Prediction of Pauses in TTS - Tamil

P. Arulmozhi
Project Associate, MILE Lab
Department of EE, IISc, Bangalore
arulmozhi@ee.iisc.ernet.in

A G Ramakrishnan
Professor, Department of EE,
IISc, Bangalore.
ramkiag@ee.iisc.ernet.in

Abstract

Text to Speech (TTS) involves the task of converting the text typed in electronic format to speech signal. In MILE lab, we are involved in making a TTS system for Tamil and Kannada. In this paper, the contribution of syntactic information such as part of speech (POS) tags in enhancing the quality of a text to speech synthesis system for Tamil is researched. The quality of a TTS system is measured by the intelligibility and naturalness of the synthesized speech. The NLP module of the TTS system (for example, text normalization) contributes not only to its intelligibility, but also to its naturalness, by improving the prosody. The stress and pause modeling can be improved using the POS and other syntactic information. In a sentence, where there should and should not be a pause needs to be identified for the naturalness of the produced speech. This is because, a sentence without any pause or with identical pause intervals between words sounds robotic. Also, pause at a wrong place makes the sentence unnatural and there is even a possibility of change of meaning. For example, take the following sentence,

avarukku inRu <P> mAlai kitaittatu.
avarukku inRu mAlai <P> kitaittatu.

<P> here indicates that there is a pause. The pause given in different places gives different meanings. Syntactic information such as parts of speech can be used for identifying the rules for pause in a sentence. A rule based POS tagger is developed for this purpose without using a root word dictionary. Currently, manual evaluation shows an accuracy of approximately 74% using only the lexical rules. The performance is expected to improve after the context sensitive rules are applied. Rules are made for predicting the insertion of pause at the right place. The manual evaluation of pause insertion shows a significant improvement in the naturalness of the produced sentence.

1. Introduction

This paper presents a rule based parts of speech tagging method in the perspective of improving the naturalness of synthesized speech. The quality of a TTS system is measured by the intelligibility and naturalness of the synthesized speech. There are two main modules in a TTS system. One is the natural language processing (NLP) module, which takes care of the production of phonetic transcription, intonation and rhythm. Another is the digital signal processing (DSP) module, which takes care of the production of the speech waveform corresponding to the given text. The stress and pause in the speech contribute majorly to the naturalness, which is controlled by the NLP module. Using this POS tagger, we try to find out the right place to introduce pause in the synthesized speech. Introducing a high degree of naturalness is theoretically possible, but the rules to do so are still to be discovered (Jonathan A. 1996). Introducing pauses at the right places in the synthesized speech is the first step in achieving this. Many linguistic aspects are analyzed (Thierry D) and syntactic information such as POS tagging is considered important to achieve a good TTS. In this paper, we present a POS tagger created for Tamil, which is a highly agglutinative and partially free word order language.

2. POS Taggers

The purpose of a POS tagger is to automatically find out the syntactic category of a word in a sentence. Different methods may be followed to do POS tagging. Most commonly used are rule based, statistical, and transformation based methods. Rule based taggers work on predefined linguistic rules for deciding the syntactic category of a word in a sentence. These rules may be lexical or context sensitive and they are language dependent. In statistical taggers, the POS tag for a particular word is decided based on its lexical and contextual probability. A training corpus is used to train the system and the input sentence is tagged based on the probability estimated using the training corpus. Transformation based taggers derive rules based on learning. Those rules are used to find the syntactic category. In English, Brill's tagger is the most commonly used TBL based tagger. There are statistical, rule based and hybrid taggers being worked on for Tamil (Arulmozhi P, Sobha L, 2006).

3. MILE POS Tagger for Tamil

3.1 Purpose of MILE tagger

While speaking, human beings naturally introduce stress and pause at the right places, so that it is easily understandable. In a TTS system, the DSP component produces the speech waveform and the NLP component is responsible for the naturalness of the produced speech. It should identify the right places to introduce the right amount of pause. The pause is introduced using the category of the words in a sentence. Other syntactic information such as shallow parsing and clause boundary identification will not only help identify the pause, but also to estimate the required intonation contour.

There are a few POS taggers available for Tamil. They have been developed for the purpose of preprocessing for NLP applications such as machine translation and information extraction (Arulmozhi P. et.al 2004). The statistical POS taggers need a huge training corpus. They provide the POS tags according to the tagset used for training. In the case of a TTS, such detailed tagging may not be needed.

3.2. Nature of the Language Tamil

Tamil is a morphologically rich, agglutinative and partially free word order language. Compound words are common in this language, where two or more words are combined to form a single word. The case and tense markers appear as inflections of the root word itself. For example, taking the word ‘varukirAn’, the root word and its inflections can be split as follows.

vA + kir + An
vA - root word
kir - present tense marker
An - 3rd person, singular, neuter gender.

Tamil is a partially free word order language because changing the word order to some extent does not affect the meaning of the sentence. However, this order change cannot occur within a phrase. For the sentence,

‘Aciriyar nanRAka paTitta mANavanukku paricae kotuttAr’
Teacher thoroughly studied student+Dat prize+Acc gave+Hon
The teacher gave the prize to the student who studied thoroughly.

It can be written variously as

‘nanRAka paTitta mANavanukku Aciriyar paricae kotuttAr’ (or)
‘Aciriyar paricae nanRAka paTitta mANavanukku kotuttAr’

without effectively changing the meaning of the sentence, but,

* ‘Aciriyar paTitta mANavanukku paricae nanRAka kotuttAr’

changes the meaning. So, within a phrase, the word order must not change.

3.3. Tagset

A tagset is the set of all the tags used by the POS tagger. Commonly, there are two levels of tags - the main and the sub tags. The main tags identify the main category of the word such as noun, verb or adjective. The subtags identify the category of the inflections such as person, number, gender, and tense. Unlike other NLP applications, we do not need very detailed tags for a TTS,. However, using only the main tags does not give sufficient information. So we need some of the sub tags too. So, as a special case, we have developed a tagset for the purpose of deciding to insert pauses at the right locations in a sentence. In our tagset, each tag is a combination of a main tag and one or more sub-tags. The nouns take the case and plural markers and the verbs take person, number, gender and tense markings. Apart from this, pronouns have person, number and gender. The clitics are suffixed to the root word to form adverbs and conjunctions. Then the dates, numbers and punctuations are also tagged separately. English POS taggers and some of the Tamil taggers (Dhanalakshmi V et. al. 2009) use monadic tags. Monadic tags do not give information on inflections, which is important for TTS. We use structured tags such as "NN+pl.acc" in which different pieces of information serve in different parts of the rules. This tag refers to a noun with the inflection for plural and accusative. We have 15 main tags and 30 subtags adding up to a total of forty five tags.

3.4 MILE Tagger

This is a rule based POS tagger. We do not use a root word dictionary. The tagger is based on a two-stage architecture. The block diagram of the POS tagger is shown in Figure 1. In the block diagram, each block explains its functionality. The first stage has the lexical rules and the second stage has the context sensitive rules. First, a sentence is taken as input and split into tokens. For each token, the suffixes are identified. Then, using the lexical rules, which work at the word level, each word is assigned a POS tag according to the suffixes identified. Then, this output is given as input to the second stage, where the context sensitive rules change the tag if it is wrongly tagged by the lexical rules. Thus, the final tagged sentence is obtained.

Separate tables are created for programming purpose with the list of suffixes identified. A lexical rule looks like,

2*1+1*1, NN+pl.acc

This means the suffixes indexed 2*1 (suffix Table 2, column 1 - kaL) and 1*1 (suffix Table 1 column 1 - ae) occur in a sequence, and the word will be tagged as Noun+Plural+Accusative. Here 'kaL' is the plural marker and 'ae' is the accusative marker. There are 13 such tables, which list 103 identified

suffixes. These suffixes are used by the lexical rules. The context sensitive rules are embedded in the system. For example, the following can be considered as a context sensitive rule.

'If a sentence starts with a verb, change it to noun'.

Since Tamil is a verb ending language, a sentence does not normally start with a verb. So, if the first word of a sentence is wrongly tagged as a verb in the first level, it will be corrected in the second level. The combinations of lexical rules including the inflections are 533 and the number of context sensitive rules are 4.

For any POS tagger to work correctly, the sentence boundaries need to be identified. We use a sentence splitter for splitting paragraphs to sentences. Input to the sentence splitter is any Tamil text such as say, paragraphs. The output is an array of sentences. This process is also embedded in the POS tagger based on our need. We use a rule based sentence splitter and the rules are heuristic in nature.

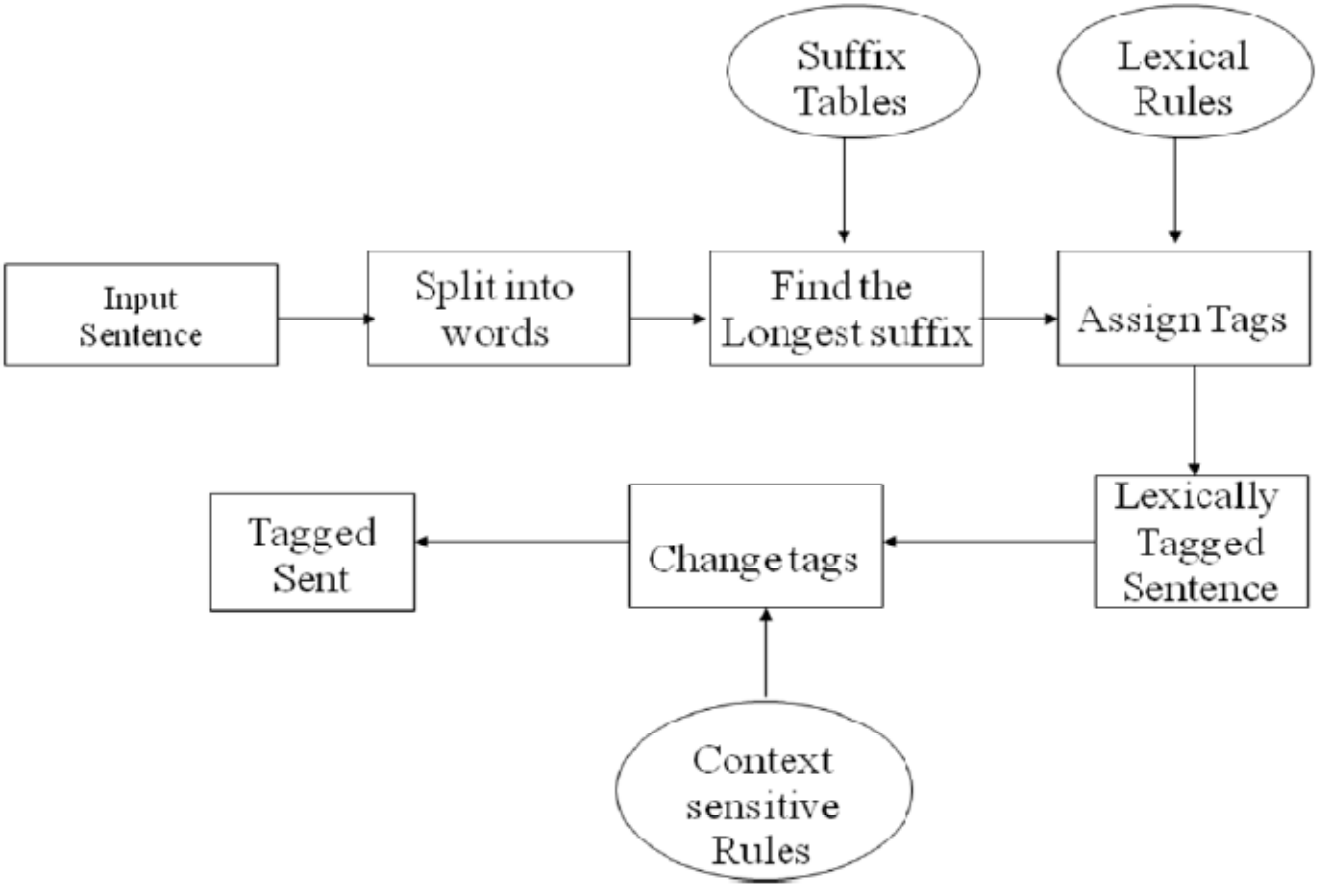

Figure 1. Block diagram of the two-stage POS tagger for Tamil TTS.

4. Pause Model

Insertion of the right amount of pauses at the right places adds to the naturalness of the synthesized speech. With a natural language text, native speakers introduce pauses with the acquired knowledge

of the language. However, in a TTS system, such appropriate pauses need to be automatically inserted by the system at the right places. For European languages such as Spanish, there are rule based pause models developed and experimented (Rafael M 2002). A wrong pause inserted between two words may make the synthesized speech unnatural. For simplicity, such an example sentence is illustrated in English. Here, the notation <np> denotes “no pause” between the words, whereas <p> refers to the required pause.

Example:

The< np>book<np>is <p> on<np>the<np>table.
The<np>book<np>is<np>on <p> the<np>table.

Speech synthesized as per the tags for the first sentence appears perceptually natural, whereas for the second sentence, the inappropriate pause between the words ‘on’ and ‘the’ makes the synthetic utterance perceptually unnatural. Hence, POS information and pause are very important in the context of TTS. In this paper, we focus on pause insertion between successive words by pause prediction from the POS tags estimated from the input text. Presently we use key words and heuristic rules for inserting pauses. Rules for inserting pauses at the right places are created according to the POS tags.

At present, in the syntactic level, only POS tags are considered for identifying pauses. It is considered as a basic preprocessing needed. On top of this, the phrase chunks may be identified, which are useful to identify the positions, where pause must be inserted. In that process, the phonological phrases must be identified finally for identifying pauses and intonation.

Six levels of pause have been identified, which determine the duration of the pause. In Chinese pause model, they use minor, major, and punctuation breaks (Fu-chiang et.al 1997). We have defined <P0>, <P1>, <P2>, <P3>, <P4>, and <PW>. P0: no pause, P1: lowest pause, P2: medium pause (ex: pause after a comma), P3: significant pause (ex: pause after a semicolon), P4: highest pause (ex. pause between sentences). The pause levels P0 to P4 are derived from the existing synthesis database. This work has been carried out for Tamil and can be extended to all the Dravidian languages with no major changes. <PW> is the common pause between each word. Wherever <P0>... <P4> is not identified, <PW> is assumed.

At the initial level, <PW> is assumed for each word, and rules are identified for converting it to <P0> to <P4>. The sample rules are given below.

1. There is no pause (or may be very minimal pause) between a number and certain words following, such as ‘mani’, ‘latcam’, ‘kOTi’. There is a list of words defined for this rule. Any noun in plural form, (NN+PL) after a number does not have a pause.
2. If the previous word has an accusative/dative marker, and the current word is a postposition, there is no pause between the current and the previous words.

Ex : avanai <P0> pola,
avanukku <P0> pin

3. Combine the words with POS tags Adjectival Participle (AJP) and Noun (NN - any number of them) occurring together. There is no pause between them.

4. There should be a pause before a quantifier (Q).

Ex : (azhakiya kiraamamum) <P3> (oru periya ooraaTciyum)

All the numbers are considered as quantifiers.

Exception: 3 Ayiram – there is no pause before Ayiram.

There are 15 such rules made for converting <PW> to <P0> - <P4>. In natural speech, we do not give any pause between most of the words. Taking that into consideration, we did another experiment. Instead of taking <PW> as default pause, we took <P0> as default ie. There is no pause between any word initially, and then inserting pauses using rules. This reduced the rule set from 15 to 8, because we had more rules for <P0>. The outputs are obtained from both models and given for evaluation.

5. Output and Evaluation

The output of the system contains the original sentence, the predicted POS tags and the predicted pause levels. All are displayed parallely. An example output is shown in Figure 2.

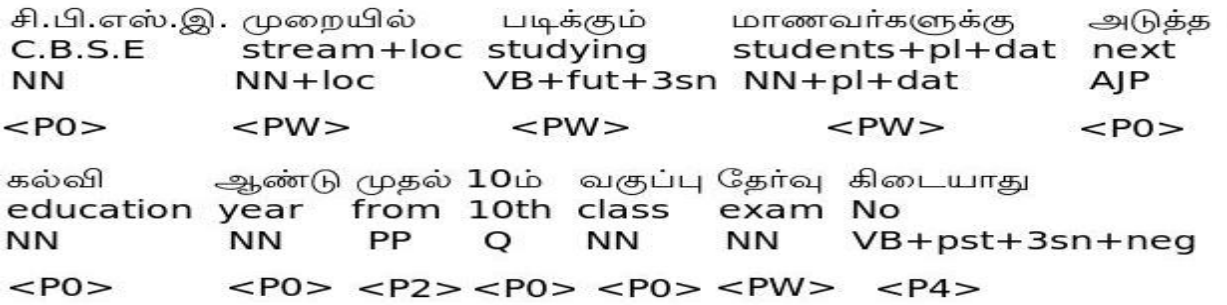

Figure 2. Example Tamil sentence and the predicted POS tags and pauses.

In the figure, the first line is the Tamil input; second line is the corresponding meaning in English; third line is the predicted POS tags and the fourth line is the pause levels identified.

This output is given to the DSP module and the wave form of the sentence obtained. The people who evaluated the outputs are native Tamil speakers, who did not have knowledge about the methods used for creating TTS outputs. Three types of outputs are given to the evaluators and their mean opinion score (MOS) is obtained. Ten sentences are created by the TTS as follows:

Without implementing the pause model.
After implementing the pause model with default as <PW>
After implementing the pause model with default as <P0>

A score of 1 to 5 is given by the evaluator according to the understandability and naturalness (1-worst, 5-best). The evaluation based on the mean opinion score gives encouraging results.

The rule based POS tagger is evaluated manually for the correctness of the tags. In a given tag, if the main tag is correct and the sub-tag is wrong, or vice versa, we take that as a wrong tag. The system gives 78% results without a root word dictionary. More context sensitive rules are to be added so as to improve the accuracy of the POS tagger.

References

1 Arulmozhi. P, Sobha. L, Kumara Shanmugam. B. 2004. Parts of Speech Tagger for Tamil. In the proceedings of Symposium on Indian Morphology, Phonology & Language Engineering, (March 19-21) Indian Institute of Technology, Kharagpur (Page 55-57).

2 Arulmozhi Palanisamy and Sobha Lalitha Devi. 2006. HMM based POS Tagger for a Relatively Free Word Order Language. A poster presentation in CICLing-06 (February 19-25) at Mexico.

3 Brill, E. 1995 Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging. Computational Linguistics, 21, 4 (Page 543-565).

4 Dhanalakshmi V, Anand kumar M and Soman K P, 2009. POS Tagger and Chunker for Tamil Language. Tamil Internet Conference, Cologne, Germany. pp. 250-255.

5 Fu-chiang Chou', Chiu-yu Tseng, Keh-jiann Chen, and Lin-shan Lee 1997. A Chinese Text-to-Speech System Based on Part-of-Speech Analysis, Prosodic Modeling and non-Uniform Units. ICASSP'97, Volume – 2, Munich, Germany.

6 Jonathan Allen 1993. Linguistic aspects of speech synthesis. Presented at a coUoquium entitled 'Human-Machine Communication by Voice. Organized by Lawrence R. Rabiner, held by the National Academy of Sciences at The Arnold and Mabel Beckman Center in Irvine, CA.

7 Rafael Marin, Lourdes Aguilar, David Casacuberta, 2002. Placing pauses in read spoken Spanish: A model and an algorithm. Language Design: Journal of Theoretical and Experimental Linguistics, pp. 49-67.

8 Thierry Dutoit. High-quality text-to-speech synthesis: an overview. Faculte Polytechnique de Mons, TCTS Lab, 31, bvd Dolez, B-7000 MONS, Belgium.