to converge contrasted with substitute criteria. On an 8-h autonomous test set of conversational speech, the proposed strategy reduces WRR by 8% with respect to a cross-entropy trained hybrid framework, while the best existing framework diminishes the WRR by 6.6% relative.

The study in [12] claims that the linear dynamic model (LDM) can model higher-order statistics and can explore correlations of features in a dynamic way. The researchers introduced a new hybrid LDM-HMM algorithm and proved that the proposed algorithm outperforms the existing HMM systems. For this purpose, they took Aurora 4 corpus which consists of original WSJ0 data. The results reveal that for clean speech data, the proposed algorithm has the WER of 11.6% compared to that of HMM alone is 13.3%. In the presence of babble noise, the proposed hybrid model achieves WER of 13.2%. The authors in [24] tried to implement the ASR system for continuous speech in South Indian languages (Kannada, Telugu, Malayalam and Tamil). The aim of this work was to extract the features of speech data using MFCC and shifted delta cepstrum (SDC) and compare the recognition performance of both the algorithms. It is believed that the SDC will improve the efficiency of recognition compared to that of the classical cepstral and delta cepstral features. The SVM is used as a classifier for all the South Indian languages. The database consists of 250 sentences from 125 people for all the aforementioned languages. The results revealed that the recognition performance using SDC outperforms the recognition performance using MFCC for all the four South Indian languages.

In [15], Branislav et al. used open-source Kaldi as the SR toolkit to implement DNN algorithms for large- vocabulary continuous speech in the Serbian language. The database consists of training and testing sets. The training set consists of 22000 utterances which are nearly equal to 90h of data. The testing set consists of 1000 utterances which are approximately equal to 5 h of data, and the lexicon consists of around 22000 Serbian words. The research here was to analyze the efficiency of hybrid GMM-HMM and DNN AMs. The results reveal that the DNN with three hidden layers has the WER of 1.86%, whereas the GMM-HMM hybrid model has the WER of 2.19%. With the urge of increasing the rate of recognition of continuous Arabic speech, the authors in [13] implemented the hybrid algorithm which is a combination of learning vector quantization (LVQ) and HMM. The database consists of 4h of continuous Arabic speech obtained from different television clips. The CMU Sphinx is used as a platform to implement the proposed algorithms. Here, the aim was to implement both LVQ AM and hybrid LVQ-HMM separately and to compare the efficiency of both. The results obtained were more convincing with 72% of recognition rate for LVQ

alone, whereas the combination of LVQ and HMM has got the recognition rate of over 89%. Thus proved that the proposed LVQ-HMM modeling technique outperforms the LVQ modeling technique alone for continuous Arabic speech corpus.

Kipyatkova and Karpov [10] stressed the significance of the language model (LM) for recognition of continuous speech. With the help of LM, they implemented the continuous SR for Russian speech corpus. The speech models were built using recurrent neural networks (RNN) with a varying number of elements in the hidden layer. The research was conducted for Russian speech database that has totally over 22 h of continuous speech data. The training set has 327 continuous spoken phrases from 50 speakers, and the testing set consists of 100 continuous spoken phrases from five speakers. The AMs were created with the Hidden Markov Model Toolkit (HTK). The proposed RNN with 500 units in the hidden layer and trigram LM yields the WER of 21.87% which has over 14% improvement in WER compared to results obtained with the trigram LM on the same test data. Again in 2017, Zohreh Ansari and Seyyed Ali Seyyedsalehi quoted that the reason for the drop in the efficiency of ASR systems is due to the improper design of the system and training procedure. To overcome this, they introduced a new concept of growing modular deep neural network (MDNN) [1]. The MDNN is trained such a way that its ability to grow makes it conceivable to accomplish spatio-temporal data of the speech frames at the input and their labels at the output layer in the meantime. The experimentation is performed on the Persian speech dataset FARSDAT. This dataset consists of 314 speakers including both male and female with different age groups and dialects. Totally, there were 6080 utterances out of which 5940 utterances from 297 speakers were used for training purpose and 140 utterances from 7 speakers were allocated for the testing phase. The results show that the proposed MDNN with three hidden layers has obtained the recognition rate of 76.5% compared to that of the hybrid GMM-HMM combination which has got the recognition rate of 73.17%.

The rest of the paper is composed as follows: In Sect. 2, the procedure followed to collect the speaker's speech data is described. The details regarding the open-source SR toolkit Kaldi are discussed in Sect. 3. Section 4 gives the procedure of building the LMs and AMs for continuous Kannada speech data using Kaldi. The system training and testing using Kaldi SR toolkit are discussed in Sect. 5. The process of building the of spoken query system to recognize the continuous Kannada speech sentences is described in Sect. 6, and at last the conclusions are stated in Sect. 7.

2 Speakers Speech Data Collection

To implement the continuous SR system in the Kannada language, the speech data are collected from the speakers of Karnataka state across different dialect regions in order to capture all possible pronunciation. The speech samples from the speakers are collected in a real-world environment, i.e., in noisy and reverberant conditions. The schematic portrayal of the block diagram used to implement an end-to-end continuous SR system for the Kannada language is shown in Fig. 1.

As shown in the figure, there are various stages in which the continuous speech recognition system is implemented. These stages are as follows:

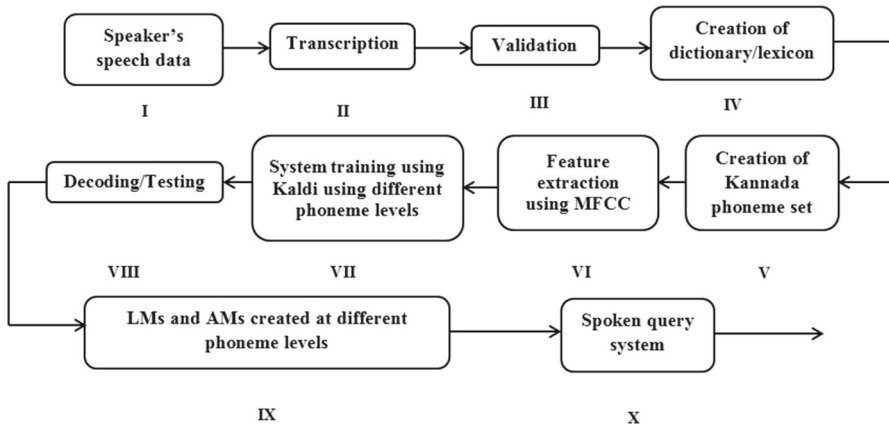


Fig. 1 The basic building block diagram of continuous speech recognition system for Kannada language

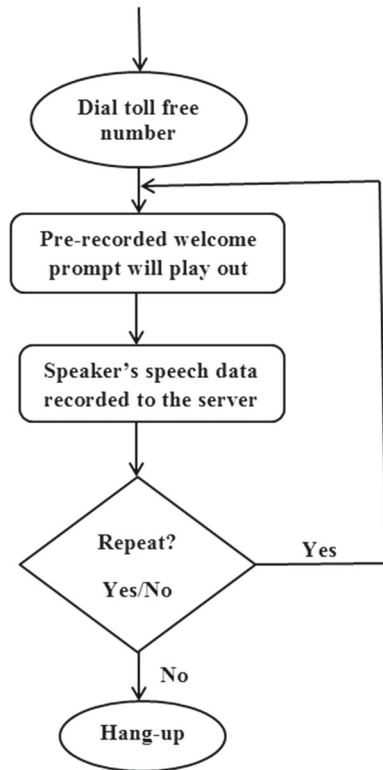
1. Speakers speech data collection under noisy and reverberant conditions.
2. The speech data collected from the speakers are subjected to transcription.
3. Authentication of transcribed speech data.
4. Building of lexicon/vocabulary.
5. Formulation of phoneme set for the Kannada language.
6. Feature extraction using MFCC.
7. Training of ASR system through Kaldi.
8. Decoding/testing.
9. Evolution of ASR models at various phoneme levels.
10. Evolution of end-to-end continuous SR system utilizing created ASR models.

The flowchart for the continuous speech data collection is depicted in Fig. 2.

The Bharat Sanchar Nigam Limited (BSNL) provided the telephone facility for IVRS call flow. The IVRS call flow took care of all the decision making processes. The continuous speech data are collected from 2400 speakers (1440 males and 960 females) of the age group in the range of 8–80 years. The speech data are obtained from various parts of the Karnataka state under real-world environment. It consists of 20,000 words that cover 30 districts. The accompanying focuses were contemplated while gathering the speech data from the speakers of the Karnataka state:

1. The state Karnataka can be divided into four physiographic landforms.
 - Northern Karnataka region: Speakers speak in northern dialects (Uttara Kannada).
 - Central Karnataka region: Speakers speak in central dialects (Are Bashe).
 - Southern Karnataka region: Speakers speak in southern dialects (Bengalooru Kannada).
 - Coastal (Karavali) Karnataka Region: Speakers speak in Coastal dialects (Karavali Kannada).
2. The ratio 60:40 (60% of male speakers and 40% of female speakers) is maintained throughout the process of data collection.

Fig. 2 Flowchart for the continuous speech data collection



3. The speech data are collected across the entire state of Karnataka covering every district in the state since there is diversity in speaking the Kannada language from region to region of Karnataka state.
4. The speech data are gathered from various mobile service providers and mobile phones, people belong to different age group, educated and unskilled individuals.

The few among the continuous speech sentences in Kannada dialects are shown in Table 1. These sentences are well known as Kannada gaadegalu/naannudigalu.

The collected speech data are subjected to manual transcription at word level and authenticated by supervisors.

3 The Open-Source Toolkit for Speech Recognition: Kaldi

Kaldi [16] is an open-source toolbox for SR scripted in C++ and authorized under the Apache License v2.0. The objective of Kaldi is to provide adaptable code that is straightforward, alter and broaden. Kaldi is accessible on SourceForge (visit <http://kaldi.sf.net/>). The tools compile on the more generally used Unix-like frameworks and on Microsoft Windows. A brief architecture of Kaldi toolkit showing various modules is depicted in Fig. 3.

Table 1 List of Kannada gaadegaluna/naannudigalu recorded from the people across the state of Karnataka

English Version of Gaadagalu	Kannada Version of Gaadegalu
veida sul:l:aadaru gaade sul:l:aagadu	ವೇದ ಸುಳ್ಳಾದರು ಗಾಡೆ ಸುಳ್ಳಾಗದು
ad:i:kege hooda maana aane kot:t:aru baaradu	ಅಡಿಕೆಗೆ ಹೋದ ಮಾನ ಆನೆ ಕೊಟ್ಟರೂ ಬಾರದು
kai kesaraadare baayi mosaru	ಕೈ ಕೆಸರಾದರೆ ಬಾಯಿ ಮೊಸರು
maatu bel:l:i mauna ban:gaara	ಮಾತು ಬೆಳ್ಳಿ ಮೌನ ಬಂಗಾರ
ad:d:a good:eya meile diipa it:t:a haage	ಅಡ್ಡ ಗೋಡೆಯ ಮೇಲೆ ದೀಪ ಇಟ್ಟು ಹಾಗೆ
akki meile aase nen:t:ara meile priiti	ಅಕ್ಕಿ ಮೇಲೆ ಆಸೆ ನೆನಟರ ಮೇಲೆ ಪ್ರೀತಿ
end nd e ban:daaga kand nd u muchchikon:d:d:an:te	ಎಣ್ಣೆ ಬಂದಾಗ ಕಣ್ಣು ಮುಚ್ಚಿಕೊಂಡಂತೆ
attegon:du kaala sosegon:du kaala	ಅತ್ತೆಗೊಂದು ಕಾಲ ಸೊಸೆಗೊಂದು ಕಾಲ
bekkige chellaat:a ilige praand a san:kat:a	ಬೆಕ್ಕಿಗೆ ಚೆಲ್ಲಾಟ ಇಲಿಗೆ ಪ್ರಾಣ ಸಂಕಟ
chin:te illadavanige san:teihuu nidde	ಚಿಂತೆ ಇಲ್ಲದವನಿಗೆ ಸಂತೆಲೂ ನಿಡ್ಡೆ
tun:bida kod:a tul:ukuvudilla	ತುಂಬಿದ ಕೊಡ ತುಳುಕುವುದಿಲ್ಲ
hani hani seiridare hal:l:a tene tene seiridare bal:l:a	ಹನಿ ಹನಿ ಸೇರಿದರೆ ಹಳ್ಳ ತೆನೆ ತೆನೆ ಸೇರಿದರೆ ಬಳ್ಳ
ati aase gati geid:u	ಅತಿ ಆಸೆ ಗತಿ ಗೇಡು
ban:daddellaa barali goovin:dana dayeyirali	ಬಂದದ್ದಲ್ಲಾ ಬರಲಿ ಗೋವಿಂದನ ದಯೆಯಿರಲಿ
beiline eddu hola meiyitan:t:e	ಬೇಲಿನ ಎದ್ದು ಹೊಲ ಮೇಯಿತಂತೆ
hand nd ele uduruvaaga chigurele nagutittu	ಹಣ್ಣೆಲೆ ಉದುರುವಾಗ ಚಿಗುರಲೆ ನಗುತ್ತಿತ್ತು
haal:uurige ul:ivadane gaud:a	ಹಾಳೂರಿಗೆ ಉಳಿದವನ ಗೌಡ
haagalakaayige beivinakaayi saakshhi	ಹಾಗಲಕಾಯಿಗೆ ಬೇವಿನಕಾಯಿ ಸಾಕ್ಷಿ
haavuu saayalilla koolu muriililla	ಹಾವೂ ಸಾಯಲಿಲ್ಲ ಕೋಲು ಮುರಿಯಲಿಲ್ಲ
hand a an:dre hend avuu baayi bid:uttade	ಹಣ ಅಂದ್ರೆ ಹೆಣವೂ ಬಾಯಿ ಬಿಡುತ್ತದೆ
huuvini:n:da naaru swarga seiritu	ಹೂವಿನಿಂದ ನಾರು ಸ್ವರ್ಗ ಸೇರಿತು
hiriyakkana chaal:i mane man:digella	ಹಿರಿಯಕ್ಕನ ಚಾಳಿ ಮನೆ ಮಂದಿಗಿಲ್ಲ
hittala gid:a maddalla	ಹಿತ್ತಲ ಗಿಡ ಮದ್ದಲ್ಲ
hut:t:u gund a sut:a:t:aru hoogolla	ಹುಟ್ಟು ಗುಣ ಸುಟ್ಟರೂ ಹೋಗೊಲ್ಲ
ollada gan:d:anige mosaralli kallu	ಒಲ್ಲದ ಗಂಡನಿಗೆ ಮೊಸರಲ್ಲಿ ಕಲ್ಲು
raatri kan:d:a baavilli hagal biddan:ge	ರಾತ್ರಿ ಕಂಡ ಬಾವೀಲಿ ಹಗಲು ಬಿದ್ದಂಗೆ
vinaasha kaalakke vipariita buddhi	ವಿನಾಶ ಕಾಲಕ್ಕೆ ವಿಪರೀತ ಬುದ್ಧಿ
yaradoo dud:du yallammana jaatre	ಯರದೋ ದುಡ್ಡು ಯಲ್ಲಮ್ಮನ ಜಾತ್ರೆ
hol:eyuvudellaa chinnavalla	ಹೊಳೆಯುವುದೆಲ್ಲಾ ಚಿನ್ನವಲ್ಲ
aapattigaadavane nijavaada gel:aya	ಆಪತ್ತಿಗಾದವನೇ ನಿಜವಾದ ಗೆಳೆಯ
in:dina soolu naal:ina geluvu	ಇಂದಿನ ಸೋಲು ನಾಳಿನ ಗೆಲುವು
gid:avaagi baggaddu maravaagi baggiitei	ಗಿಡವಾಗಿ ಬಗ್ಗಿದ್ದು ಮರವಾಗಿ ಬಗ್ಗಿತೇ
manasiddare maarga	ಮನಸಿದ್ದರೆ ಮಾರ್ಗ
uppu tin:da meile niiru kud:iyalei beiku	ಉಪ್ಪು ತಿಂದ ಮೇಲೆ ನೀರು ಕುದಿಯಲೇ ಬೇಕು
namma deivara satya namage gottu	ನಮ್ಮ ದೇವರ ಸತ್ಯ ನಮಗೆ ಗೊತ್ತು
gaayada meile bare el:edan:te	ಗಾಯದ ಮೇಲೆ ಬರೆ ಎಳಿದಂತೆ
aaseyei duh:khakke muula	ಆಸೆಯೇ ದು:ಖಕ್ಕೆ ಮೂಲ
atiyaadare amrxtavuu vishha	ಅತಿಯಾದರೆ ಅಮೃತವೂ ವಿಷ
bad:avana sit:t:u davad:ege muula	ಬಡವನ ಸಿಟ್ಟು ದವಡೆಗೆ ಮೂಲ

The toolbox relies upon two outside libraries that are additionally uninhibitedly accessible: One is OpenFst for the finite state structure, and the other is numerical algebra-based math libraries. The standard Basic Linear Algebra Subroutines (BLAS) and Linear Algebra PACKage (LAPACK) are used to build the aforementioned libraries. The library functionalities can be accessed through commands scripted in C++, and they are often called from a scripting dialect for building and running a speech recognizer. To build LMs and AMs, Kaldi is one of the most predominant SR toolkit available in recent days. A number of SR toolkits are presently available in the

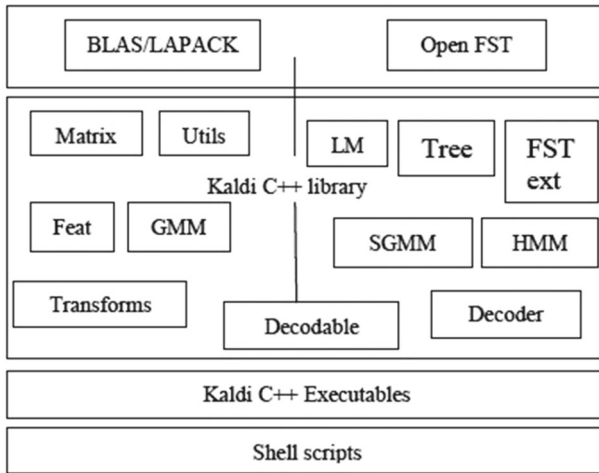


Fig. 3 Architecture of the open-source speech recognition toolkit: Kaldi

market but when compared to all other SR toolkits Kaldi stands out in a different line due to its more advantageous features such as:

- Reconciliation with Finite State Transducers (FST): With FST, the efficiency of ASR system can be improved with reduced WER.
- Large-scale reinforcement of linear algebra: The standard linear algebra packages such as BLAS and LAPACK are included to build sophisticated and strong matrix library.
- Greater flexibility in designing: The algorithms/scripts available in Kaldi are more of a general form. They can be easily modified according to the specific needs.
- Open-source license: The code is authorized under Apache v2.0, which is one of the lightest prohibitive licenses accessible.

Kaldi supports the feature extraction techniques like MFCC, perceptual linear prediction (PLP), vocal tract length normalization (VTLN), cepstral mean and variance normalization, linear discriminant analysis (LDA), maximum likelihood linear transforms (MLLT), heteroscedastic LDA (HLDA) and so on. For building AMs, the toolkit supports modeling techniques such as GMM, HMM and SGMM and it also supports speaker adaptation techniques such as maximum likelihood linear regression (MLLR) and feature-space adaptation using feature-space MLLR (fMLLR). The toolkit also supports speaker normalization using a linear approximation to VTLN which is implemented in speaker adaptive training (SAT) of the AMs.

3.1 Creation of AM

The building of the AM is based on the Kaldi procedure. Our motivation here is to prepare a DNN model, which may perform well as for the WER is concerned. For this, six diverse acoustic modeling frameworks are produced. For three of them, the emission probability of the HMM states is modeled by training GMM and others are

modeled using DNN models. To build AMs, the 39 dimensions MFCC features with first-order temporal derivative and second-order temporal derivative are used. There were three GMM-HMM AMs which are trained successfully.

- Triphone1: The first AMs were created using triphone1. These AMs were trained by directly using the features extracted using MFCC. Triphones are the combination of GMM and HMM.
- Triphone2: The second AMs were created using triphone2. Here, the AMs were trained using LDA followed by MLLT.
- Triphone3: The third AMs were created using triphone3. In third AMs, the ASR system is made speaker-independent with the help of SAT and fMLLR.

Keeping in mind the end goal to consider the impact of the setting on the acoustic realization of the phones, every one of these models is triphone based models. The third model, i.e., triphone3, has 100,000 Gaussians for 4,264 states. The DNN-HMM frameworks are prepared with the assistance from the frame-level cross-entropy, state-level minimum Bayes risk (sMBR) paradigm, the senone created from the third GMM-HMM model (triphone3) and comparing fMLLR changes. Altogether, we train three DNN models.

- DNN1 characterizes speech frames into various states of triphone, i.e., here, the probability density functions (PDFs) are estimated by DNN1. The training of DNN1 depends on the cross-entropy model.
- The building of AMs using DNN2 and DNN depends entirely on sMBR sequence-discriminative training. The contrast between the two models is the number of emphases used to prepare the model.

The neural networks are trained using sMBR sequence-discriminative training to enhance the entire sentences rather than a frame-based benchmark. The DNN models have six hidden layers, and there were 2048 nodes for each layer. There are 440 nodes (40 dimensional features of fMLLR grafted crosswise over five frames on each side of the central frame) in the input layer, and the output layer has 4,264 nodes. The quantity of parameters to appraise is around 30.6 million. The block diagram is shown in Fig. 4 sums up all the AMs used in this work.

3.2 Creation of LM

A 2-gram LMs are utilized to create the grid, and a 4-gram LM is utilized to re-score this grid. The customary training process is applied for the transcripts in the train corpus. The LMs are first trained for the continuous Kannada speech dataset; then, ideal weights are resolved so as to maximize the likelihood of the transcripts in the testing set, which has a size of around 4k Kannada words. The SRILM toolbox [28] has been used to prepare the diverse LMs and every one of them utilize Great-Turing (Katz) [9] smoothing method. It is realized that the efficiency of the Kneser–Kney smoothing [4] outwits the Katz technique. Nevertheless, in [3], the researchers proved through unique trial setup that the Katz smoothing performs much superior to the Kneser–Kney smoothing for forceful pruning administrations.

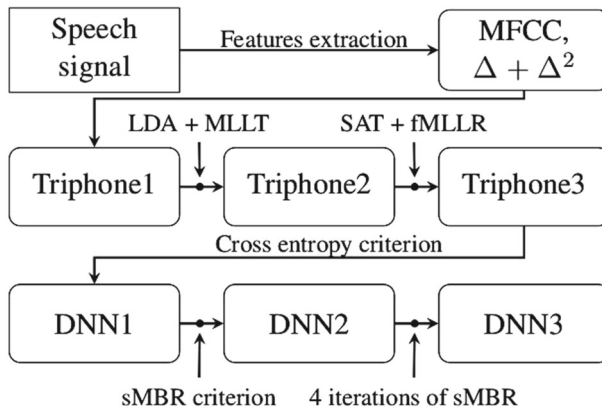


Fig. 4 Block diagram of AMs

4 Creation of ASR Models for Continuous Kannada Speech Data Through Kaldi

The gathered speech information from the speakers is utilized to construct LMs and AMs with the help of Kaldi toolkit. The accompanying hierarchy is implemented to build the LMs and AMs:

- The continuous speech data collected from the speakers are subjected to transcription.
- Validation of speech data transcribed in the previous step.
- Lexicon/vocabulary creation for the continuous Kannada speech data and Kannada phoneme set.
- Extracting the features from the speech data using MFCC.
- Training and testing of AMs and LMs through Kaldi.

4.1 Transcription of Continuous Kannada Speech Data Collected from the Speakers

The continuous Kannada speech data obtained from the speakers are subjected to transcription from word level to the phoneme level. The tool used for the transcription is the Indic transliteration (IT3 to UTF-8) tool, and the same is depicted in Fig. 5. The tags for non-lexical sounds which are also known as silence phones used during the transcription of the speech data are depicted in Table 2.

From Fig. 5, it can be noted that the speakers speech data were recorded as “veida sul:l:aadaru gaade sul:l:aagadu” and it was translated as “veida_sul:l:aadaru_gaade_sul:l:aagadu” with the help of IT3-UTF:8.



Fig. 5 The IT3 to UTF-8 tool used to transcribe the collected continuous Kannada speech data

Table 2 Tags for non-lexical sounds (Speech Disfluencies)

Tag	Explanation
<pau>	Pause or silence. It is overlooked when it is created at the beginning or toward the finish of the articulation
<aah>	Faltering sound (with mouth open)
<hmm>	Hesitation sound (with mouth closed)
<hm>	Sound of consent (yes)
<hm-hm>	Sound of disavowal (no)
<laugh>	Snickering laugh by the speaker
<horn>	Any machine produced restricted band sound, for example, horn of vehicle, beep, telephone ringing, shriek. Every one of these sounds will have the same, normal tag: <horn>
<babble>	Speech or sound created by different people that are difficult to comprehend (for example, yell, individuals talking out of sight, mumbling sound and so on)
<bang>	Sudden (impulsive) noise because of slamming of doors or sound of an applaud, the uproarious sound of protest tumbling down and so on
<bn>	Background noise is some other noisy background sound (of high energy) that cannot be labeled as either bang or horn
<vn>	Vocal noise is any non-lexical sound created by the speaker that cannot be portrayed by any of the above labels. Cases of such sounds are tongue click, making a sound as if to speak, breath clamor, hack, snuffle, lip smack, hiccups, yawns, snorts. Every such solid will have a typical, single tag: <vn>

4.2 Validation of Transcribed Speech Data

In the event that the transcriber unsuccessful to translate the speaker’s speech data in a precise manner, at that point, the validation tool is utilized to validate the transcribed continuous Kannada speech data. From Fig. 5, it is seen that the voice clip is translated as “veida_sul:l:aadaru_gaade_sul:l:aagadu” just yet it was influenced by background disturbance. Along these lines, it can be validated as “veida_sul:l:aadaru_gaade_sul:l:aagadu” <bn> as appeared in Fig. 6

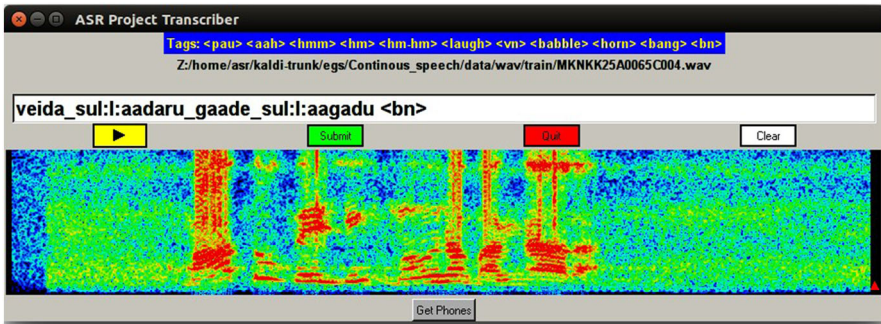


Fig. 6 The tool used to validate the continuous Kannada speech data

4.3 Building of Lexicon/Vocabulary and Phoneme Set for Continuous Kannada Speech Data

Kannada is a Dravidian dialect talked prevalently by Kannada individuals in India, mostly in the territory of Karnataka. The dialect has approximately 38 million local speakers, who are called Kannadigas (Kannadigaru). Kannada is likewise talked as the second and third dialect by non-Kannada speakers living in Karnataka, which means 50.8 million speakers. It is one of the scheduled dialects of India and the official and regulatory dialect of the territory of Karnataka. Kannada is the oldest of the four noteworthy Dravidian dialects with an abstract convention. The most established Kannada engraving was found at the little group of Halmidi and dates to around 450 CE. The Kannada script belongs to the family of the Brahmi script. It is a segmental, nonlinear letter in order content described by consonants showing up with a characteristic vowel. Each written symbol relates to one syllable, instead of one phoneme in dialects like English. Every letter in order is called as *akshara*, *akkara* or *varna*, and every letter has its own frame (aakaara) and sound (shabda) giving the unmistakable and capable of being heard portrayals. Kannada letter set is famously known as *Aksharamale* or *Varnamale*, and the current *Varnamale* list comprises of 49 characters. Keeping in mind the end goal to make the acknowledgment framework good to the prior *Varnamale* set, 50 characters have been considered in the present work. In any case, the quantities of written symbols are significantly in excess of 50 characters, as characters can join to shape compound characters prompting *Ottaksharas*. Likewise, the content is confused because of the event of different mixes of half letters or images that join to different letters in a way like diacritical imprints. The 50 essential characters are grouped into three classifications. They are *Swaras* (vowels), *Vyanjanas* (consonants) and *Vogavahakas* (part vowel, part consonants). There are fourteen vowels and are called *swaras*. Table 3 demonstrates the graphemes of vowels and the relating ITRANS (Indian dialect Transliterations). The ITRANS of the corresponding alphabets are in the brackets. The *Anusvara* and *Visarga* are the two *Yogavahakas*, and these two are additionally assembled under vowel classification. The *Vyanjanas* are ordered into structured and unstructured consonants. The structured consonants are additionally characterized into five groups as per where the tongue touches the mouth while artic-

Table 3 Classification of Kannada alphabets and their respective ITRANS

Vowels	ಅ	ಆ	ಇ	ಈ	ಉ	ಊ	ಋ	ೠ	ಎ	ಏ
	(a)	(aa)	(i)	(I)	(u)	(U)	(Ru)	(RU)	(e)	(E)
	ಐ		ಒ		ಓ		ಔ			
	(ai)		(o)		(O)		(ou)			
Yogavahakas	ಅಂ (aM)					ಅಃ (aH)				
Structured consonants	ಕ		ಖ		ಗ		ಘ		ಙ	
	(ka)		(kha)		(ga)		(gha)		(nga)	
	ಚ		ಛ		ಜ		ಝ		ಞ	
	(cha)		(Cha)		(ja)		(jha)		(nja)	
	ಟ		ಠ		ಡ		ಢ		ಣ	
(Ta)		(Tha)		(Da)		(Dha)		(Na)		
ತ		ಥ		ದ		ಧ		ನ		
(ta)		(tha)		(da)		(dha)		(na)		
ಪ		ಫ		ಬ		ಭ		ಮ		
(pa)		(pha)		(ba)		(bha)		(ma)		
Unstructured consonants	ಯ	ರ	ಲ	ವ	ಶ	ಷ	ಸ	ಹ	ಳ	
	(ya)	(ra)	(la)	(va)	(sha)	(Sha)	(sa)	(ha)	(La)	

Table 4 The labels imported from IT3 to UTF-8 for Kannada phonemes

Label set using IT3: UTF-8				Respective Kannada phonemes			
a	oo	t:h	ph	ಅ	ಓ	ಠ	ಪ್ಲ
aa	au	d	b	ಆ	ಔ	ಢ	ಬ
i	k	d:h	bh	ಇ	ಕ	ಢ	ಭ
ii	kh	nd	m	ಈ	ಖ	ಣ	ಮ
u	g	t	y	ಉ	ಗ	ತ	ಯ
uu	gh	th	r	ಊ	ಘ	ಥ	ರ
e	c	d	l	ಎ	ಚ	ದ	ಲ
ee	ch	dh	v	ಏ	ಛ	ಧ	ವ
ai	j	n	sh	ಐ	ಜ	ನ	ಶ
o	t:	p	s	ಒ	ಟ	ಪ	ಸ

ulating these characters. The unstructured consonants are those, which do not have a place with any of the organized consonants.

The order of structured consonants principally relies upon the tongue contacts the mouth palate, viz. velars, palatals, retroflex, dental and labials. The verbalization of unstructured consonants will not contact the mouth palate, and this has any sort of impact between the structured and unstructured consonants. The labels utilized from IT3 to UTF-8 for Kannada phonemes appear in Table 4. Along these lines, the Kannada ASR framework is actualized by demonstrating the 48 phonetic symbols. The labels utilized from the Indian Speech Sound Label Set (ILSL12) are appeared in Table 5.

The lexicon for Kannada dialect is made by utilizing both IT3 to UTF-8 and ILSL12 mark set appeared in Table 6. The left side of the lexicon is made by utilizing IT3 to UTF-8, and the right side of the word reference is made by utilizing ILSL12.

Table 5 The labels imported from ILSL12 for Kannada phonemes

Label set using ILSL12				Respective Kannada phonemes			
a	oo	txh	ph	ಅ	ಓ	ಠ	ಫ
aa	au	dx	b	ಆ	ಔ	ಢ	ಬ
i	k	dxh	bh	ಇ	ಕ	ಢ	ಭ
ii	kh	nx	m	ಈ	ಖ	ಢ	ಮ
u	g	t	y	ಉ	ಗ	ತ	ಯ
uu	gh	th	r	ಊ	ಘ	ಢ	ರ
e	c	d	l	ಎ	ಚ	ದ	ಲ
ee	ch	dh	w	ಏ	ಛ	ಢ	ವ
ai	j	n	sh	ಐ	ಜ	ನ	ಶ
o	tx	p	s	ಒ	ಟ	ಪ	ಸ

Table 6 Lexicon/vocabulary for Kannada dialect is built with the help of both IT3 to UTF-8 and ILSL12 label set

Label set using IT3: UTF-8	Label set using ILSL12
veida sul:l:aadaru gaade sul:l:l:aagadu	w e e d a _ s u l l x a d r u _ g a a d e _ s u l l x a a g d u
ad:ikege hooda maana aane kot:t:aruu baaradu	a d x i k e g e _ h o o d a _ m a a n a _ a a n e _ k o t t x a r u _ b a a r a d u
kai kesaraadare baayi mosaru	k a i _ k e s a r a a d a r e _ b a a y i _ m o s a r u
maatu bel:l:i mauna ban:gaara	m a a t u _ b e l l x i _ m a u n a _ b a n g g a a r a
ad:d:a good:eya meile diipa it:t:a haage	a d d x a _ g o o d x e y a _ m e i l e _ d i i p a _ i t t x a _ h a a g e
akki meile aase nen:t:ara meile priiti	a k k i _ m e i l e _ a a s e _ n e n t x a r a _ m e i l e _ p r i i t i
end~nd~e ban:daaga kand~nd~u muchchikon:d:d:ante	e n n x e _ b a n g d a a g a _ k a n n x u _ m u c c i k o n d x n t e
atregon:du kaala sosegon:du kaala	a t t e g o n d u _ k a a l a _ s o s e g o n d u _ k a a l a
bekkige chellaat:a ilige praanda san:kat:a	b e k k i g e _ c h e l l a a t a _ i l i g e _ p r a n x a _ s a n k a t a
chin:te illadavanige san:teiluu nidde	c h i n t e _ i l l a d a w a n i g e _ s a n t e l u u _ n i d d e
tun:bida kod:a tul:ukuvudilla	t u m b i d a _ k o d x a _ t u l x u k u w u d i l l a
hani hani seiridare hal:l:a tene tene seiridare bal:l:a	h a n i _ h a n i _ s e i r i d a r e _ h a l l x a _ t e n e _ t e n e _ s e i r i d a r e _ b a l l x a
ati aase gati ged:u	a t i _ a a s e _ g a t i g e i d x u
ban:daddellaa barali goovin:dana dayeyirali	b a n d a d e l l a a _ b a r a l i _ g o o w i n d a n a _ d a y e i r a l i
beiline eddu hola meiyitan:te	b e i l i n e _ y e d d u _ h o l a _ m e i t a n t e
hand~nd~ele uduruvaaga chigurele nagutittu	h a n n x e l e _ u d u r u w a a g a _ c h i g u r e l e _ n a g u t t i t t u
haal:uurige ul:ivadane gaud:a	h a a l x u u r i g e _ u l i d a w a n e i _ g a u d x a
haagalakaayige beivinakaayi saakshhi	h a a g a l a _ k a a y i g e _ b e i v i n a _ k a a y i _ s a a k s x i
haavuu saayalilla koolu muriyililla	h a a v u u _ s a a y a l i l l a _ k o o l u u _ m u r i y a l i l l a

Table 6 continued

Label set using IT3: UTF-8	Label set using ILSL12
hand~a an:dre hend~avuu baayi bid:uttade	h a n x a _ a n d a r e _ h e n x a w u u _ b a a y i _ b i d u t t a d e
huuvini:n:da naaru swarga seiritu	h u u w i n i n d a _ n a a r u _ s w a r g a _ s e i r i t u
hiriyakkana chaal:i mane man:digella	h i r i y a k k a n _ c h a a l i _ m a n e _ m a n d g e l l a
hittala gid:a maddalla	h i t t a l a _ g i d a _ m a d d a l l a
hut:t:u gund~a sut:a:t:aruu hoogolla	h u t t u _ g u n a _ s u t t a r u u _ h o o g o l l a
ollada gan:d:anige mosaralli kallu	o l l a d a _ g a n d a n i g e _ m o s a r a l l i _ k a l l u
raatri kan:d:a baaviili hagalu biddan:ge	r a a t i _ k a n d a _ b a a v i i l i _ h a g a l u _ b i d d a n g e
vinaasha kaalakke vipariita buddhi	w i n a a s h a _ k a a l a k k e _ w i p a r i i t a _ b u d d i
yaradoo dud:d:u yallammana jaatre	y a r a d o o _ d u d d u _ y e l l a m m a n a _ j a a t r e
hol:eyuvudellaa chinnavalla	h o l x e y u w u d e l l a a _ c h i n n a v a l l a
aapattigaadavanei nijavaada gel:aya	a a p a t t i g a a d a w a n e i _ n i j a v a a d a _ g e l e y a
in:dina soolu naal:ina geluvu	i n d i n a _ s o o l u _ n a a l i n a _ g e l u w u
gid:avaagi baggaddu maravaagi baggiitei	g i d a w a a g i _ b a g g a d d u _ m a r a w a a g i _ b a g g i i t e i
manasiddare maarga	m a n a s i d d a r e _ m a a r g a
uppu tin:da meile niiru kud:iyalei beiku	u p p u _ t i n d a _ m e i l e _ n i i r u _ k u d i y a l e i _ b e i k u
namma deivara satya namage gottu	n a m m a _ d e i w a r a _ s a t y a _ n a m a g e _ g o t t u
gaayada meile bare el:edan:te	g a a y a d a _ m e i l e _ b a r e _ e l e d a n t e
aaseyei duh:khakke muula	a a s e y e i _ d u h k h a k k e _ m u u l a
atiyaadare amrxtavuu vishha	a t i y a a d a r e _ a m r u t a w u u _ w i s h a
bad:avana sit:t:u davad:ege muula	b a d a w a n a _ s i t t u _ d a v a d e g e _ m u u l a

4.4 Features Extraction Using MFCC

The fundamental objective of the feature extraction module is to obtain the feature vectors that gives a minimized portrayal of the given input data. The feature extraction is done in three phases. In the primary stage, the speech data are analyzed. It performs on ranges of frequencies and analyzes the signals to produce power spectrum envelopes of short speech interims. The second stage aggregates a broadened feature vector made out of static and dynamic features. The last stage (which is not generally present) changes these broadened feature vectors into more compact and robust feature vectors that are then provided to the speech recognizer. In general, a 39-dimensional feature vector is used which is composed of first 13 MFCCs and their corresponding 13 delta and 13 delta-delta. The lower-order coefficients contain most of the information about the overall spectral shape of the source-filter transfer function. Even though higher-order coefficients represent increasing levels of spectral details, depending on the sampling rate and estimation method, 12–20 cepstral coefficients are typically optimal for speech analysis. Selecting a large number of cepstral coefficients results in more complexity in the models. MFCC depends on human hearing sense which

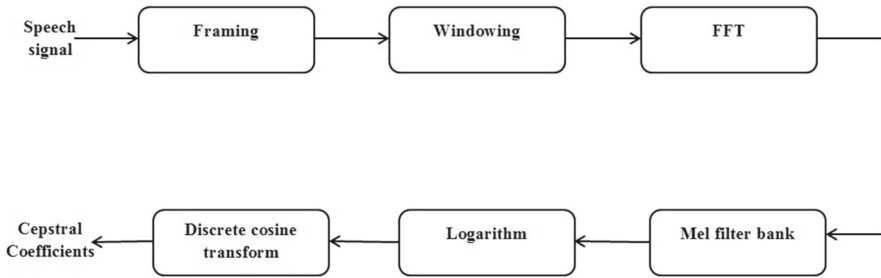


Fig. 7 The structure of MFCC processor

cannot recognize frequencies more than 1 KHz. It has two categories of filters that are linearly spaced at low frequencies beneath 1 KHz and are logarithmic separating over 1 KHz. A subjective pitch is an exhibit on the Mel-frequency scale to collect the essential characteristic of phonetic in speech. The basic building blocks of the MFCC extraction are given in Fig. 7.

The fundamental stages in extracting MFCC feature are listed as below:

- **Framing** In this progression, the continuous speech signal is split into shorter frames of N samples, with next frames isolated by M samples ($M < N$); with this, the contiguous frames are separated by $N-M$ samples. According to the literature survey, the standard value for N and M are 256 and 100 respectively. The reason for this is if the speech signals are framed too short then adequate samples are not obtained for further processing of such signals. If the frames are too large then the speech signal will be non-stationary. So the framing of speech signals should be such that it should neither be too short nor too long; it should be adequate enough to get good results.
- **Windowing** It is performed to overcome the distortions at the starting and towards the ending of each frame. Then, the frame and windowing function are multiplied together. Essentially, numerous window functions exist, for example, rectangular window, flat top window and hamming window; in any case, for the most part, the hamming window is used to perform windowing.
- **Fast Fourier Transform (FFT)** FFT is a procedure of transforming the signal from time domain to frequency domain. To acquire the frequency response of each frame, FFT is applied. By performing FFT, the output response is a spectrum or periodogram.
- **Mel filter bank** In this stage, the spectrum obtained in the previous step is mapped on Mel scale to know the estimate about the current energy at every spot with the assistance of triangular filter bank. These triangular filter banks are an arrangement of band-pass filters with dispersing alongside bandwidth chose by consistent Mel-frequency time. In this way, Mel scale encourages the way to place the filter and to ascertain how much wider it ought to be on the grounds that as the frequency moves toward the higher end, these filters are additionally become wider. The Mel-scaling mapping is to be done among the given genuine frequency scales (Hz) and the perceived frequency scale (Mels). At the time of the mapping, when

the frequency range is up to 1 KHz, the Mel frequency scaling is linear, and yet after 1 KHz, the spacing is logarithmic.

- **Cepstrum** In the final stage, the log of Mel spectrum which is in the frequency domain must be rolled back to time domain. The resultant outcome is known as the MFCCs. The cepstral portrayal of the speech spectrum is the representation of the local spectral properties of the signal for the given frame analysis. Since the MFCCs are genuine numerals (as are their logarithms), they might be restored back to the time space through the discrete cosine transform (DCT). Doing this, we got the MFCCs; thus, each utterance is transformed into a grouping of acoustic DCT. This removes the correlation between the yield estimations of the filter bank and gathers features of parameters. The acquired features are like cepstrum, and in this way, it is alluded to as the Mel scale cepstral coefficients or MFCC.

5 Training and Testing of the System Through Kaldi Speech Recognition Toolkit

The state Karnataka has distinctive lingo locales. Keeping in mind the end goal to collect every single conceivable articulation of locale, the speech information was gathered from various districts of Karnataka. The 80% and 20% of validated speech information are utilized for training and testing, respectively. The LMs and AMs were built independently for every district and for overall continuous speech data. Every speaker uttered 40 continuous sentences in the Kannada language for each session.

5.1 SGMM

The ASR systems based on the GMM-HMM structure usually involve completely training the individual GMM in every HMM state. A new modeling technique is introduced to the SR domain is called SGMM [17]. The dedicated multivariate Gaussian mixtures are used for the state-level modeling in conventional GMM-HMM acoustic modeling technique. Therefore, no parameters are distributed between the states. The states are represented by Gaussian mixtures, and these parameters distribute a usual structure between the states in the SGMM modeling technique. The SGMM consist of GMM inside every context-dependent state, and the vector $I_a \in V^r$ in every state is specified instead of defining the parameters directly. An elementary form of SGMM can be described by the below equations is as follows:

$$p(y|i) = \sum_{k=1}^L w_{ik} N(y; \mu_{ik}, \Sigma_k) \quad (1)$$

$$\mu_{ik} = M_k I_i \quad (2)$$

$$w_{ik} = \frac{\exp w_k^T I_i}{\sum_{k=1}^L \exp w_k^T I_i} \quad (3)$$

where $y \in R^D$ is a feature vector and $i \in 1, 2, \dots, I$ is the context-dependent state of speech signal. The speech state j 's model is a GMM with L Gaussians (L is in between 200 and 2000), with the matrix of covariances \sum_k which are distributed amidst states, mixture weights w_{ik} and means μ_{ik} . The derivation of $\mu_{ik}w_{ik}$ parameters is done by using I_i together with M_k , w_k and \sum_k . The detailed description of the parameterization of SGMM and its impact is given in [21]. The ASR models are developed using this modeling technique for Kannada speech database, and the least WER models could be used in SQS.

5.2 DNN

The GMM-HMM-based acoustic modeling approach is inefficient to model the speech data that lie on or near the data space. The major drawbacks of GMM-HMM-based acoustic modeling approach are discussed in [8]. The artificial neural networks (ANN) are capable of modeling the speech data that lie on or near the data space. It is found to be infeasible to train an ANN using the maximum number of hidden layers with back-propagation algorithm for a large amount of speech data. An ANN with single hidden layer failed to give good improvements over the GMM-HMM-based acoustic modeling technique. Both the previously mentioned confinements have been overwhelmed with the improvements in a previous couple of years. Different methodologies are accessible currently to train the distinctive neural nets with the most extreme number of hidden layers. The DNN consists of the maximum number of input hidden layers and an output layer to model the speech data to build ASR systems. The posterior probabilities of the tied states are modeled by training the DNN. This yielded the better performance in recognition compared to conventional GMM-HMM acoustic modeling approach. The stacking layers of the restricted Boltzmann machine are used to create the DNN. The restricted Boltzmann machine is an undirected model and is shown in Fig. 8. The model uses the single parameters set (Y) to state the joint probability variables vector (x), and hidden variables (h) through an energy E can be written as follows:

$$p(x, h; Y) = \frac{1}{Z} e^{-E(x, h; Y)} \quad (4)$$

where Z is a function of partition and is expressed as

$$Z = \sum_{x', h'} e^{-E(x', h'; Y)} \quad (5)$$

where x' and h' are the extra variables utilized for summation between the ranges of x and h . The unsupervised technique is described in detail in [25] for modeling the connection weights in deep belief networks, i.e., approximately equal to training the next pair of restricted Boltzmann machine layers. The schematic representation of context-dependent DNN-HMM hybrid architecture is shown in Fig. 9. The modeling of tied states (senones) is done by context-dependent DNN-HMM.

Fig. 8 Block diagram of restricted Boltzmann machine

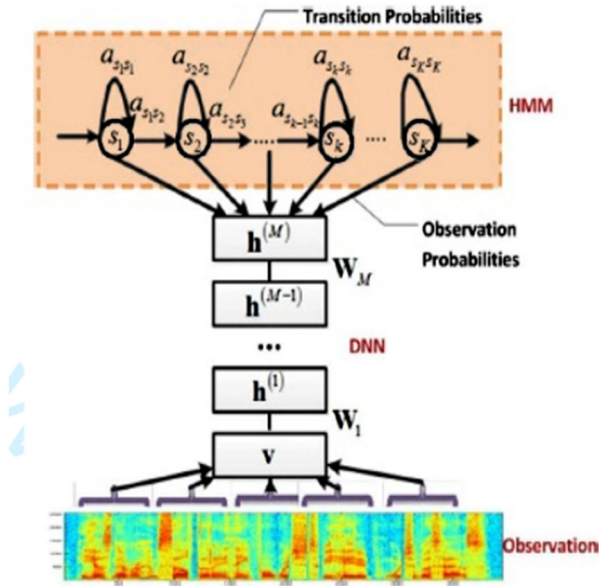
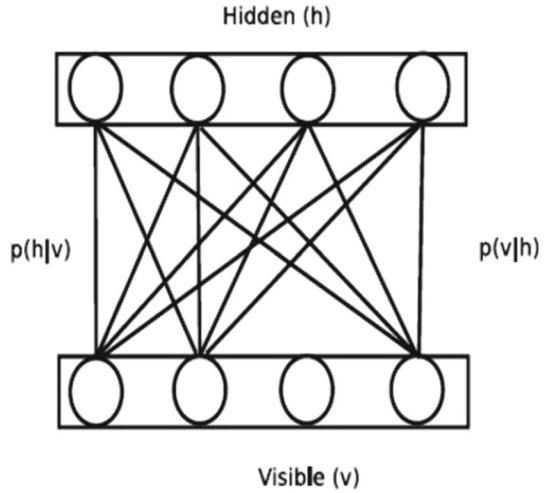


Fig. 9 Block diagram of hybrid DNN and HMM

6 An End-to-End Speech Recognition System

The reality check of the so far built continuous SR system for the Kannada language is done through by building the SQS. The SQS checks the online/real-time recognition efficiency of our continuous SR system for the Kannada language. The SQS requests an asterisk programming establishment, and it is totally accessible at Asterisk Open Source Communications (<http://www.asterisk.org/home>) on Unix/Linux plat-

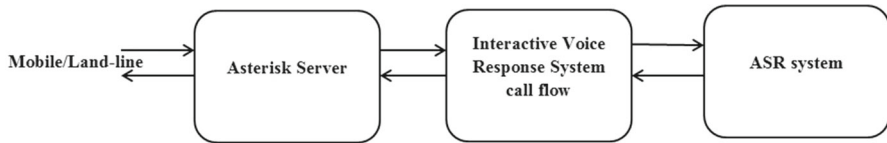


Fig. 10 The schematic of spoken query system

form that is utilized to associate the server to phone network. The asterisk server comprises of IVRS and computer system telephone interface card. The computer system telephone interface card is associated with the Integrated Services Digital Network (ISDN) and Primary Rate Interface (PRI) link line. This card is competent to help 30 synchronous time division multiplexed (TDM) channels. The asterisk server can be accessed through IP telephone, cell phone and landline through the PRI-ISDN. The essential block diagram of SQS is depicted in Fig. 10. The created framework empowers the speakers to make a query about the continuous Kannada sentences through a pre-recorded voice reaction.

6.1 Anatomy of the Call Flow for SQS

In the SQS, the client is incited to absolute Kannada sentence. On the off chance that Kannada sentence is perceived, then it will request the client to incite next Kannada sentence from our database. If that is also recognized correctly, then the system will request the client to prompt the next Kannada sentence from the database. Similarly, it will proceed until the last Kannada sentence from the database. In the event that any Kannada sentence is not perceived legitimately, this will be approached again by the client for two times. Despite the fact that on the off chance that it does not perceive, at that point, the framework says *“Too bad..! Attempt once more afterward..!”*. The schematic portrayal of SQS for recognizing continuous Kannada speech is appeared in Fig. 11.

6.2 Error recuperation Mechanisms

The error recuperation mechanisms perform a vital part in all SQS. In the created SQS, the accompanying error recuperation mechanisms are implemented.

- The voice activity detection (VAD) is utilized to identify no reactions as well as the ineffective reactions from the speakers (client). In such cases, the speakers are requested to repeat the query with much more power in the voice.
- The speaker is provoked to absolute sari/tappu (yes/no) in every phase of the decision making which depends on the recognized response.
- To boost the level of trust in the recognized response, three parallel decoders are utilized to perceive the client reaction and their yields are overviewed to build the last response.
- The three parallel decoders assist in extending the assurance level in the ASR output in contrast to the single decoder framework.

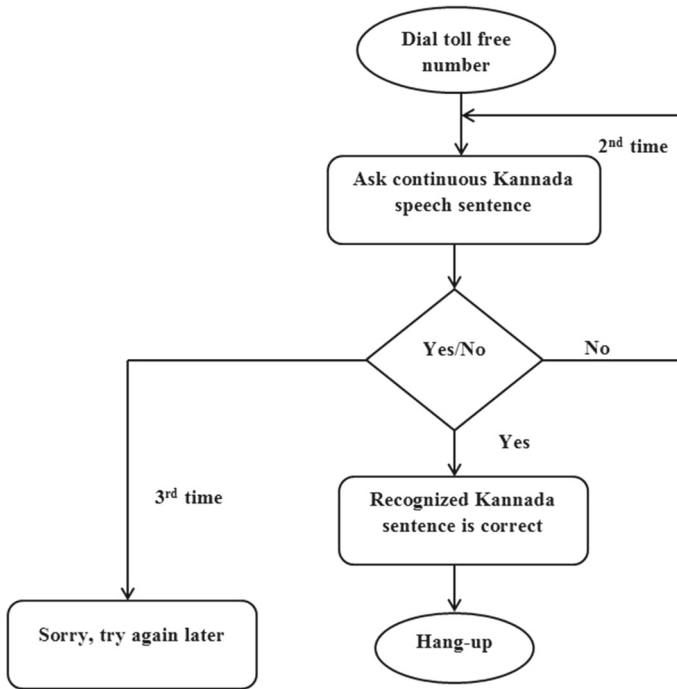


Fig. 11 The flow graph of call flow anatomy for recognizing continuous Kannada speech

7 Results and Discussion

The first and foremost thing in building the ASR system is to extract the features of the speech signal. As discussed earlier in Sect. 4.4, MFCC is the feature extraction technique used in this experiment. The parameters used for MFCC features extraction in this work are as follows:

- Window used: Hamming window.
- Window length: 20 ms.
- Pre-emphasis factor: 0.97.
- MFCC coefficients: 13 dimensional.
- Filter bank used: 21 channel filter bank.

The extracted MFCC features serve as input to the DNN input layer, and the yield of DNN is utilized with HMM, which models the sequential property of the speech. Table 7 demonstrates the speech data collected over various regions of Karnataka and the total number of speech files used for training and testing.

The guidelines used to create the LMs are as follows:

- Dictionary: The dictionary file used in Kaldi is named as lexicon.txt. It is built according to IT3-UTF:8 and ILSL12 format.
- Silence phones: The texts “sil” and “SIL” were termed as silence phones.
- Optional silence phone: The text “sil” is termed as an optional silence phone.

Table 7 Statistics of the speech files in building continuous speech recognition in Kannada language

Regions of the Karnataka state	Number of speakers		Total
	Male speakers	Female speakers	
Northern Karnataka region	300	200	500
Central Karnataka region	330	220	550
Southern Karnataka region	390	260	650
Coastal (Karavali) Karnataka region	420	280	700
Total number of speakers from the entire state of Karnataka	1440	960	2400
Total number of speech files used for training	67,200		
Total number of speech files used for testing	16,800		

- Non-silence phones: 168 non-silence phones were used to build the lexicon and the LMs.

The guidelines used to create the AMs are as follows:

- 500 leaves were used for triphone1.
- 2000 Gaussians were used for triphone1.
- 500 leaves are used for MLLT.
- 2000 Gaussians were used for MLLT.
- 500 leaves were used for SAT.
- 2000 Gaussians were used for SAT.
- 200 Gaussians were used for universal background model (UBM).
- 3000 leaves were used for SGMM.
- 3000 Gaussians were used for SGMM.
- 3 jobs were used for training and decoding.
- 2 hidden layers were used for DNN.

The AMs produced at various phoneme levels are as follows:

1. MonoPhone Training and Decoding
2. tri1: Deltas + Delta-Deltas Training and Decoding
3. tri2: LDA + MLLT Training and Decoding
4. tri3: LDA + MLLT + SAT Training and Decoding
5. SGMM Training and Decoding
6. DNN Hybrid Training and Decoding (DNN+HMM)
7. System Combination (DNN+SGMM) with iterations: 1, 2, 3, 4.
8. SGMM + MMI Training and Decoding with iterations: 1, 2, 3, 4.

The 600 senones and 4, 8 and 16 Gaussian mixtures were utilized to build the LMs and the AMs. Table 8 demonstrates the diverse WERs obtained at various phoneme levels. It was seen in the table that tri1 with 600 senones and 16 Gaussian mixtures (tri1_600_9600) has given minimum WER of 5.09%. Tri2 with 600 senones and 16 Gaussian mixtures (tri2_600_9600) has given minimum WER of 5.80%. Tri3 has a minimum WER of 5.03% for 600 senones and 16 Gaussian mixtures. Among tri1, tri2 and tri3, tri3 has the least WER for the given continuous speech data in Kannada

Table 8 The portrayal of WER at various phoneme levels for continuous Kannada speech database

Phonemes	WER_1	WER_2	WER_3	WER_4	WER_5	WER_6	WER_7	WER_8
mono	7.85	7.69	7.66	7.89	7.72	7.89	7.22	8.09
tri1_600_2400	6.32	6.29	6.45	6.54	6.59	6.78	6.31	6.68
tri1_600_4800	5.73	5.69	5.85	5.74	5.89	5.73	5.69	5.63
tri1_600_9600	5.23	5.45	5.61	5.33	5.48	5.61	5.12	5.09
tri2_600_2400	7.23	7.45	7.12	7.09	7.18	7.29	7.85	7.00
tri2_600_4800	6.14	5.96	5.98	6.06	6.18	5.99	6.25	5.97
tri2_600_9600	5.89	5.98	5.85	6.12	6.04	5.96	5.80	6.12
tri3_600_2400	6.20	6.59	6.25	6.13	6.48	6.13	6.18	6.24
tri3_600_4800	5.72	5.70	5.83	5.98	5.77	5.86	5.83	5.69
tri3_600_9600	5.42	5.04	5.15	5.32	5.03	5.86	5.47	5.19
SGMM	5.04	4.65	5.12	4.89	4.84	4.79	4.67	5.01
SGMM+MMI_it1	4.82	4.79	4.68	5.00	4.69	5.03	4.82	4.60
SGMM+MMI_it2	4.77	4.89	5.21	4.89	4.77	5.24	4.72	4.70
SGMM+MMI_it3	4.81	4.60	4.88	4.69	5.21	4.66	4.71	5.02
SGMM+MMI_it4	4.99	4.67	4.85	4.83	4.85	5.05	5.01	4.65
DNN+HMM	4.24	4.10	4.27	4.25	4.27	4.54	4.42	4.24
DNN+SGMM_it1	4.30	4.25	4.30	4.21	4.63	4.69	4.35	4.25
DNN+SGMM_it2	4.55	4.53	4.85	4.89	4.66	4.21	4.90	4.89
DNN+SGMM_it3	4.72	5.01	4.28	4.33	4.72	4.71	4.27	4.33
DNN+SGMM_it4	4.55	5.00	4.27	4.65	4.85	4.53	4.53	4.89

Bold values signifies the least word error rate (WER) among other WER in the particular row

language and thus proved the best among them. The table also shows the results for SGMM whose WER is 4.65%. The hybrid combination of (DNN+HMM) has the WER of 4.10%, and the WER corresponding to (DNN+SGMM) is 4.21%. At last, the combination of MMI + SGMM technique gives the WER of 4.60%. From the table, it is found that the hybrid combination of (DNN+HMM) has given a superior exactness contrasted and different models. The minimum WER models could be utilized as a part of SQS.

7.1 Testing of the Developed End-to-End Speech Recognition from the Speakers Across the State of Karnataka

The accuracy of the SQS developed is checked the online for SR accuracy of continuous SR system. For this purpose, the 500 speakers across the state of Karnataka are asked to test the system under an uncontrolled environment. Table 9 shows the performance evaluation of the developed SQS by the speakers from the state of Karnataka. It was observed in the table that there is a much improvement in online SR accuracy with less failure of recognizing the speech utterances compared to the earlier spoken query system. Therefore, it can be inferred that the online and offline (WERs of models) recognition rates are almost the same as shown in Table 9.

Table 9 Execution assessment of online speech recognition exactness testing by speakers in the real-world environment

Total number of Kannada sentences	40
Total number of speakers	500
First attempt	430
Second attempt	32
Third attempt	14
Total number of recognition	476
Recognition rate in percentage	95.20

8 Conclusion

A continuous Kannada speech recognition system was demonstrated in this work. The IVRS call flow structure was developed for speech data collection. The speech data were collected, transcribed and validated using the transcriber tool. The ASR models were built by using Kaldi speech recognition toolkit. All possible alternate pronunciations for particular Kannada speech sentence were included in the lexicon. By using Kaldi and Kannada dialect assets, the accomplished WERs for monophone, triphone1, triphone2, triphone3, SGMM, combination of SGMM and MMI, combination of DNN and HMM, combination of DNN and SGMM are 7.66%, 5.09%, 5.80%, 5.03%, 4.65%, 4.60%, 4.10%, 4.21%, respectively. The least WER for continuous Kannada speech data was achieved by SGMM and hybrid DNN- based modeling techniques. These least WER models (SGMM- and DNN- based models) could be utilized to develop robust ASR system. The developed ASR system was tested by different speakers under degraded conditions and compared the offline and online speech recognition accuracy. The recognition rate of the ASR system in both offline and online mode is almost the same.

References

1. Z. Ansari, S.A. Seyedsalehi, Toward growing modular deep neural networks for continuous speech recognition. *Int. J. Neural Comput. Appl.* **28**(1), 1177–1196 (2017)
2. S.S. Bharali, S.K. Kalita, Speech recognition with reference to Assamese language using novel fusion technique. *Int. J. Speech Technol.* **21**(2), 251–263 (2018)
3. C. Chelba, T. Brants, W. Neveitt, P. Xu, Study on interaction between entropy pruning and kneser-ney smoothing, in *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH)* (2010), pp. 2422–2425
4. S. Chen, J. Goodman, An empirical study of smoothing techniques for language modeling, in *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics* (1996), pp. 310–318
5. D. Dimitriadis, E. Bocchieri, Use of micro-modulation features in large vocabulary continuous speech recognition tasks. *IEEE/ACM Trans. Audio Speech Lang. Process.* **23**(8), 102–114 (2015)
6. S. Ganapathy, Multivariate autoregressive spectrogram modeling for noisy speech recognition. *IEEE Signal Process. Lett.* **24**(9), 1373–1377 (2017)
7. G. Garau, S. Renals, Template-based continuous speech recognition. *IEEE Trans. Audio Speech Lang. Process.* **16**(3), 508–518 (2008)
8. G.E. Hinton, S. Osindero, Y.W. Teh, A fast learning algorithm for deep belief nets. *Neural Comput.* **18**(7), 1527–1554 (2006)

9. S. Katz, Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Trans. Acoust. Speech Signal Process.* **35**(3), 400–401 (1987)
10. I.S. Kipyatkova, A.A. Karpov, A study of neural network russian language models for automatic continuous speech recognition systems. *Int. J. Autom. Remote Control* **78**(5), 858–867 (2017)
11. L. Lu, S. Renals, Small-footprint highway deep neural networks for speech recognition. *IEEE/ACM Trans. Audio Speech Lang. Process.* **25**(7), 1502–1511 (2017)
12. T. Ma, S. Srinivasan, G. Lazarou, J. Picone, Continuous speech recognition using linear dynamic models. *Int. J. Speech Technol.* **17**(1), 11–16 (2014)
13. K.M.O. Nahar, M.A. Shquier, W.G. Al-Khatib, H. Muhtaseb, M. Elshafei, Arabic phonemes recognition using hybrid LVQ/HMM model for continuous speech recognition. *Int. J. Speech Technol.* **19**(3), 495–508 (2016)
14. D. Palaz, M. Doss, R. Collobert, Convolutional neural networks-based continuous speech recognition using raw speech signal, in *Proceedings of 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brisbane, Australia (2015), pp. 4295–4299
15. B. Popovic, S. Ostrogonac, E. Pakoci, N. Jakovljevic, V. Delic, Deep neural network based continuous speech recognition for serbian using the kaldi toolkit, in *Proceedings 17th International Conference on Speech and Computer (SPECOM)*, Athens, Greece (2015), pp. 186–192
16. D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlce, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, K. Vesely, The Kaldi speech recognition toolkit, in *Proceedings IEEE 2011 Workshop on Automatic speech recognition and Understanding*, Hilton Waikoloa Village, Big Island, Hawaii, US (2011)
17. D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlce, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, K. Vesely, The subspace gaussian mixture model-a structured model for speech recognition. *Int. J. Comput. Speech Lang.* **25**(2), 404–439 (2011)
18. L. Rabiner, B.H. Juang, *Fundamentals of Speech Recognition* (Prentice-Hall, Inc, Upper Saddle River, 1993)
19. L. Rabiner, Applications of voice processing to telecommunications. *Proc. IEEE* **82**, 199–228 (1994)
20. S. Ravuri, Hybrid DNN-latent structured SVM acoustic models for continuous speech recognition, in *Proceedings of 015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Scottsdale, Arizona, USA (2015), pp. 37–44
21. R.C. Rose, S.C. Yin, Y. Tang, An investigation of subspace modeling for phonetic and speaker variability in automatic speech recognition, in *Proceedings ICASSP* (2011), pp. 4508–4511
22. H.B. Sailor, H.A. Patil, A novel unsupervised auditory filterbank learning using convolutional RBM for speech recognition. *IEEE/ACM Trans. Audio Speech Lang. Process.* **24**(12), 2341–2353 (2016)
23. H. Sameti, H. Veisi, M. Bahrani, B. Babaali, K. Hosseinzadeh, A large vocabulary continuous speech recognition system for Persian language. *EURASIP J.* **2011**(1), Art. No. 6 (2011). <https://doi.org/10.1186/1687-4722-2011-426795>
24. J. Sangeetha, S. Jothilakshmi, An efficient continuous speech recognition system for Dravidian languages using support vector machine, in *Proceedings of the Artificial Intelligence and Evolutionary Algorithms in Engineering Systems (ICAEES)*, New Delhi, India (2015), pp. 359–367
25. M.V. Segbroeck, H.V. Hamme, Advances in missing feature techniques for robust large-vocabulary continuous speech recognition. *IEEE Trans. Speech Audio Process.* **19**(1), 123–137 (2011)
26. P. Sharma, V. Abrol, A.K. Sao, Deep-sparse-representation-based features for speech recognition. *IEEE/ACM Trans. Audio Speech Lang. Process.* **25**(11), 2162–2175 (2017)
27. S.M. Siniscalchi, D. Yu, L. Deng, C.H. Lee, Speech recognition using long-span temporal patterns in a deep network model. *IEEE Signal Process. Lett.* **20**(3), 201–204 (2013)
28. A. Stolcke, SRILM: An extensible language modeling toolkit, in *Proceedings of the 7th International Conference on spoken language processing (ICSLP 2002)* (2002), pp. 901–904
29. R. Su, X. Liu, L. Wang, Automatic complexity control of generalized variable parameter HMMs for noise robust speech recognition. *IEEE Trans. Speech Audio Process.* **23**(1), 102–114 (2015)
30. F. Triefenbach, K. Demuynck, J.P. Martens, Large vocabulary continuous speech recognition with reservoir-based acoustic models. *IEEE Signal Process. Lett.* **21**(3), 311–315 (2014)
31. M.D. Wachter, M. Matton, K. Demuynck, P. Wambacq, R. Cools, D.V. Compennolle, Template-based continuous speech recognition. *IEEE Trans. Speech Audio Process.* **15**(4), 1377–1389 (2007)
32. E. Zarrouk, Y. Benayed, F. Gargouri, Graphical models for the recognition of Arabic continuous speech based triphones modeling, in *Proceedings of 2015 IEEE/ACIS 16th International Conference on*

Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Takamatsu, Japan (2015), pp. 1–6

33. Y. Zhao, A speaker-independent continuous speech recognition system using continuous mixture Gaussian density HMM of phoneme-sized units. *IEEE Trans. Speech Audio Process.* **1**(3), 345–361 (1993)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

P. S. Praveen Kumar¹ · G. Thimmaraja Yadava¹ · H. S. Jayanna²

P. S. Praveen Kumar
pravin227@gmail.com

H. S. Jayanna
jayannahs@gmail.com

¹ Department of Electronics and Communication Engineering, Siddaganga Institute of Technology, Tumkur, Karnataka, India

² Department of Information Science and Engineering, Siddaganga Institute of Technology, Tumkur, Karnataka, India