Letter Boundary Identification among Indic Scripts: A Kannada Case Study

Rajesha N.[1] and Manasa G.[2]

[1]n.rajesha@yahoo.co.in, [2]manasa.g84@gmail.com

Abstract

Scripts of Major Indian languages originated from the Brāhmī script. Letters are the basic rendering unit of Indian language scripts. The scripts today are encoded in electronic format using UNICODE standards. Unicode has a catalog of millions of characters. However, Indic Script such as Kannada has around 16 vowels and 36 consonants listed in the Unicode chart. In addition to this, letters are also the combinations of these vowels and consonants like CV, CCV, CCCV, etc. Along with this adding various signs like Anuswara, Visarga, Chandrabindu, Nukta and Avagraha with these combinations of vowels and consonants multiplies the complexity of finding the boundary of a single letter. Thus it can be claimed that letters in Indic scripts are multi-byte character sets which takes the combinations of vowels, consonants and various signs in all shapes and sizes. This paper presents an overview of the algorithm for text processing which can be adopted to process the Indic scripts in the context of multi-byte character sets using Kannada Script as a case study.

Key Words: Multi-Byte Expression, letter boundary, Indic script, UNICODE, ISCII, Finite State Machine Diagram

Introduction

'Letters in Indic scripts are multi-byte character sets which take the combinations of vowels, consonants and various signs in different shapes' (Rejitha K.S, 2020). In text processing it is vital to identify the boundary of letter in Indic Scripts. The single Kannada letter '/ba:/' ('ಬಾ') is a simple combination of CV, having 4 bytes, whereas, the single Kannada letter '/stri:/' ('ಸ್ತ್ರೀ') is a combination of CCCV, having 12 bytes. Unlike English characters, 'In general, a typical Kannada character could be a V, a dead consonant, a CV, a CCV, a CCCV, or a numeral' (Mahadeva Prasad M., 2014.) Simple ASCII characters and punctuations are also subsets of this complex character set. This complex clustering of multi byte character sets poses certain challenges for computer programmers.

In order to process the Indic scripts, which presents such complexity, certain procedures and algorithms have to be developed. Extracting meaningful combinations of letters is a vital process for many NLP applications like n-grams, Morphological Analyzers etc.

The Kannada script developed from the Kadamba and Cālukya scripts, descendents of Brāhmī, were used between the 5th and 7th centuries AD. These scripts developed into the Old Kannada script. The Old Kannada Script, morphed into the Middle-Kannada script by 1500AD which later evolved into contemporary Kannada script. This Kannada script is used to write Kannada, Tulu, Konkani, Kodava, Badaga and some tribal languages.

Internal representation or encoding of text in Indian Languages may be viewed as the problem of assigning codes to the letters of the languages. The complexities of the syllabic writing systems in use have presented difficulties in standardizing internal representations.

Multilingual text processing is one of the essential requirements when it comes to digitization in India. Applications developed must cater to users in different languages. Encoding standards have emerged to deal with this multilingual content for representing text in computer application. ISCII and Unicode are such standards which came into existence to cater these needs. Today, developers across the world are committing themselves to providing Unicode support in all their applications.

ISCII Standard:

ISCII was evolved as a standard for Brāhmī based Indic Scripts by 1991. Some of the Features of ISCII are.

- ISCII is a single encoding representation for all the Indian Scripts.
- Upper ASCII region (160 - 255) is used for the letters of the language.
- Matras (vowel extensions) are given separate codes

The concept of Multi-byte expression came into picture when Indic Scripts were standardized in ISCII. The multi-byte expression of letters can vary from one byte to as many as 10 bytes for a syllable.

ISCII provides the usage of '*suruli'* character to be used once or twice based to represent a conjunct or a pure consonant. This makes the processing complex since it results in more than one internal representation for the same syllable. Though representation at the level of a syllable is possible in ISCII, processing a syllable can become quite complex, i.e., linguistic processing may pose specific difficulties due to the variable length codes for syllables. Any number of arbitrary syllables can be formed in an ISCII string even though in practice it is limited in a script.

ISCII Kannada	A0	B0	C0	D0	E0	F0
0		ಓ	ಢ		ೆ	EXT
1		ಔ	ಣ	ಲ	ೇ	೦
2	ಂ		ತ	ಳ	ೈ	೧
3	ಃ	ಕ	ಥ			೨
4	ಅ	ಖ	ದ	ವ	ೊ	೩
5	ಆ	ಗ	ಧ	ಶ	ೋ	೪
6	ಇ	ಘ	ನ	ಷ	ೌ	೫
7	ಈ	ಙ		ಸ		೬
8	ಉ	ಚ	ಪ	ಹ	್	೭
9	ಊ	ಛ	ಫ	INV		೮
A	ಋ	ಜ	ಬ	ಾ		೯
B	ಎ	ಝ	ಭ	ಿ	.	
C	ಏ	ಞ	ಮ	ೀ		
D	ಐ	ಟ	ಯ	ು		
E		ಠ		ೂ		
F	ಒ	ಡ	ರ	ೃ	ATR	

Table 2: ISCII Standard for Kannada

Unicode for Indian Languages:

Unicode is an attempt to standardize encoding for multilingual documents across scripts of all languages. In respect of Indian languages Unicode almost follows ISCII. It has provided encoding only for the most basic units of the writing systems which include the vowels, consonants and the vowel modifiers.

Unlike ISCII, which has a uniform coding scheme for all the languages, Unicode has provided individual planes for the nine major scripts of India. Within these planes of 128 code values each, assignments of values in ISCII is almost retained.

Unicode suffers from the same limitations as ISCII. 'suruli', the diacritic symbol that marks the pure consonant (Patil, Vijayalaxmi F. et al, 2019), adds up to the complexity of letter boundary identification, even though not practically used, but in theory n number of consonants can be added to make a consonant conjunct on n number of consonants keeping the 'suruli' as glue. There are some questionable assignments in Unicode in respect of Matras. A Matra is not a character by itself. It is used in representation of a combination of a vowel and consonant, in other words the representation of a medial vowel. A vowel and NOT its Matra is the basic linguistic unit. Consequently linguistic processing will be difficult with Unicode with Indian languages, just as in ISCII.

In initial stages of application development in India, text rendering was a major issue as syllabic writing system in Indic scripts adds to the complexity. Earlier applications gave more emphasis on text entry and display rather than computation. Therefore the standardizations developed are mainly concerned with aspects of writing system rather than linguistic requirements.

In Indic scripts complexities of writing systems includes a large number of written shapes, but linguist content can be specified using a small set of codes for vowels and consonants. The Designers of ISCII and Unicode compromised with smaller set of code but they also incorporated codes conveying rendering information. These codes follow Devanagari writing system which is not adequate for writing systems of the south. The sorting order of the writing system is also not maintained according to the specific language script. Developers have to take additional care in handling the order in their applications.

In order to process text for Indic Scripts, certain procedures or algorithms have to be followed. This procedure is described below using Kannada script as a case study.

Unicode Kannada Block

In Unicode encoding, Kannada Block is of the range from code point 3200 to 3327 (Hexadecimal: 0C80–0CFF). It consists of following types of characters.

Vowels (V):	ಅ, ಆ, ಇ, ಈ, ಉ, ಊ, ಋ, ೠ, ಌ, ೡ, ಎ, ಏ, ಐ, ಒ, ಓ, ಔ
Consonants (C) :	ಕ, ಖ, ಗ, ಘ, ಙ, ಚ, ಛ, ಜ, ಝ, ಞ, ಟ, ಠ, ಡ, ಢ, ಣ, ತ, ಥ, ದ, ಧ, ನ, ಪ, ಫ, ಬ, ಭ, ಮ, ಯ, ರ, ಱ, ಲ, ವ, ಶ, ಷ, ಸ, ಹ, ಳ, ೞ
Vowel modifiers (VM):	◌, ◌ಂ, ◌ಃ, ◌ಁ, ◌ೱ, ◌ೲ
Suruli (S) :	◌
Matras (M):	◌ಾ, ◌ಿ, ◌ೀ, ◌ು, ◌ೂ, ◌ೃ, ◌ೄ, ◌ೢ, ◌ೣ, ◌ೆ, ◌ೇ, ◌ೈ, ◌ೊ, ◌ೋ, ◌ೌ, ◌ೕ, ◌ೖ
Nukta (N):	◌಼
Numerals (NUM)	೦, ೧, ೨, ೩, ೪, ೫, ೬, ೭, ೮, ೯
Characters found in Kannada Texts outside Unicode Kannada Block are	
Punctuations (PUNC):	Same as Latin and Devanagari Danda
Foreign Characters (FC):	Characters of Non-Kannada Unicode block. E.g. Roman Characters

Table 2: Unicode assignment for Kannada

The possible valid Kannada letters will be

- A vowel
- A vowel + Vowel Modifier
- A consonant
- One or more consonant + Matra
- One or more consonant + Nukta
- One or more consonant + Matra + Vowel Modifier
- One or more consonant + Matra + Nukta
- One or more consonant with surali at the at end of the word

Numerals and other characters like punctuations and foreign characters can be handled as per the objective of the text processing.

The Pseudo-code for Text Processing of Kannada Script:

The initial step is to read the first character.

Case 1: If the character is a numeral consider it as a letter.

Case 2: If the character is a vowel, check the next character

Case 2.1: If the character is VM then consider V+VM as a letter.

Case 2.2: If the character is not VM, then consider V as a letter.

Case 3: If the character is a consonant, check the next character

Case 3.1: If the character is any start character, concatenate all the states from start to previous state as a letter.

Case 3.2: If the character is a S then check for next character

Case 3.2.1: If the character is C then Go to Case 3

Case 3.2.2: If the character is other than C, concatenate all the states from start to previous state as a letter.

Case 3.3: If the character is N then concatenate all the states from start to current state as a letter.

Case 3.4: If the character is M, then check the next character

Case 3.4.1: If the character is any start character, concatenate all the states from start to previous state as a letter.

Case 3.4.2: If the character is N then concatenate all the states from start to current state as a letter.

Case 3.4.3: If the character is VM then concatenate all the states from start to current state as a letter.

Case 3.5: If the character is VM then concatenate all the states from start to current state as a letter

Case 4: If the character is a PUNC, Ignore and go to the next character

Case 5: If the character is a FC, Ignore and go to the next character

Finite State Machine Diagram

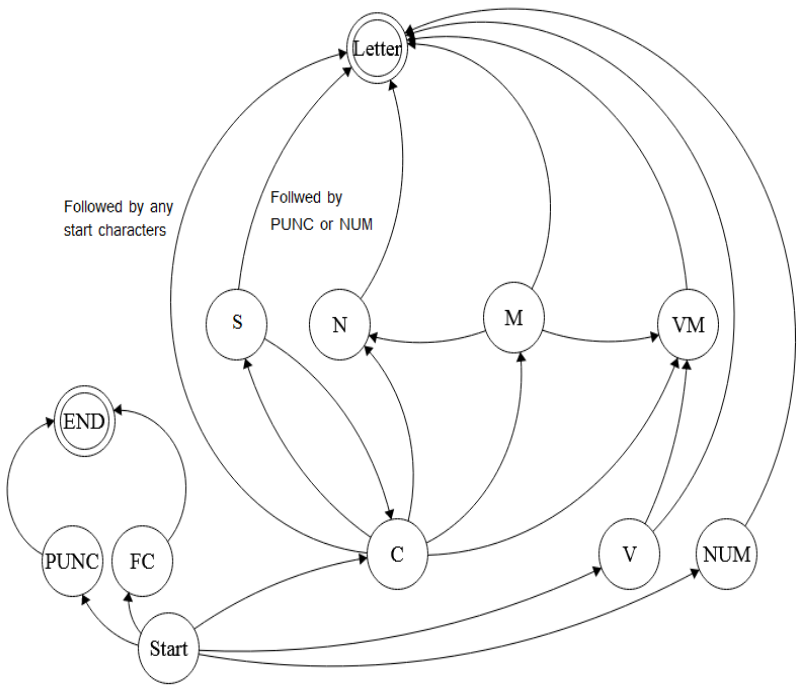

Figure 1: Finite State Machine Diagram of Kannada Letter boundary detection

Abbreviations in the diagram are; PUNC= Punctuations, FC= Foreign Character, C= Consonant, V=Vowel, NUM= Numeric Character, S=Suruli, N=Nukta, M=Matra, VM=Vowel Modifier.

Conclusion

Letter is the basic unit of Indic script, since the Indic letters are multi-byte expressions, and procedure for text processing is imperative for Indic scripts for recognizing the basic unit of the language script. This procedure is vital for some of the text processing NLP applications like N-Grams, syllable counters, Spell Checkers, Morphological analyzers, Sandhi Splitters etc.

References:

K.S. Rejitha. "Letter Based Processing of Indic Script –Malayalam Case Study" in *International Research Journal of Engineering and Technology*, Vol. 07, no. 07, 2020, pp. 2937-2940.

Mahadeva Prasad M. *Online Recognition of Isolated Handwritten Characters.* Department of Electronics, University of Mysore, Mysore, 2014.

Indian Script Standard Code for Information Interchange (ISCII), IS 13194:1991

Patil, Vijayalaxmi F., Chetan Baji, Rajesha N., Manasa G., Narayan Choudhary & L. Ramamoorthy. "Documentation of LDC-IL Kannada Raw Text Corpus" in *Linguistic Resources for AI/NLP in Indian Languages*. Central Institute of Indian Languages, Mysore, 2019, pp. 49-60.