Issues in developing LVCSR System for Dravidian Languages: An exhaustive case study for Tamil

G. Bharadwaja Kumar
Department of Computing Science & Engineering
Vellore Institute of Technology, Chennai Campus,
Chennai - 600127, India

Melvin Jose Johnson Premkumar
Department of Computer Science
Madras Institute of Technology, Anna University,
Chennai - 600044, India

ABSTRACT

Research in the area of Large Vocabulary Continuous Speech Recognition (LVCSR) for Indian languages has not seen the level of advancement as in English since there is a dearth of large scale speech and language corpora even today. Tamil is one among the four major Dravidian languages spoken in southern India. One of the characteristics of Tamil is that it is morphologically very rich. This quality poses a great challenge for developing LVCSR systems. In this paper, we have analyzed a Tamil corpora of 10 million words and have exhibited the results of a type-token analysis which implies the morphological richness of Tamil. We have demonstrated a grapheme-to-phoneme (G2P) mapping system for Tamil which gives an accuracy of 99.56%. We have shown the impact of important parameters such as absolute beam width, language weight, number of gaussians and the number of senones on speech recognition accuracy for limited vocabulary (3k). We have presented the results of large open vocabulary speech recognition task for vocabulary sizes of 30k, 60k and 100k on the speaker independent task. The Out Of Vocabulary (OOV) rates are 20.2%, 15.8%, 12.8% respectively. The accuracies are 43.59%, 47.11% and 43.52% respectively.

General Terms:

Computer Science, Artificial Intelligence

Keywords:

Speech Recognition, Tamil, Sphinx, Large Vocabulary

1. INTRODUCTION

Recently there is a growing interest in ASR for Indian languages. Initial work on large vocabulary speech recognition started with Hindi in early years of the previous decade. In [1], the authors have conducted large-vocabulary continuous speech recognition experiments in Hindi using IBM ViaVoice speech recognizer. For a vocabulary size of 65000 words, the system gives a word accuracy of 75% to 95%.

In [2], large vocabulary speech recognition for three different languages such as Marathi, Telugu and Tamil on different environments like land line and cellphone have been conducted. The vocabulary size used in these experiments varies from 14000 to 26000. They have obtained word error rates about 20.7%, 19.4% and 15.4% over land line data and 23.6%, 17.6% and 18.3% over cellphone for Marathi, Tamil and Telugu respectively. [3] used Hidden Markov Model tool kit for Bengali continuous speech recognition. They obtained an average recognition rate of 76.33% for male speakers and 52.34% for female speakers.

In [4] the authors investigate the effect of sharing the acoustic models across Tamil and English for effectively modeling the acoustic space of these languages, without having to model each of these languages separately. They conjectured that this had the effect of reducing the computational cost on the search engine as they had used only one acoustic model for many languages. They obtained word recognition accuracy of 61.61% and 64.42% for Tamil without and with adaptation respectively.

In [5], the authors have carried out experiments based on word level and triphone models for Tamil speech recognition and achieved 88% accuracy over limited data. They have also tried context independent syllable models for Tamil speech recognition [6] which under-performed when compared to context dependent phone models.

There are some attempts to build acoustic models at syllable level for Indian languages. In [7], authors proposed group delay based algorithm to automatically segment and label continuous speech signal into syllable-like units for Indian languages. The syllable recognition performance is about 42.6% and 39.94% for Tamil and Telugu respectively. The new feature extraction technique proposed by them that uses features extracted from multiple frame sizes and frame rates improves recognition performance to 48.7% and 45.36%, for Tamil and Telugu respectively.

In [8] an algorithm for segmentation based speech recognition was presented. This approach segments the words from the speech followed by characters from words. Neural networks based on back propagation algorithm was used to train and identify the segmented characters.

In [9], a modified version of the text independent phoneme segmentation algorithm proposed by Guido Aversano for their speech recognition experiments has been used. In [10], they analyzed the effect of enhanced morpheme-based trigram model with Katz back-off smoothing when compared to the word-based language models (LMs). The word error rates for word based trigram based models obtained in news and politics domain are 13.8% and 25.04% compared to 12.9% and 23.9% for morph based trigram models.

Although many experiments have been conducted to explore conventional approaches like phoneme-based models [6] and

syllable based models [5] using Sphinx and unconventional approaches like group delay based speech segmentation [7], speech recognition results in Tamil are still not comparable to work done in English either in terms of vocabulary size or in terms of corpus size. Also we can observe from the literature that many works on Indian languages reported the high accuracy results on either small vocabularies or the test data without Out Of Vocabulary words. Hence, there is a need for comprehensive experiments on large vocabulary speech recognition in Tamil.

In this paper, we have reported the results of our experiments on large vocabulary continuous speech recognition for Tamil. We use SphinxTrain for training acoustic models and Sphinx-4 as decoder for large vocabulary continuous speech recognition. We have studied the performance of the speech recognition system with varying vocabulary size, absolute beam width, number of gaussians and language weights. We also discuss the results of our grapheme-to-phoneme mapping experiments and the nature of the Tamil language.

2. NATURE OF THE TAMIL LANGUAGE

Tamil is one among the four major languages of the Dravidian Language family spoken in the southern part of India. This language is predominantly spoken in the state of Tamil Nadu of the Indian subcontinent and also in other parts of the world like Singapore, Sri Lanka and Malaysia. Dravidian languages are among the most complex languages in the world at the level of morphology, perhaps comparable only to Finnish and Turkish [11]. This is because of agglutination and complex sandhi rules. This can be attributed to the fact that a significant part of grammar that is handled by syntax in English (and other similar languages) is handled within morphology in Tamil (and other Dravidian languages). Phrases including several words (that is, tokens) in English would be mapped on to a single word in Tamil. External sandhi (that is, conflation between two or more complete word forms) and compounding add to the numbers. Even long sentences in English reduce to a single word in Tamil after applying a series of sandhi rules. In the type-token analysis of Dravidian language corpora, naturally we will see very large number of types and the type-token ratio is also expected to be very high. These are not simple concatenations or juxtapositions of complete words written without intervening spaces as is the convention in some languages of the world. These words are made up of several morphemes conjoined through complex morphophonemic processes. Figure 1 lists a few examples which illustrate the morphological complexity of Tamil.

will be going to	செல்கிறேன்
will be accepting	ஏற்றுக்கொள்வேன்
was unable to come	வரமுடியவில்லை
may not be accepting	ஏற்றுகொள்ளமுடியாது
They danced sang and enjoyed	அடிபாடிமகிழ்ந்தனர்
Are you saying that hot water is not there?	சுடுநீரில்லையா

Fig. 1. Examples of morphological complexity

In this paper, we have carried out type-token analysis of Tamil corpus. This corpus includes articles extracted from latest Wikipedia dump, CIIL Tamil corpus of 3 Million Words and news paper articles. This corpus accrues to 10 Million words (Tokens). From the Figure 2, we have shown the coverage analysis of most frequent types on the corpus. It can be seen that the number of distinct words (types) are 1.2 Million which is very high when compared to languages like English. Also, it can be observed that most frequent 100,000 types cover only 84% of the corpus. But, from the type token analysis of 100 Million word British National Corpus in English, it was noted that only most frequent 20,000 words cover 94.76% of the entire corpus [12]. It can be conjectured from the given analysis that Tamil is morphologically rich.

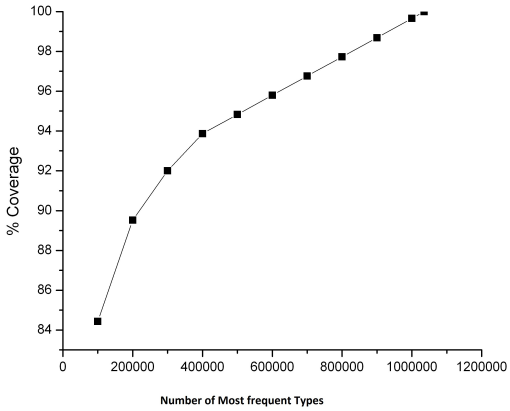

Fig. 2. Coverage analysis

Another important characteristic of Tamil is that it is a relatively free word order language. For example, the English sentence ”Rama Killed Ravana” can be translated to the following sentences without change in their meaning as shown in Figure 3.

ராமர் ராவணனை கொலை

ராவணனை ராமர் கொலை

ராவணனை கொலை ராமர்

ராமர் கொலை ராவணனை

Fig. 3. Possible Tamil Equivalents

One more characteristic of Tamil is that the grapheme-to-phoneme mapping is non trivial when compared to other Dravidian languages. Unlike many other Indian languages, Tamil script has lesser number of consonants. It has neither aspirated nor voiced stop consonants in written script. But the voiced stops are present in the spoken language as allophones. In addition, the voicing of stop consonants is governed by strict rules. They are unvoiced if they occur word-initially or in gemination.

3. EXPERIMENTAL SETUP

Here, we explain the primary modules in building our Tamil continuous speech recognition system. The training phase involves two basic tasks. The first task is creating the pronunciation lexicon and the second task is building the acoustic model. The testing phase requires the creation of language model which is used

along with the acoustic model and the pronunciation lexicon that are obtained from the training step.

3.1 Creating Pronunciation Lexicon

Phonetizers or G2P converters are required to convert the text corpus into its phonetic equivalent as well as to generate pronunciation lexicon. The lexicon is a representation of each entry of the vocabulary of the ASR system in its phonetic form [2]. Most of the Indian language scripts are phonetic in nature i.e. there exists a one-to-one correspondence between the orthography and pronunciation in these languages. Also, in Indian languages stress does not have any phonemic value and all syllables are pronounced with the same emphasis. Tamil script is also phonetic in nature but there are many exceptions. As stated in Section 2, the speech to sound mapping for Tamil is non trivial. In this paper, we have used two open-source G2P converters namely, Sequitur G2P toolkit [13] which is based on joint-sequence models and Phonetisaurus [14, 15] which is based on Weighted Finite State Transducers (WFST).

3.2 Building Acoustic Models

We have used SphinxTrain developed by CMU [16] for building our acoustic model. SphinxTrain supports the extraction of MFCC features from the audio files. We have used MFCC's, their first and second derivatives as features. Training of context dependent 3-state HMM models using SphinxTrain includes the following processes: (a) Building Context Independent HMM models, (b) Building Context Dependent HMM models (c) Building decision trees and parameter sharing by using tied states (Senones): A senone is also called a tied-state and is obviously shared across the triphones which contributed to it. This reduces the number of HMM parameters to be trained. We have trained our models with 500 senones. (d) Mixture generation: Here, the state-output distributions of HMMs are modeled using Gaussian mixture models to account for multivariate and multi-modal data due to variations in speaker, accent and gender. The number of HMM parameters to be estimated as well as the amount of training data increases as the number of Gaussians in the mixture increases. However, increasing its value also results in finer models, which can lead to better recognition. The number of Gaussians that have been used in GMM modeling is 64 in our experiments.

3.3 Building Language Models(LMs)

In the present work, we have built LMs from text corpora and not from the transcribed speech corpus. Text normalization is one of the challenging tasks for building LMs from the text corpora. The text normalization process defines what is considered to be a word by the recognition system. Initially, text has to be segmented into sentences. This has been done using the rule based sentence segmentation system developed by us. Abbreviations like ru:. (which stands for rupees) and special symbols like % (which stands for per cent) have been expanded. All the punctuation marks, special symbols have been removed except symbols related to numerals.

We have used trigram LMs in our speech recognition tasks. We have built LMs using the open source CMU language modeling toolkit [17]. It uses Good-Turing discounting method with back-off as smoothing algorithm.

3.4 Decoding

We have used Sphinx-4 decoder for our speech recognition tasks. The Sphinx-4 speech recognition framework has been developed at CMU which is designed with a high degree of flexibility and modularity [18]. There are three primary modules in the Sphinx-4 framework [19] : the 'FrontEnd', the 'Decoder', and the 'Linguist'. The 'FrontEnd' takes one or more input signals and parameterizes them into a sequence of Features. FrontEnd supports extraction of features like MFCC, PLP, LPCC etc. The 'Linguist' translates any type of standard LM, along with pronunciation information from the lexicon and structural information from one or more sets of acoustic models, into a search graph. Finally, the 'Decoder' block contains the search manager which perform search algorithms such as frame synchronous Viterbi, A*, bi-directional, and so on.The search manager uses the features from the 'FrontEnd' and the search graph from the 'Linguist' to perform the actual decoding and for generating results.

We have used word pruning breadth first search manager as search strategy in our experiments. Here, the pruning can be done by specifying values for absolute beam width and other parameters. Absolute beam width keeps only the specified number of elements in the active list. Since, it affects the number of active hypotheses, absolute beam width is an important factor which affects the speech recognition performance. We have carried out experiments with varying beam widths from 500 to 20000.

Although maximum a posterior probability estimation will be calculated using the trigram LM, in practice the language probability is raised to an exponent by using language weight during the recognition. It decides how much relative importance will be given to the actual acoustic probabilities of the words in the hypothesis. A low language weight gives more leeway for words with high acoustic probabilities to be hypothesized, at the risk of hypothesizing spurious words. Optimal values typically lie between 6 and 13 [20]. We have carried out experiments by varying language weight between 0 to 13.

4. GRAPHEME-TO-PHONEME EXPERIMENTS

In the present work, our phone set contains 41 phones. Figure 4 gives the list of vowels in Tamil and their corresponding phonetic mapping in our system. Figure 5 and Figure 6 summarize our grapheme-to-phoneme mapping convention for consonants. Some consonants have allophones based on the context. The figures also contain the list of all possible allophones of each phoneme for a given grapheme. We have manually created a pronunciation lexicon that contains a vocabulary of about 35,000 words. This lexicon has been used for training our G2P models.

Previous work on G2P conversion for Tamil was done in [21] and [22] where the authors explore rule-based and Decision Tree Learning-based approaches. In [23], the authors conclude that Sequitur [13] and Phonetisaurus [14, 15] perform better than the other existing G2P techniques for LVCSR tasks. Here we compare the performance of these two tools for Tamil LVCSR.

In Phonetisaurus, weighted finite-state transducers are used for decoding as a representation of a graphone-based n-gram LM trained on data aligned by an advanced $M : M$ alignment algorithm [14]. The n-gram can be trained using any standard LM Toolkit in which Kneser-Ney discounting with interpolation is used for smoothing. Decoding is done using OpenFST [24]. In Sequitur G2P, a joint n-gram model is used. The graphemic model $p(q_i|q_i, ..., q_1)$ is estimated using ML EM training on an existing pronunciation dictionary. For the possibly non-unique segmentation into graphones, a maximum approximation is applied.

We have tested the performance of our models on a vocabulary of 5000 words, which is independent of the training data of the

a	a:	i	i:	u	u:	e	e:	y:	o	o:	v:
அ	ஆ	இ	ஈ	உ	ஊ	எ	ஏ	ஐ	ஒ	ஓ	ஔ

Fig. 4. Vowels in Tamil and their corresponding phonetic mapping

k	g	h	x	c	s	j	x:	t	d	n:	q	f	n	n;	p	b	p;	m
க்			ங்	ச்			ஞ்	ட்		ண்	த்		ந்		ப்			ம்

Fig. 5. Consonants in Tamil and their corresponding phonetic mapping

y	r	l	v	z	l:	r:	w	w:	n
ய்	ர்	ல்	வ்	ழ்	ள்	ற்			ன்

Fig. 6. Consonants in Tamil and their corresponding phonetic mapping

Table 1. SequiturG2P vs. Phonetisaurus

Vocabulary Size	Sequitur G2P Phoneme Accuracy(%)	Phonetisaurus Phoneme Accuracy(%)
10000	97.41	98.05
18000	98.65	98.39
35000	99.56	99.05

G2P model. The 5000 vocabulary was manually phoneticized and compared with the output of our G2P model on the same vocabulary. Table 1 gives the comparison between the results obtained by both Sequitur and Phonetisaurus. It is to be noted from the table that both tools give very similar Phoneme Error Rates (PER), however, the training time taken by Phonetisaurus (minutes) was much lower than that taken by Sequitur (hours). Thus we use Phonetisaurus for building our pronunciation lexicon.

The main challenge of the G2P system was the confusions between allophones. Notice in Figures 4 and 5 the consonants that have multiple allophones. We have given a few examples of the correct phoneme representation given by our G2P model for native Tamil words in Figure 7 and borrowed words from other languages in Figure 8. In Figure 7, it has successfully resolved the confusion between 'k' and 'g' in the first word. In the second word, it has resolved the confusion between 'f' and 'q'. It can be observed from Figure 8 that our model has also learned to successfully decode borrowed words. Note that for the word "Graphics" it has successfully resolved the confusion between using 'k' and 'g'. We have provided few examples of the mistakes done by our G2P model in Figure 9. Of the 7 words for which our system went wrong, 4 are borrowed words. The problem with borrowed words is that there can be two different transcriptions for them, one an Tamil-like interpretation of them, the other the original pronunciation (English or Hindi etc) as such. This makes the transcription of borrowed words a tricky task. However, further improvement with regard to borrowed words can be done if a set of them were transcribed manually and trained with the other Tamil words. Our system has incorrectly chosen between, 'f' and 'q' twice, 'k' and 'g' twice, 'p' and 'b' once, 't' and 'd' once, and 'c' and 's' once. Table 1 outlines the performance of our system for increasing training vocabulary sizes of 10k, 18k and 35k respectively. It can seen that there is an improvement of 2% when the training vocabulary size is increased gradually from 10k to 35k. Further improvement on our G2P model can be done if we can increase the training size by including words with more than one allophonic confusion. This can make the system to address the problem of multiple allophones effectively.

அங்கீகரிக்கப்பட்ட	a x g i: g a r i k k a p p a t t a
ஆதங்கப்படுவதுண்டு	a: f a x g a p p a d u v a f u n: d u
இழிவுபடுத்தியுள்ளதாக	i z i v u p a d u q q i y u l: l: a f a: g a
இழுபறிகளுக்கிடையிலும்	i z u p a r: i g a l: u k k i d y: y i l u m
ஏற்படுத்தப்பட்டிருந்தது	e: r: p a d u q q a p p a t t i r u n f a f u
ஒதுக்கித்தரும்படி	o f u k k i q q a r u m b a d i
ஒப்படைக்கப்பட்டிருந்தது	o p p a d y: k k a p p a t t i r u n f a f u

Fig. 7. Pure Tamil Words

ஜசோடோப்புகளிலிருந்து	y: s o: t o: p p u g a l: i l i r u n f u
ஜதராபாத்	y: f a r a: b a: q
கிராபிக்ஸ்	g i r a: k: p i k s
டிரைவர்களுக்கு	d i r y: v a r g a l: u k k u
டெக்னாலஜீஸ்	t e k n a: l a j i: s
டைனமைட்	d y: n a m y: t

Fig. 8. Borrowed Words

5. EXPERIMENTS AND RESULTS

Our speech recognition experiments have been carried out on speech data recorded with a sampling rate of 16KHz and a bit rate of 16. The data collection has been done in a general lab environment and has been recorded with an ordinary microphone over the computer. The training data consists of phonetically rich sentences from newspapers and Thirukkural. Thirukkural is a Tamil classic, consisting of 1330 couplets or kurals. The speech data has been collected from 100 speakers where each speaker has spoken a minimum of 3 minutes. Finally, sentence level transcriptions were done automatically. The total speech

எகிப்து	e g i p f u
எக்ஸ்பிரஸ்தான்	e k s p i r a s q a: n
எடுக்காதீர்கள்	e d u k k a: f i: r k a l:
எடுக்கிறீர்களாமே	e d u k k i r: i: r k a l: a: m e:
எழுதும்போதே	e z u f u m p o: f e:
ஏர்லைன்சினால்	e: r l y: n c i n a: l
ஏஜன்டுகள்	e: j a n d u g a l:

Fig. 9. Mistakes by Grapheme-to-Phoneme System

data accumulates to 16 hours.

In the first stage of experiments on continuous speech recognition, we have observed the performance of the speech recognition system on limited vocabulary by varying four important parameters namely number of Gaussians, number of senones, absolute beam width and language weight. The performance of the speaker dependent recognition task has been evaluated on a test corpus of 200 sentences spoken by 10 speakers where every one of them spoke 20 sentences. The speaker independent recognition performance was evaluated on a corpus of 370 sentences spoken by 25 speakers where each person has spoken a minimum of 10 sentences.

We ascertain the importance of varying number of gaussians and number of senones in our experiments. We have gauged the performance of the system for number of gaussians 4, 8, 16, 32, 64, and 128 and for number of senones 500, 1000, and 2000. We also ascertain the importance of both the absolute beam width and language weight parameters in our experiments. We have gauged the performance of the system for absolute beam width values of 500, 5000, 10000, and 20000 and for language weights of 0, 1, 8, 11, and 13. In the case of language weights 0 and 1, there is more leeway for the recognizer to choose words with high acoustic probabilities since the importance of the LM has been curtailed. This drastically reduces the performance of the system. Increasing the absolute beam width will increase the computational cost incurred.

In these initial experiments where several parameters that can effect speech recognition performance are varied, all the sentences in the test set have been included in the LM and in the lexicon to avoid OOV (Out Of Vocabulary) words. In our result tables 'WER' is an acronym for Word Error Rate and 'Accuracy' is calculated as 100 minus 'WER' in all our calculations.

In our experimental results shown in table 2, we found that GMM model estimated with 64 gaussians gives the best performance. So we have used 64 gaussians in our further experiments. There is a drastic performance drop for GMM model with 128 gaussians because the amount of training data is not enough to estimate parameters of GMM model with 128 gaussians. From the table 3, we found that results are better for 500 senones when compared to 1000 and 2000 senones. This is due to the fact that in our training data many sentences were repeated by many speakers. The vocabulary and context in our training data is not enough to build a large number of senones. The results in table 4 establish the fact that a low language weight gives more leeway for words with high acoustic probabilities to be hypothesized, at the risk of hypothesizing spurious words. The recognition performance increases with increase in language weight from 8 to 13. Even the large absolute beam width gives more accuracy as shown in table 5, in order to achieve a trade-off between low computational cost and high recognition accuracy we have chosen the absolute beam width and language weight parameters to be 5000 and 11 respectively in all our future experiments.

In our next set of experiments, we have studied the effect of varying the vocabulary size on speech recognition accuracy. The absolute beam width and language weight values were chosen to be 5000 and 11 in this case. Here, we have carried out our experiments using the LM that has been built from the large Tamil corpus crawled from the web. This corpus contain 400k types and more than 10 million tokens. We have chosen the top 30k, 60k and 100k words for building the language model as well as vocabulary. We have used VariKN language modeling tool [25] for this task. These experiments are carried out for medium (30k), large (60k) and very large (100k) vocabulary sizes respectively.

Table 2. Performance of our Tamil Speech Recognition System (Varying Number of Gaussians)

Number of Gaussians	Accuracy (%)	Time Taken
4	91.75	44min 20sec
8	92.94	40min 57sec
16	94.88	24min 6sec
32	94.96	20min 38
64	95.52	18min 25sec
128	9.94	18min 25sec

Table 3. Performance of our Tamil Speech Recognition System (Varying Number of Senones)

Number of Senones	Accuracy (%)	Time Taken
500	95.52	44min 20sec
1000	94.21	40min 57sec
2000	91.73	24min 6sec

Table 4. Performance of our Tamil Speech Recognition System (Varying Language Weight)

Language Weight	Accuracy (%)	Time Taken
0	22.60	44min 20sec
1	53.47	40min 57sec
8	93.36	24min 6sec
11	95.52	20min 38sec
13	95.56	18min 25sec

Table 6 reports our recognition results, OOV rate and PPL for varying vocabulary sizes namely medium (30K), large (60k) and very large vocabulary (100K) on the speaker independent task. We use SRILM Toolkit [26] for all these LM experiments. Figure 10 shows some errors that occur due to the agglutinative nature of Tamil. The hypothesis contains the agglutinative form of the word in the reference or vice-versa but this may still be counted as an insertion or deletion and hence contributes to the low accuracy.

Table 5. Performance of our Tamil Speech Recognition System (Varying Absoulte Beam Width)

Absolute Beam Width	Accuracy (%)	Time Taken
1000	91.99	10min 46sec
5000	95.52	20min 38sec
10000	95.65	27min 5sec
20000	95.79	35min 32sec

Table 6. Performance on Open Vocabulary Speaker Independent Task (*Total words: 5199, Ins : Insertions, Sub: Substitutions, Del: Deletions, Acc: Accuracy)*

Vocabulary Size	Ins.	Sub.	Del.	Acc. (%)	OOV (%)	PPL
30k	514	2324	95	43.59	20.2	2629
60k	406	2196	148	47.11	15.8	3124
100k	212	2353	278	45.32	12.7	3710

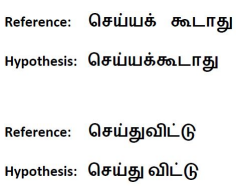

Fig. 10. Some errors due to Tamil Morphology

6. CONCLUSION AND DISCUSSION

In this paper, we present our work on LVCSR for Tamil language. We then present our G2P mapping results which measures 99.56%. Our G2P system is very successful in phoneticizing words which possess complex phonetic contexts. This high accuracy entitles our system to automatically create pronunciation dictionaries for a vocabulary of any size. We also investigate the effect of absolute beam width and language weight parameters on the recognition accuracy. The absolute beam width parameter improves the accuracy at the cost of increased computational cost. It has been noted that the role of the language model is very significant in propelling the accuracy of the recognition task.

In literature, many previous works on Indian languages have reported the results using an LM that contains the test transcriptions. This eliminates the problem of OOV which in turn drastically reduces the LM perplexity and in turn the word error rate. Our system gives an accuracy of 80.86% for the large vocabulary (65k) when there are no OOV words. This result is on par or better than the results reported in the literature for different Indian languages. But, these numbers are not real estimates of the performance of these systems in the context of large open vocabulary continuous speech recognition. Also, in the case of Dravidian languages the OOV has significant impact on the accuracy since these languages are agglutinative. Hence our experimental results are reported for the open vocabulary with the existence of OOV. The OOV rates are 20.2%, 15.8%, 12.8% for 30k, 60k and 100k vocabularies respectively. The accuracies for medium, large and very large vocabulary are 43.59%, 47.11% and 45.32% respectively. The lower accuracy for the 100k system is due to increase in search space caused by the additional words. The high perplexities reported confirm the morphological richness of Tamil. Our best system (60k) has an OOV rate of 15.8 % which is still very high and may lead to low accuracy. In future, we would like to try morpheme based LMs for Tamil and also explore ways of recognizing OOV words to improve the performance.

7. REFERENCES

[1] Kumar, M., Rajput, N., Verma, A. (2004). A large-vocabulary continuous speech recognition system for Hindi. IBM Journal of Research and Development, 48(5/6):703-710.

[2] Kumar, R., Kishore, S., Gopalakrishna, A., Chitturi, R.,Joshi, S., Singh, S., Sitaram, R.(2005). Development of Indian language speech databases for large vocabulary speech recognition systems. International Conference on Speech and Computer (SPECOM) Proceedings.

[3] Banerjee, P., Garg, G., Mitra, P., Basu, A. (2008). Application of triphone clustering in acoustic modeling for continuous speech recognition in Bengali. ICPR-2008 Proceedings. pp. 1-4.

[4] Kumar, C.S., Wei, F.S. (2003). A bilingual speech recognition system for English and Tamil. ICICS-PCM 2003 Proceedings, Singapore.

[5] Thangarajan, R., Natarajan, A.M., Selvam, M. (2009). Syllable modeling in continuous speech recognition for Tamil language. International Journal of Speech Technology, 12(1):47-57.

[6] Thangarajan, R., Natarajan, A.M., Selvam, M. (2008). Word and triphone based approaches in continuous speech recognition for Tamil language. WSEAS Trans. Sig. Proc, 4(3):76-85.

[7] Sarada, G.L., Lakshmi, A., Murthy, H.A., Nagarajan, T. (2009). Automatic transcription of continuous speech into syllable-like units for Indian languages. Sadhana, 34(2):221-233.

[8] Chandrasekar, M., Ponnavaikko, M. (2008). Tamil speech recognition: a complete model. Electronic Journal on Technical Acoustics.

[9] Saraswathi, S., Geetha, T.V. (2010). Design of language models at various phases of Tamil speech recognition system. International Journal of Engineering, Science and Technology, 2(5):244-257.

[10] Saraswathi, S., Geetha, T.V. (2007). Comparison of performance of enhanced morpheme-based language model with different word-based language models for improving the performance of Tamil speech recognition system. ACM Transactions on Asian Language Information Processing (TALIP), 6(3):9.

[11] Kumar, G.B., Murthy, K.N., Chaudhuri, B.B. (2007). Statistical analysis of Telugu text corpora. IJDL, 36(2):71-99.

[12] Kumar, G.B. (2007). UCSG Shallow Parser: A Hybrid Architecture for a Wide Coverage Natural Language Parsing System. PhD thesis, Department of Computer & Information Sciences, University of Hyderabad, Hyderabad, India.

[13] Bisani, M., Ney, H. (2008). Joint-sequence models for grapheme-to-phoneme conversion. Speech Communication, 50(5):434-451.

[14] J. Novak, D. Yang, N. Minematsu, K. Hirose, "Initial and Evaluations of an Open Source WFST-based Phoneticizer", The University of Tokyo, Tokyo Institute of Technology.

[15] D. Yang, et. al., "Rapid development of a G2P system based on WFST framework", ASJ 2009 Autumn session, pp. 111-112, 2009.

[16] Group, R. (2008). Robust Group Tutorial. http://www.speech.cs.cmu.edu/sphinx/tutorial.html.

[17] Clarkson, P., Rosenfeld, R. (1997). Statistical language modeling using the CMU-Cambridge toolkit. ESCA Eurospeech Proceedings, 2707-2710.

[18] Lamere, P., Kwok, P., Walker, W., Gouva, R., Singh, R., Raj, B., Wolf, P. (2003). Design of the CMU sphinx-4 decoder. 8th European Conf. on Speech Communication and Technology (EUROSPEECH) Proceedings.

[19] Walker, W., Lamere, P., Kwok, P., Raj, B., Singh, R., Gouvea, E., Wolf, P., Woelfel, J. (2004). Sphinx-4: A flexible open source framework for speech recognition. Technical report, Sun Microsystems.

[20] Mosur, R. (2008). Sphinx-3 s3.X Decoder. http://www.cs.cmu.edu/ãrchan/s_info/Sphinx3/doc/s3_description.html.

[21] C. S. Kumar, Shunmugom V., Udhyakumar Nallsamy and Srinivasan R., "Automatic grapheme to phoneme converter for Tamil using rules", in proceedings International Conference On Speech and Language Technology, 2004.

[22] Udhyakumar Nallasamy, C. S. Kumar, Srinivasan R. and Swaminathan R., "Decision tree learning for automatic grapheme to phoneme conversion for Tamil", in proceedings SPECOM, 2004.

[23] S. Hahn , P. Vozila , M. Bisani, "Comparison of Grapheme-to-Phoneme Methods on Large Pronunciation Dictionaries and LVCSR Tasks", in Proceedings of Interspeech, 2012.

[24] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri, "OpenFst: a general and efficient weighted finite-state transducer library", Prague, Czech Republic, Jul. 2007, pp. 11?23.

[25] Vesa Siivola, Mathias Creutz and Mikko Kurimo: "Morfessor and VariKN machine learning tools for speech and language technology", Proceedings of the Interspeech, 2008.

[26] Andreas Stolcke, "SRILM - An Extensible Language Modeling Toolkit", in Proc. Intl. Conf. Spoken Language Processing, Denver, Colorado, September 2002.