FORMULATION OF SYLLABLE BASED PRONUNCIATION MODELS FOR TAMIL TEXT-TO-SPEECH SYNTHESIZER

Vaibhavi Rajendran[1] and G Bharadwaja Kumar[2*]

[1,2]School of computing Science and Engineering, Vellore Institute of Technology, Chennai Campus TamilNadu, India, 600127

_E-mail: vvaibavi@gmail.com[1], bharadwaja.kumar@vit.ac.in[2] (corresponding author)

DOI: https://doi.org/10.22452/mjcs.vol33no4.3

ABSTRACT

The primary aim of Human-Computer Interaction (HCI) is to deliver the power of computers and communication systems to people in an easily accessible and understandable form. HCI in a person's native/first language is always invigorate. Developing a Tamil Text-To-Speech (TTS) system will facilitate a convenient medium of interaction for people who speak Tamil language. This paper emphasizes on the development of pronunciation models, a vital component of a Tamil TTS. Developing a pronunciation model for Tamil is more arduous when compared to other languages due to the non-triviality between the letter to sound correspondence. Veritably, two syllable-based pronunciation models developed by us are discussed in this paper. First, is a syllable-centric rule-based pronunciation model that generates a well-founded training data which is ingrained into the second, Conditional Random Field (CRF) enforced model. It is evident that both of these models are dominions with a high Mean Similarity Score of 0.97 and 0.94 respectively in comparison to the other existing rule driven and data driven models in the literature. These syllable-based pronunciation models will enrich the performance of a Tamil TTS.

Keywords: Tamil Letter to Sound, Text To Speech, Conditional Random Fields, Machine Learning

1.0 INTRODUCTION

Technologies such as graphical user interface, speech recognition & synthesis, gesture & handwriting recognition, multimedia presentation and cognitive models help in establishing Human-Computer Interaction (HCI). English which is referred as a 'lingua franca', is a commonly accepted bridging language among people. Due to it's compliance as the global communication language, English is the widely preferred language in HCI. Tremendous efforts have been carried out to develop user interfaces in English, but alongside substantial efforts should be taken for developing user interfaces in a person's native language too. Primarily, a person usually learns to 'think, understand and communicate' with the help of his/her native language. Moreover, a native/first language is the one in which a person communicates the best as he/she has been exposed to it right from birth. Hence, HCI in a person's native/first language is always the most optimal choice. Tamil is the first language of communication for most of the people in Tamil Nadu, Pondicherry, Andaman & Nicobar in India. Tamil is also an official language of communication in countries like India, Sri Lanka, Malaysia and Singapore. Hence, developing a Tamil Text-To-Speech (TTS) system, a speech interface will facilitate a convenient medium of interaction for people in these regions. HCI in Tamil will remove the restraints on people who are not connected to a computer due to the misfortune of deprivation from English. The recent enforcement of digital revolution in India is a major probing factor to increase digital information management in the regional languages for the benefit of people of India. A Tamil TTS provides an ardent way to foster the digital information to all realms of people. As commonly quoted, "Knowledge is power" and if a person is repressed from gaining knowledge only due to language barrier, then it can be counterbalanced with HCI in his/her native language. The focal point of this paper is on the development of pronunciation models for Tamil language, a vital component of a Tamil TTS.

A 'word' is a meaningful element of written or spoken form in a language. The written form of the word is referred to as 'spelling' and the spoken form as 'pronunciation'. The spelling and pronunciation of a word are not always the same. Hence, a model which maps the spelling of a word to it's pronunciation is a requisite. The pronunciation model is subjected to the spelling of a word from which it has to render it's pronunciation using either a 'phone' or a 'syllable' as the basic sound unit. The orthographic representation of a phone and syllable are called as a 'phoneme' and 'syllabary' respectively. For a written form of the word, a sequence of phonemes or syllabaries will be rendered using a pronunciation model. On pronouncing/producing these phonemic or syllabic sequence the spoken form of the word is obtained. A pronunciation model developed based on phones

is termed as a Grapheme-To-Phoneme (G2P) converter and the one based on syllables is termed as a Letter-To-Sound (LTS) converter.

Depending on the nature of the language, one among the two sound units can be chosen to model the pronunciation of words. In general, Indian languages are syllable-timed languages where the syllables take approximately equal amounts of time to pronounce [1, 2, 3]. Since stress does not have any phonemic value in Indian languages, a syllable ingrained pronounciation model would be more appropriate.

Interestingly, the mapping between the characters of a word to it's appropriate sound unit representation is similar to a sequence labelling problem in pattern recognition. A sequence labelling problem can be branched down to several individual classification tasks between the characters and the sound unit representation. One algorithm which has proven to perform exquisitely well on sequence labelling problems is Conditional Random Fields (CRF). Owing to the supremacy of CRF in solving sequence labelling problems [4] and it's phenomenal performance in resolving G2P conversion problems [5], we have chosen CRF for developing a syllable ingrained pronunciation model for Tamil.

Since, the goal of current work is to develop a data driven pronunciation model using machine learning algorithms, a valid well-founded training data is essential to build the model. We have used a rule-based pronunciation model to obtain the training data. The output from the syllable-centric rule-based pronunciation model was fed to the data driven model developed using Conditional Random Fields. Veritably, two syllable-based pronunciation models developed by us are discussed in this paper:

1. A syllable-centric rule-based pronunciation model for Tamil

2. A syllable ingrained Conditional Random Field enforced pronunciation model for Tamil.

There are two factors which attribute to the novelty of the current work.

- First, a machine learning based pronunciation model for Tamil is still unsubstantial. A decision tree based approach developed on phonemes in [6] is the only indication of work towards machine learning based Tamil pronunciation model in the literature. Moreover, a 'syllable' ingrained machine learning based pronunciation model is still void for Tamil.

- Second, the training data fed to the data driven approach is generated using a syllable-centric rule-based model formulated by us, which discerns the need for hand - refinement of the words. Typically, the training data used for any data driven/machine learning approach is generally scrutinized by a hand-refinement process for it's correctness. Since the syllable-centric rule-based model formulated by us generates word pronunciations with a very low Word Error Rate (WER) and high Mean Similarity Score (MSS), it eliminates the need of a hand-refinement process.

The rest of the paper composes of the following sections: the next section holds a discussion of the related work, section 3 briefs on the characteristics of Tamil language, section 4 helps us view Tamil LTS conversion in a pattern analysis perspective, section 5 details on the usage of CRF in the development of the data driven pronunciation model for Tamil, section 6 holds the results and analysis on the syllable ingrained pronunciation models and finally the last section holds our conclusion and future work.

2.0 RELATED WORK

The application of Letter-To-Sound conversion is not just bound to a speech synthesizer, it's other noted applications are Automatic Speech Recognition (ASR), speech-based generation of spelling and predictive search. A significant amount of work has been carried out to resolve this letter to sound conversion problem. Some of the approaches in the literature are effective but with certain restrictions like a closed set of words. An approach with no restrictions and openness to input words is a requisite. Most of the existing approaches operate on phoneme as the sub-word unit whereas work on syllable as the sub-word unit is very minimal.

A very primitive approach is a dictionary look-up approach, where a set of words of a language is placed along with it's pronunciation given as a sequence of strings or symbols. The input word is looked up in the dictionary and pronounced. But the main drawback of this approach is the size of the dictionary. Words in any language are countless and a dictionary cannot hold all words in the language. A dictionary look-up approach used for Tamil is discussed in [7], but the approach suffered from instances of Out-Of-Vocabulary (OOV) word [8]. From a

single root word, hundreds of words can be formed in Tamil due it's highly inflectional and agglutinative nature [9, 10], thus making the usage of a dictionary look-up even more difficult for Tamil. To overcome this setback much more sophisticated approaches came into usage. These approaches can generally be categorized into rule driven and data driven approaches. Some of the popular rule driven approaches are discussed in [11] for English, [12] for Urdu and in [13], [14], [15] for Tamil. In a rule driven approach, a possible set of rules are enlisted with the help of a linguist. For each letter in the input word, the list of rules in the rule list is skimmed through and the matching rule is applied onto the letter for producing the pronunciation [16,17]. It is the simplest approach for resolving the LTS problem. However, the implausibility in framing a complete set of rules for any language is the major obstacle. Even if we confide in a linguist to form a rule base, forming a complete rule base is not plausible for any language. This frailty directed the researches towards robust machine learning algorithms.

A data driven approach is totally dependent on the training instances for learning. A data driven approach may follow a top-down, bottom-up or a mixed strategy to solve the LTS conversion problem. Decision tree [16] and Bayesian networks based G2P [18] were the two initial approaches with a top-down strategy for resolving the pronunciation of words and they work fairly well when input words are similar to the training set. If the similarity between the input word and training set deteriorates so does the performance [8]. A decision tree was experimented on syllables for English but due to improper syllable boundary detection, the results were not as expected [19]. Researchers then explored the bottom-up strategy to generate the pronunciation, one example of this strategy is Pronunciation by Analogy (PbA) approach. In this approach, even new or rare words has a higher probability of getting pronounced properly or at least getting a pronunciation a little near to the actual pronunciation. Sometimes similarly written words tend to have a totally different pronunciation. But the pronunciation probability of such words has a cutting edge on the robustness as the approach depends on the similarity measure between spelling of the words [12]. Another approach which is similar to PbA, applied to solve G2P and which follows the bottom-up strategy is pronunciation by latent analogy [20]. Latent analogy also experiences the same drawback as PbA and is more suitable for pronunciation of rarer context words.

A joint-grapheme-phoneme n-gram approach [21] is a mixed approach and it tries to balance out the drawbacks of top-down and bottom-up approaches [22]. The major constraint in this joint-sequence model is the need of a huge data set. Moreover, a joint sequence model requires two additional models such as: Expectation Maximization based alignment model - which performs mapping between the letter and sound unit [23]; translation model - which generates the n-gram model using a Weighted Finite State Transducer (WFST) [24]. These additional models in a joint sequence model creates an overhead for resolving the G2P complexity. On the contrary, Long-Short Term Memory (LSTM) neural network model eliminates the overhead of constructing such additional models [25]. Neural networks and Recurrent Neural Networks [26] have also been proposed for solving LTS conversion problems. In fact, a Recurrent Neural Network Toolkit is available for performing G2P conversion and is discussed in [27]. Investigations proved that multi-layer perceptron performed better among the different types of neural networks. But the questionable factor about the LSTM neural network and other neural networks is the amount of time it takes to process the data. LSTM takes nearly one whole day to train a large model on phonemes itself and hence applying it for syllables may highly extend the training time. To accomplish better results investigations on hybrid models such as joint n-gram with Conditional Random Field [28] and joint n-gram model with a decision tree model [29] are being carried out. Though numerous models have been explored, one unsettling fact is that there is still a need for a robust LTS conversion model and the conception of syllable based LTS conversion models are yet to be focused.

When we shift the focus towards Tamil LTS models, the existing work revolves mostly around rule-based systems [13, 14, 30, 31]. A decision tree based on phonemes for Tamil, discussed in [6] is the solitude work using data driven approach. The decision tree in [6] was centered on phonemes and if a decision tree is developed for syllables it would undergo over-fragmentation issues due to the increased tree size. G2P tools like Phonetisaurus and Sequitur based on joint sequence models were adapted for Tamil, these tools were developed with phoneme as the sub-word unit. These tools are not able to cope when subjected to syllables, due to the increase in the length of the sub-word unit (syllables are longer in length than phonemes). The total number of phonemes in Tamil language is only 41 [14] whereas the total number of syllables is definitely greater than the phonemes (number of syllables in any language will be in thousands) [32]. This increase in the total count of sub-word unit and the usage of a bi-gram prediction model hinders the performance of these data driven toolkits. Although work on syllable as a sub-word unit in pronunciation models is very minimal, the experimentation of syllable-based TTS for Tamil confirms the fact that syllable will be a better choice of pronunciation unit for Tamil [33, 30, 31] than a phoneme. Since, the existing G2P approaches and toolkits are not able to cope when subjected to syllables, we decided to experiment on Conditional Random Fields for syllable-based pronunciation model in Tamil. CRF is widely used in NLP, Speech Technology and Information

Retrieval as they allow us to automatically build language independent, reusable NLP modules by incorporating linguistic information resulting in a high accuracy when compared to Hidden Markov Model, Naive Bayes classifier and other generative models [34, 35]. CRF has already been experimented for LTS conversion for other languages such as English and has been quite successful [4].

3.0 CHARACTERISTICS OF TAMIL LANGUAGE

Tamil is one of the longest surviving Dravidian language with a very complex phonology, rich morphology and syllable isochrony. On analyzing the characterization of Tamil language with other languages, we observe that:

- The morphological richness of Tamil is presumably comparable with languages like Finnish and Turkish [36];

- The Dravidian languages like Telugu, Malayalam and Kannada also exhibit similar characteristics to Tamil excluding the phonology.

The major complexity in Tamil phonology is due to the presence of aspirated and voiced consonants in the spoken language despite it's absence in the written script [3]. Hence mapping between the letter to sound unit is mislaid in Tamil. Though there is an ambiguity in the mapping between the written and spoken script, Tamil letter to sound units also exhibit a certain pattern of similarity among the irregularities which helps in resolving the complexity and is elaborated in the next section.

4.0 PATTERN ANALYSIS FOR TAMIL LTS CONVERSION

The main reason for the non-triviality between the letter to sound units in Tamil is to redeem the lesser number of characters in the written script to sound units. In Tamil language's written script, there are 12 vowels, 18 consonants and 1 special character. In addition to the 18 consonants, 5 additional consonants called 'granthas' were included for scripting the loan/foreign words which has invaded into Tamil language in the due course of it's existence. The combination of these vowels and consonants combine to form compound characters named as 'uyirmei ezhuthukal' in Tamil, where 'uyir' refers to vowels, 'mei' to consonants and the word 'ezhuthukal' refers to characters. The vowels and the special character are given in Table 1, the consonants and granthas are given in Table 2 and Table 3 respectively. The consonants in Tamil scripting system can be split into four categories: vallinam, mellinam, idaiyinam and granthas as shown in Table 4. Literally, the naming convention followed in Tamil represents the nature of the consonants. The word 'vallinam' can be split into 'val+inam', where 'inam' implies to a 'category' or a 'group' in Tamil language. 'val' is a shortened transformation of the root word 'vanmai' which means hard in Tamil. Hence, the word 'vallinam' refers to the hard category of consonants in Tamil language. The word 'mellinam' can be split into 'mel+inam', where 'mel' is a shortened transformation of the word 'menmai' which means soft in Tamil. Thus, 'mellinam' refers to a group of soft consonants. In a similar way, 'idaiyinam' can be split into 'idai+inam' and 'idai' is a shortened transformation of the word 'idaimai' which means 'medial' in Tamil. Therefore, 'idaiyinam' indicates a medial category of consonants. The vallinam, mellinam (also called as nasals) and idaiyinam hold a group of six consonants each respectively while the granthas hold a group of 5 consonants.

Table 1: Tamil vowels

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

Table 2: Tamil consonants

க்	ங்	ச்	ஞ்	ட்	ண்	த்	ந்	ப்	ம்	ய்	ர்	ல்	வ்	ழ்	ள்	ற்	ன்
k	N;	s	N:	D	n:	dh	N	p	m	y	r	l	v	z:	L	R	n

For understanding the non-triviality in the letter to sound correspondence in Tamil consider the examples given in Table 5. In Table 5, Tamil words and their character-wise representation are given in column one and column two respectively. The pronunciation of these words is given using romanization in the third column. The

rationale for using a romanization to represent the pronunciation is due to the one-to-many correspondence between the letter-to-sound units in Tamil. Although, the first word's first character and the second word's third character are the same (in the second column, the character is underlined in both the words to highlight it), the pronunciation of these words show two different sound representations of the same character (in the third column, the two different representations of the same character 'p' are underlined in both the words to highlight it). The vallinam p has only one written representation (as given in Table 2) but has two different spoken representations (as given in Table 6) p and b, where p is unvoiced and b is voiced

On analysis, it is evident that all the vallinam/hard consonants in the written script unfolds to more than one sound unit in Tamil. There exists a persistent role of the preceding and succeeding letters in deciding the pronunciation of the current letter in a word. A concealed regularity is observed in the pattern of this one-to-many mapping between the written to spoken unit. When vallinam/hard consonants are either preceded with or succeeded by mellinam/soft or idaiyinam/medial consonants it transits to either a voiced or aspirated sound unit which is contrarily absent in the written script. In some cases, the occurrence of a vowel or a grantha also effects the transition of the vallinam/hard consonants. Conspicuously, the co-occurrence of two vallinam consonants also invokes the transition of the vallinam consonants. The sound variants of the hard consonants dealt in the current work are given in Table 6. The contextual analysis led to the realization of specific patterns which triggers the transition of hard consonants and can be represented using a mapping function: $L_p[L_c]L_n = S_c$ [37]. Here, L_c represents the current Letter under analysis; L_p represents the previous letter; L_n represents the next letter and S_c is the current letter's sound representation (symbol/sound unit). The patterns which trigger the transition of the hard consonants from it's default sound unit to another sound variant fall into eight categories and is given in Table 7. The default sound unit of the hard consonants are mentioned in Table 2. The primal objective of our mapping function for Lc is to perform a pattern analysis and to check if there is any match to the patterns given in Table 7. If a match is found, then that particular rule in accordance to the pattern can be applied to resolve the letter to sound mapping. The term 'Any/None' in Table 7 under the L_p column entails two possibilities. First, the vallinam/hard consonant can be preceded by any letter (any) and second, the hard consonant under analysis is the initial letter of a word (none). Similarly, the term 'Any/None' under the L_n column also entails two possibilities. First, the vallinam/hard consonant can be followed by any letter (any) and second, the hard consonant under analysis is the last letter of the word (none). These 8 categories are taken into consideration for forming a rule-based pronunciation model for Tamil and is discussed in detail in section 5.1.

Table 3: Granthas

ஜ்	ஶ	ஷ	ஸ	ஹ்
J	s;	s:	S	h

Table 4: Consonant Categories

Vallinam/Hard	க்	ச்	ட்	த்	ப்	ற்
Mellinam/Soft/Nasal	ங்	ஞ்	ண்	ந்	ம்	ன்
Idaiyinam/Medial	ய்	ர்	ல்	வ்	ழ்	ள்
Granthas	ஜ்	ஶ	ஷ்	ஸ்	ஹ்	-

Table 5: Examples of One-to-many correspondence in Tamil

பலகை	ப ல கை	pa la gai
கம்பம்	க ம் ப ம்	ka m ba m

Table 6: One-to-many correspondence of vallinam/hard consonants to sound variants

க்		ச்			ட்		த்		ப்		ற்		
k	g	s	ch	j	D	T	dh	th	p	b	R	tR	dR

Table 7: Pattern List - Effecting the Transition of Vallinam/Hard Consonants

Pattern List		
Previous Letter (*Lp*)	Current Letter (*Lc*)	Next Letter (*Ln*)
Any/None	Vallinam / Hard consonant	Vowels
Any/None	Vallinam / Hard consonant	Mellinam / Soft consonant
Any/None	Vallinam / Hard consonant	Vallinam / Hard consonant
Vowels	Vallinam / Hard consonant	Any/None
Mellinam / Soft consonant	Vallinam / Hard consonant	Any/None
Idaiyinam / Medial consonant	Vallinam / Hard consonant	Any/None
Granthas	Vallinam / Hard consonant	Any/None
Vallinam / Hard consonant	Vallinam / Hard consonant	Any/None

5.0 SYLLABLE INGRAINED DATA DRIVEN PRONUNCIATION MODEL FOR TAMIL TTS

The current section describes the development of a data driven syllable ingrained pronunciation model for Tamil Text-To-Speech system. A machine learning based approach specific to Tamil pronunciation model is still un-evolved and has been emphasized already in section 2. Developing a data driven pronunciation model for Tamil is more arduous when compared to other languages due to the non-triviality between the written and spoken script as discussed already in section 4. Since, Tamil is a syllable-timed language where the syllables take approximately equal amounts of time to pronounce [1, 2, 3], we have chosen a syllable as the basic sound unit for developing our pronunciation model. We have posed the mapping between the letters and its sound units (syllables) as a sequence labelling problem and conscripted a CRF to model the pronunciation of Tamil words owing to the supremacy of CRF in solving sequence labelling problems [4]. This syllable-centric CRF based pronunciation model for Tamil is an inception towards syllable-driven machine learning based pronunciation model for Tamil.

In letter to syllable conversion task, the sequence of letters in a word are dealt as successive subsequence of letters, each subsequence is mapped to a pronunciation label. Here, the pronunciation label is a syllabary which represents the spoken syllable using a sequence of written symbols. Any classifier can be used to label the letters to syllabaries, but the main concern here is the inclusion of sequential information which will increase the accuracy of the labeler. In order to determine the current letter's pronunciation label, we need to know the previous and next letter's information. Consider an example, where the letter 's' has three possible pronunciation labels based on the context. If the previous letter is 'N:', then 's' will take the label 'j', else if the previous letter is also 's', then the current 's' will be labelled as 'ch', else the label is 's'. Hence, our requirement of a classifier which incorporates this sequential information is fulfilled impeccably by a CRF. The preparation of training data and the training process are discussed in detail in the following subsection.

5.1 Preparation of Training Data

To perform pronunciation labelling using CRF we need a well-founded training data with syllabic representation. For any machine learning algorithm, having the right amount and right mix of data instances in the right format is very essential. The training data for a syllable ingrained pronunciation model should contain a list of Tamil words with it's pronunciation given using syllabaries. A dictionary/lexicon of this sort is not available online for Tamil language, hence we need to construct the training data. Generally, rule-based pronunciation systems are used to generate the seed lexicon for training the data driven model.

5.1.1 Syllable-centric Rule Based Approach for Tamil

Conventionally, rule-based systems are used to generate the training data for any data driven approach. Typically, this data generated using a rule-based system is scrutinized [6] by a linguist to ensure the correctness

of the data. We intend to discern this hand - refinement process as it is time consuming and deters the automation of the pronunciation model development. The pronunciation generated by the existing rule-based systems are not very accurate and professedly we need a more robust rule-based model. We implemented the existing rule-based systems given by [13] and [14] and thoroughly analyzed the words which deteriorates the pronunciation performance. We ascertained that the ambiguity in pronunciation generation prevailed only around the vallinam/hard consonants in Tamil. Although rules were already designed to tackle this ambiguity, the one-to-many correspondence [1] of a single vallinam character to more than one sound unit made it more arduous.

The analysis of the Tamil characters helped us uncover a concealed pattern of regularity for formulating the LTS conversion rules and has already been discussed in section 4. We identified 8 patterns (given in Table 7) which triggers the pronunciation of a vallinam from it's default sound (given in Table 8) to it's other sound production (allophone - given in Table 6). From Table 7, perceptible evidence to a strong codification in triggering the sound variants of vallinam/hard consonants due to the occurrence of any one of the other consonant groups or vowels is clear. Conspicuously, the co-occurrence of two vallinam consonants also invokes the transition of the hard consonants. We formulated a set of 11 new rules which consecutively summated to a set of 29 rules for performing the Tamil LTS conversion. The other 18 rules also comprehend to the 8 categories in Table 7 and are taken from [13] and [14]. The 11 rules formulated by us are given in Table 8 and the 18 rules taken from the existing systems are given in Table 9.

Table 8: Newly formulated set of 11 rules

L_p	L_c	L_n	S_c
*	k	N;	g
y	k	*	g
z:+hal	k	*	g
k	k	*	k
s	s	*	ch
D	s	*	ch
Grantha (except J)	D	*	T
D	D	*	T
*	D	S	T
th	th	*	th
S(Grantha)	th	*	th

Table 9: Existing set of 18 rules

L_p	L_c	L_n	S_c
[]	k	v:	g
Nasal	k	*	g
a;	k	*	g
r+hal	k	*	g
l+hal	k	*	g
L+hal	k	*	g
Vowel	k	Vowel	g
N:	s	*	j
*	D	Vowel	D
Nasal	th	*	dh
Vowel	th	*	dh
y+hal	th	*	dh
[]	th	*	dh
Nasal	p	*	b
y+hal	p	*	b
p	p	*	p
R	R	*	tR
Nasal	R	*	dR

In order to affirm the usage of the best rule-based system to generate the training data, we compared the existing rule-based systems in the literature with our newly formulated syllable-centric rule-based system. We have given a comparison between four systems: System-I developed from [13], System-II from [14] and System-III from [15] in the literature. System-IV is the syllable-centric rule-based system developed by us. A test set of 5353 randomly selected words from Wikipedia has been formed. The pronunciation of these words has been constructed with the help of linguists to endorse it as a benchmark pronunciation test set for our experimentation. The details of the linguists who helped us frame the test set is given in the acknowledgement section. The pronunciation constructed by the linguists for these words are termed as the 'trusted pronunciation' and the output of the rule-based systems are termed as 'generated pronunciation'. The output generated by all the four systems are converted to one common romanization format as followed in the test set to obviate the differences in the romanization followed by each system.

Table 10: Performance Evaluation of Rule based systems

System	CP	IP	Mean WER	MLD	MSS
I	1019	4334	13.6760	1.7299	0.9053
II	1953	3400	8.9380	1.2488	0.9169
III	1333	4020	12.0952	1.5922	0.8968
IV	3999	1354	2.3941	0.3178	0.9775

The pronunciation generated for each word in the test set are compared using 3 metrics: Word Error Rate (WER), Mean Levenshtein Distance (MLD) and Mean Similarity Score (MSS) for each system. The comparative results of these four systems are given in Table 10. The CP and IP in Table 10 stand for the number of Correct Pronunciations (CP) and number of Incorrect Pronunciations (IP) generated by a system. A CP implies that each and every basic unit of the word has been generated correctly by the rule-based system. The WER, Levenshtein Distance and Similarity Score is calculated for each word with respect to the trusted pronunciation. To calculate the WER, first compare the generated pronunciation against the trusted pronunciation and keep a count of number of words inserted, deleted or substituted with respect to the trusted pronunciation. Then these insertions, deletions, substitutions are divided against the length of the generated pronunciation and represented as a score out of 100 in percentage. The Levenshtein Distance and Similarity Score are calculated by devising the Levenshtein Distance algorithm [38] with the source and destination strings replaced with the trusted and generated pronunciations respectively. The value of WER ranges from 0 to 100 in percentage, Levenshtein Distance ranges from 0 to the length of the string and Similarity Score ranges from 0 to 1. The mean WER, MLD and MSS are considered for comparing the performance of each system. The prefiguration to a good pronunciation model is a low score of the mean WER & MLD and a high score of MSS.

CPs have a WER of 0 (the least possible score of WER), a Levenshtein Distance of 0 (least possible score of Levenshtein Distance), a Similarity Score of one (maximum possible score of Similarity) and hence, high CPs generally yield a high MSS. System-IV has 3999 CPs which is the highest amongst the four systems and consecutively also has a high MSS of 0.97. System-IV also has the least WER of 2.39% which is very minuscule when compared to13.67%, 12.09% or even 8.93% generated by other systems. Hence System-IV outperforms the other 3 existing systems in the literature by providing enhanced pronunciation. The adherence to the prefiguration of a low mean WER, MLD and a high MSS is incredible in System-IV and hence it is chosen for generating the training data.

One of our goals, which is discerning the need of the hand-refinement process on words generated from rule-based system seemed feasible with such low WER, MLD and remarkably high MSS scores. The training data for CRF was generated using this new syllable-centric rule-based system (System-IV) for around 34,000 randomly selected words from 'Ponniyin Selvan', a Tamil Novel.

5.2 Training Process

The input function for our CRF based pronunciation model constitutes of:

- a word, w

- position of the current letter, i

• current letter, L_c

• the label of the current letter, S_c

• previous letter at i-1, L_p

• next letter at i+1, L_n.

Here, the linguistic information required to obtain the pronunciation label are obtained from the sequences in a word, separated across contextual overlapping frames. CRF assigns a pronunciation label (S_c) for each sequence of letters in a word using it's knowledge acquired from the training instances. CRF can handle overlapping context of features with a relative cohesion in an excellent way

Apparently, to train CRF a training data with appropriate alignment and a template with details on the contextual frame size is required. One detriment in using CRF is the need of a monotonic alignment between the source and target symbol for resolving the one-to-many alignment [39]. This alignment process is not an integral part of a CRF and should be fabricated separately. The alignment between the letters and syllables should be done very cautiously as the performance of the CRF model depends extensively on it [39]. To build a training data with a fitting alignment for CRF, perform the steps mentioned in Algorithm 1.

Algorithm 1 Pre-processing

```
for lines in corpus do
    Pre-process the training data
    Tokenize the training data into words
    Extract only Tamil words
    for each word do
        Syllabalize the word using Tamil language's spoken syllable scripting rules
        Tokenize the character/group-of-characters constituting to a syllable
        Place one token in a line
        Use a new line as a demarcation between words
    end for
end for
for each syllabalized and tokenized word do
    Subject the sequence of letters to the rule-based system to obtain it's syllabary
    Place the syllabary (pronunciation) against the sequence of letters in a word with a uniform tab alignment
end for
```

Once, this alignment process is over, CRFs are applied to this training data from which it learns the pronunciation labels for the letters in a word by aggregating the contextual syllabic information. To construct the contextual template, we experimented with the usage of syllable unigrams and varied the frame size between 3 and 5. Finally, a contextual frame of 5 over the syllable unigrams was chosen.

Finalizing the number of training instances was very tricky due to two reasons. First, data driven approaches learn from each training instance and try to replicate the acquired knowledge on test instances of both known and unknown type. Hence, if the training data is less the approach will not work well on all possible instances, if the training data is more or repetitive then the approach will suffer over-fitting issues. In both the cases the performance of the approach will get derailed. Second, choosing a large training data would mean higher computational intensity. Accounting to these facts, we increased the training data size from 16,000 to 34,000 words and applied both the variants of contextual template for further observations. After some experimentation conclusively, CRF was applied on a data set of around 34,000 words with a contextual frame size of 5 to obtain the final model.

5.3 Testing data and processing

The test data can have instances either totally similar or different from the training data. For similar instances, performance is better but for dissimilar instances the performance cannot be promising. To have an impartial testing data, a set of 5353 randomly selected words from Tamil Wikipedia has been chosen. The details on the

construction of this test set is already mentioned in section 5.1.1. To perform the testing process on the CRF model, the test set should also be aligned appropriately. The alignment process discussed in section 5.2. is applied excluding the fourth step for generating the test set in accordance to the requirement of the CRF model. The aligned test data is subjected to the CRF model for obtaining the pronunciation labels (syllabaries) for the letters in a word.

The CRF model is imposed on the test set of 5353 words and the metrics WER, MLD and MSS already discussed in section 5.1.1 are benefacted again to evaluate the performance of the CRF based pronunciation model. The generated pronunciations are compared against the trusted pronunciation and the results are tabulated in Table 11.

Table 11: Comparison of CRF and Phonetisaurus

System	CP	IP	Mean WER	MLD	MSS
CRF	3388	1965	6.5349	0.8205	0.9433
Phonetisaurus	399	4954	18.2221	2.7198	0.8725

5.4 Comparative Results of Syllable ingrained CRF Model and Phonetisaurus

CRF based pronunciation model using syllable as the sub-word unit is an incipient towards the development of syllable ingrained machine learning model. Although a syllable ingrained data driven model is still unsubstantial for Tamil, conducive to provide an exemplification of the efficiency of our CRF based pronunciation model we compared the results with Phonetisaurus. Phonetisaurus is a data driven tool developed using the joint-grapheme- phoneme n-gram approach [24] with phonemes as the sub-word unit. The data used to train CRF model was provided to Phonetisaurus with phonemes as the sub-word unit. The pronunciation model using Phonetisaurus was obtained after performing the alignment process and framing the FST (Finite State Transducer) and is already discussed in section 2. Radically, the pronunciation of a word is our prime concern and hence we overlook the difference in the sub-word unit for analyzing the performance. To charter justice to both the sub-word units, we finalized 'words' to be the basic unit for performance evaluation. Both the systems were presented the same test set and the generated pronunciation were viewed word wise and analyzed using WER, MLD and MSS metrics. In section 5.1., the description and formulation of these metrics is already discussed. The results for our CRF based pronunciation model (mentioned as CRF) and Phonetisaurus are given in Table 11.

The CP in Table 11 refers to instances where the pronunciation model was successful in producing a correct pronunciation of an entire word. A CP means each and every sub-word unit was determined correctly by the pronunciation model. Phonetisaurus was able to produce correct pronunciations of only 399 words out of the 5353 words falling way behind our CRF based pronunciation model with 3388 correct pronunciations. A prefiguration of a good pronunciation model should have a low mean WER and MLD. The mean WER and MLD of CRF based pronunciation model is 6.53% and 0.82 while it is 18.22% and 2.71 for Phonetisaurus. A higher value of MSS is a clear implication of a better pronunciation model. The MSS of Phonetisaurus is 0.87 while the MSS of CRF based pronunciation model is 0.94. Hence, the syllable ingrained CRF enforced pronunciation model proposed by us certainly outpaces the performance of Phonetisaurus.

6.0 RESULTS & DISCUSSION

In this paper, we have developed a syllable-centric rule-based pronunciation model and a syllable ingrained CRF enforced pronunciation model. The first model follows a rule driven approach while the second is a data driven approach.

6.1 Rule driven pronunciation model

The need for well-founded data to develop a data driven model and the defalcation in pronunciation accuracy by the existing rule-based systems in the literature propelled us to develop an enhanced and robust rule-based approach. This enhanced rule-based approach outperformed the existing rule-based systems in the literature with a high MSS of 0.97, a low MLD of 0.31 and a low WER score of 2.39% (given in Table 10). The words were modelled using syllable as the sub-word unit in order to generate the training data for developing a pronunciation model concentric on a machine learning approach.

A comparison of the results of these four rule driven pronunciation systems have been illustrated further with two graphs. The first graph holds (Fig. 1) a comparison of the CP and IP while the second graph (Fig. 2) portrays the comparison between the metrics (WER, MLD and MSS).

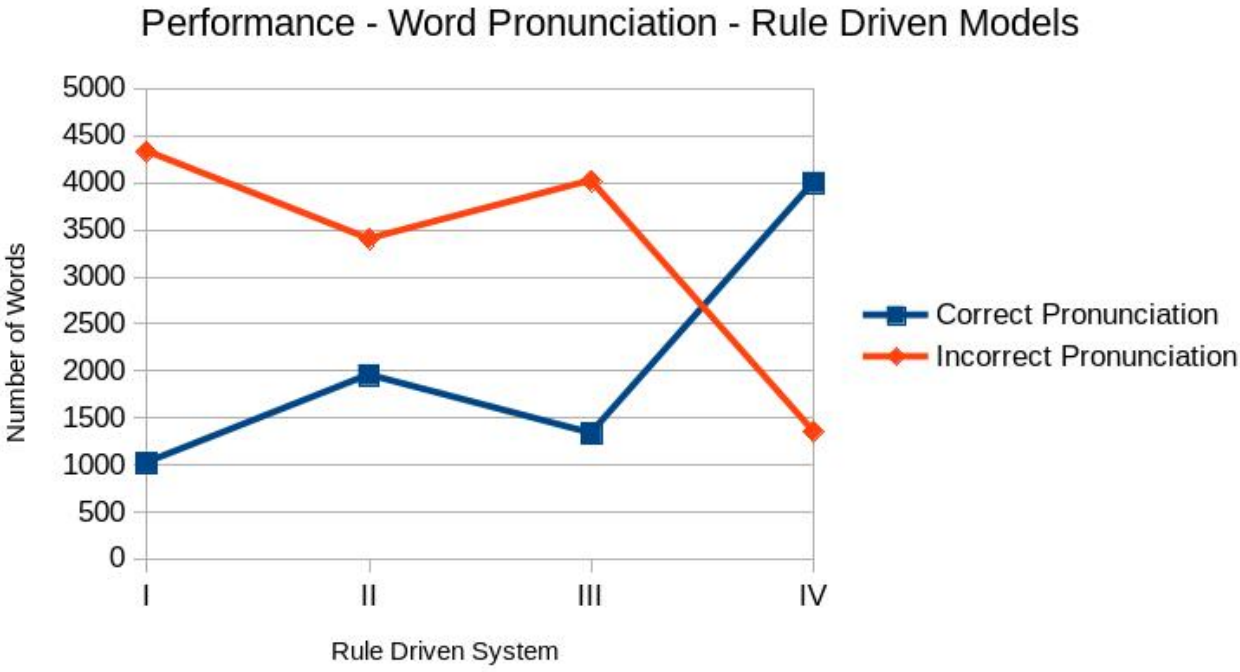

Fig. 1: Comparison of CP vs IP – Rule Driven Systems

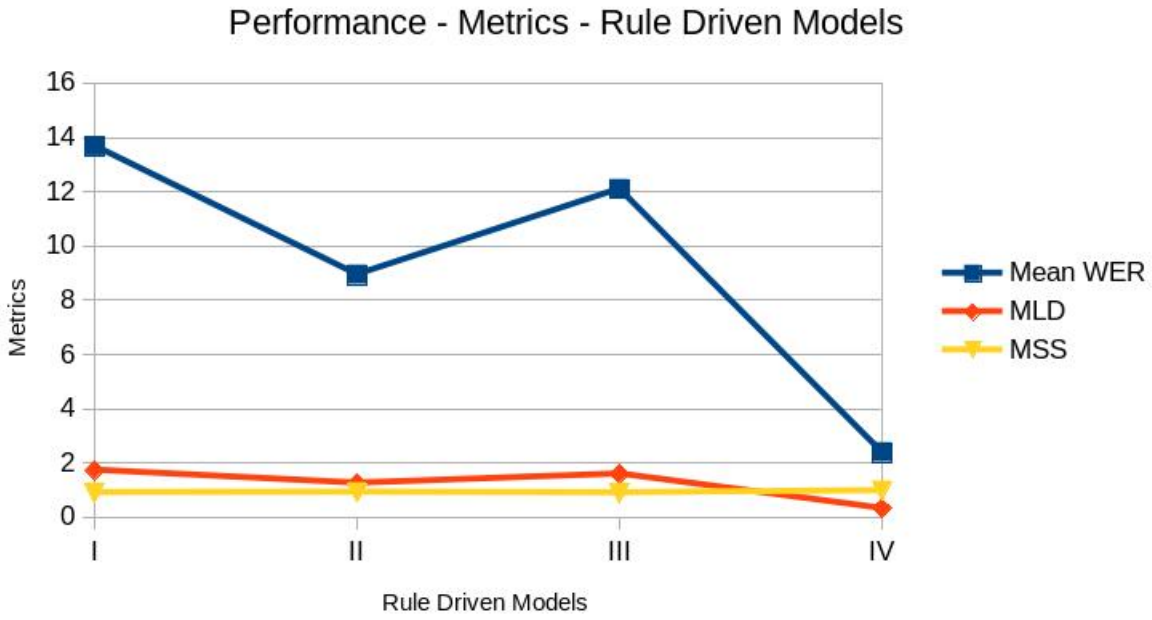

Fig. 2: Performance comparison of Rule Driven Models

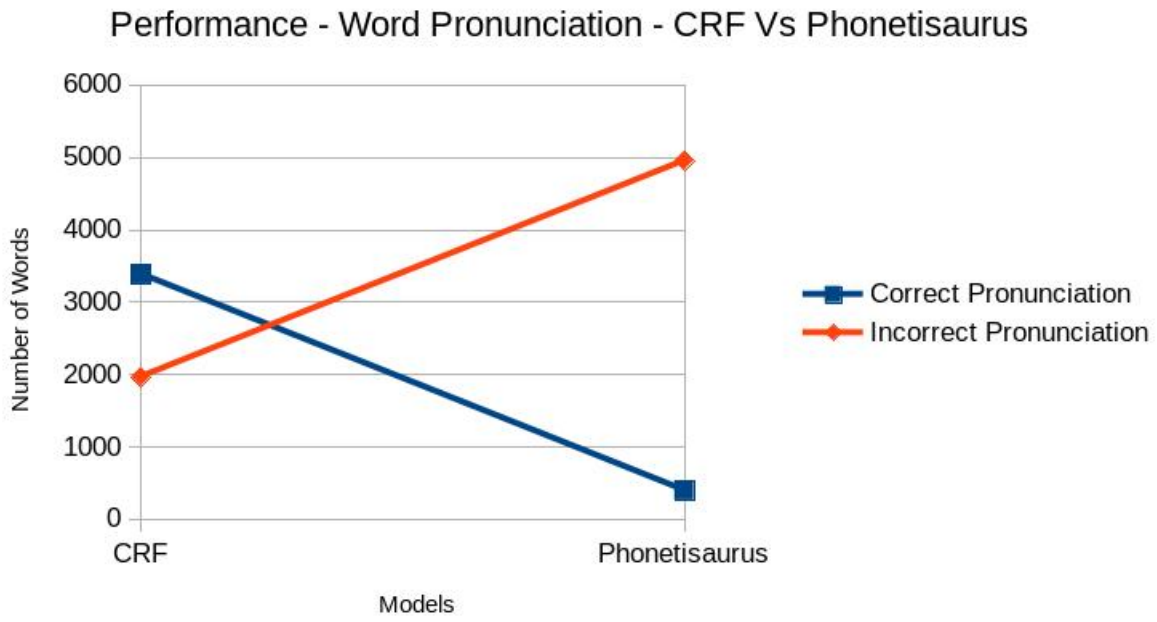

Fig. 3: Comparison of CP vs IP – Data Driven Models

6.2 Data driven pronunciation model

We have developed a syllable ingrained pronunciation model using CRF, as CRFs provide phenomenal performance for NLP related tasks by outranking most of the generative models such as Naive Bayes and HMM [34, 35]. A data driven model with syllable as a sub-word unit is nearly non-existent in the literature and perhaps the reason can be the requirement of protracted training time. Contemplating on the syllabic nature of Tamil language we decided to overlook the training time as it is a one-time process. We seed the conception of a syllable ingrained, data driven Tamil pronunciation model. The results of the syllable ingrained CRF enforced pronunciation model indemnifies the longer training time and encourages to explore further. The results of the syllable ingrained CRF enforced pronunciation model was compared with Phonetisaurus, a very popular data driven G2P tool. The results were analyzed in a word wise perspective to evade the difference in the basic unit used for modelling (CRF based model is syllable-centric and Phonetisaurus is phoneme-centric). CRF based model excelled over Phonetisaurus with a high MSS of 0.94, a low MLD of 0.82 and a low WER of 6.53% (given in Table 11). The results of CRF enforced pronunciation model and Phonetisaurus have also additionally been illustrated with two graphs. The first one (Fig.3) shows the difference in the correct and incorrect pronunciation of words while the second graph (Fig.4) shows the comparison of the two models with respect to the metrics discussed (WER, MLD, MSS).

6.3 Observations on the rule driven and data driven syllable-centric pronunciation models

When we compared the performance of our syllable-centric rule driven and syllable ingrained data driven pronunciation models, the rule driven model outperformed the CRF based data driven model and is quite amusing. The reason for the difference in performance can be due to the lesser occurrence of certain contextual syllable instances in the training data. Also, the training data is comprised to 34,000 words only due to the computational complexity involved in training the CRF models. A rule-based approach has an advantage that it does not depend upon the number of training instances but rather only on the rules. Optimal training data with respect to syllable instances will improve the efficiency of the CRF model further.

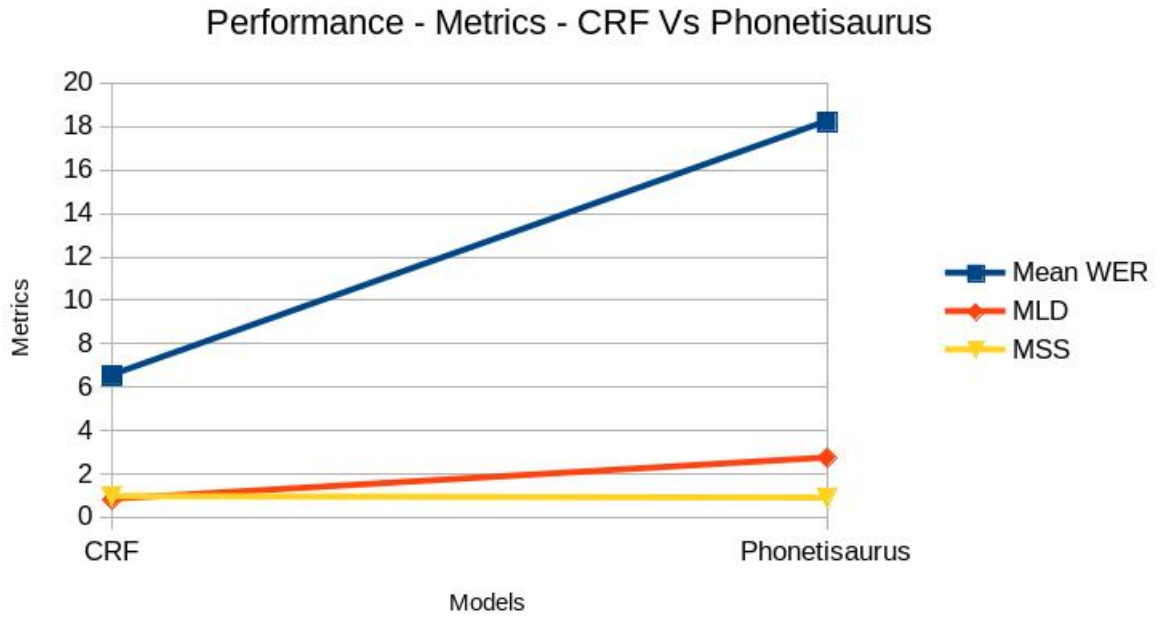

Fig. 4: Performance comparison of Data Driven Models

7.0 CONCLUSION & FUTURE WORK

The need for a mapping between the written (spelling) to spoken (pronunciation) form of a word is indispensable in speech synthesis. Technologies like speech recognition, speech-based spelling generation & predictive search also require a mapping between the pronunciation and spelling. Diligently, two syllable-based pronunciation models have been developed for Tamil in this paper: (a) a syllable-centric rule-based pronunciation model and (b) a syllable ingrained Conditional Random Field enforced pronunciation model. Both these models are dominions in comparison to the other existing rule driven and as well data driven models in the literature with a high MSS of 0.97 and 0.94 respectively. The performance of the syllable ingrained CRF enforced pronunciation model for Tamil can be improved further with optimal training data with respect to syllables. These syllable ingrained pronunciation models developed by us will help in enriching the Tamil TTS.

8.0 ACKNOWLEDGEMENTS

We express our fervent gratitude to Dr.Va.Mu.Se. Muthuramalinga Andavar, Associate Professor in PG and Research Department of Tamil, Pachaiyappas College, Chennai, Tamilnadu, India; Dr.S.Ganesh, Assistant Professor, Department of Tamil, Arul Anandar College, Karumathur, Madurai, Tamilnadu, India; and Dr.R.Vimala Devi, Assistant Professor, Department of Tamil, Chellammal Womens College, Chennai, Tamil Nadu, India for their valuable help in building the pronunciation test set. We also thank the Tamil native speakers who actively took part and shared their opinion in the analysis of the pronunciation generation.

REFERENCES

[1] P. Bhaskararao, "Salient phonetic features of Indian languages in speech technology". *Sadhana*, Vol. 36, No.5, 2011, pp. 587-599.

[2] H. Sirsa and M. A. Redford., "The effects of native language on Indian English sounds and timing patterns". *Journal of phonetics*, Vol 41, No.6, 2013, pp 393-406.

[3] K. R. Krishnan, S. A. Shanmugam, A. Prakash, G. R. Kasthuri and H. A. Murthy..,"IIT Madras's submission to the blizzard challenge 2014". *In Proc. Blizzard Challenge 2014, Satellite workshop of Interspeech'14*, 2014.

[4] E. Fosler-Lussier, Y. He, P. Jyothi and R. Prabhavalkar.,"Conditional random fields in speech, audio, and language processing". *Proceedings of the IEEE*, Vol. 101, No. 5, 2013, pp. 1054-1075.

[5] S. Ash and D. Lin.,"Grapheme to phoneme translation using conditional random fields with re-ranking". *In International Conference on Text, Speech, and Dialogue*. September 2016, pp. 314-325.

[6] N. Udhyakumar, C. S. Kumar, R. Srinivasan, and R. Swaminathan., "Decision tree learning for automatic grapheme-to-phoneme conversion for Tamil". *In 9th Conference Speech and Computer*. 2004

[7] H. A. Murthy, A. Bellur, V. Viswanath, B. Narayanan, A. Susan, G. Kasthuri, and K. Prahallad., "Building unit selection speech synthesis in Indian languages: An initiative by an Indian consortium". *Proceedings of COCOSDA, Kathmandu, Nepal*. 2010.

[8] A. W. Black, K. Lenzo and V. Pagel., "Issues in building general letter to sound rules". *International Speech Communication Association*. 1998

[9] M. Anand Kumar, V. V. Dhanalakshmi and S. Rajendran.,"A novel data driven algorithm for Tamil morphological generator". *International Journal of Computer Applications*, Vol. 12, 2010, pp. 52-56.

[10] I. M. Kalith, D. Asirvatham, A.Khatibi and S. Thelijjagoda.,"Comparison of Syllable and Phoneme Modelling of Agglutinative Tamil Isolated Words in Speech Recognition". *Current Journal of Applied Science and Technology*. 2018, pp. 1-10.

[11] R. I. Damper, Y. Marchand, M. J. Adamson and K. Gustafson.,"Comparative evaluation of letter-to-sound conversion techniques for English text-to- speech synthesis". *In The Third ESCA/COCOSDA Workshop (ETRW) on Speech Synthesis*. 1998.

[12] H. Sarmad.,"Letter-to-sound conversion for Urdu text-to-speech system". *In Workshop on Computational Approaches to Arabic Script*. 2004, pp. 74-79.

[13] A. G. Ramakrishnan, L. N. Kaushik and M. Laxmi Narayana., "Natural language processing for Tamil TTS". *Proc. 3rd Language and Technology Conference, Poznan, Poland*. 2007, pp. 192-196.

[14] S. Yuvaraja, V. Keri, S. C. Pammi, K. Prahallad, and A. W.,"Building a Tamil Voice using HMM segmented labels". International Institute of Infor- mation Technology, Hyderabad, India Language Technologies Institute, Carnegie Mellon University, USA, communication. 2010.

[15] A. Baby, N. L. Nishanthi, A. L. Thomas and H. A. Murthy.,"A unified parser for developing Indian language text to speech synthesizers". *In International Conference on Text, Speech, and Dialogue* September 2016. pp. 514-521.

[16] A. K. Kienappel and R. Kneser.,"Designing very compact decision trees for grapheme-to-phoneme transcription". *In Seventh European Conference on Speech Communication and Technology*. 2001. pp. 1911-1914.

[17] V. Rajendran and G.B. Kumar.,"A Robust Syllable Centric Pronunciation Model for Tamil Text To Speech Synthesizer". *IETE Journal of Research. Taylor & Francis*, 2018. pp. 1-12.

[18] C. Ma, M. A. Randolph and J. Drish,.,"A support vector machines-based rejection technique for speech recognition". *In Acoustics, Speech, and Signal Processing*, 2001. Proceedings.(ICASSP'01). Vol. 1, 2001. pp. 381-384.

[19] L. Jiang, H. W. Hon, H. W and X. Huang.,"Improvements on a trainable letter-to-sound converter". *In Fifth European Conference on Speech Communication and Technology*. 1997.

[20] J. R. Bellegarda.,"Unsupervised, language-independent grapheme-to-phoneme conversion by latent analogy". *Speech Communication*. Vol. 46, No. 2, 2005. pp. 140-152.

[21] M. Bisani and H. Ney.,"Joint-sequence models for grapheme-to-phoneme conversion". *Speech communication*, Vol.50, No.5, 2008. pp. 434-451.

[22] L. Galescu and J. F., "Bi-directional conversion between graphemes and phonemes using a joint n-gram model". *In 4th ISCA Tutorial and Research Workshop (ITRW) on Speech Synthesis*. 2001.

[23] S. Jiampojamarn and G. Kondrak.,"Phoneme alignment: An exploration*". In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. 2010. pp. 780-788.

[24] J. R. Novak, N. Minematsu and K. Hirose.,"WFST-based grapheme-to-phoneme conversion: Open source tools for alignment, model- building and decoding". *In Proceedings of the 10th International Workshop on Finite State Methods and Natural Language Processing*. 2012. pp. 45-49.

[25] K. Rao, F. Peng, H. Sak and F. Beaufays.,"Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks". *In Acoustics, Speech and Signal Processing (ICASSP),* April 2015. pp. 4225-4229.

[26] H. Sak, A. Senior, and F. Beaufays., "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition". 2014. *arXiv preprint arXiv*:1402.1128.

[27] T. Mikolov, S. Kombrink, A. Deoras, L. Burget and J. Cernocky., "RNNLM-Recurrent neural network language modeling toolkit". *In Proc. of the 2011 ASRU Workshop*. December 2011. pp. 196-201.

[28] K. Wu K. Hall., M. Riley and B. Roark.,"Encoding linear models as weighted finite-state transducers". *In Fifteenth Annual Conference of the International Speech Communication Association*, 2014, pp. 1258-1262.

[29] S. Hahn, P. Vozila and M. Bisani.,"Comparison of grapheme-to-phoneme methods on large pronunciation dictionaries and LVCSR task". *In Thirteenth Annual Conference of the International Speech Communication Association*, 2012, pp. 2538-2541.

[30] H. A. Patil, T. B. Patel, N. J. Shah, H. B. Sailor, R. Krishnan, G. R. Kasthuri, T. Nagarajan, L. Christina, N. Kumar,, V. Raghavendra and S. P. Kishore..,"A syllable-based framework for unit selection synthesis in 13 Indian languages". *In Oriental COCOSDA held jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE), IEEE International Conference*, November 2013, pp. 1-8.

[31] A. Parlikar, S. Sitaram, A. Wilkinson and A. W. Black.,"The festvox indic frontend for grapheme to phoneme conversion". *In WILDRE: Workshop on Indian Language Data-Resources and Evaluation*, 2016.

[32] E. V. Raghavendra, S. Desai, B. Yegnanarayana, A. W. Black and K. Prahallad.,"Global syllable set for building speech synthesis in Indian languages". *In Spoken language technology workshop, IEEE*, December 2008, pp. 49-52.

[33] S. P. Kishore, R. Kumar and R. Sangal.,"A data driven synthesis approach for indian languages using syllable as basic unit". *In Proceedings of Intl. Conf. on NLP (ICON),* 2002, pp. 311-316.

[34] D. Klein, K. Toutanova, H. T. Ilhan, S. D. Kamvar and C. D. Manning.,"Combining heterogeneous classifiers for word-sense disambiguation". *In Proceedings of the ACL-02 workshop on Word sense disambiguation: recent successes and future directions*, Vol. 8, July 2002, pp. 74-80.

[35] X. Wang and K. C. Sim.,"Integrating conditional random fields and joint multi-gram model with syllabic features for grapheme-to-phone conversion". *INTERSPEECH*, 2013, pp. 2321-2325.

[36] G. B. Kumar, K. N. Murthy and B. B. Chaudhuri.,"Statistical analysis of Telugu text corpora". *International journal of Dravidian linguistics*, Vol. 36, no. 2, 2007, pp. 71-99.

[37] J. Lee and G. G. Lee.,"A data-driven grapheme-to-phoneme conversion method using dynamic contextual converting rules for Korean TTS systems". *Computer Speech & Language*, Vol 23, no. 4, 2009, pp. 423-434.

[38] B. Babych.,"Graphonological levenshtein edit distance: Application for automated cognate identification". *Baltic Journal of Modern Computing*, Vol 4, no. 2, 2016, pp. 115-128.

[39] P. Lehnen, S. Hahn, A. Guta and H. Ney., "Incorporating alignments into conditional random fields for grapheme to phoneme conversion". *In Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference*, May 2011, pp. 4916-4919.