Automatic Syllabification Rules for Bodo Language

Jyotismita Talukdar[1] Chandan Sarma[2,] Prof.P.H Talukdar[3]

[1]Asian Institute of Technology, Gauhati University, India,

Abstract

Syllabification performs the task of Identifying syllables in a word or in a sentence. Most of the syllabification tasks are done manually. As the syllabification rules vary from language to language so it is difficult to design a common syllabification rules or algorithm to fit all the languages. On the other hand Syllabification rules are the basic backbone for any task related to text-to-speech conversion system. An attempt has been made through this paper to design an automatic syllabification rules for Bodo language . An algorithm has been developed for automatic syllabification of Bodo language and the algorithm is evaluation on 5000 phonetically rich words obtained from different sources like Text books, Newspapers and Radio News. The result is compared with the same words manually syllabified. The algorithm performs well upto about 95.5 %

Key Words: Text-to speech, phoneme, syllable , monophthongs etc.

1. 0: INTRODUCTION

A Syllable is a unit between phoneme and word. It is larger than phoneme and smaller than word[1]. Many theories are available in phonetics and phonology to define a syllable [2]. In phonetics, the syllables are defined based upon the articulation [3]. But in phonology, the syllables are termed as the different sequences of the phonemes.

In Syllabification , a word is divided into its constituent syllables [4]. Syllabification is an important component in TTS system [5]. It helps the implementation of certain letter-to-phoneme rules or Grapheme-to-Phoneme(G2P) rules. Detecting the syllable correctly helps in modeling duration and improve the synthesized speech intonation. The syllabification has a big role in enhancing the quality of synthetic speech. The quality of a TTS system is measured by the extent of naturalness of the synthetic speech produced by the system [6]. Syllabification helps to improve the naturalness of the synthetic speech [7].

Text-to-speech (TTS) systems are considered as a very mature technology in the speech area [3]. It permits automatic synthesis of speech from text. To make the systems more robust, efficient and reliable, it is crucial to have a good

Pre-processing module for the word corpus to be used in the training of a TTS system and for the texts to be synthesized. The pre-processing module , also known as **front-end**, is composed of three stages: **text analysis, phonetic analysis and prosodic generation**.

In this paper an algorithm to syllabify Bodo words into syllables is proposed. The algorithm was tested on a text corpus which contains representative words for each grammatical rule. The degree of success was measured in terms of the percentage of correctly syllabified words.

2.0: BODO LANGUAGE AND ITS PHONOLOGICAL STRUCTURE

Bodo language belongs to the branch of Barish section under Baric division of the Tibeto-Burman languages. It is spoken by the Bodo people of north-eastern India, part of North Bengal and Nepal. The language is one of the official languages of the Indian state of Assam, and is one of the 22 scheduled languages .The inventory of Bodo phonemes consists of the number of (a) **Segmental phonemes** being consonants and vowels and (b) **Suprasegmental phonemes** being tone, juncture and contour, co-occurring with them as extra sound features used in the language [9]. Bodo language contains 22 segmental phonemes; **six pure vowels** or **monophthongs** and **sixteen consonants** including **two semi vowels.**

The Bodo vowels and consonants are shown in **Table 1.0** and **Table2.0**

Table 1.0: Bodo Vowels

	Front	**Central**	**Back**
Close	**I**		**u w**
Mid	**E**		**ɔ**
Open		**a**	

Table 2.0: Bodo Consonalts

Manner of articulation	Bilabial		Alveolar		Alveolo-Palatal		Velar		Glottal
	Vl	Vd	Vl	Vd	Vl	Vd	Vl	Vd	Vd
STOP			b		d			g	
		p^h		t^h			k^h		
Nasal			m		n			ŋ	
Fricative					s	z			h
Trill					r				
Lateral					l				
Semi-vowel		w				j			

2. Syllabfication

Phonological rules and constraints basically apply within syllables or at syllable boundaries , so the linguists view syllables as an important unit of prosody [10]. Apart from purely linguistic significance, syllables play an important role in speech synthesis and recognition [11]. For a given phoneme the pronunciation tends to vary depending on its location within a syllable. While actual implementations vary, text-to-speech (TTS) systems must have, at minimum, three components [12] : **a letter-to-phoneme (L2P) module, a prosody module, and a synthesis module**. Syllabification play important roles in all these three components.

3.1:Methodology

The methodology followed in the present study was (i) To examine the Bodo Syllable structures from the Linguistics literature (ii) Gathering the opinions of scholars from the various linguistic traditions. (iii) Dialects variations. Based on these three issues we have developed syllabification rules and an algorithm to syllabify Bodo words automatically.

3.2: Syllable Structure in Bodo.

Bodo language is highly **monosyllabic.** A Bodo word may be either a monosyllabic or a polysyllabic, co-occurring with either rising tones or falling tones.

The syllables are described as sequences of phonemes in segments of the vowels(V) and the consonants(C) and also of the clusters in consonants. The Bodo syllable structures may be divided into the following types based on the distribution of the segmental phonemes and consonantal clusters-

1. V
2. VC
3. CCV
4. CCVC
5. CVCVCC
6. CVCCCVC
7. CVCCVCCVCCV

3.3:Syllabification Procedure

In Bodo language every syllable has at least one vowel sound. So, number of vowel sounds in a word equals to the number of syllable. A monosyllabic word (e.g **/ai/)** need not to be syllabified and consonants blends and digraphs (/**kh/, /ph**/) also need not to be syllabified. The basic rules of syllabification in Bodo are detailed below :.

1) A word with single vowel syllable boundary assigned at the end of the word.(**Rule #1).**

2) For vowels with different sounds in a word, the syllable boundary is marked after the first vowel.ie if VV then **V/V. (Rule #2).**

3) For a word with **consonant-vowel structure** like **VCV** then mark the syllable boundary after the first vowel.ie if **VCV then V/CV. (Rule #3).**

4) For a word with consonant-vowel structure like VCCV means two consonants exist between two vowels then mark the syllable boundary between the consonants.ie if VCCV then **VC/CV**.(**Rule #4).**

5) With three consonants comes between two vowels in a word like **VCCCV** then mark the syllable boundary after the first consonants.ie if VCCCV then **VC/CCV.(Rule #5)**

6) Aa word with **consonant-vowel structure** like **CVVC** means two vowels come between two consonants then mark the syllable boundary between the vowels forming two syllables.ie if CVVC then CV/VC.(**Rule #6)**

7) A word with **consonant-vowel structure** like **CVVCV** , mark the syllable boundary after two consecutive vowels.ie if CVVCV then **CVV/CV.(Rule #7).**

8) If a word has consonant-vowel structure like VCVC or VCVV means VCV followed by either C or V then mark syllable boundary after the first vowel.ie if VCVC or VCVV then V/CVC or V/CVV.(**Rule #8).**

9) If a word has consonant-vowel structure like CVCVV or CVCVC means CVC followed by either VV or VC then mark the syllable boundary between first vowel and the consonant followed by it.ie if CVCVV or CVCVC then CV/CVV or CV/CVC. **(Rule# 9).**

10) If a word has consonant-vowel structure like CVCCV means if CVC is followed by CV then mark the syllable boundary between two consecutive consonants.ie if CVCCV then CVC/CV.(**Rule #10) .**

3.4 Syllabification Action

All the rules, as mentioned above, were implemented in the algorithm that executes as many iterations as possible to have all the syllables of a given word separated. The Algorithm analyzed a word from left to right. On each iteration, the algorithm attempts to find a new syllable in the portion of the word that has not yet been analyzed by trying to match one of the rules with this entire portion or just part of it. The algorithm follow the hierarchical sequence for the verification of these rules defined above. When a rule is matched, depending on its definition, the algorithm extracts a new syllable from that part of the word which is currently under analysis.

4.0: Syllabification Algorithm:

In this section, the Bodo syllabification rules identified in the section (3.3) are presented in the form of a formal *algorithm*. The function ***syllabify()*** accepts an array of phonemes generated, along with a variable called ***current_index*** which is used to determine the position of the given array currently being processed by the algorithm.

Initially the ***current_index*** variable will be initialized to 0. The ***syllabify()*** functionis called recursively until all phonemes in the array are processed. The function ***mark_syllable_boundary(position)*** will mark the syllable boundaries of an accepted array of phonemes. The other functions used within the ***syllabify()*** function are described
below.

- ***total_vowels(phonemes)****:* accepts an array of phonemes and returns the number of vowels contained in that array.
- *is_vowel(phoneme):* accepts a phoneme and returns true if the given phoneme is a vowel.
- *count_no_of_consonants_upto_next_vowel(phonemes, position):* accepts an array of phonemes and a starting position; and returns the count of consonants from the starting position of the given array until the next vowel is found.

The complete listing of the algorithm is as follows:

```
function  syllabify (phonemes, current_index)

if total_vowels(phonemes) is 1 then
   mark_syllable_boundary(at_the_end_of_phonemes)
else
   if  is_vowel(phonemes[current_index]) is true then
         total_consonants=
         count_no_of_consonants_upto_next_vowel
          (phonemes,current_index)
       if   total_consonants is 0 then
          if  is_vowel(phonemes[current_index+1]) is true then
```

```
            if is_vowel(phonemes[current_index+3) is true
            then
               mark_syllable_boundary(current_index+1)
               syllabify(phonemes,current_index+2)
            end if
            mark_syllable_boundary(current_index)
            syllabify(phonemes, current_index+1)
         end if
      else
         if total_consonants is 1 then
           mark_syllable_boundary(current_index)
           syllabify(phonemes, current_index + 2)
         end if
         if no_of_consonants are 2 then
            mark_syllable_boundary(current_index+1)
            syllabify(phonemes, current_index+3)
          end if
          if  total_consonants are 3 then
             mark_syllable_boundary(current_index+1)
             syllabify(phonemes,current_index+4)
        end if
     end if
  else
      syllabify(phonemes,current_index+1)
  end if
```

5.0: Results and Discussion

The above algorithm was tested on 5000 distinct words extracted from a Bodo corpus and then compared with manual syllabification of the same words to measure accuracy.

Heterogeneous nature of texts obtained from the News Paper, Feature Articles Text books, Radio news etc was chosen for testing the algorithm due to. A list of distinct words was first extracted, and the 5000 most frequently occurring words chosen for testing the algorithm.

The 5000 words yielded some **18,755 syllables** .The algorithm achieves an overall accuracy of **97.05%** when compared with the same words manually syllabified by an expert.

An error analysis revealed that foreign words directly encoded in Bodo produces error.

6. Conclusion

Syllabification is an important component of many speech and language processing systems, and this algorithm is expected to be a significant contribution to the field, and especially to researchers working on various aspects of the Sinhala language.

REFERENCES

[1.] ChandanSarma,U.Sharma,C.K.Nath,S.Kalita,P.H.Taluk dar, **Selection of Units and Development of Speech**

[2.] **Database for Natural Sounding Bodo TTS System**, CISP Guwahati ,March 2012.

[3.] Parminder Singh, Gurpreet Singh Lehal, **Syllables Selection for the Development of Speech Database for**

[4.] **Punjabi TTS System** , IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 6, November 2010.

[5.] R.A. Krakow, **Physiological organization of syllables: a review**, Journal of Phonetics, Vol. 27, 1999, pp. 23-54.

[6.] Susan Bartlett, Grzegorz Kondrak, Colin Cherry, **On the Syllabification of Phonemes**, Human Language

[7.] Technologies: The 2009 Annual Conference of the North American Chapter of the ACL, pages 308– 316,Boulder, Colorado, June 2009. c 2009 Association for Computational Linguistics.

[8.] Y. A. El-Imam, **Phonetization of arabic: rules and algorithms**, Computer Speech & Language, vol. 18, pp. 339–373, October 2004.

[9.] A.W. Black and K.A. Lenzo,**Building synthetic voice**, http://festvox.org/bsv/, 2003

[10.] R. Dale et al. (Eds.), **A Rule Based Syllabification Algorithm for Sinhala**, IJCNLP 2005, LNAI 3651, pp. 438 – 449, 2005.© Springer-Verlag Berlin Heidelberg 2005.
[11.] Couto I., Neto N., Tadaiesky, V. Klautau, A. Maia, R.2010. **An open source HMM-based text-to-speech**
[12.] **system for Brazilian Portuguese**. Proc. 7th International Telecommunications Symposium Manaus.
[13.] Madhu Ram Baro, **Structure of Boro language**,2008
[14.] Juliette Blevins,**The syllable in phonological theory**,1995
[15.] George Kiraz and Bernd M¨obius, **Multilingual syllabification using weighted finite-state transducers.**In Proceedings of the 3rd Workshop on Speech Synthesis,1998.
[16.] Robert Damper. 2001. **Learning about speech from data: Beyond NETtalk.** In Data-Driven Techniques in
[17.] Speech Synthesis, pages 1–25. Kluwer Academic Publishers.