

# Interpreting Text for Indian Language TTS

Kalika Bali

Microsoft Research Labs India  
“Scientia”, 196/36, 2<sup>nd</sup> Main, Sadashivnagar, Bangalore, India  
kalikab@microsoft.com

**Abstract**— This paper highlights some of the commonly faced issues while interpreting text for Indian Languages. Spelling variations, lack of a standard for transliteration schemes as well as the existence of code-mixed multilingual text can pose a problem for developers of speech technology with little specialized knowledge of speech and/or linguistics. The paper hopes to generate discussion on how these can be resolved through proper support in common standards for all Indian Languages.

**Keywords**— Text-to-speech Synthesis, Indian languages, Interpretation of Text, Multilingual Text, Standards

## I. INTRODUCTION

Interpreting Indic language text for Text-to-Speech Synthesis (TTS) is seen as a relatively uncomplicated task as most Indian languages have a near one-to-one mapping between orthography and sound. Though this is largely true, there are still certain issues in text processing and normalization that are specific to Indian languages and require special treatment [3]. SSML [1] with its aim to help TTS interpret standard orthographic text correctly in a given context, can assist in better processing of the input text in these areas and hence, enhance the output of the TTS. This paper would like to focus on a few specific issues which might require extension to existing SSML tag set.

## II. ISSUES IN INTERPRETING TEXT FOR INDIAN LANGUAGE TTS

In this section some of the prominent issues with respect to text interpretation for Indian language text-to-speech synthesis are discussed.

### A. Spellings

Indic scripts are syllabic in nature, that is, an orthographic character represents a single syllable. Thus, when a word needs to be spelt out it is usually spelt out in syllables. For example, a word in Hindi /bhar t / “India”, will be spelt out as broken in syllables as [bha] [r ə] [t ə]. If the word is to be spelt out as phonemes, there would be no indication that the first long vowel is written as vowel ligature on the consonant and not as an independent vowel. If the word is to be spelt out in alphabets, it would require a lengthy description of all the ligatures. Thus, /bhar ə t / would require a complete

description as “bhə with the vowel *matra* for /a/, r ə , t ə ”. This can become even more complicated with words containing conjuncts and series of special ligatures.

### B. Transliteration Schemes

By and large, phonetization of most Indic language text is not as complicated as languages like Chinese, for example, and is mostly pronounced as it is written. However, there are certain cases, in Tamil for instance, where a single grapheme is mapped to several phonemes. A TTS should be able to discriminate between the alternate pronunciations either through phonotactic rules or lexicons. In reality, creating an exhaustive lexicon of exceptions is not an easy task. Thus, being able to provide a phonetic transliteration of the input text is useful. Though IPA alphabet or other specific phonetic alphabets like SAMPA would indeed be ideal in such cases, these, especially IPA, require special training as most developers are not familiar with these. In the Indian context, certain transliteration schemes like ITRANS are in common use though not a standard. Support for these widely accepted transliteration schemes would provide an easy way to input phonetized text in Indian languages. Though a standard transliteration scheme for all Indian languages would be ideal

### C. Multilingual Text Documents

Code-mixing refers to instances where lexical items and grammatical features of two languages appear in a single sentence and is a common and well-studied phenomenon in a multilingual society [5]. In Romanized transliteration of Indian languages, especially during informal communications like mail or chat, interspersed English words and phrases, are extremely prevalent. Codemixing can be viewed as a form of noise caused due to multilinguality in text.

In a country with 18 official languages including English, it is no surprise that multilingual documents are in common use. A single document can contain text in more than one language running together. It is fairly common for texts in Indic languages to be interspersed with English words, written in Roman alphabet. For example, it is not uncommon to find sentences such as “यह एक तेज missile है” in a Hindi text. This code-mixing of distinct languages requires different engines for proper rendering and requires multilingual support in

SSML whereby portions of text in distinct languages can be parsed and tagged with separate language tags.

#### D. Compounding

Compounding is an extremely productive process in Indian languages like Hindi, where almost any two (or more) words can be clubbed together to get a compound. Compounds in Hindi require special treatment in terms of certain morphophonemic rules, like schwa-deletion [2]. Each consonant in written Hindi is associated with an “inherent” schwa, which is not explicitly represented. This schwa is sometimes pronounced and sometimes it is deleted based on certain morphophonemic rules which may work for certain words [4] but are not very productive for compounds.

Further, it is well known that compound words differ in their prosodic structure as well. The problem is complicated by the fact that in Hindi (as in other Indian languages) compounds are not separated by either space or any other separator like a hyphen [2]. Thus, compounds in Hindi, like /loks ə b ha/ “lower house of parliament”, need to be correctly identified and interpreted to be rendered correctly as /loks ə b ha/ and not /lok ə s b ha/. Ideally, this should be done through a compound word lexicon but again this would require either appropriately tagged corpus or special morphological analysers which are not available for these languages. Thus, SSML support for marking compound words in Indian

languages would facilitate the correct interpretation of these words

### III. CONCLUSIONS

The aim of this paper is to highlight certain issues which are commonly encountered in the development of TTS systems for Indian languages. The hope is that this would generate discussion on how these can be resolved through proper support in SSML and other standards. This would make it easier for developers with no specialized knowledge in either speech or linguistics to rapidly develop high quality voice-based applications.

### REFERENCES

- [1] W3C, “Speech Synthesis Markup Language (SSML) Version 1.1”, <http://www.w3.org/TR/speech-synthesis/>
- [2] Deepa. S.R., A.G. Ramakrishnan, Kalika Bali, and Partha Pratim Talukdar, 2004. “Automatic Generation of Compound Words Lexicon for Hindi Speech Synthesis”, Language Resources & Evaluation Conference (LREC), 2004.
- [3] Kalika Bali, Partha Pratim Talukdar, N. Sridhar Krishna, A.G. Ramakrishnan, 2004 “Tools for the Development of a Hindi Speech Synthesis System”, pp.109-114, 2004
- [4] Monojit Choudhury and Anupam Basu A., 2002. “A Rule-based schwa deletion algorithm for Hindi” Proceedings of the International Conference On Knowledge-Based Computer Systems, Vikas Publishing House, Navi Mumbai, India, pp. 343 - 353. 2002.
- [5] Suzanne Romaine and Braj Kachru. 1992. “Code-mixing and code-switching.” In T. McArthur (ed.) *The Oxford Companion to the English Language*. Oxford University Press. pp. 228-229