

Acoustic Modeling for Development of Accented Indian English ASR

Partho Mandal, Gaurav Ojha, Anupam Shukla and S.S. Agrawal

Abstract This paper investigates Indian English from the point of view of a speech recognition problem. A novel approach towards building an Automated Speech Recognition System (ASR) for Indian English using PocketSphinx has been proposed. The system was trained with a database of English words spoken by Indians in three different accents using continuous as well as semi-continuous models. We have compared the performances in each case and the optimum case performance comes close to 98 % accurate. Based on this study, we tweaked the original PocketSphinx Android application in order to incorporate our results and present it as an Indian English-based SMS sending application. We are working further on this approach to identify ways of successfully training a speech recognition system to recognize a much wider variety of Indian accents with much more significant accuracy.

Keywords Automatic speech recognition · Indian English · Discrete HMMs

1 Introduction

Nowadays, the application of information processing machines have become widespread. From desktop computers to handheld devices and even web applications can do a fair bit of speech processing. But most of this speech recognition technology has become more of an adjust-to-available-options rather than speak-however-you-feel-convenient. Speech recognition technology currently supports English spoken in a rather peculiar way. This leaves a lot of people still fumbling to make the device understand what they are speaking [1]. Given that

P. Mandal · G. Ojha (✉) · A. Shukla
Department of Information Technology, ABV-IIITM, Gwalior, India
e-mail: ojha.iiitm@gmail.com

S.S. Agrawal
KIIT Group of Colleges, Gurgaon, India

English is an official language of India, and over 200 million English speakers exist in India, it becomes essential to modify the speech recognition systems accordingly rather than try to adapt to the ones currently in use which support British or American accents more comfortably. We are highly motivated by this and wish to build something that caters to the need of Indians since they form the second largest country of English-speaking people [2].

Many institutions are working on making speech recognition more and more effortless for people across the globe and we have attempted to do our bit in this research. In this project, we are working on developing a system that performs better speech recognition for Indian accents. Our approach involves studying outputs of various methods and trying to find the optimal approach for training by simply varying the inputs till we get to the point where results could not be further optimized.

2 Previous Work

Quite a few studies have been done in this field of customized speech recognition for regional accents and languages.

One of the initial studies in this context [3, 4] provides a good idea about how speech recognition works and how to get an automatic speech recognition (ASR) system up from scratch. They give a lot of relevant information regarding the working of a speech recognition system and two models for the same, viz., acoustic model and language model. Relevant work in this scenario has been done in [5] and a lot more can still be done.

Researchers at Carnegie Mellon University have developed PocketSphinx [6] which proves to be a boon for speech recognition systems especially in handheld devices. PocketSphinx happens to be the first such system and it comes with an open-source license. They had used a 1000-word vocabulary system which worked quite well on a handheld device operating at 206 MHz. PocketSphinx is revolutionary in that it is nearly real time and several times faster than the baseline system under consideration.

Further work in ASR for accented Indian English has been done at Siemens collaboratively at Bangalore and Germany [7]. The results show the effect of variability in accents on the performance of a system processing Indian English and are quite impressive. The test vocabularies were trained using Hidden Markov Models (HMMs) specifically on accent-specific data. The study suggests that the first task should be to identify the accent and then use the selected accent for further speech recognition. This approach maybe effective but the primary task, viz., identifying the accent still remains somewhat ambiguously mysterious.

In a study contemporary with that done at Siemens, researchers at Carnegie Mellon University proposed that currently available speech recognition systems can be improved using additional data [8] which can be used to create new duration and

pronunciation models. They have essentially tried to build a better sounding Indian English voice with some additional speech data over the existing systems. The technique can prove to be effective but that is something yet to be examined.

A more recent study in the domain has been done on the phonetic segmentation of Hindi speech using HMMs [9]. The system mimics the way humans understand and identify spoken words. The study concludes that best performance was obtainable through a combination of two different Gaussian mixtures and five HMM states. Although there are certain errors in the automatic segmentation process especially concerning some consonants, the study throws light on the fact that it is possible to work on the idea to develop more friendly recognition systems which can outperform the current ones. On similar lines, [10] attempts to highlight the significance of appropriate selection of Gaussian mixtures toward improving the results of a speech recognition problem. The experimental results of the study have proven an improvement of 3–4 % over the current recognition rates due to the inclusion of a third-order derivative of the speech features. Further it states that for a medium-sized vocabulary of about 600 words, the system required 8 Gaussian components to give the optimal results.

In this paper, we have used the CMU PocketSphinx and trained an Indian English database spoken by speakers of various dialects using the continuous as well as the semi-continuous models by changing various parameters and comparing their performance.

3 Proposed Methodology

We start by analyzing the ASR architecture and the procedure of speech analysis and feature extraction. After that we discuss the Continuous Density Hidden Markov Model (CDHMM) for speech recognition and then we state our experiments.

3.1 ASR Architecture

A conventional ASR system is made up of four primary structures which comprise of the following—a. Speech analysis and feature extraction, b. Feature reduction, c. Phonetic transcription, and d. Acoustic model. Figure 1 explains this structure.

3.1.1 Speech Analysis and Feature Extraction

The module for feature extraction computes the various acoustic feature vectors that are used to state the speech signal. Empirical studies in speech analysis [11] have shown that since a speech signal evolves from the lungs with the help of the vocal

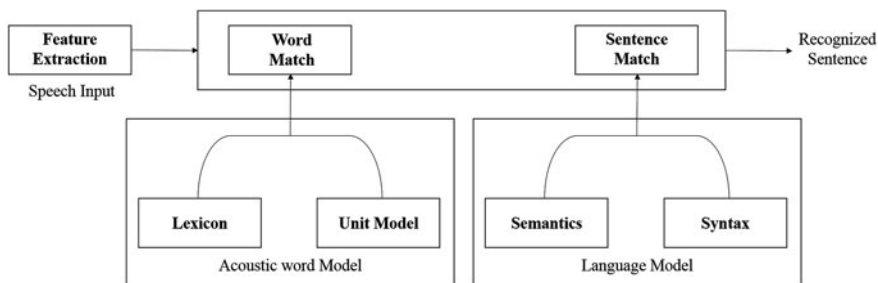


Fig. 1 Working of a continuous speech recognizer

tract, it is best to discriminate between sounds based on the response of the vocal tract component. It has also been observed that including the psychophysical process attributes of how humans perceive speech improves the accuracy of recognition. We have used the mel frequency cepstral coefficients (MFCC) which is a widely used feature based on how humans hear and perceive pitch. The samples of speech are divided into several overlapping frames of small durations, usually 20–30 ms and the frame rate is 10 ms. Hamming window is then used and the log magnitude of each frame is calculated. In order to simulate the human auditory system, the subject spectrum is filtered using triangular band-pass filters which are based on the mel scale. This gives the vector of log energies, a relation which is governed by the following equation where f is the normal frequency and $\text{mel}(f)$ is its counterpart on the mel scale:

$$\text{mel}(f) = 2595 \times \log_{10} \left(1 + \frac{f}{700} \right) \quad (1)$$

3.1.2 Feature Reduction

Feature reduction refers to the methods and techniques which utilize mathematical and statistical tools to reduce the complexity of features while attempting to retain as much information as possible [12]. Majority of these methods rely on schemes involving linear transformation such as the Linear Discriminate Analysis (LDA) and Maximum Likelihood Linear Transform (MLLT). LDA focuses on much more significant differences in variance along a specific direction which helps achieve an optimum value of the probability that all training data in the transformed space and every training sample is an equal contributor to the final objective function. MLLT supposes one transformation matrix to be tied with a group of covariance matrices (CM) belonging to (an) individual state(s) of a HMM. It introduces a new form of a CM which allows sharing a few full CM over many distributions [13].

3.1.3 Phonetic Transcription

The next task is to write down a phonetic structure of the recorded signal. This includes whole word models, syllable-based models, and phone-based models. The former is the simplest of all and often yields good results for small databases which are usually utilized in closed vocabulary systems. Phone-based models are complex but are more flexible and are called context-independent (CI) models. Each phoneme can have one of many possible acoustic realizations which is context-dependent (CD). For a properly trained system, it is essential to take into consideration this co-articulation effect as well. Senones are generated to model these CD sub-word units, which are combinations of phonemes with their preceding and succeeding phonemes. These senones are then represented using the HMMs. In our study, we have used the Arpabet [14] phonetic transcription code.

3.1.4 Acoustic Model

During this modeling process, HMMs which represent the basic phonetic units of the training data are created which are free from quantization errors and model continuous signal representation in a much better and useful way than other techniques. For each model it is necessary to evaluate the highest likelihood of the observation sequence. This is done using the forward algorithm. Given a set of feature vectors, it is possible to find the highest likely word sequence by mapping all possible state sequences that could be generated from all possible word sequences. This can be done using the Baum–Welch algorithm. Finally, the forward–backward algorithm is used to solve the learning problem. In this, the HMM corresponding to the sequence of words in each spoken sentence is found out and another HMM is generated which represents the statement. The system then utilizes the results from the language model to understand the sequence of words which is most likely to occur. In the end, the recognizer takes the final decision as to what word was spoke by taking into consideration the results given out by the previous blocks, viz., the word sequences which are likely and ones which have the highest matching score.

3.2 CDHMM and SCHMM

The continuous density HMMs come into picture here and are based on some probability density functions (PDFs). The multivariate Gaussian mixture density function has been employed for the requirement of a probability density function in our analysis. It is a parametric probability density function represented as a weighted sum of Gaussian component density [15] which can be expressed using the following equation:

$$p(x|\gamma) = \sum_{i=1}^M w_i g(x|\mu_i, \Sigma i) \quad (2)$$

Here, γ is a D-dimensional continuous feature vector, w_i where $i = 1 \dots M$ are the mixture weights and $g(x|\mu_i, \Sigma i)$ represents component Gaussian densities from $i = 1 \dots M$. The component densities are D-variate Gaussian functions which can be represented by the following equation:

$$g(x|\mu_i, \Sigma i) = \frac{e^{-\left(\frac{1}{2(x - \mu_i)' \Sigma_i^{-1} (x - \mu_i)}\right)}}{\sqrt{(2\pi)^D |\Sigma i|}} \quad (3)$$

Here μ_i represents the mean, Σi represents the covariance and $\sum_{i=1}^M w_i = 1$ represents the constraint on the mixture weight where covariance can be full or diagonal. To initialize the estimation of the GMM parameters, maximum a posteriori parameter estimation (MAP) is used. The number of mixture components per state which is given by M forms an important parameter for the overall performance of ASR.

The semi-continuous HMM models, a mixture of Gaussians is shared by all the HMM state densities with different mixture weights assigned to them. The weights are further shared by the senones in these models [16].

The likelihood that state s would be observed for frame \bar{X} is represented by the following:

$$L_s(\bar{X}) = \sum_{i=1}^N \lambda_{s_i} \times N_i(\bar{X}) \quad (4)$$

Wherein N is the size of the Gaussian set, λ_{s_i} is the weight for the i th Gaussian in the state s and $N_i(\bar{X})$ represents the likelihood of that Gaussian [17].

4 Results and Discussion

We have used an Indian English database with three speakers (D1, D2, and D3) for our ASR system. Each speaker data consists of 620 sentences. A short description of these speakers is as follows:

D1: The speaker is female, native language is Hindi. It consists of a vocabulary of 1267 words.

D2: The speaker is female, native language is Kannada. It consists of a vocabulary of 1228 words.

D3: The speaker is male, native language is Punjabi. It consists of a vocabulary of 1300 words.

The speech utterances were recorded in 16 kHz through two recording channels: a headset and a desktop-mounted microphone. The speech data was recorded in 16 kHz stereo 16-bit format.

The recognition system is an HMM-based speaker dependent speech recogniser. We have used three—state HMMs for each model, these models are left to right with no skip state. For all the speakers, n-gram statistical language model was used with a language weight of 11.5. We tried experimenting with both open set models and closed set models.

For the open set, we used 80 % of the data, i.e., 496 sentences as training data, and 20 %, i.e., 124 sentences as testing data. The acoustic model uses HMMs as explained earlier. We trained the data using both continuous (Table 1) as well as semi-continuous (Table 2) model.

For the closed set, we used all the 620 sentences as training data, and 20 %, i.e., 124 sentences as testing data for each of the speaker utterances. HMMs were used and the data was trained using both continuous (Table 3) as well as semi-continuous (Table 4) models.

In case of the Continuous model, the input features consisted of a single independent stream “1s_c_d_dd”. The initial tied Gaussian density was 1. The initial tied Gaussian density should be less than final tied Gaussian density. The data was trained using both 8 as well as 16 tied Gaussian densities. The accuracy was tested by comparing the data trained using different tied Gaussian Mixture Models (Senones).

Table 1 Error rates of all speaker utterances in open set continuous model

Database name	Error (%)	Senones: tied Gaussian density				
		70:8	100:8	100:16	120:8	120:16
D1	SER	37.1	37.1	27.4	40.3	32.3
	WER	7.5	7.5	5.4	8.9	6.5
D2	SER	36.3	36.3	31.5	35.5	33.1
	WER	7.2	7.2	5.2	7.7	6.2
D3	SER	29.0	29.0	27.4	30.6	28.2
	WER	3.9	3.9	3.5	4.3	3.9

Table 2 Error rates in the open set semi-continuous model

Database name	Error (%)	Senones: tied Gaussian density	
		100:128	100:256
D1	SER	31.5	24.2
	WER	7.2	5.2
D2	SER	29.0	28.2
	WER	5.8	5.3
D3	SER	30.6	27.4
	WER	5.2	4.6

Table 3 Error rates of all speaker utterances in closed set continuous model

Database name	Error (%)	Senones: tied Gaussian density	
		100:8	100:16
D1	SER	35.0	20.2
	WER	5.5	3.4
D2	SER	37.1	25.0
	WER	7.4	4.5
D3	SER	25.8	12.1
	WER	3.2	1.5

Table 4 Error rates in the closed set semi-continuous model

Database name	Error (%)	Senones: tied Gaussian density	
		100:128	100:256
D1	SER	22.6	21.7
	WER	4.9	4.2
D2	SER	23.4	20.2
	WER	3.5	2.8
D3	SER	21.0	17.7
	WER	2.7	2.2

In case of the semi-continuous model, the input features consisted of feature vector with four independent streams “s2_4x”. The initial tied Gaussian density is same as final tied Gaussian density. The data was trained using both 128 as well as 256 tied Gaussian densities and accuracy was tested by comparing data trained using different Senones.

It was observed that the optimal number of senones to be used for training data should be 100. As it can be seen that in each database the result of data trained using 70 senones and 100 senones gave the same accuracy, senones less than 100 were not used. The more senones the model has, the more precisely it discriminates the sounds. But on the other hand if we have too many senones, it may not be generic enough to recognize any unseen (new) speech. That means that the WER will be higher on unseen data. That is why it is important to not overtrain the models. This is evident from experiments—as the number of senones is increased from 100 to 120, both SER as well as WER increase. Formula used for calculating the recognition score is as follows:

$$\frac{\text{No. of Recognized Words/Sentences}}{\text{Total no. of Words/Sentences in the Testing Set}} \times 100 \quad (5)$$

It was also observed that more the number of final tied Gaussian density, better is the accuracy. Usually, 8–32 densities are used for each mixture in a typical CDHMM system but in the experiments they have been limited to 16, as increasing

the number of densities also increases the computational time. Another interesting feature observed is that the accuracy of the male speaker is better than that of the female speakers. Here, the data has been considered as having 100 senones and 16 tied Gaussian states.

It has been observed that the optimal number of senones to be used for training data should be 100 in continuous model. So we took the number of senones as 100 for semi-continuous model as well. It has also been observed that in the continuous model, greater number of final tied Gaussian density is better for accuracy. The number of mixture Gaussians is usually 128–2048 in SCHMM system, which is much larger than the 16–32 densities used for each mixture in a typical CDHMM system. The number of Gaussian densities was thus limited to 256, as increasing the number of densities also increases the computational time. Here, the data has been considered as having 100 senones and 256 tied Gaussian states.

Table 5 shows the overall comparison between the models discussed earlier. These are error % change b/w 8 tied Gaussian states and 16 tied Gaussian states in continuous and b/w 128 tied Gaussian states and 256 tied Gaussian states in semi-continuous when the number of senones is 100. The summary of overall database accuracy is shown in Table 6.

Experiments were repeated with inclusion of LDA/MLLT along with MFCC. LDA/MLLT was utilized to improve the accuracy of the results and those observations were recorded as well. The LDA dimension used is 32. For semi-continuous models, LDA/MLLT feature transform is not supported in PocketSphinx. The data of MFCC+LDA/MLLT is summarized in Table 7.

Table 5 Overall comparison of tested models

DB/model	D1		D2		D3	
	Δ SER	Δ WER	Δ SER	Δ WER	Δ SER	Δ WER
O/C	26.14	28.00	13.22	27.77	5.54	10.23
O/S-C	23.17	27.77	2.76	8.62	10.46	11.54
C/C	42.28	38.18	32.61	39.18	53.11	53.12
C/S-C	3.98	14.28	13.67	20.0	15.71	18.52

Table 6 Overall best % accuracy for each database using MFCC

DB/model	D1		D2		D3	
	S-A	W-A	S-A	W-A	S-A	W-A
O/C	72.6	94.6	68.5	94.8	73.4	96.5
O/S-C	75.8	94.8	71.8	94.7	72.6	95.4
C/C	79.8	96.6	75.0	95.5	87.9	98.5
C/S-C	78.3	95.8	79.8	97.2	82.3	97.8

Table 7 Best % accuracy using MFCC+LDA/MLLT

DB/model	D1		D2		D3	
	S-A	W-A	S-A	W-A	S-A	W-A
O/C	78.9	95.9	76.4	96.1	79.5	97.3
C/C	84.6	97.4	81.2	96.6	90.9	98.3

5 Conclusion and Future Work

It can be concluded that in terms of accent, the Punjabi accented HMMs (i.e., database D3) gives the best performance compared to other two accented HMMs. The percentage change in error is higher in continuous models when data is trained from 8 tied Gaussian states to 16 tied Gaussian states than in semi-continuous models when data is trained from 128 tied Gaussian states to 256 tied Gaussian states. Better improvement in terms of change in percentage error is seen in the case of continuous models based on closed data set (SER in the range of 32–56 %) as compared to continuous models based on open data set (SER in the range of 5–26 %). Overall, the optimum number of senones for each of the database is found to be 100 and the optimal number of final tied Gaussian for continuous model chosen is 16. For semi-continuous model, the number of final tied Gaussian is found to be 256.

It can also be deduced that the accuracy of a male speaker is overall better than that of female speakers, with exception in semi-continuous model in open set data where SER of female speaker is more than male speakers. Also, on an average, the overall accuracy of closed set data is better than the accuracy of open set data. The sentence accuracy of closed set data is 7–15 % more than open set data in case of continuous model and 3–10 % more than open set data in case of semi-continuous model. The word accuracy of closed set data is 0.7–2 % more than open set data in case of continuous model and 1–3.5 % more than open set data in case of semi-continuous model. It was observed that LDA/MLLT could bring about nearly a 6–8 % improvement in sentence accuracy and 1–2 % in word accuracy. LDA/MLLT brought more improvement in data sets that had less sentence and word accuracy compared to data sets that had more sentence and word accuracy.

Based on this study, we tweaked the original PocketSphinx Android application in order to incorporate our results and present it as an Indian English-based SMS sending application [18].

These are only the initial results and more work needs to be done in order to completely implement these ideas in practical applications.

References

1. Discussion Forum about Siri on the official website of Apple Inc. <https://discussions.apple.com/thread/3390280?tstart=0>.
2. List of Countries by English Speaking Population—Wikipedia. http://en.wikipedia.org/wiki/List_of_countries_by_English-speaking_population.

3. Samudravijaya K. Automatic Speech Recognition. Tata Institute of Fundamental Research Archives. 2004.
4. Samudravijaya K. Speech and speaker recognition—a tutorial. Tata Institute of Fundamental Research Archives. 2004.
5. Samudravijaya K, Rao PVS, Agrawal SS. Hindi speech database. In: the Proceedings of the International Conference on Spoken Language Processing ICSLP00, Beijing, 2000; CDROM: 00192.pdf.
6. Huggins-Daines D, Kumar M, Chan A, Black AW, Ravishankar M, Rudnicky AI. Pocketsphinx: a free, real-time continuous speech recognition system for hand-held devices. In: The proceedings of the IEEE international conference on acoustics, speech and signal processing (ICASSP), France, 2006.
7. Kulkarni K, Sengupta S, Ramasubramanian V, Bauer JG, Stemmer G. Accented Indian english ASR: some early results. In: The proceedings of the IEEE spoken language technology workshop, India, 2008.
8. Kumar R, Gangadharaiiah R, Rao S, Prahallad K, Rosé CP, Black AW. Building a better Indian english voice using ‘more data’. In: The proceedings of the 6th ISCA workshop on speech synthesis, Germany, 2007.
9. Balyan A, Agrawal SS, Dev A. Automatic phonetic segmentation of Hindi speech using hidden Markov model 27:543–549, AI & Soc, Springer: London; 2012.
10. Sinha, S, Agrawal, SS, Jain, A. Continuous density hidden markov model for context dependent hindi speech recognition. In: The proceedings of the international conference on advances in computing, communication and informatics (ICACCI), India, 2013.
11. Picone J. Signal modeling techniques in speech recognition. In: Proceedings of the IEEE international conference, June 1993.
12. Geirhofer S. Feature reduction with linear discriminant analysis and its performance on phoneme recognition. Department of Electrical and Computer Engineering: University of Illinois at Urbana-Champaign; 2004.
13. Psutka JV. Benefit of maximum likelihood linear transform (MLLT) used at different levels of covariance matrices clustering in ASR systems., Lecture Notes in Computer ScienceBerlin Heidelberg: Springer; 2007.
14. Arpabet. <http://en.wikipedia.org/wiki/Arpabet>.
15. Reynolds DA. A Gaussian mixture modeling approach to text-independent speaker identification. Ph.D. thesis, Georgia Institute of Technology, 1992.
16. Raux A, Singh R. Maximum-likelihood adaptation of semi-continuous HMMs by latent variable decomposition of state distributions. In: The proceedings of the 8th international conference on spoken language processing (ICSLP), South Korea, 2004.
17. Duchateau J, Demuynck K, Van Compernelle D. Fast and accurate acoustic modelling with semi-continuous HMMs. Speech Commun. 1998;24(1):5–17.
18. Indian English SMS Sending App—PocketSphinx Derivative. <https://github.com/parthoiiitm/smsforindeng>.