Language Processing in Vani

Harsh Jain Varun Kanade Kartik Desikan

1 Introduction

Any Text to Speech Synthesizer, has two essential components[3] viz. language processing and signal processing. In this report, the language processing techniques used in Vani, will be described.

2 Language Processing in Indian Languages

Most Indian languages are phonetically fairly sound. Hindi is no exception to this rule. What this essentially means is that there is no difference between what we speak and what we write. It is because of this that very little language processing is required for Indian Languages. Though Sanskrit, the root of most modern Indian languages is phonetically perfect, the languages derived from this are not completely phonetically perfect. There are some situations where there is deviation from what we speak and what we write. In Hindi this situtaion typically occurs when the inherent vowel 'a' is supressed in many cases. As an example in the word 'kamala', the actual pronounciation would be done as 'kamal', where the 'a' following the 'l' is supressed. These procedure is known as schwa deletion.

3 Schwa Deletion

Hindi is almost phonetically sound, but there are few instances when there is a difference. Each consonant in Hindi is associated with an inherent schwa, namely 'a'. The schwa is pronounced in some cases but omitted in others. Without schwa deletion, speech will sound very unnatural, but it will also be incomprehensible in many cases. There has been some work in this area in Hindi. The two main works in this area are the first by B. Narsimhan, R. Sproat and G.Kiraz [1] . This work combines morphological analysis, with finite state transducers and cost models. The accuracy of this is about 89%. The second work in this area is by Monojit Choudhary and Anupam Basu [2]. This is a rule based approach and is much simpler to implement. Also the accuracy in this case is 96.12% without a morphlogical analyser and 99.89% without one. Currently in Vani 2.0 we plan to implement the algorithm without the morphological analyser. There are some cases when morphological analyser becomes important, eg. ”asamaya” without a morphological analyser, it will be pronounces as ”asmay”, where as it should be pronounced as ”asamay”. This is because ”asamaya” is actually constructed from ”samaya” by prefixing ”a”. The meaning of ”samaya” needs to be preserved in this case. So actually the algorithm should delete schwas from "samaya" which would give "samay" and then prefix "a" to it. Since Hindi is close to being phonetically perfect, we can proceed with such an algorithm without a morphological analyser. The next section will briefly describe the algorithm by Monojit Choudhary and Anupam Basu [2].

4 The Algorithm

Input : Word string of alphabets (graphemes)

Output: input word with some schwas deleted.

1. **Mark all the full vowels, viz. vowels not associated with consonants as full and also all consonants followed by vowels other than the inherent schwa and all "h"s as full, unless explicitly marked half by use of halant.**

   This is because of the empirical observation that the schwa following h is always retained. Mark all consonants foolowed by consonants or halants as half. Mark all remaining as undetermined.

2. **If in the word, y is marked as undetermined and is preceded by i,I,ri,u or U, mark it as full.**

   The consonant "y" is a glide from a high vowel to a medium vowel. Therfore if schwa is deleted in the context when "y" is preceded by a high vowel, this glide will be lost, and y will not be appropriately pronounced. Eg. in "tRîtIya" the "a" following "y" needs to be retained. But it may be deleted from "hoya".

3. **If "y","r","l" or "v" are marked as undetermined and preceded by consonants marked half, then mark them as full.**

   This is because of phonotactic constraint. Eg. "kAvya", "samprati", "ashva".

4. **If consonant marked as undetermined is followed by a full vowel, then mark it as full.** This is to maintain lexical distinctions. Eg. if from "baDhaI", the schwa after Dh is deleted, then "badhaI" will be indistinguishable from "baDhi".

5. **While traversing from left to right, if a consonant marked undetermined is encountered before any consonant or vowel marked full, then mark it as full.**

   The schwa following the first syallable is never deleted. This may result in illegal consonant clusters and may change the identity of the word. Eg. "kalama", if the schwa following "k" is not retained it will be "klama". This changes the identity of the word.

6. **If the last consonant is marked undetermined, mark it half.**

   This is again empirically observed. Eg. ”kalama” may be ”kalam”, ”banda” should be ”band” and so on.

7. **If any consonant is marked undetermined and immediately followed by a consonant marked half, then mark it as full.**

   This is mainly because of phonotactic constraints. Eg. ”sAphalya”, the schwa after ”y” needs to be produced, to make the word sound. This is because pronouncing ”phly”is not possible.

8. **While traversing from left to right, for every consonant marked undetermined, mark it half if it is preceded by full, and followed by undetermined or full, otherwise mark it as half**

9. **For all consonants marked half, if it is followed by a schwa in the original word, then delete the schwa from the word. The resulting word is the required ouput.**

5 Conclusion

In this article, we have seen the language processing that is employed in Vani 2.0. The language processing may be improved by using a morphological analysyer as indicated in the paper[2]. This would take the accuracy from 96.12% to 99.89%. The point again is stressed, that such a simple language processing module is possible in Vani, because of the phonetic nature of the language. In case of English, language processing without morphological analysis cannot even be considered, because the same string has completely different pronounciations in different contexts.

References

[1] N. Bhuvana, S. Richard, and K. George. Schwa deletion in hindi text-to-speech synthesis. In *Workshop on Computational Linguistics in South Asian Languages*. 21st SALA, October 2001.

[2] M. Choudhary and A. Basu. A rule based schwa deletion algorithm for hindi. In *IKON 2002*. IKON, December 2002.

[3] H. Jain, V. Kanade, and K. Desikan. Vani - an indian language text to speech synthesizer.