Journal of
Computational and Theoretical Nanoscience
Vol. 17, 4054–4058, 2020

RESEARCH ARTICLE

Metric Analyzer for Kannada Verse Using Rule-Based Approach

R. J. Prathibha*, K. H. Manju Skanda, S. Juned, Anup R. Shetty, and B. V. Shashank

Department of Information Science and Engineering, Sri Jayachamarajendra College of Engineering, Mysore 570006, Karnataka, India

Metrical poetry in any language is called *Chandassu* (ಛಂದಸ್ಸು). It generates rhythm to poem when the predefined metric rules are properly followed. The classification of *Chandassu* is done with the help of syllables known as *Laghu* (ಲಘು), *Guru* (ಗುರು) and *Gana* (ಗಣ). The proposed metric analyzer for Kannada verse is a rule-based teaching and learning tool devised to identify and classify the *Chandassu* of input Kannada poem. This tool also contains an exercise module to test the level of understanding of learners about metric analysis. Accuracy obtained by the proposed system is very good.

Keywords: *Kannada* Verse, *Vrutta*, *Chandassu*, Metric Analysis, *Laghu*, *Guru*, *Gana* Pattern.

1. INTRODUCTION

The ancient and medieval Kannada poetry was composed of three distinct prosodic modes viz., (i) *Akshara Chandassu* (ಅಕ್ಷರ ಛಂದಸ್ಸು), (ii) *Maatra Chandassu* (ಮಾತ್ರ ಛಂದಸ್ಸು) and (iii) *Amsha Chandassu* (ಅಂಶ ಛಂದಸ್ಸು). They are based on different principles of measuring the phonetic units of poetry. The time taken to utter a short vowel is the basic unit of these prosodic systems. This unit is referred to as one '*Maatraa*.' There is no change in the time required if a consonant is added to a vowel or vice versa, without a break either before it or after it. Hence 'a' (ಅ) and 'ka' (ಕ) are both worth one '*Maatra*.' Such vowels and vowel-consonant combinations are designated as '*Hrasva Akshara*' (short letter). The time taken to utter a long vowel either independently or in combination with a consonant is double the time required to utter a short vowel. Consequently, they are worth two '*Maatraas*.' Long vowels and their combinations with consonants are called '*Deergha Akshara*' (long letter). A syllable worth one *Maatraa* is called '*Laghu*' and that which is worth two *Maatraas* is designated as '*Guru*.' There are some basic rules to decide whether a letter is '*Laghu*' or '*Guru*.' The '*Laghu*' and '*Guru*' are represented with the symbols 'U' and '_' respectively.

1.1. *Gana* Classification

There are mainly three types of *Gana* viz., (i) *Akshara Gana* (ಅಕ್ಷರ ಗಣ), (ii) *Maatraa Gana* (ಮಾತ್ರ ಗಣ) and (iii) *Amsha Gana* (ಅಂಶ ಗಣ). A group of letters irrespective of the number of *Maatraas* is called '*Akshara Gana*.' The classification *Akshara Gana* is based on *Gana* which is the pattern of *Laghu* and *Guru* in a sequence of three syllables. There are eight different *Akshara Ganas* viz., (i) *yagana* (ಯಗಣ: U_ _), (ii) *magana* (ಮಗಣ: _ _ _), (iii) *tagana* (ತಗಣ: _ _U), (iv) *ragana* (ರಗಣ: U _ U), (v) *jagana* (ಜಗಣ: U _ U), (vi) *bagana* (ಬಗಣ: _ U U), (vii) *nagana* (ನಗಣ: U U U) and (viii) *sagana* (ಸಗಣ: U U _).

A group of letters containing a particular number of *Maatraas* irrespective of the number of letters is called either a '*Maatraa Gana*' or an '*Amsha Gana*.' *Maatraa Gana* takes the count of *Maatraas* in each *Gana*. *Amsha Gana* is another approach where it is dependent on the way the letters in the poem are pronounced.

1.2. Categories of *Kannada Chandassu*

There are three categories in *Kannada Chandassu* based on *Maatraas* viz., (1) *Kanda Padya* (ಕಂದಪದ್ಯ), (2) *Shatpadi* (ಷಟ್ಪದಿ) and (3) *Ragale* (ರಗಳೆ). Detailed description about these three *Chandassu* is given below.

(i) *Kanda Padya*: The main characteristics of *Kanda Padya* are:

- A verse written with 4 *Maatras* in each *Gana*. The verse should not contain any *jagana* (ಜಗಣ: U_U) pattern at even places.
- The second line should end with a *Guru*.
- Only 6th and 14th *Gana* can contain *jagana* (ಜಗಣ: U_U) pattern.

(ii) *Shatpadi*: *Shatpadi* is a native meter in *Kannada* prosody that has been used extensively in *Kannada* poetry. *Shatpadi* can usually have 6 *Paadas* of syllables, divided

*Author to whom correspondence should be addressed.

 1546-1955/2020/17/4054/005 doi:10.1166/jctn.2020.9009

into groups of various fixed number of *Maatraa* in each line. It should not contain any ಜಗಣ (*jagana*: U_U) pattern. It was most efficiently employed by the great medieval *Kannada* poets such as Raghavanka, Kumaravyasa, and Lakshmisha. Following are the different types of *Shatpadi* available in *Kannada*.

a. *Shara Shatpadi* (ಶರ ಷಟ್ಪದಿ): *Shara Shatpadi* has 6 *Paadas*. The lines 1, 2, 4 and 5 are similar having 2 *Ganas* of 4 *Maatraas* each. 3rd and 6th line has 3 *Ganas* of 4 *Maatraas* each.

b. *Kusuma Shatpadi* (ಕುಸುಮ ಷಟ್ಪದಿ): The lines 1, 2, 4 and 5 are similar having 2 *Ganas* of 5 *Maatraas* each. 3rd and 6th line has 3 *Ganas* of 4 *Maatraas* each.

c. *Bhoga Shatpadi* (ಭೋಗ ಷಟ್ಪದಿ): The lines 1, 2, 4 and 5 are similar having 2 *Ganas* of 3 *Maatraas* each. 3rd and 6th line has 6 *Ganas* of 3 *Maatraas* each.

d. *Bhamini Shatpadi* (ಭಾಮಿನಿ ಷಟ್ಪದಿ): The 3rd and 6th line has 3 *Ganas* of 7 *Maatras* each and a *Guru* at last. Rest of the lines has 2 *Ganas* of 7 *Maatraas* each. The 7 *Maatraas* should be in a 3 + 4 pattern.

e. *Vardhaka Shatpadi* (ವಾರ್ಧಕ ಷಟ್ಪದಿ): The lines 1, 2, 4 and 5 are similar having 4 *Ganas* of 5 *Maatraas* each. 3rd and 6th line has 3 *Ganas* of 4 *Maatraas* each.

f. *Parivardhini Shatpadi* (ಪರಿವರ್ಧಿನಿ ಷಟ್ಪದಿ): The lines 1, 2, 4 and 5 are similar having 4 *Ganas* of 4 *Maatraas* each. 3rd and 6th line has 6 *Ganas* of 4 *Maatraas* each.

(iii) *Ragale*: This meter can usually have as many *Paadas* of syllables divided into two groups of a various fixed number of *Maatraa* in each line. The three different types of *Ragale* are explained below.

(i) *Lalitha Ragale* (ಲಲಿತಾ ರಗಳೆ): In this variation of meter, each *Paada* has 4 syllable groups (*Gana*). Each syllable group has 5 *Maatraas*.

(ii) *Mandanila Ragale* (ಮಂದಾನಿಲ ರಗಳೆ): In this variation of meter, each *Paada* has 4 syllable groups. Each syllable group has 4 *Maatraas*.

(iii) *Utsaha Ragale* (ಉತ್ಸಾಹ ರಗಳೆ): In this variation of *Ragale* meter, each *Paada* has 4 syllable groups. Each syllable group has 3 *Maatraas*.

1.3. Categories of *Kannada Vrutta*

The poem written on the basis of *Akshara Gana* is called *Vrutta*. In *Kannada*, there are 6 types of *Vruttas* viz., (i) *Utpala Maala* (ಉತ್ಪಲಮಾಲಾ), (ii) *Champaka Maala* (ಚಂಪಕಮಾಲಾ), (iii) *Shardula vikridita* (ಶಾರ್ದೂಲ ವಿಕ್ರೀಡಿತ), (iv) *Mattebha Vikridita* (ಮತ್ತೇಭ ವಿಕ್ರೀಡಿತ), (v) *Sragdharaa* (ಸ್ರಗ್ಧರಾ), (vi) *Mahaa Sragdharaa* (ಮಹಾ ಸ್ರಗ್ಧರಾ). Each *Vrutta* has its own characteristics.

2. LITERATURE REVIEW

Rama and Meenakshi Lakshmanan [1] gave an insight into the computational logic of the *Chandassu* for *Sanskrit* language using rule-based approach. In this paper, just identification of metric is carried out. Sekhar Reddy and Humera Khanam [2] presented a paper on *Chandassu* recognizer for *Telugu* poems using rule-based approach for classification, identification and recognition of *Telugu* poems. Diwakar Mishra and Girish Nath Jha [3] have designed a model for metric analysis of *Sanskrit* poetry text. Literature shows that no attempt has been made on the development of metric analyzer for *Kannada* verses. Hence there is a scope for the design of metric analyzer for *Kannada* poems.

3. PROPOSED SYSTEM

3.1. Architecture

Architecture of the proposed metric analyzer for *Kannada* verse is given in Figure 1. This system has three main modules viz., (i) Text Normalizer, (ii) *Akshara* (letter) splitter and (iii) Identification of *Laghu*, *Guru*, *Gana*, *Matraas* and determination of *Chandassu/Vrutta*. The major functionality of each module is given below.

(i) *Text Normalizer*: This module removes all line spaces, extra spaces, special characters and non-*Kannada* letters that are present in the given input *Kannada* poem.

(ii) *Akshara splitter*: This module splits the given verse into *Aksharas* or letters using the rule-based approach.

(iii) *Identification of Laghu, Guru, Gana, Maatraas and Determination of Chandasssu/Vrutta*: Once the individual *Aksharas* of input poem are extracted, identification of *Laghu* and Guru for each *Akshara* is carried out using rule-based approach. On the basis of the identified *Laghu* and *Guru*, their respective *Maatras* and *Ganas* are determined, grouped and *Chandassu/Vrutta* of input poem is identified.

3.2. Methodology

The proposed metric analyzer for *Kannada* verse is designed as a teaching and learning tool in order to help the students, researchers, linguists, and language enthusiasts. This tool is also designed to make the user to understand *Chandassu* to a greater extent with examples and exercises. This tool is developed using python, html, CSS, bootstrap, and JavaScript. The input *Kannada* script is represented in Unicode which is a computing industry standard for encoding and handling of text in regional languages of India. The proposed system is designed using

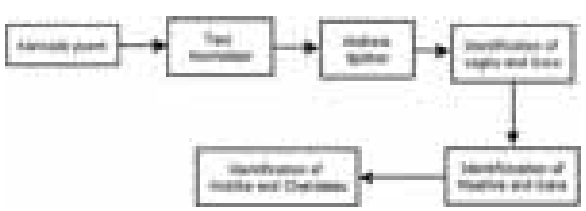

Fig. 1. Architecture of the proposed metric analyzer for Kannada verse.

RESEARCH ARTICLE

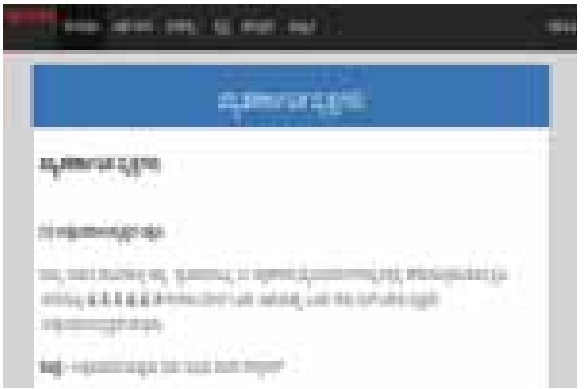

Fig. 2. Introduction about *Khyaathakarnataka Vrutta*.

rule-based approach. For each input *Akshara*, a set of predefined rules are applied to determine whether it is a *Laghu* or *Guru*. For *Laghu* and Guru, *Weightage* 1 and 2 is given respectively. On the basis of this determined *Laghu* and *Guru*, their respective *Ganas* are identified and grouped. The *Ganas* follow a particular pattern of occurrence of elements. Thus, the *Chandassu/Vrutta* is determined. The rules that are used to identify *Vrutta/Chandassu* is given in the introduction section.

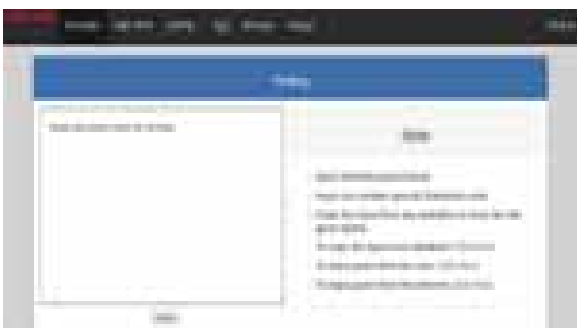

Fig. 3. Options to input *Kannada* poem from the user.

Fig. 4. Different categories of input poems from local database.

4. EXPERIMENTAL RESULTS AND DISCUSSION

The proposed metric analyzer for *Kannada* verse mainly has three phases viz., (i) Introduction, (ii) Testing and (iii) Exercise. The detailed description about each phase is given below.

(i) *Introduction phase*: Introduction page gives brief information related to different rules applied to determine *Laghu*, *Guru*, *Gana* for the input poem in *Kannada* and also the characteristics of different *Chandassu* and *Vrutta*. An example for introduction about *khyaathakarnataka vrutta* (ಖ್ಯಾತಕರ್ನಾಟಕ ವೃತ್ತ) in *Kannada* is given in Figure 2.

(ii) *Testing phase*: The user has to give input poem in *Kannada* to test the performance of the system. There are 3 ways using which the user can input *Kannada* poem viz., (i) from existing database which is created in this proposed work, (ii) using *English* to *Kannada* transliteration tool (http://kannada.changathi.com), or (iii) selecting from internet source like (http://padyapaana.com/). The screenshots for the selection of input poem by the user by copying poem from local database is given in Figures 3–5.

Given input *Kannada* poem can contain special characters, non-*Kannada* letters or words, extra spaces,

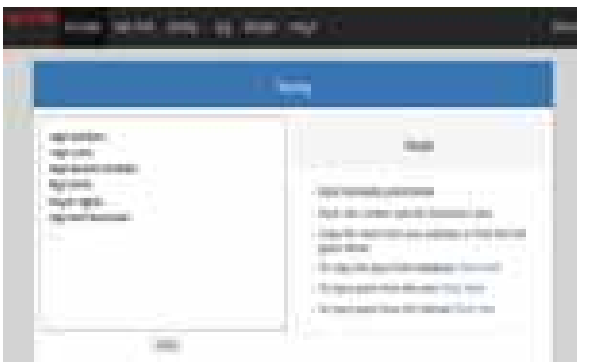

Fig. 5. Selected input *Kannada* poem from local database.

Fig. 6. Normalizer output for the given input poem.

RESEARCH ARTICLE

Fig. 7. Split *Aksharas* for the given sample input poem.

punctuation marks, etc. These types of characters are removed by the normalizer module. The output obtained by the normalizer for the selected sample input poem is shown in Figure 6.

Akshara Splitter module splits the given poem into set of Aksharas or letters. The split *Aksharas* for the given sample input poem is shown in Figure 7.

Identification of *Laghu*, *Guru*, *Gana* and *Chandassu/Vrutta* is carried out by the proposed system using rule-based approach. The given sample input *Kannada* poem belongs to *Shara Shatpadi*. The screenshot for the output obtained by the system for the given sample input poem is given in Figure 8.

If the given *Kannada* poem does not belong to any of the predefined *Chandassu/Vrutta*, then the proposed metric analyzer determines *Laghu*, *Guru* for each *Akshara* and

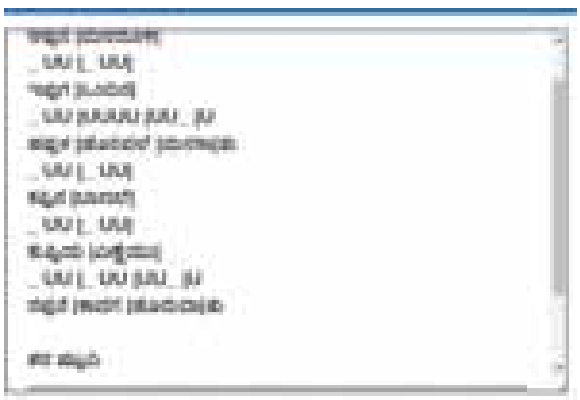

Fig. 8. Output for the selected sample input poem.

Fig. 9. Output for the poem does not belong to any *Chandassu/Vrutta*.

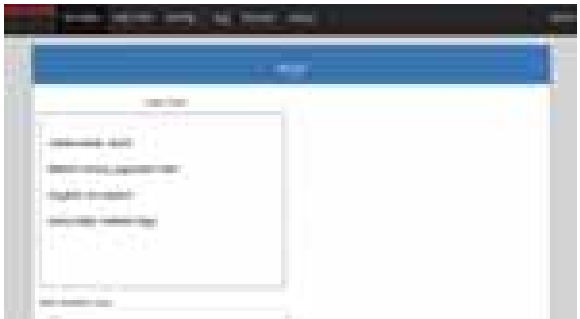

Fig. 10. Poem given by the system to test the user.

displays *Chandassu* as Invalid. The screenshot for the output obtained by the system for the selected poem which does not belong to any of the predefined *Chandassu/Vrutta* is given in Figure 9.

Fig. 11. *Laghu*, *Guru* and *Chandassu* provided by the user.

Fig. 12. Checking user input with the system's output.

(iii) Exercise Phase

This is one of the important phases in the proposed metric analyzer for *Kannada* verse, where the understanding level of the student or learner about metric analysis is verified. In this phase, a poem is selected randomly by the system and displayed in the user input box where the user is required to insert the respective *Laghu*, *Guru* for each *Akshara* and predict the correct *Chandassu/Vrutta* to which the given poem belongs. The screenshots of the steps followed in exercise phase are given in Figures 10–12.

Accuracy of the proposed metric analyzer for *Kannada* poem is extensively appreciable. Almost all poems written by ancient poets and modern poets are collected from different websites and tested the proposed system and the obtained results are very good.

5. CONCLUSION AND FUTURE ENHANCEMENT

The proposed metric analyzer for *Kannada* verse/poem is a teaching and learning tool for the students and teachers to learn how to determine *Laghu* and *Guru* for each *Akshara* in the given Kannada poem and identify its *chandassu* or *vrutta*. Input poem may contain special characters, numbers and non-*Kannada* letters, punctuation marks, etc. In such case, the normalizer module, removes all these undesired characters and provides only Kannada letters. The *Akshara* splitter, splits the normalized poem into individual and independent *Akshara* and determines *Laghu*, *Guru* and *Gana,* and identifies the *Chandassu/Vrutta* of the poem. In exercise phase, users' level of understanding about the metrics will be tested by providing different type of *Kannada* poems and asked the user to determine *Laghu*, *Guru*, *Gana* and *Chandassu* or *Vrutta*. If the given input poem belongs to any one of the predefined *Chandassu/Vrutta*, then the accuracy of the system is very good. If the input poem does not follow any of these predefined patterns, then the system will determine *Laghu* and *Guru* of the *Aksharas* present in the input poem and gives invalid *Chandassu/Vrutta* as output.

The output obtained by the proposed system can be used as training dataset and in future metric analyzer for *Kannada* verse can be built using machine learning approaches. This approach can be used to build metric analyzer tool for poems written in different Indian languages. New variety of poem can also be included by providing the rules for pattern of new poem. This system can also be extended to predict the actual poets of the given input poems. And also this system can be integrated with *Kannada* text-to-speech convertor to make the blind people to learn about *Kannada Chandassu/Vrutta* patterns.

References

1. Rama, N. and Meenakshi Lakshmanan, **2010**. A computational algorithm for metrical classification of verse. *International Journal of Computer Science Issues, 7*(2), pp.46–53.
2. Sekhar Reddy, M. and Humera Khanam, **2016**. Chandhassu recognizer for Telugu poems. *International Journal of Computer Science and Information Technologies, 7*(3), pp.1425–1434.
3. Diwakar Mishra and Girish Nath Jha, **2006**. Strategies for Metrical Analysis of Sanskrit Text. *Conference Proceedings in National Seminar on Perspectives in Linguistics*.

Received: 23 September 2019. Accepted: 10 October 2019.