A Finite State and Rule-based Akshara to Prosodeme (A2P) Converter in Hindi

Somnath Roy
Centre for Linguistics, Jawaharlal Nehru University, Delhi

Abstract

This article describes a software module called Akshara to Prosodeme (A2P) converter in Hindi. It converts an input grapheme into prosodeme (sequence of phonemes with the specification of syllable boundaries and prosodic labels). The software is based on two proposed finite state machines—one for the syllabification and another for the syllable labeling. In addition to that, it also uses a set of nonlinear phonological rules proposed for foot formation in Hindi, which encompass solutions to schwa-deletion in simple, compound, derived and inflected words. The nonlinear phonological rules are based on metrical phonology with the provision of recursive foot structure. A software module is implemented in Python. The testing of the software for syllabification, syllable labeling, schwa deletion and prosodic labeling yield an accuracy of more than 99% on a lexicon of size 28664 words.

Index Terms: Syllabification, Prosodic Labeling, Syllable Weight, Schwa-deletion

1. Introduction

The function of Akshara to Prosodeme (A2P) converter is similar to Grapheme to Phoneme (G2P) converter. Moreover, the former also includes the details of syllable boundaries and prosodic labels. Past studies on Hindi has mainly focused on the implementation of G2P converters for speech synthesis programme [1, 2, 3]. These G2P converters are based on linear phonological rules proposed by [4]. However, syllable is known to be a better unit for Hindi speech synthesis as reported in [5, 6]. The usefulness of syllable as the basic linguistic unit in the context of speech recognition system has been explored in English [7] and Tamil [8]. Similar work for Hindi requires a software for syllabification. The automatic syllabification would be more useful if it could also predict the prosodic labels in words of natural speech as this would facilitate synthesis and recognition. This work fulfills that need. The A2P converter is based on the following novel contributions.

• A finite state machine (FST) is proposed for both syllabification and syllable labeling. The use of FSTs yield faultless syllabification and syllable labeling at underlying phonemic form. The syllabification at underlying phonemic form is called as I-level syllabification in this work.
• A new foot formation rule is proposed. The proposed rules assume the extrametricality of foot unlike syllable as proposed in [9] and allow superfoot by prosodic recursion described in [10, 11].
• The proposed foot formation rules are autonomous to morphological boundaries. The rule set performs better than the previous works for schwa deletion in simple, compound, derived and inflected words.

Rest of this paper is organized as follows. Section 2 describes the salient points of metrical phonology relevant to this work. Section 3 describes the process of syllabification and syllable labeling. Section 4 describes the proposed foot formation rule. Section 5 describes schwa deletion and prosodic labeling. Section 6 describes the testing and evaluation report. The conclusion is written in Section 7.

2. Theoretical Background

metrical phonology is based on nonlinear arrangement of the constituents of a phrase [12, 13, 14, 15, 16, 17]. The nonlinear arrangement is realized in the form of a tree with nodes as the constituents of a phrase. The constituents are syllable, foot, phonological word, phonological phrase and intonational phrase. Syllable is the lowest unit in the hierarchy dominated by foot, which in turn is dominated by a phonological word. The higher units such as phonological phrase and intonational phrase are not relevant in the current work. Syllable functions as a domain for segmental phonological rules. In non-linear phonology, the rules are written on the basis of interaction among syllables under the domain of higher constituents. A syllable has obligatory rhyme and optional coda. The syllables are also described by the moraic weight in quantity-sensitive languages such as Hindi [18]. Foot as a domain is used for describing stress and re-syllabification due to deletion of segment like schwa in languages such as French and Hindi. The foot is used as a musical meter and the concept is borrowed to non-linear phonology as a constituent [14, 12]. Most of the quantity sensitive languages have binary foot, but some also allow degenerate foot. A binary foot is erected on either two syllables or on a single syllable having at least two moras. A single syllable having one mora, if projected as a foot, is called degenerate foot [13, 12]. Phonological word, also known as prosodic word, is a constituent unit of prosodic hierarchy above syllables or foot and below phonological phrases. Prosodic word takes into account the non-isomorphism between morphology and phonology [19]. In other words, a morphological word may not be a phonological word. Prosodic word also accommodate derivational affixes and compounding, and characterized as the domain for syllabification and stress rules.

3. I-Level Syllabification

I-level syllabification is derived from the underlying phonemic form (UPF), which in turn is derived from orthography using the following mapping rules.

• Each consonant grapheme in Hindi is inherently associated with the mid-central vowel called schwa (“ə”).
• If a consonant is followed by a vowel diacritic mark, or a diacritic called halant, the inherent schwa is deleted.
• The inherent schwa is not realized in case of consonant at word final position.
• Two or three consonant together can form a ligature.
• A short vowel at word final position is lengthened.

For example, कमल /k m l/ (Lotus) becomes kaml, कमान /k m A l/ (Wonder) becomes kaman, मस्त /s t j/ (Truth) becomes sotj.

The process of syllabification in Hindi was explored by [4] and [18, 9]. Their analysis do not talk about the role of maximal onset principle for syllabification. The present analysis for syllabification follows maximum onset principle [20]. The maximum onset principle is a sufficiency condition which can be demonstrated by the following examples.

\[ \text{मृत्युंजय (A name)} \rightarrow [\text{mṛt}] [\text{jun}] [\text{d̄ɔj}] \rightarrow *[\text{mṛt}] [\text{jun}][d̄ɔj]\]

\[ \text{अभिषेक (Capture)} \rightarrow [\text{a]} [\text{d̄i}] [\text{gra}] [\text{han}] \rightarrow *[\text{a}] [\text{d̄i}][\text{g}] [\text{ra}] [\text{han}] \rightarrow *[\text{ad̄i}][\text{g}] [\text{ra}] [\text{han}] \]

In the above examples, the right hand side shows the potential syllable structures for a word. The square bracket denotes the syllable boundary. An asterisk before the syllable structure indicates that this potential syllable sequence is incorrect. Moreover, the above examples show that either onsets are maximized or is equal to number of coda consonants. Therefore, maximum onset principle need to be applied to ensure correct syllabification. Based on many such examples, following regular expressions are proposed for syllabification in Hindi.

• \( v \rightarrow [v] \)
• \( vv \rightarrow [v][v] \)
• \( c^* \text{cv} \rightarrow [c^* v] [cv] \)
• \( c^* \text{cv}1 \text{cv} \rightarrow [c^* \text{cv}] [\text{cv}1] [\text{cv}] \)
• \( c^* \text{cv}1 \text{cv}2 \rightarrow [c^* \text{cv}] [\text{cv}1] [\text{cv}2] \)
• \( c^* \text{cv}1 \text{cv} \rightarrow [c^* \text{cv}] [\text{cv1}] [\text{cv}] \)
• \( c^* \text{cv}2 \text{cv}2 \rightarrow [c^* \text{cv}2] [\text{cv2}] \)
• \( c^* \text{v} \text{cv} \text{v} \rightarrow [c^* \text{cv}] [\text{ccv}] \)

In the above expressions, \( v \) denotes a vowel, \( "c1" \) denotes a stop consonant, \( "c2" \) represents a semivowel (r, l, \( j \)) and \( "c" \) at intervocalic position denotes a consonant that is neither a stop nor a semivowel. An asterisk in superscript denotes the kleene star.

The finite state machine for the I-level syllabification is shown in Fig 1. It contains seventeen states with the start state as I and the final state as F. The orthography of an input word is transliterated into a sequence of consonants and vowels. The 8 syllabification rules are applied to this sequence to derive a symbol sequence in terms of \( c, c1, c2 \) and \( v \). The symbol sequence is the input to the FSM in Fig 1. An arc between a pair of states in FSM is associated with a label (a pair of symbols separated by \( '/' \)). Suppose the symbol pair associated with an arc is \( "x/y" \). This indicates that whenever a symbol \( "x" \) is fed to the state at the beginning of the arc, the system makes a transition along the arc and outputs the symbol \( "y" \). The label \( e/c \) symbolizes null input and null output for a transition. If part of a symbol string reaches to the final state F, then it is consumed, and a transition from F to I with arc label e/b takes place, where e is null and b denotes the syllable boundary of the consumed string. The remaining part of the string repeats the same process from initial state I until everything is consumed.

### 3.1. Syllable Labeling

Hindi is a quantity sensitive language. Therefore, syllables in Hindi are also described based on an attribute called moraic weight [15, 18]. The stress assignment rules for English proposed by [21] reports moraic weight (syllable weight) as the most influential parameter. Based on moraic weight, the following rules are used for syllable labeling in Hindi.

1. Each short vowel (like a, u, i) and each coda consonant of a syllable are assigned a weight of one mora, while a long vowel (like a:, u:, i:) is assigned a weight of two moras.
2. The syllables with one, two and three moras are called weak (w), heavy (h) and superheavy (sh) syllables respectively [18, 22, 23, 24].

The examples of syllable labels are shown below in the Table 1 and Table 2. The Table 1 shows monosyllabic words and Table 2 shows bisyllabic words with syllable boundary denoted by square bracket and syllable labels in terms of w, h, and sh.

| Word | Type | Syllable Label | Gloss |
|------|------|----------------|-------|
| ki   | w    | [k]i1          | that  |
| man  | h    | [m]an1         | soul  |
| gali | sh   | [gali1]        | cheek |

| Word | Type | Syllable Label | Gloss |
|------|------|----------------|-------|
| kalam| w+h  | [ka]1[lo]      | pen   |
| rasa | l+h  | [ra]1[sa]      | silk  |
| aaram| h+sh | [a]1[ram]      | comfort|
| devn | sh+sh| [dev1][nar]    | A name|

A finite state machine for syllable labeling is shown in Fig 2. The machine consists of one initial state (state I), seven non-final states and three final states (F1, F2 and F3).
sylabified string (i.e., the syllable boundary marked as b) is given as input to the initial state. Since, each short vowel gets one mora and long vowels get two moras, therefore, vowel type distinction is essential at syllable labeling stage. Each coda consonants get one mora and the onset consonants do not contribute to the moraic weight of syllables in Hindi. Therefore, consonant type distinction is not required at this stage. The symbol c, v_s and v_l stand for consonants, short vowels and long vowels respectively. The output symbol along an arc is either a syllable label or the reflection of the input itself. There is a null transition from each final state to the initial state so that the process can be repeated for the remaining part of the string. The FSM assigns the label "w" to a syllable with phoneme sequences v_s, cv_s, ccv_s, ccc'v_s and "h" to v_sc, c v_sc, c'v_l, and "sh" to c'v_scc, c'v_lc.

Figure 2: Finite State Machine for Syllable Labeling

4. Foot Formation

This paper describes a new approach to foot formation in Hindi. The approach is based on three assumptions, six rules and one convention as described below.

4.1. Assumptions

• Foot is formed using the labeled syllables of a word. The process of syllable labeling is described in the section 3.

• Foot is generally binary branching. A bimoraic foot and degenerate foot projected on a single syllable are also permissible.

• A superfoot is formed either between a syllable and a foot, or between two foot.

4.2. Rules

The six rules for foot formation are listed below:

• Weak to Weak Affinity Rule (WW AR): Two adjacent weak syllables form a binary foot and results into a bimoraic foot as shown in Fig 3. The foot \( <\sum_s> \) is the extra metrical foot, which never bears any stress.

• Heavy to Weak Affinity Rule (HW AR): A heavy and a weak adjacent syllables form trimoraic binary foot as shown in Fig 4.

• Weak to Heavy Affinity Rule (WHAR): This kind of foot is either formed in bisyllabic Hindi words or in loan words as shown in Fig 5.

• Heavy to Heavy Affinity Rule (HHAR): Two adjacent heavy syllables also form a binary foot. In this case trisyllabic words having left most syllable as heavy is downgraded as weak and bisyllabic words having right most node as heavy is downgraded as weak. The foot structure is shown in Fig 6

• Superheavy to Others Affinity Rule (SOAR): The superheavy syllables always projected as a foot as shown below in Fig 7. The superfoot \( \sum_s \) is formed using one syllable and one foot. The projected foot constituent in \( \sum_s \) could be either a new syllable after schwa deletion as shown in Fig 3 or a syllable which inherently bear stress i.e., the superheavy syllable as shown in Fig 7.

• List Affinity Rule (LAR): LAR is devised for handling words having same syllable structure at underlying phonemic form but realized differently at surface level. In such overlapping cases different foot formation rules are applied. These rules make categorical distinction based on the segmental information in a syllable. The foot formation rules shown in Fig. 8 describes four cases with overlapping syllable structure. These four cases represent four different lists of words.

4.3. Convention

The rules apply in the direction from left to right (LR).
5. Schwa Deletion and Prosodic Labeling

Schwa deletion is an optional phenomenon in Hindi. This means that schwa can be deleted or retained in the same environment, and the choice solely depends on the speaker and the context being used. Schwa deletion phenomena in Hindi helps speakers to utter a word quickly, i.e., the process of schwa deletion reduces the overall effort in terms of duration. It enables stress shift from one syllable to other. The following rules describe the contexts in which schwa gets deleted.

\[ \sigma \rightarrow \Phi / \left[ \sigma^* \sigma^w \right] \sum \]
\[ \sigma \rightarrow \Phi / \left[ \sigma^h \sigma^h \right] \sum \]
\[ \sigma \rightarrow \Phi / \left[ \sigma^w \sigma^w \right] \sum \]

In other words, if the right most node of a foot is a weak syllable having schwa then schwa can be deleted.

Application of the above schwa deletion (SD) rules and consequent re-syllabification of exemplar words are shown in Table 3. The process of schwa deletion and re-syllabification occur at foot level. In the process of re-syllabification, the bare consonant(s) after schwa deletion are assigned as coda consonant(s) to the preceding syllable. The re-syllabified words are again fed to the finite state machine for syllable labeling. The syllable label associated to a syllable after re-syllabification is called the prosodic label. If prosodic label of a syllable is heavy or superheavy then it is stressed otherwise not stressed. The foot structure for the examples in Table 3 can be found in the Section 4.

6. Testing and Evaluation

Two databases are used for the testing and evaluation — BBC Hindi corpus and Hindi Wordnet [25]. Hindi wordnet is publicly accessible. It is first cleaned i.e., digits, hyphen and other special characters are removed. Long but meaningful words especially compounds, derived and inflected words are picked up from different lexical categories like Noun, Verb, Adjective and Adverbs. Finally, a list of 5817 words is collected in which schwa deletion rule applies at least once. The databases used for the testing and evaluation by the previous systems like [1],[2],[3] and [26] are not publicly available. Therefore, an another word list of size 22847 words is also used for the testing and evaluation. These words are extracted from the BBC Hindi corpus and it contains the same 16000 words used in [9]. The word lists from both the databases are fed to the software and resultant output is verified against the list annotated by an expert at four levels; a) Syllabification b) Syllable Labeling c) Schwa deletion and d) Stress Alignment using Prosodic Label. The summary of test result is shown below in Table 4.

7. Conclusion

In this paper, a software module A2P converter is described. The software is based on two finite state machines and non-linear phonological rules. The A2P converter shows an accuracy of more than 99% at all the testing level. The current implementation cannot be compared with existing systems because the past systems are only for G2P conversion. However, the A2P converter shows an improvement of more than 2% for G2P conversion over the best existing G2P converters.
8. References

[1] K. Bali, P. P. Talukdar, N. S. Krishna, and A. Ramakrishnan, “Tools for the development of a hindi speech synthesis system,” in Fifth ISCA Workshop on Speech Synthesis, 2004.

[2] B. Narasimhan, R. Sproat, and G. Kiraz, “Schwa-deletion in hindi text-to-speech synthesis,” International Journal of Speech Technology, vol. 7, no. 4, pp. 319–333, 2004.

[3] M. Choudhary, “Rule-based grapheme to phoneme mapping for hindi speech synthesis,” in 90th Indian Science Congress of the International Speech Communication Association (ISCA), Bangalore, India, 2003.

[4] M. Ohala, Aspects of Hindi phonology. Motilal Banarsidass Publisher, 1983, vol. 2.

[5] A. Bellur, K. B. Narayan, K. Raghava Krishnan, and H. A. Murthy, “Prosody modeling for syllable-based concatenative speech synthesis of hindi and tamil,” in Communications (NCC), 2011 National Conference on. IEEE, 2011, pp. 1–5.

[6] S. P. Kishore and A. W. Black, “Unit size in unit selection speech synthesis.” in INTERSPEECH, 2003.

[7] A. Ganapathiraju, J. Hamaker, J. Picone, M. Ordowski, and G. R. Doddington, “Syllable-based large vocabulary continuous speech recognition,” Speech and Audio Processing, IEEE Transactions on, vol. 9, no. 4, pp. 358–366, 2001.

[8] A. Lakshmi and H. A. Murthy, “A syllable based continuous speech recognizer for tamil.” in INTERSPEECH, 2006.

[9] P. Pandey, “Akshara-to-sound rules for hindi,” Writing Systems Research, vol. 6, no. 1, pp. 54–72, 2014.

[10] J. Ito and A. Mester, “Recursive prosodic phrasing in japanese,” Prosody matters: Essays in honor of Elisabeth Selkirk, pp. 280–303, 2012.

[11] V. Martínez-Paricio, “Superfeet as recursion,” in 30th West Coast Conference on Formal Linguistics (WCCFL 30), University of California, Santa Cruz, USA, 2012.

[12] B. Hayes, Metrical stress theory: Principles and case studies. University of Chicago Press, 1995.

[13] M. Liberman and A. Prince, “On stress and linguistic rhythm,” Lingustic inquiry, vol. 8, no. 2, pp. 249–336, 1977.

[14] E. O. Selkirk, “The role of prosodic categories in english word stress,” Linguistic inquiry, vol. 11, no. 3, pp. 563–605, 1980.

[15] B. P. Hayes, “A metrical theory of stress rules,” Ph.D. dissertation, Massachusetts Institute of Technology, 1980.

[16] E. O. Selkirk, Phonology and syntax: the relationship between sound and structure. MIT press, 1986.

[17] D. Apoussidou, The learnability of metrical phonology. Netherlands Graduate School of Linguistics, 2006.

[18] P. K. Pandey, “Word accentuation in hindi,” Lingua, vol. 77, no. 1, pp. 37–73, 1989.

[19] T. A. Hall and U. Kleinhenz, Studies on the phonological word. John Benjamins Publishing, 1999, vol. 174.

[20] E. O. Selkirk, “On the major class features and syllable theory,” 1984.

[21] K. Church, “Stress assignment in letter to sound rules for speech synthesis,” in Proceedings of the 23rd annual meeting on Association for Computational Linguistics. Association for Computational Linguistics, 1985, pp. 246–253.

[22] B. Hayes, “Compensatory lengthening in moraic phonology,” Linguistic inquiry, vol. 20, no. 2, pp. 253–306, 1989.

[23] P. K. Pandey, “Hindi schwa deletion,” Lingua, vol. 82, no. 4, pp. 277–311, 1990.

[24] Y. Kashru, Hindi. John Benjamins Publishing, 2006, vol. 12.

[25] P. Bhattacharyya, “Indowordnet,” in In Proc. of LREC-10. Citeseer, 2010.

[26] A. A. Raj, T. Sarkar, S. C. Pammi, S. Yuvaraj, M. Bansal, K. Prahallad, and A. W. Black, “Text processing for text-to-speech systems in indian languages,” in SIS, 2007, pp. 188–193.