Word and Triphone Based Approaches in Continuous Speech Recognition for Tamil Language

R. THANGARAJAN
Assistant Professor, Information Technology Department
Kongu Engineering College
Perundurai – 638 052, Erode
INDIA
thangs_68@yahoo.com

A.M. NATARAJAN
Chief Executive and Professor
Bannari Amman Institute of Technology
Sathyamangalam – 638 401, Erode
INDIA

M. SELVAM
Assistant Professor, Information Technology Department
Kongu Engineering College
Perundurai – 638 052, Erode
INDIA

Abstract: - Building a continuous speech recognizer for the Indian language like Tamil is a challenging task due to the unique inherent features of the language like long and short vowels, lack of aspirated stops, aspirated consonants and many instances of allophones. Stress and accent vary in spoken Tamil language from region to region. But in formal read Tamil speech, stress and accents are ignored. There are three approaches to continuous speech recognition (CSR) based on the sub-word unit viz. word, phoneme and syllable. Like other Indian languages, Tamil is also syllabic in nature. Pronunciation of words and sentences is strictly governed by set of linguistic rules. Many attempts have been made to build continuous speech recognizers for Tamil for small and restricted tasks. However medium and large vocabulary CSR for Tamil is relatively new and not explored. In this paper, the authors have attempted to build a Hidden Markov Model (HMM) based word and triphone acoustic models. The objective of this research is to build a small vocabulary word based and a medium vocabulary triphone based continuous speech recognizers for Tamil language. In this experimentation, a word based Context Independent (CI) acoustic model for 371 unique words and a triphone based Context Dependent (CD) acoustic model for 1700 unique words have been built. In addition to the acoustic models a pronunciation dictionary with 44 base phones and trigram based statistical language model have also been built as integral components of the linguist. These recognizers give very good word accuracy for trained and test sentences read by trained and new speakers.

Key-Words: - Acoustic Model, Context Dependent, Context Independent, Continuous Speech Recognition, Hidden Markov Model, Tamil language, Triphone.

1 Introduction

AUTOMATIC SPEECH RECOGNITION (ASR) deals with automatic conversion of acoustic signals of a speech utterance into text transcription. Even after years of extensive research and development, accuracy in ASR remains a challenge to researchers. There are number of well known factors which determine accuracy. The prominent factors are those that include variations in context, speakers and noise in the environment. Therefore research in ASR has many open issues with respect to small or large vocabulary, isolated or continuous speech, speaker dependent or independent and environmental robustness.

ASR for western languages like English and Asian languages like Chinese is well matured. But similar research in Indian languages is still in its infancy stage. Another major hurdle in ASR for

Indian language is resource deficiency. Annotated speech corpora for training and testing the acoustic models are scarce. Recently there is a growing interest in ASR for Tamil and other Indian languages. There are speech recognition works for Tamil language which are targeted towards low and restricted vocabulary task [1]. There are some funded research works in spoken digit recognition [2]. Others have attempted to speech recognition for isolated word recognition in Tamil using Artificial Neural Network (ANN) [3]. Some research has been reported to tackle the issue of resource deficiency by means of cross language transfer and cross language adaptation techniques [4].

There are other literatures concerning large vocabulary continuous speech recognition (LVCSR) in Indian languages including Tamil. These works are mainly concentrated and targeted towards exploiting the syllabic nature of Tamil and other Indian languages. Basically there are three approaches to speech recognition with respect to the choice of sub-word units namely, word based, phone based and syllable based recognition. The criterion for the choice of sub-word units is dealt with in detail in the section on literature survey.

This paper is organized into eight sections including the introduction. Section 2 covers the literature survey of various related works in Tamil language with respect to choice of sub-word units. Section 3 gives the problem formulation. Section 4 describes the Tamil language and its inherent features, pronunciation and its phone set. Section 5 describes the construction of acoustic models using the proposed methodology of the paper. Section 6 describes the deployment of the acoustic models in real-time speech decoder namely, the Carnegie Mellon University's (CMU) Sphinx 4.0. Section 7 shows the results of the experimentation and section 8 gives the discussion and conclusion of the paper.

2 Literature Survey

Speech recognition requires segmentation of speech waveform into fundamental acoustic units. Word is the preferred and natural unit of speech because ultimately it is the word that one is trying to recognize. Moreover, word units have their acoustic representation well defined. Acoustic variability occurs mainly in the beginning and end of a word i.e. at word boundaries. However using word as a speech unit in LVCSR introduces several serious problems. Since each word has to be trained individually and there cannot be any sharing of parameters among words, one has to have a very large training set so that all words in the vocabulary are adequately trained. The second problem lies with memory requirement which grows linearly with the number of words. Hence word models are not practical for LVCSR systems. But word models have been successfully used for limited vocabulary ASR [5] [6]. This was the main motivating factor to employ word models in our experimentation for small vocabulary task.

A smaller or sub-word unit is the next choice because one has to have sharing of parameters across models in order to save computing resources. The most popular sub-word units are the phones, as used in English. Since there are only about 50 phones in English and other languages, phone models can be sufficiently trained for a reasonable size of training corpus. However there are other problems in phone models also. It is a well known fact that the same phone in different words has different realizations. This is because the articulators cannot move from one position to another instantaneously. Hence the realization of a phone is strongly affected by its adjacent phones or in other words, phones are highly context dependent. Some phones are aspirated when they occur in the beginning of a word and the same phones are not aspirated when they occur at the end of a word. Therefore the acoustic variability of basic phonetic units due to context is sufficiently large and not well understood in many languages. It has been shown that word based Dynamic Time Warping (DTW) performs significantly better than phone based HMM [7] [8]. Hence, it can be observed that while word models don't generalize, phone models over-generalize.

This problem of over-generalization can be overcome by employing phone-in-context. A context refers to immediate left and/or right neighboring phones. Phone-in-context can be either a left-context phone or a right-context phone or both. The third category is also known as triphone which includes both the left and right contexts. If two phones have the same identity but different left or right contexts, then they are considered as different triphones. Triphone models are powerful sub-word models because they account for the left and right phonetic contexts. Triphones have been enormously successful in acoustic modeling of LVCSR systems [9]. Compared to word-model and phone-model, triphone model reduced word error rate (WER) by more than 50% [10] [11]. However there are problems with triphones also. For any language, there are large numbers of triphones in the training set. This leads to memory wastage since a model is created for a triphone even if that triphone is observed only once in the training set.

There are other deviant approaches in the literature which use larger sub-word units of speech in order to model the co-articulator effects of speech production. The most popular unit under this category is the syllable. A syllable is a larger unit than a phone since it encompasses two or more phone clusters. These phone clusters account for the severe contextual effects. A syllable is composed of three parts: the onset, the nucleus and the coda or rhyme. The nucleus or central part does not have any contextual dependencies while the onset and the coda are still susceptible to some contextual effects [12].

Tamil and other Indian languages share phonological features which are rich in vowel and consonant realizations. Pronunciations are mainly based on syllables. With these features in mind, several leading researchers in India have focused their work on the syllable as a speech unit. These works also address the problem of deficiency in annotated speech corpora. In [13] [14], an approach is proposed for automatically segmenting and annotating continuous speech without the use of manually annotated speech corpora. Using the short-term energy as the magnitude spectrum, the continuous speech signal is segmented into syllable-like units. Subsequently similar syllables are clustered using an unsupervised incremental clustering technique and the syllables are labeled manually. Models are then created for the syllable clusters which are trained and used to transcribe continuous speech. Similar approach has been used in [15] where both continuous speech signal and the text are segmented into syllable-like units and models are created. However the major problem with syllable is that there are roughly around 20,000 syllables for languages like English [9] and Tamil, and without parameter sharing trainability is poor. The performance of these systems is around 53% accuracy [16].

3 Problem Formulation

Fundamentally, the problem of speech recognition can be stated as follows. When given with acoustic observation $X = X_1X_2...X_n$, the goal is to find out the corresponding word sequence $W = w_1w_2...w_m$ that has the maximum posterior probability $P(W|X)$ expressed using Bayes Theorem as shown in equation (1).

$$W = \arg\max_{w} P(W \mid X) = \arg\max_{w} \frac{P(W)P(X \mid W)}{P(X)} \quad (1)$$

Where $P(W)$ is the probability of word W uttered and $P(X|W)$ is the probability of acoustic observation X when word W is uttered. $P(X|W)$ is also known as class conditioned probability distribution. $P(X)$ is the average probability that the observation X will occur. Since the maximization of equation (1) is done with variable X fixed, to find the word W it is enough to maximize the numerator alone.

$$W = \arg\max_{w} \left(P(W)P(X \mid W) \right) \quad (2)$$

The first term in equation (2), $P(W)$, is computed with the help of a language model. It describes the probability associated with a hypothesized sequence of words. The language model incorporates both the syntactic and semantic constraints of the language and the recognition task. Generally the language model may be of the form of a formal parser, syntax analyzer, N-gram model or a hybrid model [17].

The second term in equation (2), $P(X|W)$, is computed using an acoustic model which estimates the probability of a sequence of acoustic observations conditioned on the word W. The recognizer needs to know the class conditioned probability $P(X|W)$ from the acoustic model in order to compute the posteriori probability $P(W|X)$. HMM has become the common structure of acoustic models because HMM can normalize speech signal's time-variation and characterize speech signal statistically thus helping to parameterize the class conditioned probabilities. Thus the acoustic model forms the core knowledge base representing various parameters of speech in the optimal sense. Even though speech decoding with other classification models like neural networks and support vector machines are also reported in the literature[18], at present, all state-of-the-art commercial and most laboratory speech recognition systems are based on HMM that give very low WER when tested on standard speech databases [19][20].

Therefore in this paper the authors have attempted to build HMM based acoustic models for small and medium vocabulary continuous speech recognition using word and triphone as units and compare their performances in terms of WER, speed and memory footprints for a new language like Tamil.

4 The Tamil Language

Tamil is a Dravidian language spoken predominantly in the state of Tamilnadu in India and Sri Lanka. It is the official language of the Indian state of Tamilnadu and also has official status in Sri Lanka and Singapore. With more than 77 million speakers, Tamil is one of the widely spoken languages of the world.

4.1 Tamil alphabet

Some of the phonological features which are of interest to speech recognition research are as follows. Tamil vowels are classified into short, long (five of each type) and two diphthongs. Consonants are classified into three categories with six in each category: hard, soft (a.k.a nasal), and medium. The classification is based on the place of articulation. In total there are 18 consonants. The vowels and consonants combine to form 216 compound characters. The compound characters are formed by placing dependent vowel markers on either one side or both sides of the consonant. There is one more special letter *aytham* (ஃ) used in classical Tamil and rarely found in modern Tamil. Summing up there are 247 letters in standard Tamil alphabet. In addition to the standard characters, six characters taken from the *Grantha* script which is used in modern Tamil to represent sounds not native to Tamil, that is, words borrowed from Sanskrit and other languages. Even though Tamil is characterized by its use of retroflex consonants similar to the other Dravidian languages, it also uses a unique liquid which is Tamil equivalent to *zh.* Extensive research has been reported in articulation of liquid consonants in Tamil [21].

4.2 Pronunciation in Tamil

Tamil has its unique letter to sound rules. There are very restricted numbers of consonant clusters. Tamil has neither aspirated nor voiced stops. Unlike most other Indian languages, Tamil does not have aspirated consonants. In addition, the voicing of plosives is governed by strict rules. Plosives are unvoiced if they occur word-initially or doubled. The Tamil script does not have distinct letters for voiced and unvoiced plosives, although both are present in the spoken language as allophones.

Generally languages structure the utterance of words by giving greater prominence to some constituents than others. This is true in the case of English: one or more phones standout as more prominent than the rest. This is typically described as word stress. The same is true for higher level prosody in a sentence where one or more constituent may bear stress or accent. As far as Tamil language is concerned, it is assumed that there is no stress or accent in Tamil at word level and all syllables are pronounced with the same emphasis. However there are other opinions that the position of stress in the word is by no means fixed to any syllable of individual word. In connected speech the stress is found more often in the initial syllable. Detailed study on pronunciation in Tamil can be found in [22] [23]. In our experiment, stress on syllable is ignored because we are dealing with read speech.

4.3 Phone set of Tamil Language

In word based recognition, the word itself is treated as a phonetic unit whereas in phone based approach, a well defined phone set is required. The authors have proposed a phone set for Tamil language which contains 44 phones and other acoustic events. The total phone set along with co-articulation details is shown in Appendix-A. As discussed in section 4.1 and 4.2, Letter to sound rules are not very rigid in Tamil as in other Indian and Dravidian languages. There are many occurrences of allophones and pronunciation variation based on position of the letter in the word. Hence a pronunciation dictionary becomes necessary to aid the recognition process. A pronunciation dictionary for 1700 unique Tamil words was built.

5 Building Word and Triphone Tamil Words

As a first step towards building a LVCSR system for Tamil language, in this paper the authors have attempted to build a small vocabulary continuous speech recognizer using CI word model and a medium vocabulary continuous speech recognizer using triphone model. The important modules in speech recognition are acoustic model, dictionary and language model.

5.1 Development of Language model and Dictionary

Statistical tri-gram language models were built using the CMU Statistical Language Modeling toolkit for word and phoneme model with 341 and 1700 unique words respectively. Dictionary for word model has been created by mapping every word in the lexicon to itself. Dictionary for triphone model has been created by mapping every word in the lexicon to the string of phones. Same word can have several pronunciations which are represented by multiple entries indexed with numbers.

5.2 Development of Speech Corpus

Since speech corpora are not available for Tamil, a speech corpus was created in-house. The corpus containing 22.5 hours of continuous read speech of 13 males and 12 females for training and 7.5 hrs of speech of 75 males and 75 females for testing has been created. The recording was carried out in a

noise free lab environment. Finally, sentence level transcriptions were done manually.

All the speakers spoke from a set of unique sentences compiled by the authors using the text corpus. In the training data, all 25 speakers spoke the same 550 sentences.

However, in the test set, every speaker spoke 4 sentences whose lexicon is covered in the training set. Test sentences for the triphone model partly covered the training set and partly new.

5.3 Training the Acoustic Model

The HMM based acoustic model trainer from CMU, *SphinxTrain*, has been employed. The input file format and details of front-end processing are summarized in table 1.

Table 1. Front-end Processing Details

Parameter	Value
Input File Format	Wav (Microsoft) File
Sampling Rate	16,000 Hz
Depth	16 bits
Mono/Stereo	Mono
Window Length	0.025625 S
No. of FFT	512
No. of Filters	31
Min. Frequency	200 Hz.
Max. Frequency	3500 Hz.
No. of Ceptrums	13
Output	Mel frequency Ceptral Co-efficient (MFCC)

Other details of the training parameters are summarized in table 2. For word model, the number of states in the HMM is 20 since duration of words are longer than phones. Word model used CI training. For triphone or CD phone model, a 3-state HMM was used. There are three basic problems one can address with HMM.

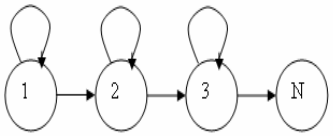
a. Evaluation: Given a model λ and observation sequence O calculate $P(O|\lambda)$
b. State Sequence: Given a model λ and observation sequence O, find state sequence S^* such that
$$P(S^* \mid O, \lambda) = max_S P(S \mid O, \lambda)$$
c. Learning: Given a set of observation sequence $\{O^k\}$, find a model λ^* such that
$$P(X \mid \lambda^*) = max_\lambda P(X \mid \lambda)$$

The first problem is of evaluation. Here one can determine the likelihood of an observation sequence O being generated by a model λ. The algorithm used is called *forward* algorithm.

The second problem is to determine the state sequence S^* which could have generated the observation sequence O for a given model λ. This is also known as the decoding problem. Here the *Viterbi* algorithm is used.

The third problem pertains to model training. For a given set of observation sequences $\{O^k\}$, find the model λ, which maximizes $P(X \mid \lambda)$. The algorithm used for training is called the *Baum-Welch* algorithm or the *forward-backward* algorithm.

Figure 1 shows a 3-state HMM along with its transition matrix. A fourth non-emitting node is automatically added to the HMM. From the transition matrix it is evident that the HMM is a left to right HMM.



1.0 1.0 0.0 0.0
0.0 1.0 1.0 0.0
0.0 0.0 1.0 1.0

Figure 1. A 3-State HMM with its Transition matrix.

Table 2. Training Parameters

Parameter	Word based model	Triphone based model
Type of Training	Context Independent (Continuous Density)	Context Dependent (Continuous Density)
Input Features	Mel frequency Ceptral Co-efficient	Mel Frequency Ceptral Co-efficient
Feature Type	Ceptra, Delta and Double Delta	Ceptra, Delta and Double Delta
Dimensions	13	13
No. of States in HMM	20 and one Non-emitting node	3 and one Non-emitting node
No. of Gaussians	1	8

Triphone based model follows a generic strategy for training. The training procedure comprises the following processes.

1. **Flat-start monophone training:** Generation of monophone or CI seed models with nominal values, and re-estimation of these models using reference transcriptions. This is also called flat initialization of CI model parameters

2. **Baum-Welch training of monophones:** Adjustment of the silence model and re-estimation of single-Gaussian monophones using the standard Viterbi alignment process.
3. **Triphone creation:** Creation of triphone transcriptions from monophone transcriptions and initial triphone training. This step creates CD untied model files and flat initialization of model files.
4. **Training CD untied models:** Again the Baum-Welch algorithm is iteratively used. This takes 6 – 10 iterations.
5. **Building decision trees and parameter sharing:** A group of similar states is called a senone. Senone is also called as a tied state. Then the senones are trained.
6. **Mixture generation:** Splitting single Gaussian distributions into mixture distributions using an iterative divide-by-two clustering algorithm and re-estimation of triphone models with mixture distributions.

Word based model follows only the first two steps in its training procedure. After the training is over, *SphinxTrain* generates the parameter files of the HMM namely, the probability distributions and transition matrices of all the HMM models.

6 Implementation

Both word and triphone based models are implemented on Sphinx-4 which is a state-of-art HMM based speech recognition system. It is being developed on open source since February 2002. Sphinx-4 is the successor of Sphinx-3 and Sphinx-2 designed jointly by Carnegie Mellon University, Sun Microsystems Laboratories and Mitsubishi Electric Research Laboratories, USA. It is implemented in Java programming language and thus it is portable across a growing number of computational platforms [24].

6.1 The Sphinx-4 Framework

The Sphinx-4 framework has been designed with a high degree of flexibility and modularity. Figure 2 shows the overall architecture of the system. Each labeled element represents a module that can be easily replaced, allowing researchers to experiment with different module implementations without needing to modify other portions of the system. There are three primary modules in the Sphinx-4 framework: the *Front-End*, the *Decoder*, and the *Linguist*.

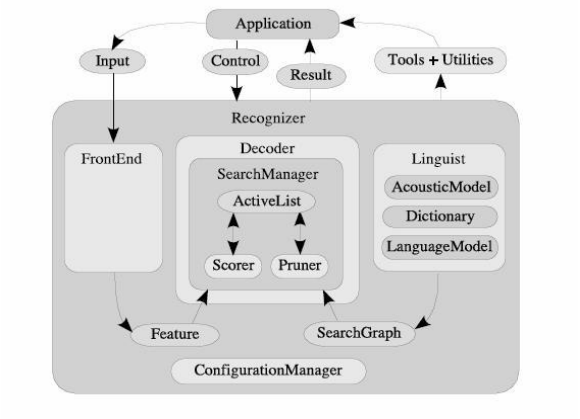


Figure 2. Sphinx-4.Architecture.
[Source: Carnegie Mellon University]

The Linguist comprises one or more *Acoustic models*, a *Dictionary* and a *Language Model*. Depending upon the linguist, different modules can be plugged into the system. This is done through the *Configuration Manager* module.

6.2 Decoding Continuous Tamil Speech

The language model, dictionary and the acoustic model developed in section 5.3 were deployed on the Sphinx-4 decoder. Sphinx-4 was configured to operate for both of the models with the components shown in table 3.

Table 3. Components of ASR

Component	Word based model	Triphone based model
Linguist	Flat Linguist	LexTree Linguist
Dictionary	Full Dictionary	Fast Dictionary
Search Manager	Simple Breath First Search Manager	Simple Breath First Search Manager

6.2.1. Flat linguist

This is a simple form of a linguist. A flat linguist takes a grammar graph and generates a search graph for the grammar. The following assumptions are made

- Zero or one word per grammar node
- No fan-in allowed
- Only unit, HMM state and pronunciation states are allowed
- Only valid transitions are allowed
- No tree organization of units

6.2.2. LexTree Linguist:

Due to large size of the vocabulary, a LexTree Linguist is used. The characteristics of LexTree Linguist are

- Dynamic: It generates search states on the fly, greatly reducing the required memory footprint
- Tree topology: It represents the search space as an inverted tree. Units near the root of words are shared among many different words.
- HMM sharing: Because of state tying in the acoustic models, it is often the case that triphone units that differ in the right context actually are represented by the same HMM. This linguist recognizes this case and will use a single state to represent the HMM instead of two states. This can greatly reduce the number of states generated by the linguist.
- Small-footprint: It uses a few other techniques to reduce the overall footprint of the search space. One technique is particularly helpful to share the end word units (where the largest fan-out of states occurs) across all the words.
- Quick loading: It can compile the search space very quickly. A 60K vocabulary can be made ready in less than 10 seconds.

6.2.3. Full dictionary

This component creates a dictionary by reading the Sphinx-3 format dictionary. The full dictionary will read all the words and their pronunciations at startup. Therefore, it is suitable for low vocabulary task.

6.2.4. Fast Dictionary

Since the triphone model uses medium size vocabulary, fast dictionary is used. Fast dictionary quickly adds words from the dictionary file to a hash table, assuming that the most words are not going to be used immediately. Only when a word is actually used, its pronunciation array is actually built. This helps in creating dictionary quickly.

6.2.5. Simple Breadth First Search Manager

With the acoustic features and linguist as input, this module performs simple breadth first search on the search graph rendered by the respective linguist used based on the model.

7 Results

The hypothesis word sequences from the decoder are aligned with reference sentences. The result is generated in terms of WER and word accuracy. Word errors are categorized into number of insertions, substitutions and deletions. Finally, the accuracy is computed by equation (3).

$$Accuracy = \frac{CorrectLength - Subs - Dels - Ins}{CorrectLength} \times 100 \quad (3)$$

Other performance measures are speed and memory footprints. Both word and triphone models were tested in batch mode with three trails. The results are tabulated in tables 4, 5 and 6.

Table 4. Trained voice on trained sentences

Details	Word- model	Triphone- model
Words	110	127
Errors	6 (Sub: 0 Ins: 0 Del: 6)	8 (Sub: 6 Ins: 0 Del: 2)
Accuracy	94.55 %	93.70%
Sentences	13	22
Time	Audio: 26.80 s Processing: 46.53 s	Audio: 33.68 s Processing: 21.48 s
Speed	1.74 × Real time	0.64 × Real time
Memory	Average: 22.51 MB Max: 26.40 MB	Average: 33.50 MB Max: 38.00 MB

Table 5. Trained voice on test Sentences

Details	Word- model	Triphone- model
Words	387	617
Errors	114 (Sub: 35 Ins: 2 Del: 77)	71 (Sub: 58 Ins: 2 Del: 11)
Accuracy	71.05%	88.82 %
Sentences	50	45
Time	Audio: 86.36 s Processing: 232.79 s	Audio: 337.99 s Processing: 219.58 s
Speed	2.70 × Real time	0.65 × Real time
Memory	Average: 21.93 MB Max: 27.69 MB	Average: 40.35 MB Max. 50.07 MB

Table 6. Results with new voice on test sentences

Details	Word-model	Triphone-model
Words	341	617
Errors	103 (Sub: 35 Ins: 1 Del: 67)	51 (Sub: 41 Ins: 2 Del: 8)
Accuracy	70.08 %	92.06 %
Sentences	50	45
Time	Audio: 80.54 s Processing: 198.26 s	Audio: 381.32 s Processing: 254.72 s
Speed	2.46 × Real time	0.67 × Real time
Memory	Average: 22.14 MB Max: 27.65 MB	Average: 42.50 MB Max. 55.25 MB

8 Discussion and Conclusion

Obviously the accuracy of both of the models is very high for the trained sentences with trained voices. In scenarios where the vocabulary is limited, repeatability of sentences is more and speakers are limited, the word based recognizer with trained voice on trained sentences is highly suitable. In the same scenario, when the vocabulary is high and

speakers are limited, triphone based model is suitable. The WER shows a majority of deletions errors in word based model and substitution errors in triphone model. The speed of recognition process is lower in word based model than triphone based model. As far as word models are concerned, CI word modeling yields reasonable accuracy for vocabulary size of 350.

For medium and large vocabulary, a triphone based approach is best suited. This is due to the fact that the number of phones is limited and larger training corpus is used. Hence more training is provided to the triphones in the acoustic model thereby increasing the accuracy. Hence it is concluded that performance of both the models is on par with the recognizers for resource rich languages like English. But still there are problems and issues to be addressed. Tamil being a highly inflected language, there are many occurrences of homonyms which account for WER. Moreover, Tamil language is agglutinative in nature i.e. words are composed of root and affixes. Most of the affixes are suffixes. The length of agglutination is more in Tamil resulting very long words.

Hence further work could be aimed at the inherent features in pronunciation of Tamil language which could be exploited in acoustic modeling. It is believed that larger sub-word units like syllable could improve system performance. Many attempts have been made for English language. But initially, there is an increase in WER [25][26]. In English pronunciation variation is high and syllabification is fuzzy. Even with increasing WER, syllable still remains a primary focus of research in speech recognition. But on the contrary, Tamil has well defined syllabification and *sandhi* rules which could help in syllable modeling which will in turn increase the recognition rates.

Appendix-A: Tamil Phone-set

S. No	Phone	V / C	VL	VH	VF	LR	TC	PA	CV	Remarks
1	A	+	s	3	3	-	NA	NA	NA	
2	Aa	+	l	3	3	-	NA	NA	NA	
3	Ih	+	s	1	1	-	NA	NA	NA	
4	Iy	+	l	1	1	-	NA	NA	NA	
5	Uh	+	s	2	3	+	NA	NA	NA	
6	Uw	+	l	1	3	+	NA	NA	NA	
7	Eh	+	a	2	2	-	NA	NA	NA	
8	Ee	+	d	2	1	-	NA	NA	NA	
9	Ay	+	d	3	2	-	NA	NA	NA	
10	Oh	+	s	2	2	+	NA	NA	NA	
11	Oo	+	d	2	2	+	NA	NA	NA	
12	Aw	+	d	3	2	+	NA	NA	NA	
13	F	-	NA	NA	NA	NA	f	b	-	
14	K	-	NA	NA	NA	NA	s	v	-	
15	G	-	NA	NA	NA	NA	s	v	+	
16	Ch	-	NA	NA	NA	NA	a	p	-	
17	T	-	NA	NA	NA	NA	s	a	-	
18	D	-	NA	NA	NA	NA	s	a	+	*
19	N	-	NA	NA	NA	NA	n	p	+	
20	Th	-	NA	NA	NA	NA	f	d	-	
21	Dh	-	NA	NA	NA	NA	f	d	+	*
22	N	-	NA	NA	NA	NA	n	a	+	
23	P	-	NA	NA	NA	NA	s	l	-	
24	B	-	NA	NA	NA	NA	s	l	+	*
25	M	-	NA	NA	NA	NA	n	l	+	
26	Y	-	NA	NA	NA	NA	a	p	+	
27	r	-	NA	NA	NA	NA	a	a	+	
28	L	-	NA	NA	NA	NA	l	a	+	
29	V	-	NA	NA	NA	NA	f	b	+	
30	L	-	NA	NA	NA	NA	l	a	+	
31	R	-	NA	NA	NA	NA	a	a	+	
32	S	-	NA	NA	NA	NA	f	a	-	
33	Sh	-	NA	NA	NA	NA	f	p	-	*
34	J	-	NA	NA	NA	NA	a	p	+	Gr
35	H	-	NA	NA	NA	NA	f	g	-	Gr
36	ksh	-	g	NA	NA	NA	s	p	-	Gr
37	sri	-	g	NA	NA	NA	f	a	-	Gr
38	Kh	-	NA	NA	NA	NA	s	v	-	*
39	Gh	-	NA	NA	NA	NA	s	v	+	*
40	Td	-	NA	NA	NA	NA	s	a	-	*
41	Tth	-	NA	NA	NA	NA	f	d	-	*
42	ddh	-	NA	NA	NA	NA	f	d	+	*
43	Bh	-	NA	NA	NA	NA	s	l	+	*
44	Zh	-	NA	NA	NA	NA	f	l	+	

Legend

V/C	Vowel / Consonant	(+) Vowel, (-) Consonant
VL	Vowel Length	(s)hort, (l)ong, (d)ipthong, (s)chwa, (g)eminate
VH	Vowel Height	(h)igh, (m)id, (l)ow
VF	Vowel Front ness	front, mid, back
LR	Lip Rounding	(+) Yes, (-) No, NA Not Applicable
TC	Type of Consonant	stop, fricative, affricative, nasal, liquid
PA	Place of Articulation	(l)abial, (a)lveolar, (p)alatal, (l)abio-dental, (d)ental, (v)elar, (g)lottal
CV	Consonant Voicing	(+) Yes, (-) No, NA Not Applicable
Gr	*Grantha*	Phones borrowed from Sanskrit
*		Phone not available in Tamil

Acknowledgement

This project is sponsored by Tamil Virtual University, Chennai under the Tamil Software Development Funding (TSDF) scheme.

The authors would like to thank Central Institute of Indian Languages (CIIL), Mysore, India for providing the Tamil text corpus.

References

[1] A. Nayeemulla Khan and B. Yegnanarayana, Development of Speech Recognition System for Tamil for Small Restricted Task, *Proceedings of National Conference on Communication*, India, 2001.

[2] M. Plauche, N. Udhyakummar, C. Wooters, J. Pal, and D. Ramachadran, Speech Recognition for Illiterate Access to Information and Technology, *Proceedings of First International Conference on ICT and Development*, 2006.

[3] S. Saraswathi and T. V. Geetha, Implementation of Tamil Speech Recognition System Using Neural Networks, *Lecture Notes in Computer Science*, Vol. 3285, 2004.

[4] C. S. Kumar and Foo Say Wei, A Bilingual Speech Recognition System for English and Tamil, *Proceedings of Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing*, 2003 and the *Fourth Pacific Rim Conference on Multimedia*, vol. 3, 2003, pp. 1641 – 1644.

[5] R. P. Lippmann, E. A. Martin, and D. P. Paul, Multi-style training for robust isolated-word speech recognition, Proc. *IEEE Int. Conf. Acoustics, Speech, Signal Processing*, April 1987, pp. 705-708.

[6] L. R. Rabiner, J. G. Wilpon, and F. K. Soong. High performance connected digit recognition using hidden Markov models, presented at the *IEEE Int. Conf. Acoustics, Speech, Signal Processing*. April 1988.

[7] L. R. Bahl, P. F. Brown, P. V. De Souza, and R. L. Mercer, Acoustic Markov models used in the Tangora speech recognition system, presented at the *IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Apr. 1988.

[8] D. B. Paul and E. A. Martin, Speaker stress-resistant continuous speech recognition, presented at the IEEE *Int. Conf. Acoustics, Speech, Signal Processing*, Apr. 1988.

[9] Kai-Fu Lee, Context Dependent Phonetic Markov Models for Speaker Independent Continuous Speech Recognition, *IEEE Transactions on, Acoustics, Speech and Signal Processing*, Volume 38, No. 4, 1990, pp 599-609.

[10] L. R. Bahl, R. Bakis. P. S. Cohen, A. G. Cole, F. Jelinek. B. L. Lewis, and R. L. Mercer, Further results on the recognition of a continuously read natural corpus, presented at the *IEEE lnt. Conf. Acoustics, Speech, Signal Processing*, Apr. 1980.

[11] R. M. Schwartz, Y. L. Chow, S. Roucos. M. Krasner. and J. Makhoul, Improved hidden Markov modeling phonemes for continuous speech recognition, presented at the *IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Apr. 1984.

[12] Aravind Ganapathiraju, Jonathan Hamaker, Joseph Picone, Mark Ordowski and George R. Doddington, Syllable Based Large Vocabulary Continuous Speech Recognition, *IEEE Transactions on Speech and Audio Processing*, Volume 9, No. 4, pp 358-366, 2001.

[13] T.Nagarajan, Hema A.Murthy and Rajesh M.Hegde, Segmentation speech into syllable-like units, EUROSPEECH-2003, pp. 2893-2896.

[14] Sarada, G., L., Hemalatha, N., Nagarajan, T.,Murthy, Hema A., Automatic Transcription of Continuous Speech using Unsupervised and Incremental Training, ICSLP-2004, October, Korea, 2004

[15] T.Nagarajan, V.Kamakshi Prasad and Hema A.Murthy, The minimum phase signal derived from the magnitude spectrum and its applications to speech segmentation, in Sixth Biennial Conference of Signal Processing and Communications, July 2001

[16] Lakshmi. A, Hema A. Murthy, A Syllable based continuous speech recognizer for Tamil, *SPCOM* December 2004

[17] Frederick Jelinek, *Statistical Methods for Speech Recognition*, MIT Press, (ISBN 0262100665), 1997

[18] Mari Ostendorf, Vassilios V. Digalakis, and Owen A. Kimball, From HMM To Segment Models: A Unified View of Stochastic Modeling for Speech Recognition, *IEEE Transactions on Speech and Audio Processing*, Volume 4, No. 5, 1996, pp 360-378.

[19] Xuedong Huang, Alex Acero and Hsiao-Wuen Hon, *Spoken Language Processing – A Guide to Theory, Algorithm and System Development*, Prentice Hall PTR (ISBN 0-13-022616-5), 2001

[20] Daniel Jurafsky, James H. Martin, *Speech and Language Processing*, Pearson Education, (ISBN 8178085941), 2002

[21] Shrikant Narayanan, Dani Byrd and Abigail Kaun, Geometry, Kinematics and Acoustics of Tamil Liquid Consonants, *Journal of Acoustical Society of America*, 106(4), Pt. 1, 1999, pp 1993-2007.

[22] Harold F. Schiffman, *A Reference Grammar of Spoken Tamil*, Cambridge University Press (ISBN-10: 0521027527), 2006

[23] Balasubramanian T, Timing in Tamil, *Journal of Phonetics*, Volume 8, 1980, pp 449 – 467.

[24] Paul Lamere, Philip Kwok, William Walker, Evandro Gouvea, Rita Singh, Bhiksha Raj and Peter Wolf, Design of the CMU Sphinx-4 Decoder in *EUROSPEECH 2003*, 2003

[25] Herve Bourlard, Hynek Hermansky and Nelson Morgan, Copernicus and the ASR Challenge - Waiting for Kepler, *Proceedings of ARPA Speech Recognition Workshop*, Arden House, New York, 1996

[26] Herve Bourlard, Hynek Hermansky and Nelson Morgan, Towards Increasing Speech Recognition Error Rates, *Speech Communication, Elsevier*, No. 18, 1996 pp 205-231.