

# Hidden Markov Model for Speech Recognition System—A Pilot Study and a Naive Approach for Speech-To-Text Model

S. Rashmi, M. Hanumanthappa and Mallamma V. Reddy

**Abstract** Today's advancement in the research field has brought a new horizon to design the state-of-the-art systems that produce sound utterance. In order to attain a higher level of speech understanding potentiality, it is of utmost importance to achieve good efficiency. Speech-to-Text (STT) or voice recognition system is an efficacious approach that aims at recognizing speech and allows the conversion of the human voice into the text. By this, an interface between the human and the computer is created. In this direction, this paper introduces a novel approach to convert STT by using Hidden Markov Model (HMM). HMM along with other techniques such as Mel-Frequency Cepstral Coefficients (MFCCs), Decision trees, Support Vector Machine (SVM) is used to ascertain the speakers' utterances and catalyse these utterances into quantization features by evaluating the likelihood extremity of the spoken word. The accuracy of the proposed architecture is studied, which is found to be better than the existing methodologies.

**Keywords** Bayesian classifier · Context-dependency · Decision trees  
Classification · Hidden Markov Model (HMM) · Phonetics  
Support Vector Machine (SVM)

---

S. Rashmi (✉) · M. Hanumanthappa  
Department of Computer Science and Applications, Bangalore University,  
Bangalore 560056, India  
e-mail: rashmi.karthik123@bub.ernet.in

M. Hanumanthappa  
e-mail: hanu6572@hotmail.com

M. V. Reddy  
Department of Computer Science, Rani Channamma University,  
Vidyasangam, Belgaum 591156, India  
e-mail: mallammantreddy@gmail.com

## 1 Introduction

Speech is the flow of thoughts in the form of natural language which is produced by articulating the sounds that are generated. Speech includes the formation of words and sentences. Perhaps speech is a perfect blend of rhythm and prosody, and hence, Concatenative Speech Analysis (CSA) has become extremely popular [1]. The primary target of CSA is to produce the phonetic structures and prosody models for the speech.

Speech-to-Text (STT) is a computer-based system that enables the user to enter the data in the form of speech, and then, it is converted into the textual form of data. Such a process automatically works without the human intervention. Over the past few decades, there is a tremendous amount of improvement in this arena, and it is becoming famous commercially as well. However, STT systems demand high quality, precision and accuracy. The coherence of STT mainly depends on the vocabulary size, speaker dependent versus independent, algorithms used, rate of speech and various other language constraints, and thus, its accuracy varies from system to system. This research paper focuses on studying the phonetic models and its components. We also aim to develop an accurate STT synthesizer by applying Data Mining and Natural Language Processing techniques in order to achieve improved efficiency as compared to the existing STT systems.

Speech is characterized by its temporal structure rather than spatial features; henceforth, speech always results in spectral vectors that span the audio frequency range. Furthermore, speech is characterized by the statistical models. In the persistence of the above-said fact, Hidden Markov Model (HMM) is a powerful framework that helps to construct the sound structure models more efficiently and effectively. HMM is considered as one of the substantial technique that is bound within every modern speech recognition system. It is because of this fact that it can be called as heart of the speech synthesizer systems. In the upcoming sections, evolution of STT, architecture of STT using HMM-based recognizer, implementation mechanisms and various challenges for this implementation have been discussed.

## 2 Literature Survey

The HMM is a famous decision-making technique that is most widely used in speech recognition systems. The available speech synthesizers using HMM are ATRECCS and TC-STAR. However, these incur a lot of time and expense. The history of speech synthesizer way dates back to 2002, and the final output was released in the year 2005 and was working for three languages: English, Spanish and Mandarin. The speech rate was 10 Hz, and the recording precision was set to 96 kHz/24 bit. In the year 2004, one more speech synthesizer was developed and was named as 'Blizzard challenge'. This could pronounce 1200 phonetics

utterances each having 1.5 Hz. Over the years, there has been a lot of improvement in this field. All such inventions are the major motivation for this work. The current work is focusing on STT by applying HMM techniques [2]. The core subject of HMM is to estimate the probability of word sequence which is achieved with the help of huge training set text. By maximizing the probability of the feature quantization vector series of the phonemes, the recognition hypothesis will be made.

The agglomerative clustering procedure for generating the text for multiple phonemes is explained here

- Initiate the HMM synthesizer for each pair of phone
- By this, a cluster of phonemes is formed
- Search for the phone-pairs which are closely related and merge together
- Look for the phonetic dictionary for the phoneme match
- Repeat the above steps for every word in the cluster

The current work is divided into three modules. First, the characteristics of the acoustic models are studied. Among them, we have chosen prosody and rhythm. On the other hand, the second module describes the construction of phonetic dictionary mitigating the issues related to this and finally, the third module discusses the feature selection approach for language identification.

### 3 Architecture of STT Using HMM-Based Speech Recognizer

The proposed architecture is shown in Fig. 1. It also shows the primary ingredients of a speech identification system. The principle aim of this research paper is to examine and analyse the core structure of STT and then describe the various milestones to achieve the state-of-the-art accomplishment. This is attained by using HMM. The acoustic models of the different variants of speech input are put forward.

The input for the system is the audio waveform. The input can be any recorded speech or the recorded voice using a microphone. The wave structure of the input audio is transformed into a series of fixed size vectors which are characterized by the acoustic features. This process is called as Quantization/Feature Extraction. Next is the decoder. The decoder is distinguished by three components: (1) language identification, (2) speech/acoustic models and (3) pronunciation dictionary. Furthermore, the decoder aims at identifying the words that are most likely to be indicated by the feature vectors, i.e. decoder produces the pronounced word as shown by the following equation (Eq. 1).

$$\hat{w} = [\max\{P(w/X)\}]_{[m\hat{x}_n]} \quad (1)$$

where  $w = \text{words } \{w_1, w_2 \dots w_n\}$ ,  $X = \text{Feature vector } \{X_1, X_2, X_n\}$ .

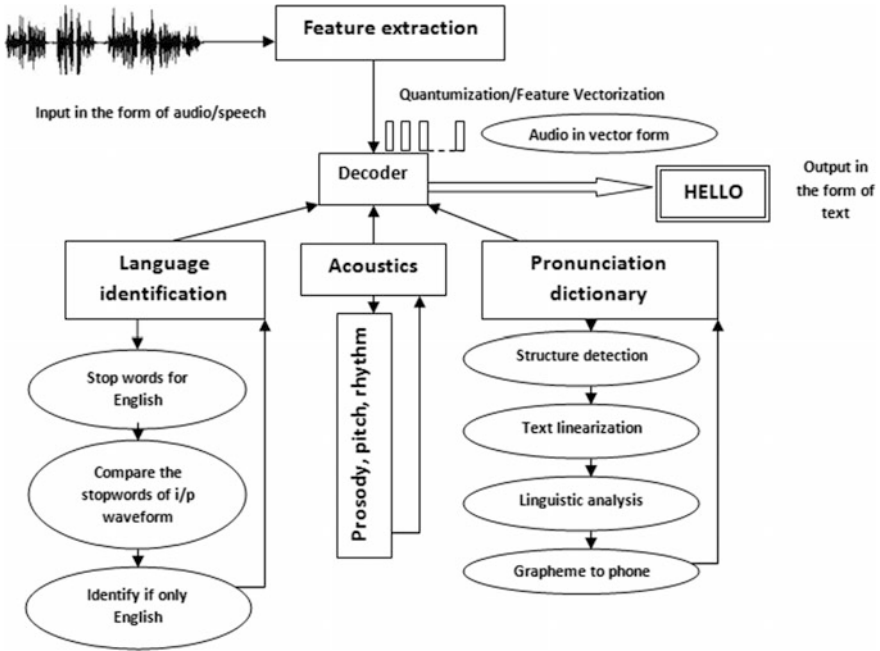


Fig. 1 Architecture of Speech-to-Text using HMM

However, the probability of  $P(w/X)$  is extremely tough to predict by using a brute force strategy. Therefore, by using a Bayesian transformation rule, Eq. 1 can take the equivalent form which is much easier to find the solution.

$$\hat{w} = \sum_{x=1}^N \frac{\max [P(w/X)]}{p(w) \cdot p(x)} \cdot p(w) \cdot p(x) \Big|_{(m \times n)} \tag{2}$$

The likelihood of the probabilities shown in Eq. 2 is designated by using the acoustic model in the form of phonemes. Phonemes are the basic language unit, each of which is represented in the language model. Phonemes are composed of ‘phone’, a single unit in phoneme. Such phone represents the association of a gigantic phoneme structure. For an instance, consider the word ‘beautiful’. This word is composed of four phones ‘bjʊ:-/tr-/fʊl,-/f(ə)l’. There are about 45 such distinguished phonemes in English dictionary. The phonetic structure of a spoken word can be generated by concatenating all the phonemes. Since the conversion of every grapheme (written form) into the equivalent phoneme (spoken form) is based on its antecedent, the phoneme model is considered as N-gram model where the output of  $n$ th level is dependent on the  $N - 1$  predecessor.

In the following section, the paradigms of the above components are explained in detail.

### 3.1 Feature Extraction/Quantization

A novel representation of the speech with the appropriate wave form is put forward. The major challenge in this realm is to hold the meaning of the word from getting lost during the intermediate conversion. Feature vectors are accomplished using one of the encoding schemes, Mel-Frequency Cepstral Coefficients (MFCCs). MFCCs are a famous and a standard encoding scheme typically used for large audio files with distinct parameter fluctuation in terms of bit rate and sampling rate [3]. The speech input signal is divided into window modelling where the size of the window ranges with the size of the word and falls between 10 and 25 ms. With this, the discrete Fourier transformation is computed which is given by

$$y(f) = \sum_{N=0}^{N-1} w(m)x(m) \exp(i2\theta f(m/N)) \quad (3)$$

where  $N$  is the overall length of the window and  $m$  is the length of one discrete-time signal and  $f$  represents frequency that varies between  $0 \dots N$ . Then, the word length [ $w(x)$ ] and magnitude of every word corresponding to the time signal [ $x(m)$ ] are calculated logarithmically by using Mel Filter, giving

$$y'(\theta) = \ln \left[ \sum_{N=0}^{N-1} (|y(f)|) \cdot \mathcal{M}(p, \theta) \right] \quad (4)$$

This equation is nonlinear for the predefined frequencies. The end result of the quantization process is a sequence of feature vectors whose dimensionality is almost decorrelated. When such feature vectors are concatenated in an orderly fashion, we arrive at delta and theta parameters [4]. These parameters make a heuristic attempt for finding regression coefficients. Therefore,  $\Delta x_t^v$ , the delta parameter is evaluated by the following equation,

$$\Delta x_t^v = \frac{\sum_{i=1}^N w_i(x_{t+i}^v - y_{t-i}^v)}{\pi \sum_{i=1}^N \theta(w_i)} \quad (5)$$

$\theta$  in Eqs. 3, 4 and 5 represents the angular velocity of the movements of feature vectors.

### 3.2 Decoder

Once the feature vectors arrive at the decoding segment, these vectors are distributed in a nonlinear manner across the speech spectrum. As noted earlier, for any

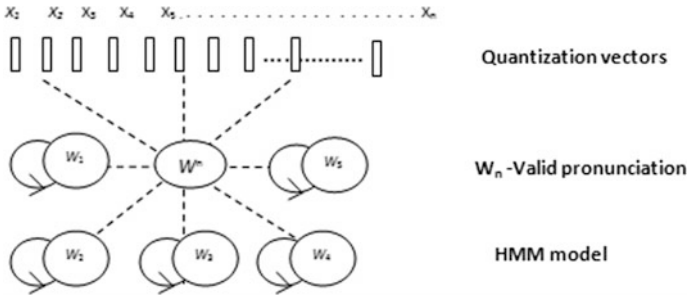


Fig. 2 Transition from feature vectors to word mapping using HMM-based phoneme model

word ‘ $w$ ’, there are a series of sound model produced called phonemes. Let these series be named as  $R_w$ . The probability of such likelihood can be expressed as Eq. 1. By using Eq. 1, the overall genuine and correct pronunciation can be tractable using the following equation

$$P(R/w) = \sum_{i=1}^L p(w/R_w), w^n \tag{6}$$

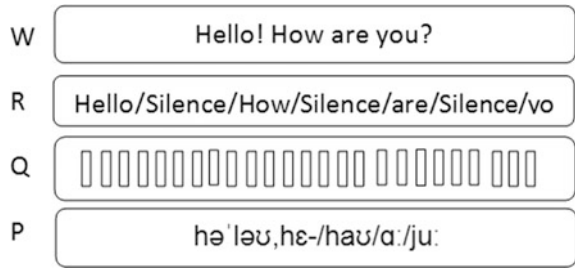
$w^n$  indicates a valid pronunciation. The transition of such probabilistic measure is described by HMM using a transition diagram. This resolves the round boundary set  $R_w$  making the transition from its present state to all the values of  $w^n$ , for each value of  $n$ .

Furthermore, from Fig. 2, it is evident to make the following discussions.

- $w^n$  can be generated with the help of all the independent values of  $w$
- $w^n$  is independent of  $X$ ; however, the values of  $w^n$  are correlated with the value of  $X$
- Many feature vectors can be discarded considering as noise, which also includes the millisecond gap during the intermediate phones generated by the audio input
- The order of sound utterance must be preserved

The partitioning of feature vectors into the phonemes is a major concern as its distributions are dependent on the likelihood of  $w$ ’s and in turn  $X$ ’s. Such an approach demands a high-level context-dependent covariance which is commonly referred as Beads-On-A-String (BDAS) [5]. This is because all the combinations of a valid pronunciation arrive at the interval of  $w^n$  by concatenating the sequence of  $w$  together. This imposes a large degree of context-dependency. For example, observe the value for  $w^n$ , Loot, School, Wool and Reel. The repetitive letters ‘oo’ or ‘ee’ have to be pronounced though it is same yet differs when one of them is omitted. The mapping of context-dependency is demonstrated in Fig. 3. The figure uses the conventions where  $W$  is the word spoken/input,  $Q$  represents the quantization vectors,  $P$  denotes the phonemes,  $L$  is the language representation and  $R$  is

**Fig. 3** Formation of phoneme modelling using content dependency structure



the logical modelling. The output dissemination can be constructed by using Gaussians distribution. According to Gaussians [6],

$$f(x/\mu_X,\sigma_X) = \prod_{x=1}^n \sigma_x \sigma_y \frac{1}{\sigma_X \sqrt{2\pi}} \int_{x=1}^{n-1} e^{-\frac{(x-\mu_X)^2}{2\sigma_X^2}} \tag{7}$$

In Eq. 7,  $\mu_X$  is the mean and  $\sigma_X$  represents variance. This equation gives the distribution of feature vectors for the normal deviation. This is diagonal in nature.

The feature vectors result in a series of phonetic transcription by word-to-word mapping, and these are then plotted in a look-up table. This table contains the phonetic word dictionary [Logical model]; finally, the phonemes are translated to English words [7]. The association between the logical and physical model is bound together through the states of the transition. Such transitions require the usage of decision trees for every phone that has been formulated using the above-said mechanism. All the phonemes are tied at the root nodes combining the value of each phone for the state ‘i’ of which the nodes are later chopped into various levels until the leaf nodes. This is a greedy approach, and it is iterative in nature. Figure 4 illustrates the decision tree for this greedy approach.

In this figure, an example of nasal sounds is shown. In English, the sounds of /m/, /n/, /ng/ are nasal, produced by generating the airstream through node; the example words ‘bringing’ and ‘hanging’ as ‘bri-ng + ing’ and ‘ha-ng + ing’ are shown in the form of decision tree in Fig. 4.

From the observation made using HMM recognizer, the Bayesian Classifier describes the topology and its construction. This is depicted in Fig. 5. The following HMM topology notations are assumed. The circle represents the discrete variable and empty circle stands for loss of phonemes, whereas filled circle shows the transition of phonemes that are to be considered. Square gives the continuous values for the phone structure, empty is for ‘no’ in the decision tree and ‘yes’ is given for a filled square. The triangle shows the constraint satisfaction and conditional transition. HMM contains many hidden states. A state is said to be hidden because when we traverse through the HMM synthesizer, these states will make the transition from hidden to visible. The number of states in a HMM model depends on the sequence of tokens in the input string. This grows recursively when the

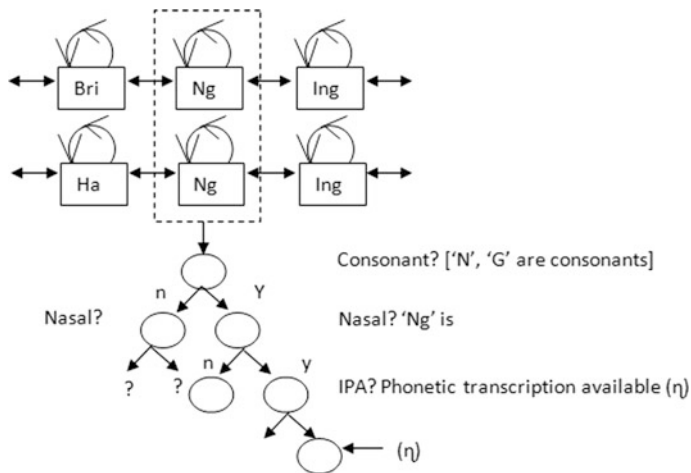


Fig. 4 Transition from feature vector to word mapping using HMM-based phoneme model

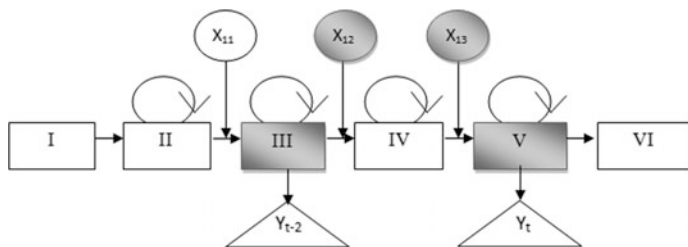


Fig. 5 Bayesian network for HMM topology

speech data increases. The lower bound on the number of states should be at least one (minimum requirement for the speech to be converted into text); however, there is no upper bound as this significantly grows with the input.

### 3.3 Pronunciation Dictionary

Predominantly, all the speech recognition system uses corpora which contain the phonetic transcriptions for the words of the native language. Such corpora are called the phonetic dictionary. This forms the training data. Nevertheless, even a clearly defined lexicon fails to provide the phonemes for all the pronunciations made by human. Besides, if an attempt is made to provide a dictionary which contains all such phonemes, then the size of the dictionary will be excessively large. Support Vector Machine (SVM) along with the rule-based classifier across the phoneme



model is proven to be much coherent for devising the phonetic corpora with not much modification in the dictionary. Using SVM, the phonemes model will generate new paths, and this is to automate the creation of new phonemes. The phonetic dictionary can be implemented in two ways: (1) dictionary-based and (2) rule-based classifier based on the SVM.

### 3.3.1 Dictionary Based

In this method, all the words and the corresponding phonemes are gathered. However, this could lead to a vast dictionary. This method has one more drawback, when a new word is encountered which is not found in the dictionary the output is not rendered.

### 3.3.2 Rule-Based Classifier

It was shown by Swamy [8] that 70% of English words can be deduced by using a subset of 2000 words only. This forms the base for our hypothesis, and henceforth, a phonetic dictionary was constructed consisting of 2500 randomly chosen, basic yet key words in English language. However, for a new word, the dictionary is trained using one of the supervised learning methods called as Support Vector Machine (SVM). With thorough literature survey, it was discovered that SVM is an optimal approach to implement phonetics. SVM, a classification technique, requires a trained/labelled data using which it categorizes a hyper plane that is favourable and optimum. The first step in SVM is to draw a line that linearly separates the points on the plane. In the next step, draw a line of equal distance between two boundaries where the line was linearly separated. This line should not be too close to the samples. If so, the points on this line will be eliminated as noise, and the related phones are abandoned. All the labelled samples that fall on an optimal linear bar form the support vectors. If the line is not 'linearly separable', then it is called as 'perceptron'. Assume a labelled data across  $M$  and  $N$  coordinates such that  $M_i$  and  $N_i$  are given by  $1, 2, \dots, Z$ .  $M \in E$  where  $E$  is the edit distance that calculates the level of similarity by calculating the number of edits needed to transform one text into another.  $N \in -1$  to  $+1$ , this provides the scope of the feasible efforts to bring the required phonemes for a given input. Hence, the function  $f(M, N)$  is given as  $f(M, N) \rightarrow$  Case i:  $\geq 0$  ( $N$  as positive coordinates). Case ii:  $< 0$  ( $N$  as negative coordinates). Accordingly, for a precise classification  $f(M, N) \geq 0$  should hold true. If this classification exists, then it is named as 'Linear Separable'. This is shown in Eq. 8.

$$f(M, N) = E[(\text{Weight}_{\text{Factor}})]_M^N + e. \quad (8)$$

Here,  $E$  represents the edit distance for all the  $N - 1$  points on the hyper plane;  $e$  is the noise error that is almost negligible. For example, the word ‘HALF’ contains one character ‘L’ as noise error since it is silent.

### 3.4 Language Recognition

The font type and the coded language have to be identified when a speech is given as an input. Therefore, the fundamental step is to identify the language spoken in the input speech.

#### 3.4.1 Classification

A technique used to forecast the correct label for an input data is called as classification [9]. To issue loan, the bank manager must inspect the available data (training set) of a customer in order to know whether granting loan to the applicant is safe or not. Thus, a proper supervision is required to manifest a clear boundary as this technique always incurs a question of uncertainty. The likelihood of data is either they belong to a trained class or it might be rejected [10]. Classification process contains building a classifier/training data model. A list of stop words is constructed. Stop words form the basic fundamental unit which is distinctive for a language dialect (the, to, is, I, am and so on). This acts as data for the supervised learning method. The characters are compared against the training data by using IF-THEN association rules. Table 1 shows the working of proposed Rule-Based-Classifier (RBC) algorithm.

**Table 1** RBC algorithm

<p><b>Algorithm: RBC: Rule Based Classifier</b></p> <p>Input → Set of words, <math>W_i \leftarrow D</math>, Set of stop words <math>SW</math>, <math>AVALUE \leftarrow 0</math> represents the attributes in the given input text. <math>Count \leftarrow 0</math>, Number of words in the input text.</p> <p>Output → Language recognized as English</p> <pre> Def ← Rule Set = { <math>\emptyset</math> }     Rules discovered so far is null.     For all values in <math>W_i</math> do         Def ← Rule<sub>i</sub>             if &lt; <math>W_i</math> &gt; = &lt; <math>SW</math> &gt; //each word in <math>W_i</math> is compared with every other word in <math>SW</math>.         Set <math>W_i</math> to <math>AVALUE</math>             <math>AVALUE \leftarrow AVALUE ++</math>         End if         IF <math>AVALUE = Count</math>             Then Set Language ← English         Else             Do not claim         End if     End for                 </pre>
--

## 4 Implementation

The Phonetics Language Processor is an interface that the entire research project will be interpreted on. The processor includes four tabs, namely Language detector, Audio/Speech-to-Text, Text-to Audio/Speech and Grammar check. The tab, Language detector shows the identification of language as explained in the Sect. 3.4. In the second segment, Audio/Speech-to-Text is taken care off. The techniques explained in the present work are administered and executed in the fragment Audio/speech-to-Text. The third and the fourth component have been defined to address Text-to-Audio and Grammar check which is beyond the scope of this research paper.

The statistical approach for STT using HMM was showed in the earlier section. In this section, the techniques explained so far have been implemented by designing an interface using .Net platform. Figure [6] is the final output which showcases all the features explained so far. The audio/speech file is provided as input for this interface. The file has to be in .wav form only. On successful loading of file, preview button is pressed. The output in the textual form will be seen in the layout.

## 5 Results and Discussion

A basic strategy to obtain the phonetic transcription is by using the available morphological analyser; however, the efficiency which the analyser provides does not cross above 75% in a huge volume of dictionary which contains 300 k words. By these, we can conclude that the pronunciation must be hand built by generating rules. In order to deduce the system with such rule is a major challenge. In the present work, a small number of hand cribbed words were added to the phonetic dictionary that includes 2500 words and 1500 sentences. In order to reduce the complexity of dictionary, many variations such as ‘ya, yep, gonna, wanna’ were

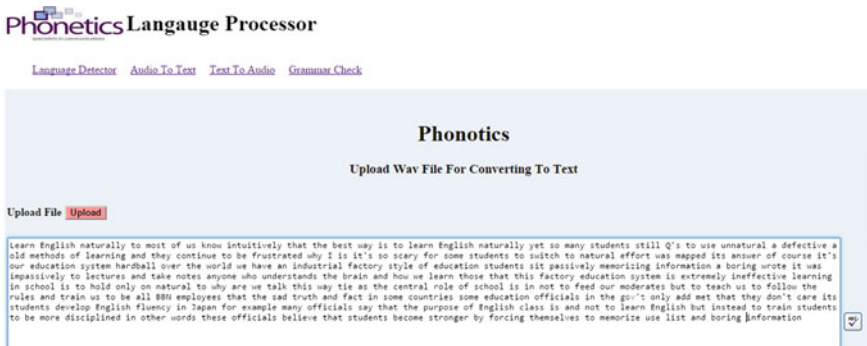


Fig. 6 Interface showing the conversion of Speech-to-Text

mapped to their base forms. The phonetics has an average of 4.3 variants per word in English. This shows the importance of pronunciation dictionary which maps one-to-one modelling. In reality, this results in a huge vocabulary; hence, by adopting the HMM-based speech synthesizer, this number can be reduced.

To summarize, the acoustic speech model was studied with the help of HMM and Gaussian distributions whilst decision tree supported the assumptions drawn on these. The overall study showcases the following key features.

- Monophonic mapping was deduced by HMM–Gaussian model that calculates mean and variance of the training data. Later, these monophonic transcriptions are mapped onto the phonetic dictionary that was built beforehand as explained in Sect. 3.3.
- With each monophonic word, biphonemes, triphonemes and multiphonemes are transformed into phonetic transcription and once again re-estimated using context-dependent model structure.
- The language of all these phonemes must be in English. This language identification is done to ensure that the input audio/speech was in English. If otherwise, the interface does not provide the output. The reason for this is that the phonetic dictionary was built only for the English language.
- The output is rendered in the form of text.

The performance of the STT synthesizer was evaluated for a different range of speech input. In practice, it was found that the efficiency of the speech recognition system varies with the size of the vocabulary. When a set of audio files are interpolated as the input, the equivalent text was received as the output. Table 2 evaluates the performance of the proposed architecture. The input speech was named as S1, S2, S3, S4 and S5. Each of this input consisted of all the different type of speech variants. S4 was the size of 20 min, and it was expected to be much harder when compared to other speech files. It carried disruption such as background music and embodied other kinds of interference which included multilingual context. However, the system gave the correct results by identifying only the English language. The results achieved by our approach are fascinating and were found to be

**Table 2** Result evaluation on the interface

Input	Total words	Words correctly identified	Words that are identified wrong	Words that are not identified	% correctness	Of recall (%)	Precision (%)
S1	200	195	3	2	97.5	98.1	98.3
S2	18	14	2	2	77.7	87	87
S3	570	497	38	35	87.19	93	92
S4 (English only)	8478	7023	97	358	94.63	95	98
S5	4396	3762	363	271	85.57	93	91

**Table 3** Comparison of the results of existing systems

Sl. No	Features	Techniques/ Methodologies	Accuracy (%)	Drawbacks
1	MFCCs	Rule based, PRLM	82	Concentrates only on the feature extraction
2	Acoustic models	HMM	79	Generation of the acoustic features results in chopping of essential elements
3	Phonotactics, acoustic, prosodic information	Trigram model	NA	Does not show the conversion of Speech-to-Text
4	Articulation of sounds	SVM	83	Time taken for the sound apprehensions is high
5	Phonetic features, nazal, articulation features	HMM & N-gram model	78	Concentrates on the phonemes generation; however, the system does not show the performance for the huge data

88.51% accurate and efficient. Table 3 provides a comparative study of the proposed architecture with existing methodologies and algorithms.

## 6 Conclusion

In this research paper, STT paradigm using HMM is put forward. HMM is an excellent technique for resolving many computational language challenges in the field of speech recognition. The intention of this work was to develop an interface using the acoustic models. It was found that the output text was being trained automatically on the input speech. The various feature distribution and their effect on the output were studied at the same time. In terms of chief investigation in the STT, it was discovered that the model can be extended for multiple languages by building the phonetic dictionary of the same along with some modifications in the phonemes. HMM-based model adopts several assumptions on the feature quantization, training data and context-dependency. Conventionally, a few of those presumptions can be compromised to some extent. Finally, it should be noted that despite the advantages of HMM and its superiority, many expostulate that HMM is

blemished. This is of course true under many circumstances as the system gets vulnerable with the speaking styles, frequency, dialects and accents. Perhaps there has been no good alternative for HMM and it is because of this that HMM is still undeniably the best approach for implementing STT.

## References

1. Alias, F., et al.: Towards high-quality next generation text-to-speech synthesis: a multi domain approach by automatic domain classification. *IEEE Trans. Audio Speech Lang. Process.* **16**(7) (Sept 2008)
2. Abushariah, A.A.M., et al.: English digits speech recognition system based on Hidden Markov Models. In: *IEEE Conference 2010, ICCCE*. doi:[10.1109/ICCCE.2010.5556819](https://doi.org/10.1109/ICCCE.2010.5556819)
3. Hossan, M.A., et al.: A novel approach for MFCC feature extraction. In: *IEEE Conference 2010, ICSPCS*. doi:[10.1109/ICSPCS.2010.5709752](https://doi.org/10.1109/ICSPCS.2010.5709752)
4. Bsyne, W.: Minimum Bayes risk estimation and decoding in large vocabulary continuous speech recognition. *IEEE* **E89-D**(3), 900–907 (2006)
5. Patel, I., et al.: Speech recognition using hidden Markov model with MFCC-subband technique. In: *IEEE Conference (2010)*. doi:[10.1109/ITC.2010.45](https://doi.org/10.1109/ITC.2010.45)
6. Duan, W., et al.: Weighted naive Bayesian classifier model based on information gain. In: *IEEE Conference, (ISDEA)*. doi:[10.1109/ISDEA.2010.226](https://doi.org/10.1109/ISDEA.2010.226)
7. Gales, M., et al.: The application of hidden Markov models in speech recognition. *Found. Trends Signal Process.* **1**(3), 195–304 (2008)
8. Swamy, S., et al.: An efficient speech recognition system. *Comput. Sci. Eng. Int. J. (CSEIJ)* **3** (4) (Aug 2013)
9. Kholghi, M., et al.: Classification and evaluation of data mining techniques for data stream requirements. In: *IEEE Conference on Computer Communication Control and Automation (3CA)*. doi:[10.1109/3CA.2010.5533759](https://doi.org/10.1109/3CA.2010.5533759)
10. Shahrokhi, N., et al.: Targeting customers with data mining techniques: classification. In: *2011 International Conference on User Science and Engineering (i-USER)*, IEEE, New York. doi:[10.1109/iUSER.2011.6150567](https://doi.org/10.1109/iUSER.2011.6150567)