JSS MAHAVIDYAPEETHA
Sri Jayachamarajendra College of Engineering, Mysuru-570006
An Autonomous Institute
Affiliated to
Visvesvaraya Technological University, Belagavi

"Metric Analysis and Identification of Metrics for Given Kannada Verse using Rule-Based Approach"

Report submitted in partial fulfilment of curriculum prescribed for the award of the degree of

BACHELOR OF ENGINEERING
IN
INFORMATION SCIENCE AND ENGINEERING

Submitted by
Anupa Rathnakar Shetty [4JC15IS005]
B V Shashank [4JC15IS009]
Juned S [4JC15IS015]
Manju Skanda K H [4JC15IS019]

Under the Guidance of
Dr R J Prathibha
Assistant Professor,
Department of Information Science and Engineering
SJCE, Mysuru.

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING
May 2019

JSS MAHAVIDYAPEETHA

SRI JAYACHAMARAJENDRA COLLEGE OF ENGINEERING

MYSURU-570006

CERTIFICATE

This is to certify that the work entitled "**Metric Analysis and Identification of Metrics for Given Kannada Verses using Rule-Based Approach**" is a bonafide work carried out by **Anupa Rathnakar Shetty, B V Shashank, Juned S, and Manju Skanda K H** in partial fulfillment of the award of the degree of **Bachelor of Engineering** in **Information Science and Engineering of Visvesvaraya Technological University, Belgaum during the year 2018-19**. It is that all corrections/suggestions indicated during Continuous internal evaluation have been incorporated in the report. The report has been approved as it satisfies the academic requirements in respect of project work prescribed for Bachelor of Engineering degree.

Guide	**Head of the Department**	**Principal**
Dr. R J Prathibha, Assistant Professor, Dept. of IS&E, SJCE, Mysuru	**Dr. D S Vinod,** Associate Professor & Head, Dept. of IS&E, SJCE, Mysuru	**Dr. T N Nagabhushan,** Principal, SJCE, Mysuru

Examiners: 1…….

Place: Mysuru 2.……

Date: 3…….

Declaration

We, **Anupa Rathnakar Shetty**, **B V Shashank**, **Juned S**, and **Manju Skanda K H** bearing the USN 4JC15IS005, 4JC15IS009, 4JC15IS015, 4JC15IS019 respectively, hereby declare that the dissertation work entitled "**Metric Analysis and Identification of Metrics for Given Kannada Verses using Rule-Based Approach**" was independently carried out by us under the guidance and supervision of **Dr. R J Prathibha,** Assistant Professor, Department of Information Science and Engineering, Sri Jayachamarajendra College of Engineering, Mysore.

The dissertation work is submitted in partial fulfillment of the requirements for the award of Degree of Bachelor of Engineering in Information Science during the academic year 2018-19.

Anupa Rathnakar Shetty B V Shashank Juned S Manju Skanda K H

Place: Mysuru

Date:

Acknowledgment

With the immense pleasure, we acknowledge the help, guidance, and support received from various quarters during the course of our project. We wish to express our sincere gratitude to **Dr. T N Nagabhushan,** Principal, SJCE and **Dr. D S Vinod**, Head of the Department, Information Science and Engineering, SJCE for the facilities, and support provided throughout the project work. We are profoundly grateful to our project guide and mentor **Dr. R J Prathibha** for her expert guidance and continuous encouragement throughout. We would also like to thank all the faculty members, panelists and non-teaching staff for being a source of constant support.

Anupa Rathnakar Shetty [4JC15IS005]

B V Shashank [4JC15IS009]

Juned S [4JC15IS015]

Manju Skanda K H [4JC15IS019]

Abstract

Kannada is an ancient and one of the Dravidian languages. In Kannada verses are written using combinatorial methods which brings rhythm to the poem. Metrical poetry in the Kannada language is called ***Chandas*** or ***Chandassu***. The metric analyzer for Kannada poems is a tool that is devised to identify and classify *Chandassu* ***(ಛಂದಸ್ಸು)***. *Chandassu* generates rhythm to the literature when the rules are properly followed. The classification of *Chandassu* is done with the help of syllables known as ***Laghu (ಲಘು)*** and ***Guru (ಗುರು)***. The pattern of *Laghu* and *Guru* in a sequence of three is called **Gana(ಗಣ).** The identification and classification of the metric system follow the set of rules using *Ganas* **(ಯಮಾತಾರಾಜಬಾನಸಲಗಂ).** The Kannada script is represented in Unicode which is a computing industry standard for encoding and handling of text expressed in the Indian writing system. In the proposed metric analyzer, the input is fed into the computer system using the Unicode format. This tool consists of four modules. The first module called normalizer removes all embedded spaces, lines, special characters, and non-Kannada characters and also splits the given poem into a set of letters called Kannada-Akshara. The second module assigns *Laghu* and *Guru* to each letter (Akshara) based on the set of rules. The third module identifies and calculates *Matra* and *Ganas*. The *Ganas* are typically made as a group of three, four or five *Matras*. Module four classifies and determines *Chandassu* based on the outputs given by other modules. In Kannada, there are mainly three types of *Chandassu* namely *Kanda Padya, Shatpadi* and *Ragale*. This further includes some sub-classes also. The communication is done with the help of front-end which includes a user interface. This tool provides innovative exercises to the users which test the capability and understanding of the metric analysis by the learner.

This proposed module can be used as a teaching and learning tool in schools and other educational institutions. This can be also used as a reference module by professors, students, linguists, and other language enthusiasts. This will reduce the complexity in learning the complex rules of *Chandassu* prevailing in Kannada language and will help the learner.

TABLE OF CONTENTS

CERTIFICATE **i**

DECLARATION **ii**

ACKNOWLEDGEMENT **iii**

ABSTRACT **iv**

1. Introduction **1**

1.1 Rules for Laghu(ಲಘು) 1

1.2 Rules for Guru(ಗುರು) 2

1.3 Rules for identification of Gana (ಗಣ) 3

1.4 Categories in Chandassu 4

1.4.1 Kanda Padya (ಕಂದಪದ್ಯ) 4

1.4.2 Shatpadi (ಷಟ್ಪದಿ) 4

1.4.3 Ragale (ರಗಳೆ) 8

1.5 Problem Statement 9

1.6 Aim of the project 9

1.7 Motivation 10

1.8 Objectives 10

1.9 Scope 10

2. Literature Survey **11**

2.1 Previous works 11

3. Requirement Specification **12**

3.1 System Features 12

3.2 Functional requirements 14

3.3 Non-Functional requirements 15

3.4 Software Requirements 15

3.5 Hardware Requirements 15

4. Design **16**

4.1 High-level design 16

4.2 Low-level design 17

5. Implementation **20**

5.1 Introduction 20

5.2 Modular specification 21

5.2.1 Normalizer 21

5.2.2 Text (Akshara) splitter 21
5.2.3 Determination of Laghu, Guru and respective Ganas 22
5.2.4 Identification of Chandassu of input poems 23

6. Experimental Result and Analysis 25
6.1 Introduction Page 26
6.2 Testing 26
6.3. Exercise 28

7. Conclusion and future enhancements 31
7.1 Conclusion 31
7.2 Future enhancements 31

ANNEXURE A

REFERENCES

LIST OF FIGURES

Fig 1.1: Rules for Laghu(ಲಘು) 1
Fig 1.2 : Rules for Guru(ಗುರು) 2
Fig 1.3 : Rules for identification of Gana (ಗಣ) 3
Fig 1.4: An example of Kanda Padya 4
Fig. 1.5: An example for Shara Shatpadi 5
Fig. 1.6: An example for Kusuma Shatpadi 6
Fig. 1.7: An example on Bhoga Shatpadi` 6
Fig. 1.8: An example on Bhamini Shatpadi 7
Fig. 1.9: An example on Vardhika Shatpadi 7
Fig. 1.10: An example on Parivardini Shatpadi 8
Fig. 1.11: An example on Lalitha Ragale 8
Fig. 1.12: An example on Mandanila Ragale 9
Fig 1.13: An example on Utsaha Ragale 9
Fig 4.1: The architecture of the proposed Kannada Metric System 16
Fig 4.2 : Function of a normalizer 17
Fig 4.3 : Akshara splitter 18
Fig 4.4 : Laghu-Guru identifier 18

Fig 4.5 : Mathra and Gana identifier 19
Fig 4.6 : Chandassu identifier 19
Fig 5.1: Result obtained by Normalizer module 22
Fig 5.2 : Output obtained by Akshara splitter module 23
Fig 5.3 : Output obtained by Laghu-Guru module 24
Fig 5.4 : Snapshot of output of Shara Shatpadi after classification. 26
Fig 6.1: UI of our project 27
Fig 6.2: The user console waiting for testing verse input 28
Fig 6.3: The local directories to get the input verse 29
Fig 6.4: Input from dataset 29
Fig 6.5: Input verse to the user 30
Fig 6.6: The output for the given input verse 30
Fig 6.7 The user console showing the exercise tool 31
Fig 6.8 Exercise page for the user 31
Fig 6.9 User input and system output 32

LIST OF TABLES

Table 5.1: Kannada Unicode representation table 21

ANNEXURE A

Fig 1: Testing input verse for Shara Shatpadhi 32
Fig 2: Output for the given Shara Shatpadhi verse 32
Fig 3: Testing input verse for Kusuma Shatpadhi 32
Fig 4: Output for the given Kusuma Shatpadhi verse 33
Fig 5: Testing input verse for Bhamini Shatpadhi 33
Fig 6: Output for Bhamini Shatpadhi verse 33
Fig 7: Testing input verse for Vardhika Shatpadi 34
Fig 8: Output for the given Vardhika Shatpadhi verse 34
Fig 9: Testing input verse for Mandanila Ragale 34

Fig 10: Output for the given Mandanila Ragale verse 35

Fig 11: Testing input verse for Utsaha Ragale 35

Fig 14: Output for the given Utsaha Ragale verse 35

Chapter 1

Introduction

Metrical poetry in any language is known as Chandas(ಛಂದಸ್) or Chandassu(ಛಂದಸ್ಸು). Chandassu generates rhythm to the literature when the rules are properly followed. The classification of Chandassu is done with the help of syllables known as Laghu(ಲಘು) and Guru(ಗುರು). The pattern of Laghu and Guru in a sequence of three is called Gana(ಗಣ).

The technique used in this project is a rule-based approach. Classification of Chandassu (ಛಂದಸ್ಸು) can be done based on Gana(ಗಣ). Maatra Gana (ಮಾತ್ರ ಗಣ)is used for the identification of the metrics. Maatra meters are based on Maatras. A maatra (ಮಾತ್ರ) is a unit of time. One maatra is denoted as U and is called Laghu(ಲಘು). Two Maatra is denoted as _ and is called Guru (ಗುರು). There are a set of rules to decide if a letter in a poem takes one or two Maatra times.

The classification based on Gana(ಗಣ) is done by Laghu (ಲಘು) (One Syllable) and Guru (ಗುರು) (Two Syllable).

1.1 Rules for Laghu(ಲಘು)

U	U	U	U	U	U
ಅ	ಇ	ಉ	ಋ	ಎ	ಒ
U	U	U	U	U	U
ಕ	ಕಿ	ಕು	ಚ	ಟ	ತ
U	U	U	U	U	U
ಕೆ	ಕೊ	ಸು	ಸೊ	ಸೃ	ಕೃ

–U	–U	–U	–UU
ಕಲ್ಲು	ಮಣ್ಣು	ನಿಲ್ಲು	ಮೆತ್ತಗೆ

Fig. 1.1: Rules for Laghu(ಲಘು)

The ancient and medieval Kannada poetry was composed in three distinct prosodic modes. They are called respectively ‘akshara chandassu’, ‘matra chandassu’ and ‘amsha chandassu’. (Chandassu= Prosody, Metrics) They are based on different principles of measuring the phonetic units of poetry. However, all of them take the time elapsed during the enunciation of a phonetic unit in to consideration. The stress on a particular syllable does not have any prosodic significance. This is contrary to the practice of English prosody. The time taken to utter a short vowel is the basic unit of these prosodic systems. This unit is referred to as one ‘mAtre’.(ಮಾತ್ರೆ) 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 (ಕ) are both worth one ‘matre’. Such vowels and Vowel-consonant combinations are designated as ‘Hrusva Akshara’(Short letter). These kind of Aksharas are designated as Laghu.

1.2 Rules for Guru(ಗುರು)

ಲಕ್ಷಣ	ಉದಾಹರಣ
ದೀರ್ಘಾಕ್ಷರ	_ U ಶಾಲೆ
ಒತ್ತಕ್ಷರದ ಹಿಂದಿನ ಅಕ್ಷರ	_ U U U ಒ ತ್ತಿ ನ ಣೆ
ಅನುಸ್ವಾರದಿಂದ ಕೂಡಿರುವ ಅಕ್ಷರ	_ U U ಬಂ ದ ನು
ವಿಸರ್ಗದಿಂದ ಕೂಡಿರುವ ಅಕ್ಷರ	_ U ದುಃಖ
ವ್ಯಂಜನಾಕ್ಷರದಿಂದ ಕೂಡಿದ ಅಕ್ಷರ	U U _ ಮನದೊಳ್
ಐ ಸ್ವರವಿರುವ ಅಕ್ಷರ	_ U U ಕೈ ಮು ಗಿ
ಔ ಸ್ವರವಿರುವ ಅಕ್ಷರ	_ U ಮೌ ನ
ಷಟ್ಪದಿಯ ಮೂರು ಮತ್ತು ಆರನೆಯ ಪಾದದ ಕೊನೆಯ ಅಕ್ಷರ	

Fig 1.2: Rules for Guru(ಗುರು)

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 ‘matre’s. Long vowels and their combinations with consonants are called Deergha Akshara (Long letter) A syllable worth one matre is called ‘Laghu’ and that which is worth two matres is designated as ‘Guru’. The letter that precedes a consonant cluster is designated as ‘guru’ and it is worth two matres. There are some more rules to decide whether a letter is ‘laghu’ or ‘guru’. Very rarely one

comes across a syllable which requires three matras for enunciation and they are called ‘Pluta’s. In any combination of letters or phonetic units one finds different patterns of ‘Laghu’ and ‘Guru’ syllables.The rules for Laghu and Guru are shown in Figure 1.1 and Figure 1.2.

1.3 Rules for identification of Gana (ಗಣ)

A group of letters irrespective of the number of matres is called ‘akshara gana’. A group of letters containing a particular number of matres irrespective of the number of letters is called either a ‘matra gana’ or an ‘amsha gana’.

There are mainly three types of gana classification:

1. Akshara Gana
2. Matra Gana
3. Amsha Gana

We use Akshara Gana (ಅಕ್ಷರ ಗಣ) and Maatra Gana (ಮಾತ್ರ ಗಣ) as the rule for the identification of Gana (ಗಣ). The Matra Gana takes into account the classification based on the count of maatras in each gana. This is further enhanced by naming a particular group by an Akshara which are discussed in the table. The Amsha Gana is another approach where it is dependant on the way the letters in the poem are pronounced. Akshara Gana has 8 types and they are given in Figure 1.3.

ಗಣ	ಅಕ್ಷರಗಳು	ಪುಸ್ತಾರ
ಯಗಣ	ಯಮಾತಾ	U _ _
ಮಗಣ	ಮಾತಾರಾ	_ _ _
ತಗಣ	ತಾರಾಜ	_ _ U
ರಗಣ	ರಾಜಭಾ	_ U _
ಜಗಣ	ಜಭಾನ	U _ U
ಭಗಣ	ಭಾನಸ	_ U U
ನಗಣ	ನಸಲ	U U U
ಸಗಣ	ಸಲಗಂ	U U _

Fig 1.3:Rules for identification of Gana (ಗಣ)

1.4 Categories in Chandassu

There are 3 categories in Chandassu. They are,

1. Kanda Padya
2. Shatpadi
3. Ragale

1.4.1 Kanda Padya (ಕಂದಪದ್ಯ)

This is a verse written with 4 matras in a gana. This is known as “ಚಾತುರ್ಮಾತ್ರಾಗಣಗಳಗತಿ” .

The verse should not contain any ಜಗಣ (i.e, U_U) at even places.

● The second line should end with a guru (ಗುರು).

● Only 6th and 14th gana can contain ಜಗಣ or ಸರ್ವಲಘುಂಗಳಿರುವ ವಿನ್ಯಾಸ (U, UUU), if it is ಸರ್ವಲಘುಂಗಳಿರುವ ವಿನ್ಯಾಸ then the yathi should be taken after the first laghu.

Example:

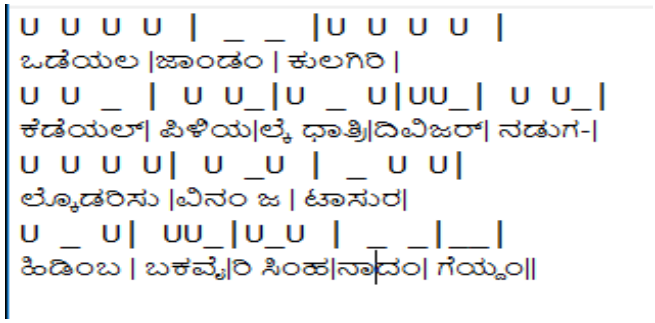

Fig 1.4: Example of Kanda Padya

1.4.2 Shatpadi (ಷಟ್ಪದಿ)

Shatpadi (**ಷಟ್ಪದಿ** Shatpadi) is a native meter in Kannada prosody that has been used extensively in Kannada poetry. It meter can usually have six padas of syllables, divided into groups of a various fixed number of matra (beats) in each line. It was most efficiently employed by the great medieval Kannada poets such as Raghavanka, Kumaravyasa, and Lakshmi. The Shatpadi has 6 types,

1. Shara Shatpadi(ಶರ ಷಟ್ಪದಿ)
2. Kusuma Shatpadi(ಕುಸುಮ ಷಟ್ಪದಿ)
3. Bhoga Shatpadi(ಭೋಗ ಷಟ್ಪದಿ)

4. Bhamini Shatpadi(ಭಾಮಿನಿ ಷಟ್ಪದಿ)
5. Vardhaka Shatpadi(ವಾರ್ಧಕ ಷಟ್ಪದಿ)
6. Parivardini Shatpadi(ಪರಿವರ್ಧಿನಿ ಷಟ್ಪದಿ)

1.4.2.1 Shara Shatpadi (ಶರ ಷಟ್ಪದಿ)

Shara Shatpadi has 6 padas. The lines 1,2,4 and 5 are similar having 2 ganas 4 matras each. 3rd and 6th line has 3 ganas of 4 matras each. It does not contain any ಜಗಣ (i.e, U_U). Example for Shara shatpadi is given in Figure 1.5

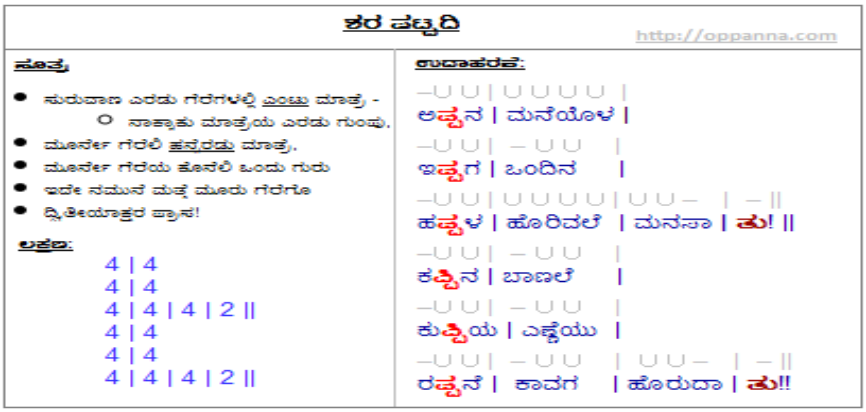

Fig. 1.5: Example for Shara Shatpadi

1.4.2.2 Kusuma Shatpadi (ಕುಸುಮ ಷಟ್ಪದಿ)

This is similar to **Shara Shatpadi (ಶರ ಷಟ್ಪದಿ)**. The lines 1,2,4 and 5 are similar having 2 ganas 5 matras each. 3rd and 6th line has 3 ganas of 4 matras each. It does not contain any ಜಗಣ (i.e, U_U). Example of kusuma shatpadi is given in Figure 1.6.

Fig. 1.6: Example for Kusuma Shatpadi

1.4.2.3 Bhoga Shatpadi(ಭೋಗ ಷಟ್ಪದಿ)

This is similar to other Kusuma Shatpadi. The lines 1,2,4 and 5 are similar having 2 ganasmatras each. 3rd and 6th line has 6 ganas of 3 matras each. It does not contain any ಜಗಣ(i.e,U_U). Example of Bhoga shatpadi is given in Figure 1.7

Fig. 1.7: Example for Bhoga Shatpadi

1.4.2.4 Bhamini Shatpadi(ಭಾಮಿನಿ ಷಟ್ಪದಿ)

This has 6 padas. 3rd and 6th line has 3 ganas of 7 matras each and a guru at last. Rest of the lines have 2 ganas of 7 matras each. The 7 matras should be in a 3+4 pattern. Example of kusuma shatpadi is given in Figure 1.8

ವೇದ| ಪುರುಷನ | ಸುತನ| ಸುತನ ಸ
ಹೋದ|ರನ ಹೆ|ಮ್ಮಗನ| ಮಗನ ತ
ಳೋದ|ರಿಯ ಮಾ|ತುಳನ| ಮಾವನ|ನತುಳ|ಭುಜಬಲ|ದಿ
ಕಾದು| ಗೆಲಿದನ|ನಣ್ಣ|ನವ್ವೆಯ
ನಾದಿ|ನಿಯ ಜಠ|ರದಲಿ| ಜನಿಸಿದ
ನಾದಿ|ಮೂರುತಿ | ಸಲಹೊ| ಗದುಗಿನ | ವೀರ|ನಾರಯ|ಣ

Fig. 1.8: Example for Bhamini Shatpadi

1.4.2.5 Vardhaka Shatpadi(ವಾರ್ಧಕ ಷಟ್ಪದಿ)

This is similar to Shara Shatpadi. The lines 1,2,4 and 5 are similar having 4 ganas 5 matras each. 3rd and 6th line has 3 ganas of 4 matras each. It does not contain any ಜಗಣ (i.e, U_U).

Example for Vardhaka Shatpadi is given in Figure 1.9.

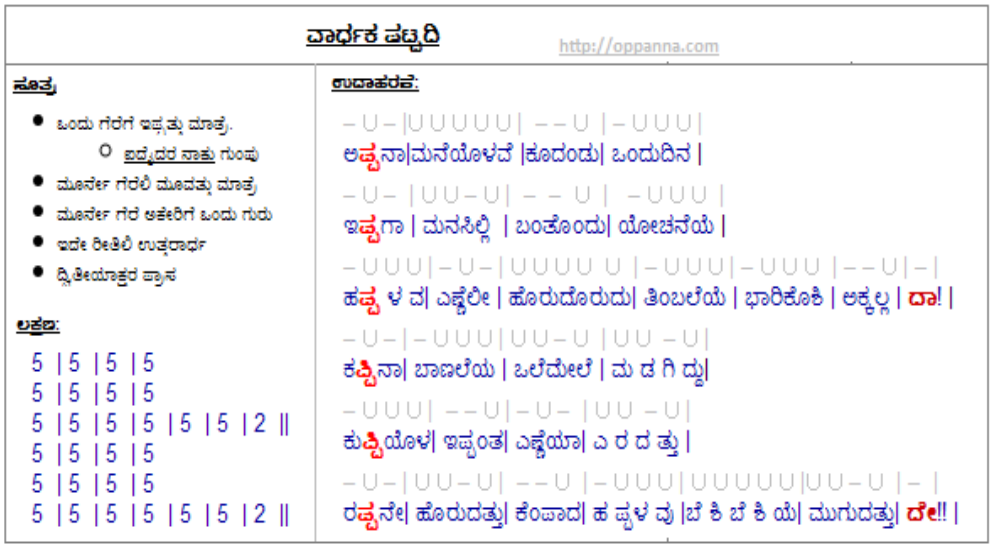

Fig. 1.9: Example for Vardhaka Shatpadi

1.4.2.6 Parivardhini Shatpadi(ಪರಿವರ್ಧಿನಿ ಷಟ್ಪದಿ)

It is similar to Kusuma Shatpadi. The lines 1,2,4 and 5 are similar having 4 ganas 4 matras each. 3rd and 6th line has 6 ganas of 4 matras each. It does not contain any ಜಗಣ (i.e, U_U).

Example for Parivardhini Shatpadi is given in Figure 1.10.

Fig. 1.10: Example for Parivardini Shatpadi

1.4.3 Ragale (ರಗಳೆ)

It is a type of meter in Kannada prosody that is used in Kannada poetry. This meter can usually have as many padas of syllables divided into two groups of a various fixed number of matra in each line. Different types of Ragale are explained below:

1.4.3.1 Lalitha Ragale(ಲಲಿತ ರಗಳೆ)

In this variation of Ragale meter, each pada has four, syllable groups (Gana). Each syllable group has five matras (time units). Example for Lalitha Ragale is given in Figure 1.11.

ಕುಣಿಯುತ್ತೆ |ಮಣಿಯುತ್ತೆ |ಧೂಪಮಂ |ನಲಿದಿತ್ತು|
ತಣಿಯುತ್ತೆ |ತೂಗುತ್ತೆ |ಆರತಿಗಳಂ| ಪೊತ್ತು|
ಆರೋಗ |ಣೆಯನಿತ್ತು |ವೀಳೆಯಂ |ಗಳನಿತ್ತು|
ಪೊಸಗಬ್ಬಮಂ| ಗದ್ಯಪ|ದ್ಯಂಗಳಂ| ಪೇಳ್ದು|

Fig. 1.11: Example for Lalitha Ragale

1.4.3.2 Mandanila Ragale(ಮಂದನೀಲ ರಗಳೆ)

In this variation of Ragale meter, each pada has four, syllable groups (Gana). Each syllable group

has four matras (time units). Example for Mandanila Ragale is given in Figure 1.12.

ಜಡೆಗಳ್| ಬೆಂಬಳಿ |ಎಡಿದಾ|ಡುತ್ತಿರೆ|
ಮುಡಿಯೊಳ್| ಸುರನದಿ |ತುಳ್ಕಾ|ಡುತ್ತಿರೆ|
ಜಡೆಯೊಳ್| ಶಶಿಕಳೆ |ಯಲ್ಲಾ|ಡುತ್ತಿರೆ|
ಕಡುಚೆ |ಲ್ವಳಕಂ |ಕುಣಿದಾ|ಡುತ್ತಿರೆ|

Fig. 1.12: Example for Mandanila Ragale

1.4.3.3 Utsaha Ragale(ಉತ್ಸಾಹ ರಗಳೆ)

In this variation of Ragale meter, each pada has four, syllable groups (Gana). Each syllable group has three matras (time units). Example for Utsaha Ragale is given in Figure 1.12.

ಕುಳಿವ |ಪೊಗೋ|ಳಂಗ|ಳಲ್ಲಿ|
ತಳಿರ |ಕಾವ|ಣಂಗ|ಳಲ್ಲಿ|
ತುಂಬಿ |ವಿಂಡಿ|ನಂತೆ |ಪಾಡಿ|
ಜಕ್ಕ|ವಕ್ಕಿ |ಯಂತೆ |ಕೂಡಿ|

Fig. 1.13: Example for Utsaha Ragale

1.5 Problem Statement

To build a computational linguistic tool for metric analysis and identification of metrics (**ಛಂದಸ್ಸು**) for given Kannada verses using a rule-based approach.

1.6 Aim of the project

The aim of the project is to,

1. Create a Unicode Kannada dataset to test the system.
2. Identification, generation of rules and classification for Chandassu.
3. Building a learning tool that could help students, researchers, and other language enthusiasts.
4. Building a self-tutor exercise with user-interface that would help and check the knowledge of the learner.

1.7 Motivation

- Kannada is an ancient language with a history dating to the 5th century. The old Kannada poems are made with the same combinatorics.
- The verses when written using the methods brings rhythm to the poem
- The computer implementation is done only for Sanskrit and not for any other natural languages
- This as a learning tool helps and assists teachers and students to learn metric analysis

1.8 Objectives

- Creation of testing dataset (poems in Unicode standard).
- Identification and generation of rules for metric (Laghu-ಲಘು and Guru-ಗುರು).
- Classification of metrics.
- Identification of Chandassu (ಛಂದಸ್ಸು) using a rule-based approach.
- Build a learning tool for students, researchers, and other language enthusiasts

1.9 Scope

The scope of Metric Analyzer System is to provide details about normalizer, text splitter, determination of Laghu and Guru, grouping and identification of Ganas, determination of Chandassu and identification of poets. Using this system, the user can learn and teach about Chandassu and can make the system identify and classify Chandassu.

Chapter 2

Literature Survey

2.1 Previous works

Rama N and Meenakshi Lakshmanan[1] gave an insight into the computational logic of the Chandassu for Sanskrit language. This system uses a rule-based approach for the classification of Sanskrit Chandassu. In this paper, just identification of the metric is done, but no scope for identification of Chandassu.

Sekhar Reddy and M.Humera Khanam[6] gave an insight into the Chandassu recognizer for Telugu poems. This system follows a rule-based approach for the classification, identification and recognition of Telugu poems. In this paper the classification is done. The paper further enhances the method of identification of Telugu chandassu.

Literature shows that no attempt has been made on the development of metric analyzer for Kannada verses.

Chapter 3

Requirement Specification

3.1 System Features

This section deals with organizing the functional requirements for the product by system features, the major services provided by the product.

3.1.1 System Feature 1

Normalizer: This module removes all the line spaces, extra spaces, special characters and non-Kannada characters that are not required in identification and classification of metric of given Kannada verse.

Stimulus/Response Sequences: The system removes the special characters and replaces it with no space.

Functional Requirements: The system must take in the raw Unicode data and normalize it with the required format.

Req1: Normalizer

3.1.2 System Feature 2

Text (*Akshara*) splitter: This splits the constituting *Aksharas* or letters without effecting the meaning. This in turn gives input for the classifier.

Stimulus/Response Sequences:

This splits the output poem into *Aksharas* and makes it an array of *Aksharas*.

Functional Requirements:

The system should recognize and split the respective *Aksharas*.

Req 1: Unicode data

Req 2: Text splitter

3.1.3 System Feature 3

Determination of Laghu, Guru and respective Ganas:

Once the *Akshara* is obtained the respective rules are applied for determining *Laghu* and *Guru*.

Stimulus/Response Sequences:

The *Laghu* and *Guru* are applied according to the rules specified. The system then must group it into *Ganas*.

Functional requirements:

This must categorize the data into *Laghu* and *Guru* based on the rules specifies and the letters present. Later, again on the rules it must give out the respective *Ganas*.

Req 1: *Laghu- Guru* determiner

Req 2: *Gana* grouping

3.1.4 System Feature 4

Identification of *Chandassu*:

The *Ganas* follow a particular pattern of occurrence of elements. Thus, on this basis the *Chandassu* is determined.

Stimulus/Response Sequences:

The *Ganas* have a pattern now according to the pattern the *Chandassu* is defined.

Functional requirements:

The *Chandassu* is the last part of the program which determines the pattern present in a particular *Chandassu* based on the rules specified.

Req 1: System feature 1

Req 2: System feature 2

Req 3: System feature 3

3.2 Functional requirements

Normalizer:

This module removes all the line spaces, extra spaces, special characters and non-Kannada characters that are not required in identification and classification of metric of given Kannada verse.

Text splitter:

This module splits the given verse into Aksharas or letters using the rule-based approach.

Determination of Laghu, Guru and respective Ganas:

Once the Aksharas are obtained the respective rules are applied for determining Laghu and Guru. On the basis of this determined Laghu and Guru, their respective Ganas are identified and grouped.

Identification of Chandassu:

The Ganas follow a particular pattern of occurrence of elements. Thus, on this basis, the Chandassu is determined.

3.3 Non-Functional requirements

Performance Requirements

1. The product shall be based on the language that the program is done.

2. The CPU clock time shall play a role in case of a larger data set.

3. The performance shall depend upon hardware components of the computer.

3.4 Software Requirements

The software mainly deals with the interaction of Kannada Unicode data with the program designed. The program takes the raw data as input, then classifies, modifies and gives out the result as expected by the user.

3.5 Hardware Requirements

As the system being designed does not have any designated hardware system, this makes it to no direct designated hardware system. The interface and the working of the classes and functions are provided by the underlying software system.

Chapter 4

Design

4.1 High-level design

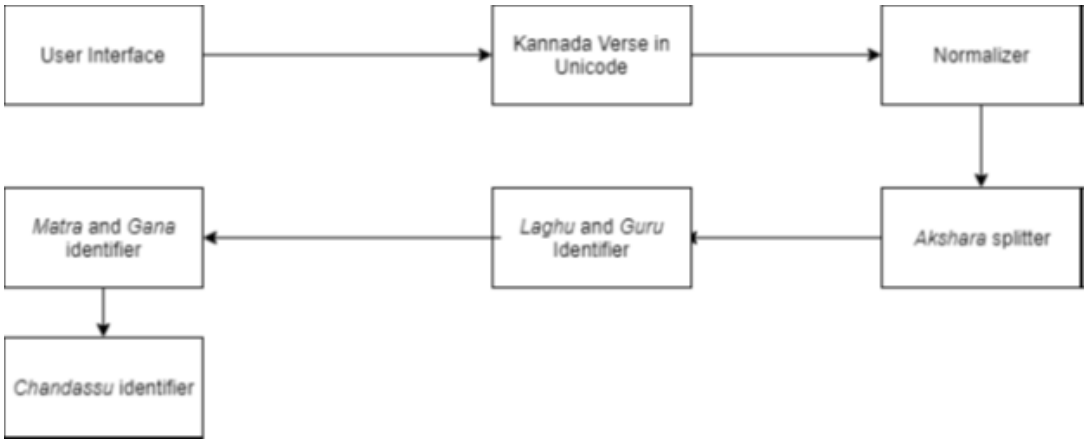

Fig 4.1: Architecture of the proposed Kannada Metric System

The modules of the proposed system are explained below:

User Interface:

The startup page lands at introduction about Laghu, Guru and other metric norms. The page further runs down to types of Chandassu such as Kanda Padya, Shatpadi and Ragale. The other section of exercise helps the learners to experiment with the verses by testing it against the rules of identification and classification of the metric.

Normalizer:

This module removes all the line spaces, extra spaces, special characters and non-Kannada characters that are not required in identification and classification of metric of given Kannada verse.

Text splitter:

This module splits the given verse into Aksharas or letters using the rule-based approach.

Determination of Laghu, Guru and respective Ganas:

Once the Aksharas are obtained the respective rules are applied for determining Laghu and Guru.

On the basis of this determined Laghu and Guru, their respective Ganas are identified and grouped.

Identification of Chandassu:

The Ganas follow a particular pattern of occurrence of elements. Thus, on this basis, the Chandassu is determined.

4.2 Low-level design

Normalizer:

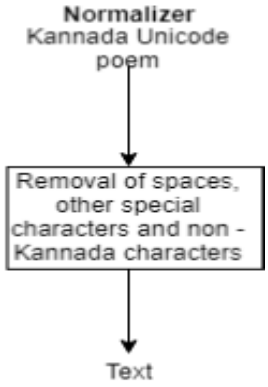

Fig 4.2: Function of a Normalizer

This is the basic system that takes in the Kannada Unicode poem as input then it checks for the spaces and tabs present in the given text. Then it removes all the special characters (such as ?, -,:,; etc.,) and non-Kannada characters and gives out text as normalized data.

Akshara (Letter) splitter:

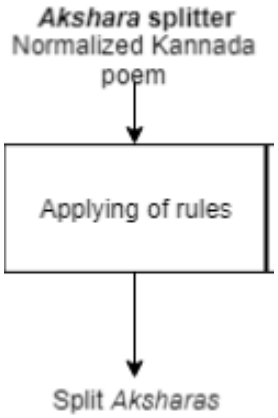

Fig 4.3: Akshara splitter

The normalized data from the normalizer is the input for the Akshara splitter. Here individual Akshara or letter is identified and separated. This process is done by the identification of respective Aksharas by the rules provided.

Laghu and Guru Identifier:

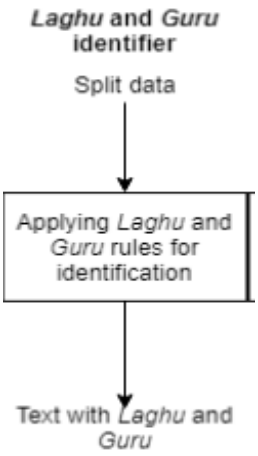

Fig 4.4:Laghu-Guru identifier

On this the Akshara is split, the system determines whether to apply Laghu or Guru based on the rules. This is mainly done on the basis of the syllable.

Matra and Gana identifier:

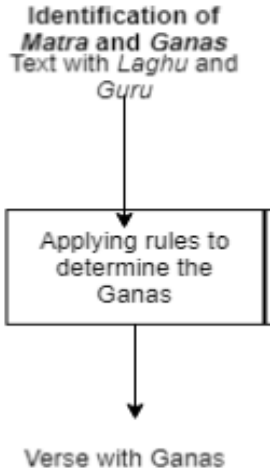

Fig 4.5: Mathra and Gana identifier

The poem with Laghu or Guru is fed to this module which calculates the Matra (is one for Laghu

and two for Guru) and accordingly groups it as Ganas (typically a group of three, four or five based on the calculation of the Matras). This Ganas is made and the verse is made from initial text to a text with Ganas.

Chandassu identifier:

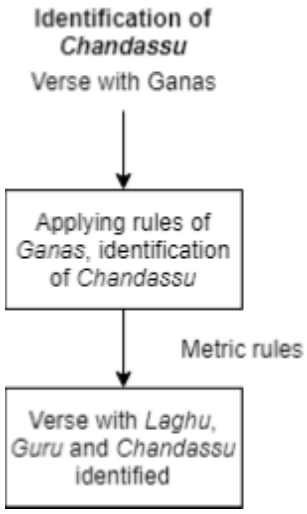

Fig 4.6 : Chandassu identifier

The input poem with Ganas is fed into the module which sees into the pattern of the occurrence of the Ganas and classifies it into the respective Chandassu. The identification of Chandassu is again done based on the rules specified. This, in turn, gives the complete result with Laghu, Guru, Matra, Ganas, and Chandassu identified for a given verse.

Chapter 5

Implementation

5.1 Introduction

The project mainly relies on two entities:

- **The Unicode representation**
- **The rules specified for classification of Chandassu**

Kannada[1]

Unicode.org chart (PDF)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
U+0C8x			ಂ	ಃ		ಅ	ಆ	ಇ	ಈ	ಉ	ಊ	ಋ	ಌ		ಎ	ಏ
U+0C9x	ಐ		ಒ	ಓ	ಔ	ಕ	ಖ	ಗ	ಘ	ಙ	ಚ	ಛ	ಜ	ಝ	ಞ	ಟ
U+0CAx	ಠ	ಡ	ಢ	ಣ	ತ	ಥ	ದ	ಧ	ನ		ಪ	ಫ	ಬ	ಭ	ಮ	ಯ
U+0CBx	ರ	ಱ	ಲ	ಳ		ವ	ಶ	ಷ	ಸ	ಹ			ಽ	ಾ	ಿ	ೀ
U+0CCx	ು	ೂ	ೃ	ೄ		ೆ	ೇ	ೈ		ೊ	ೋ	ೌ	್			
U+0CDx						ೕ	ೖ								ೞ	
U+0CEx	ೠ	ೡ					೦	೧	೨	೩	೪	೫	೬	೭	೮	೯
U+0CFx		ೱ	ೲ													

Notes

1.^ As of Unicode version 6.0

Table 5.1: Kannada Unicode representation table

The input Kannada text in Unicode is taken from Kannada Transliteration tool which is available in website(http://kannada.changathi.com/)[2] data is done using the transliterate tool that helps in converting the English lettered words into Kannada words. The rules specified requires a Python code that takes the Unicode data converts it into subsequent modules. The modules are,

1. Normalizer
2. Text splitter
3. Determination of Laghu, Guru and respective Ganas

4. Identification of Chandassu of input poems

5.2 Modular specification

5.2.1 Normalizer

This module removes all the line spaces, extra spaces, special characters and non-Kannada characters that are not required in identification and classification of metric of given Kannada verse.

Algorithm:

Step1: Read input poem line by line

Step2: Remove all extra spaces, special characters and all other characters other than Kannada Unicode characters.

Step3: Split input line into a list of letters or Akshara**s**

Example:

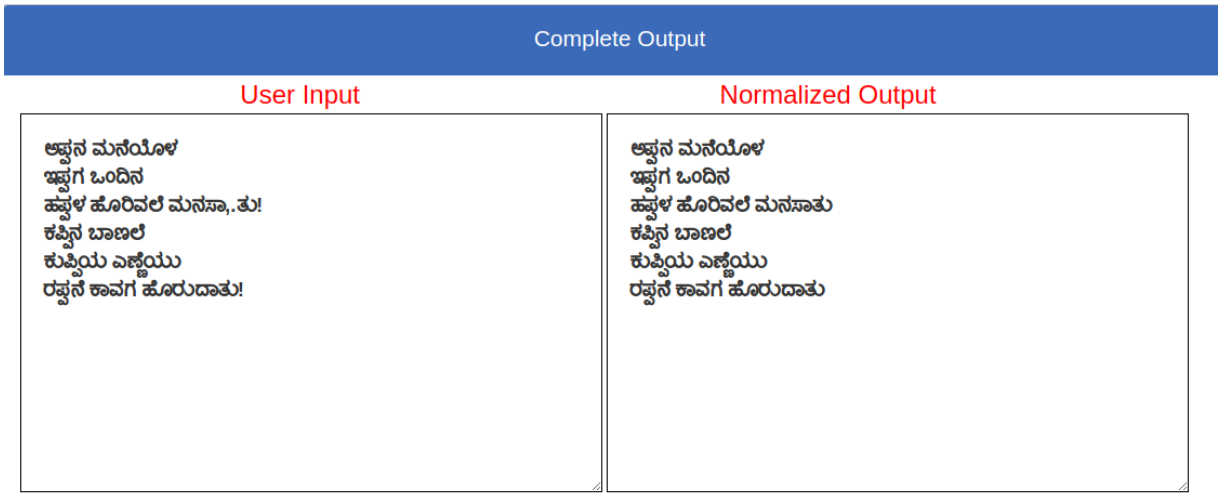

Fig 5.1: Result obtained by normalizer module.

5.2.2 Text (Akshara) splitter

This module splits the given verse into Aksharas or letters using the rule-based approach.

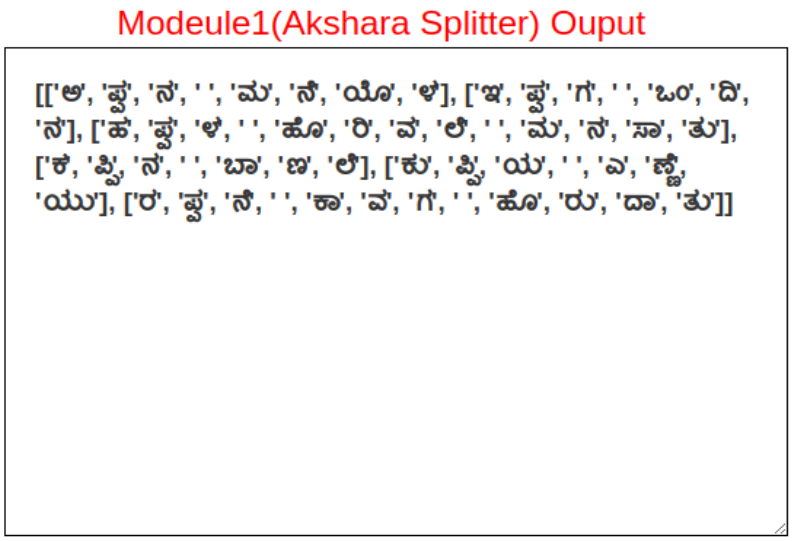

Fig 5.2: Output obtained by Akshara-splitter module

5.2.3 Determination of Laghu, Guru and respective Ganas

Algorithm:

Step1: Read single letter or Akshara at a time from the normalized output.

Step2: If the letter is a single syllable then mark it as LAGHU (U) else mark it as GURU (_).

Step3: If the letter has Halant character and there is a letter next to that then mark it as GURU and give 2 as its weightage and mark the letter beside it as LAGHU and give 1 as its weightage.

Step4: If the letter containing the next letter as Halant then mark as 2 combining both letters.

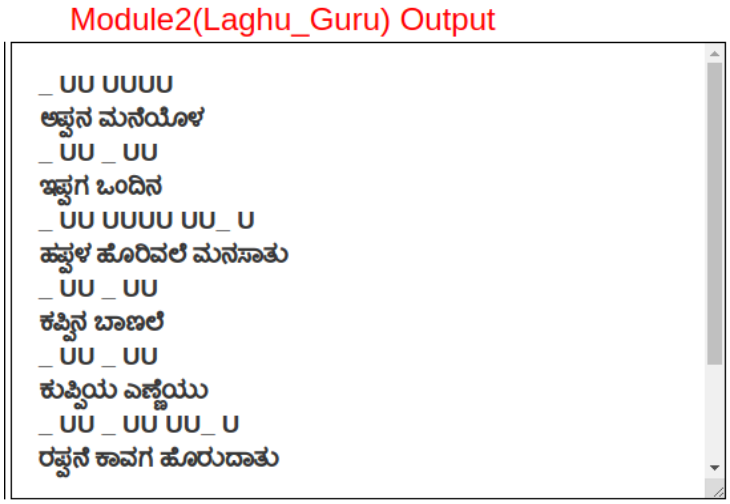

Fig 5.3: Output obtained by Laghu-Guru module

Once the Aksharas are obtained the respective rules are applied for determining Laghu and Guru. On the basis of this determined Laghu and Guru, their respective Ganas are identified and grouped.

5.2.4 Identification of Chandassu of input poems

The Ganas follow a particular pattern of occurrence of elements. Thus, on this basis, the Chandassu is determined.

Algorithm:

Step1: If the number of lines is 2 or 4 and sum of each line (weightage) is 12,20,12,20 and pattern ‘U_U’ should not come in the odd places(1,3,5,7,9,11,15) and pattern 'UUUU’ or ‘U_U’ should come in the 6th and 14th places of Gana similarly '_ _’ or 'UU_’ should come in the 8th and 16th places of Gana then it is classified as Kanda Padya.

Else

Step 2: If the number of lines is 3 or 6 and

• 1,2,4,5 lines contain 2 Gana of 4 syllables each and 3,6th line contains 3 Gana of 4 syllables and one Guru at last and there is no pattern of 'U_U’ then it is classified as Shara Shatpadi.

• 1,2,4,5 lines contain 2 Gana of 5 syllables each and 3,6th line contains 3 Gana of 5 syllables and one Guru at last and there is no pattern of 'U_UU’,'U_ _’ then it is classified as Kusuma Shatpadi

• 1,2,4,5 lines contain 4 Gana of 3 syllables each and 3,6th line contains 6 Gana of 3 syllables and one Guru at last and there is no pattern of 'U_’ then it is classified as Bhoga Shatpadi

• 1,2,4,5 lines contain 4 Gana of 3,4,3,4 and 3,6th line contains 3,4,3,4,3,4 and one guru at last and there is no pattern of 'U_U’ then it is classified as Bhamini Shatpadi

• 1,2,4,5 lines contain 4 Gana of 4 syllables each and 3,6th line contains 6 Gana of 4 and one Guru at last and there is no pattern of 'U_U’ then it is classified as Parivardhini Shatpadi

• 1,2,4,5 lines contain 4 Gana of 5 syllables each and 3,6th line contains 6 Gana of 5 and one Guru at last then it is classified as Vardhaka Shatpadi

Else

Step 3: if the number of lines is 4 and

• Each line contains 4 Gana of 3 syllables each or 3 Gana of 3 syllables each and one Guru then it is classified as Utsaha Ragale

• Each line contains 4 Gana of 4 syllables each or 4 Gana of 3,5,3,5 then it is classified as Mandanila Ragale

• Each line contains 4 Gana of 5 syllables each as Lalitha Ragale.

• Else Sarala Ragale.

Else

Invalid Chandassu

The Ganas follow a particular pattern of occurrence of elements. Thus, on this basis, the Chandassu is determined.

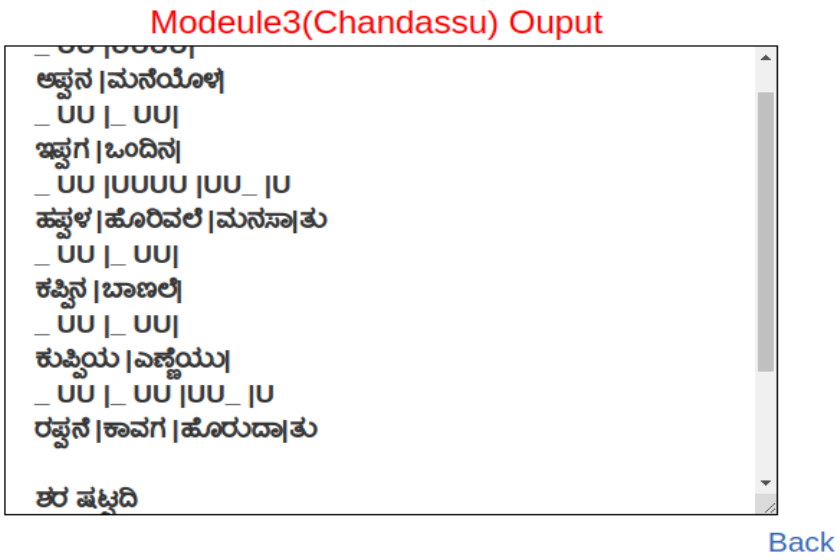

Fig 5.4: Snapshot of output of Shara Shatpadi after classification.

Chapter 6

Experimental Result and Analysis

A learning tool has been developed in order to help students, researchers, linguists, and other language enthusiasts. The user interface is designed using html,css, bootstrap, and javascript in such a way that it makes the user understand Chandassu to a greater extent with examples. The user interface has three main modules. They are,

1. The introduction page
2. Testing tool
3. Exercise tool

6.1 Introduction Page

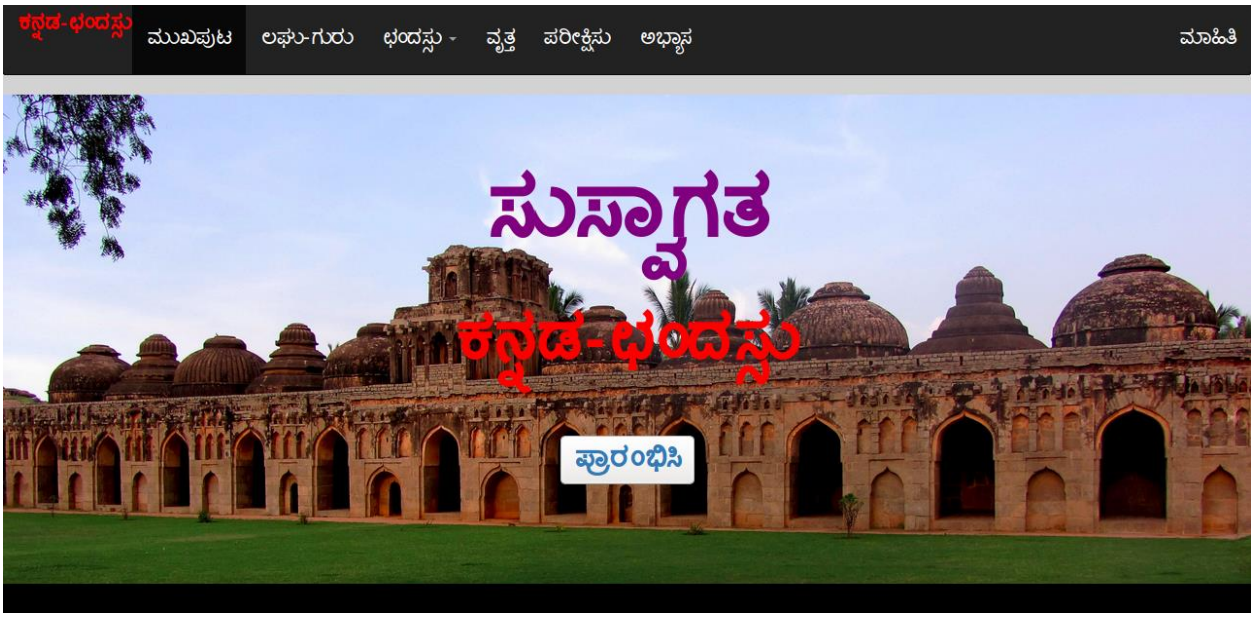

Fig 6.1: UI of our proposed project

The startup page lands at introduction about *Laghu*, *Guru* and other metric norms. The page further runs down to types of *Chandassu* such as *Kanda Padya*, *Shatpadi* and *Ragale*. The other section of exercise helps the learners to experiment with the verses by testing it against the rules of identification and classification of metric.

6.2 Testing

In testing tool, user has to give a input poem/verse and need to follow few instructions as follows:

a) Given input poem/verse can contain any special characters, line spaces, extra spaces as well.

b) User can copy the poem/verse from any website or from the system itself using the link given below.

c) For English to Kannada transliteration use the given link (http://kannada.changathi.com)[2]

d) For extracting from online source (http://padyapaana.com/)[4]

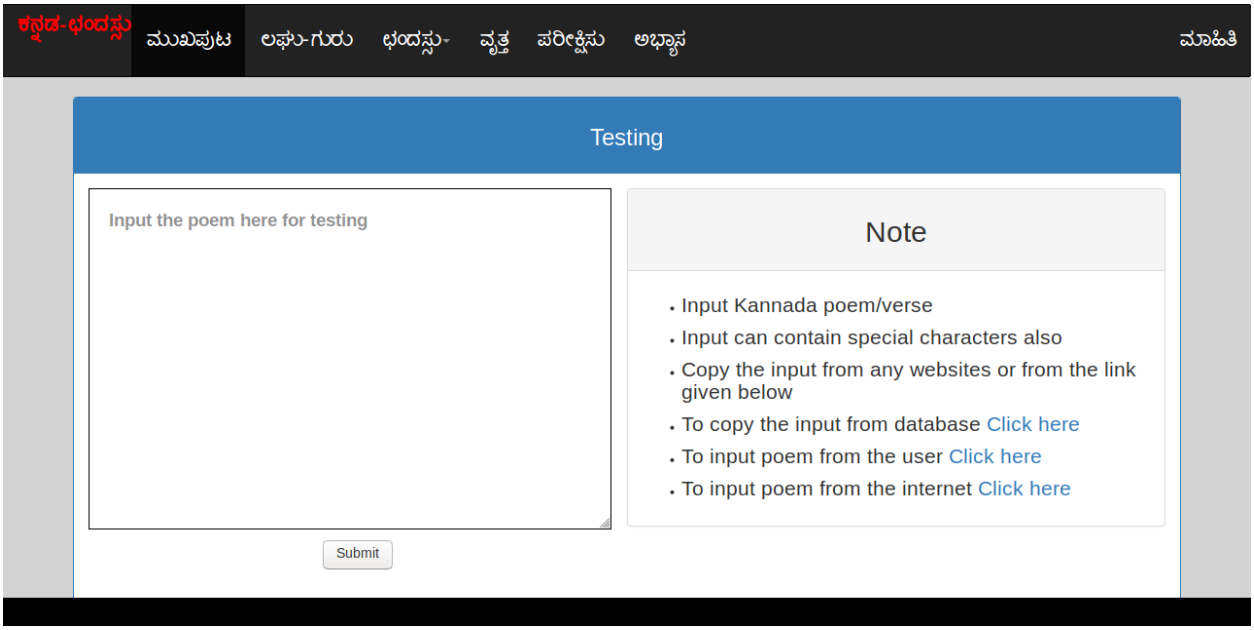

Fig 6.2: The user console to test input verse

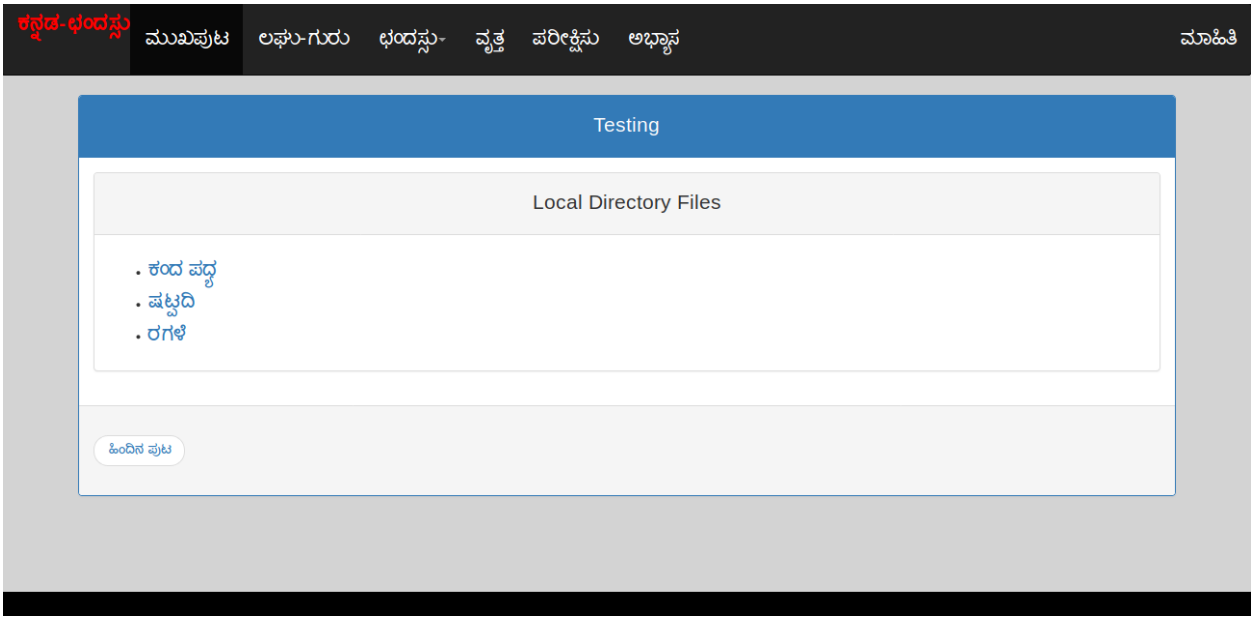

Fig 6.3: The local directories to get the input verse

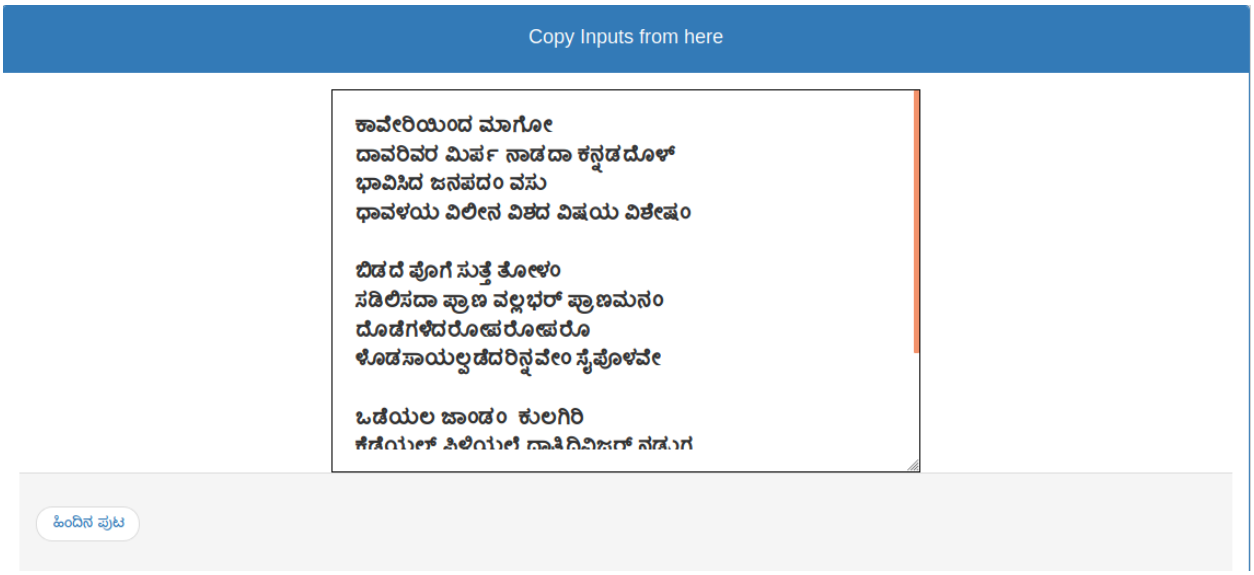

Fig 6.4: Input from dataset

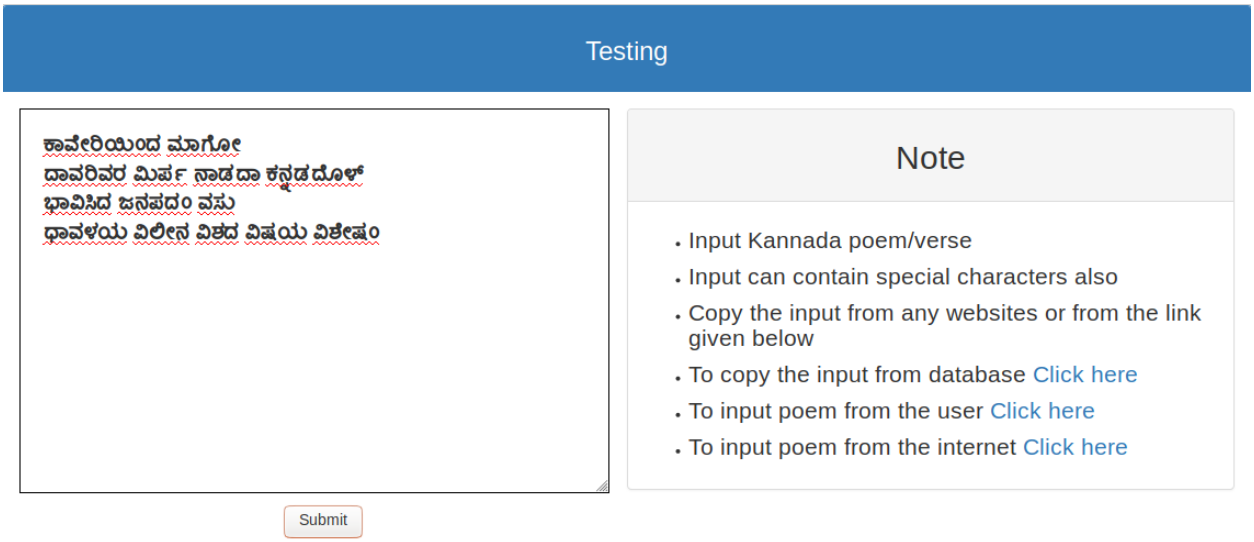

Fig 6.5: Input verse to the user

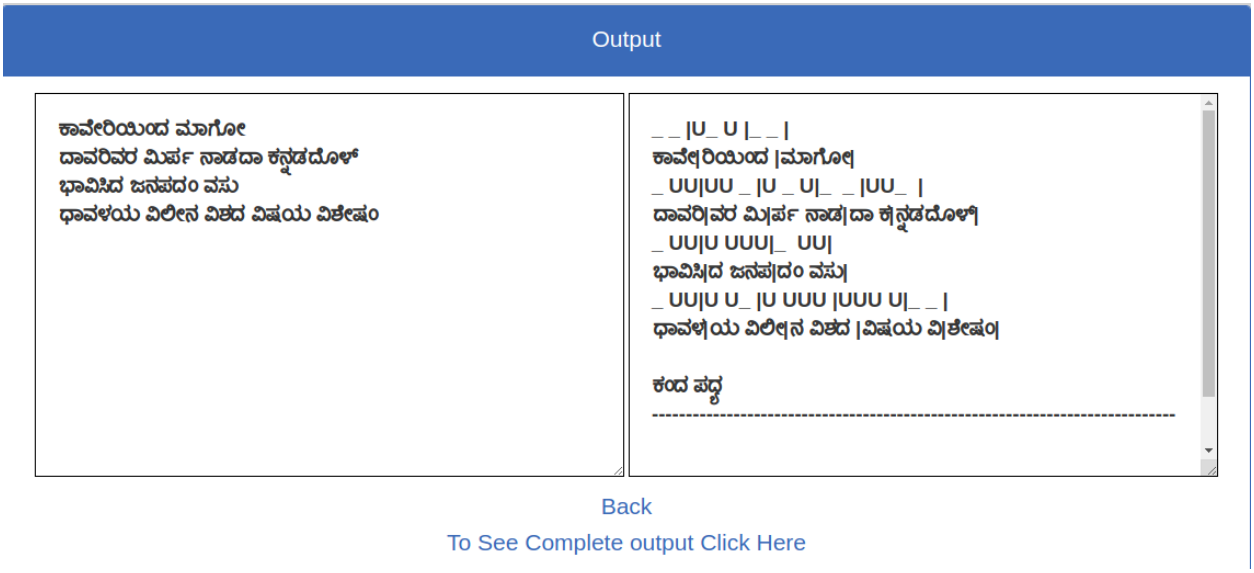

Fig 6.6: The output for the given input verse

6.3. Exercise

In the exercise tool, a random poem/verse is appeared in the user output box where a user needs to insert respective laghu-guru to each aksharas and predict the correct chandassu type to which chandassu the poem belongs to.

Exercise tool is first appeared as shown where the user inserts laghu-guru and select chandassu-

type as follows:

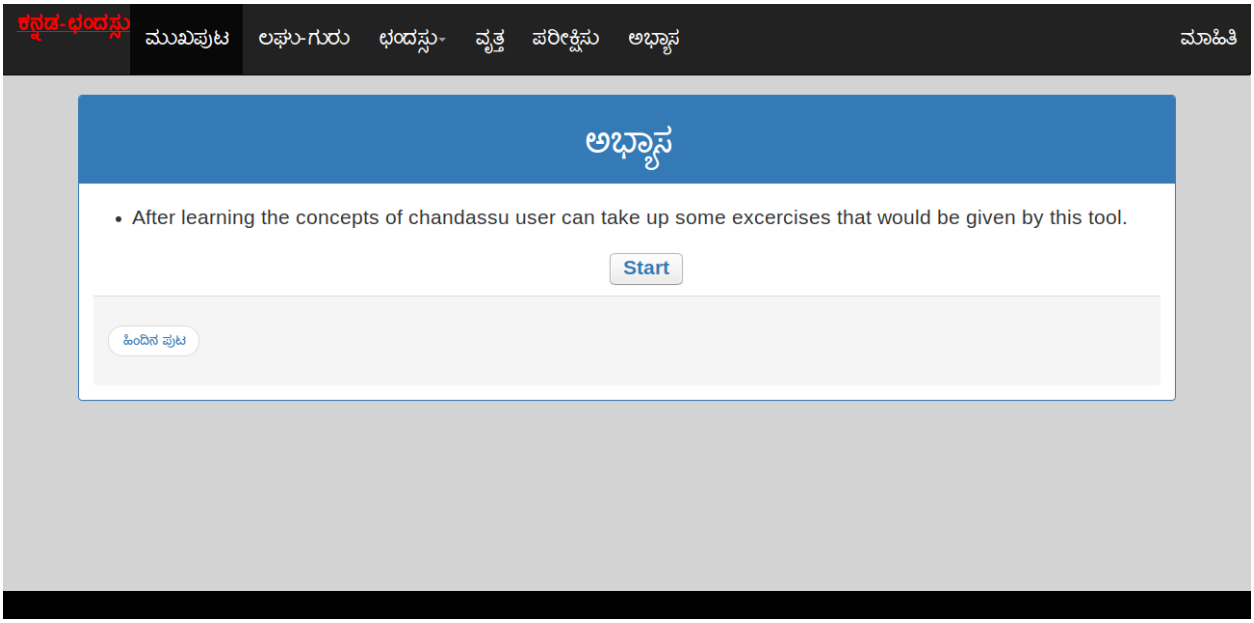

Fig 6.7 The user console showing the exercise tool

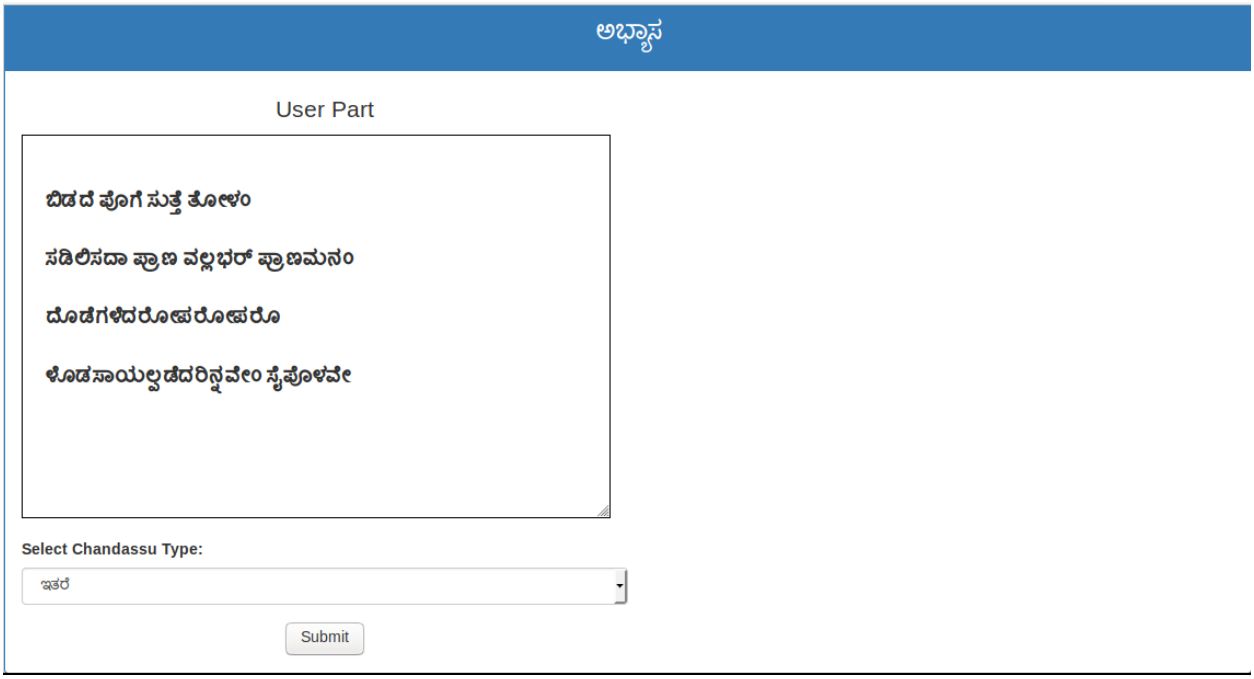

Fig 6.8 Exercise page for the user

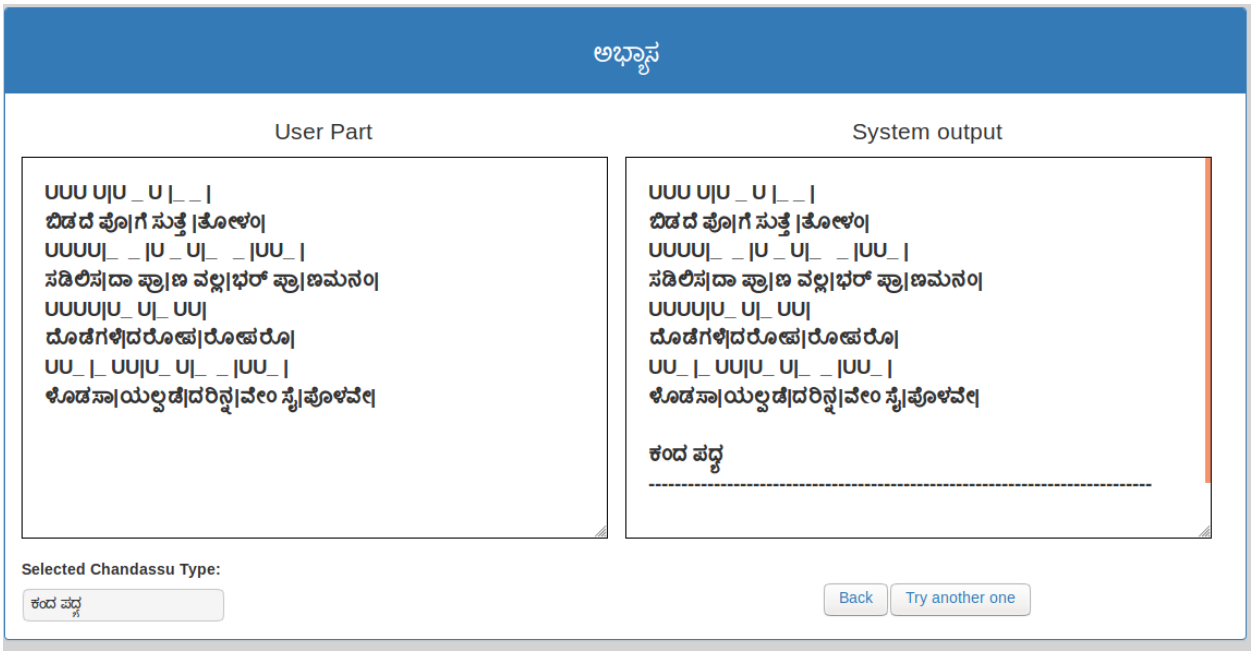

Fig 6.9 User input and system output

System output is shown when submit is pressed and the system output shows the correct application of laghu-guru for the given poem/verse and if chandassu-type selected by the user is incorrect then it shows invalid type.

Further, the user can try with the other poems similarly or go back to the previous page.

Chapter 7

Conclusion and future enhancements

7.1 Conclusion

Metric Analyzer project is a complete implementation of identification of Chandassu (ಛಂದಸ್ಸು) for a given Kannada verse using rule-based approach. The Graphical User Interface acts as an effective teaching as well as learning tool containing introduction to all the types of Chandassu like Kanda Padya(ಕಂದಪದ್ಯ), Shatpadi(ಷಟ್ಪದಿ), Ragale(ರಗಳೆ) and further having its own testing and exercise modules to test the system accuracy and user knowledge in Kannada prosody respectively. The testing module gets input poems from three different sources like database provided by the system, online transliteration tool called Kannada Changathi(http://kannada.changathi.com/)[2], online poems (http://padyapaana.com/)[4] or the user can also give his own poem from his personal directory. The system tests the input poem based on the rules specified and gives the desired output.

The tool also includes an exercise part that provides a Kannada verse randomly to the user to insert Laghu and Guru to each Akshara and predict the correct Chandassu type for the poem thereby checking the user about his knowledge in Kannada prosody. This project containing introduction about the topic and also having integrated with a testing and exercise tool makes the system a good learning tool for students, professors, researchers, linguists, and other language enthusiasts to learn more interactively and user-centric.This tool thus gives scope for creating newer types of metrics for new Kannada verses. This tool can be integrated with Kannada text-to-speech conversion module and help blind people in learning metric analysis.

7.2 Future enhancements

This project can even be enhanced by the identification of poets for the given Kannada verse. The dataset obtained by this project can also be used and enhanced to make machine learning approaches in future enhancements.

Annexure A

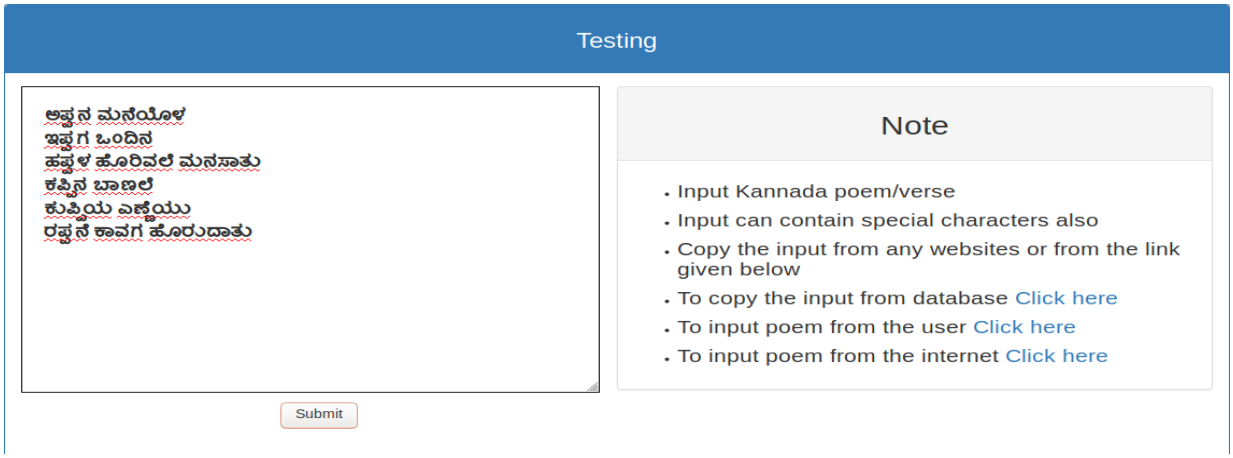

Fig 1: Testing input verse for Shara Shatpadhi.

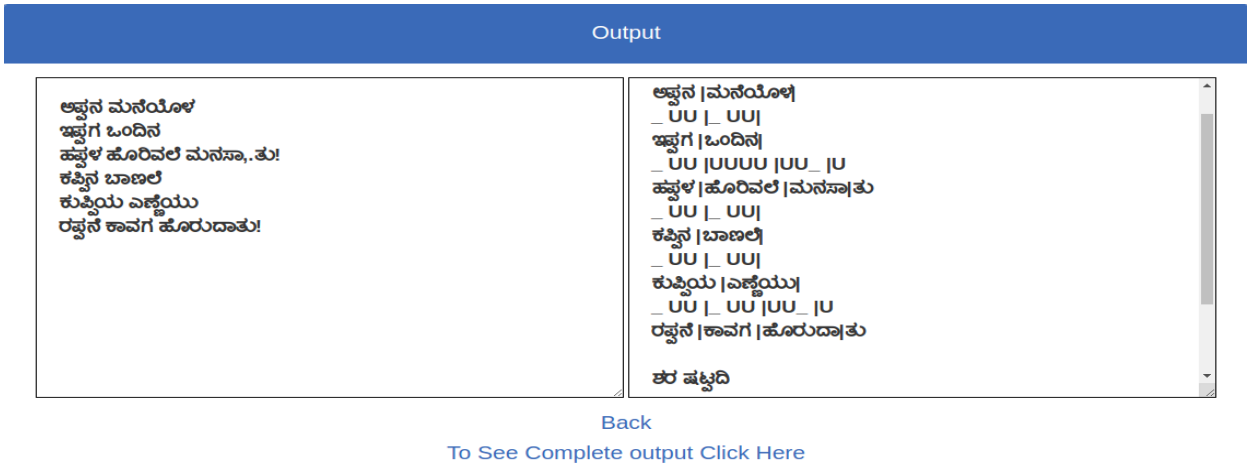

Fig 2: Output for Shara Shatpadhi verse.

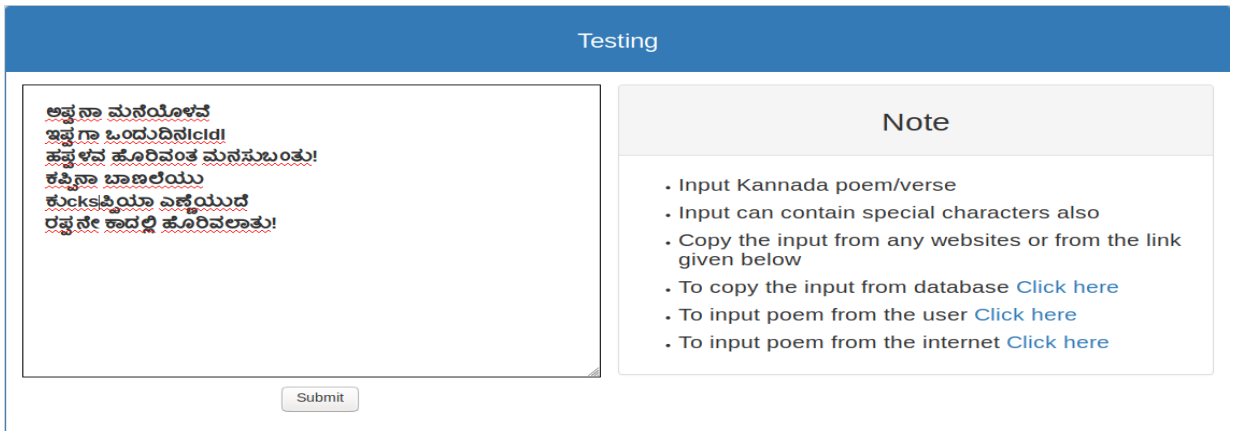

Fig 3: Testing input verse for Kusuma Shatpadhi.

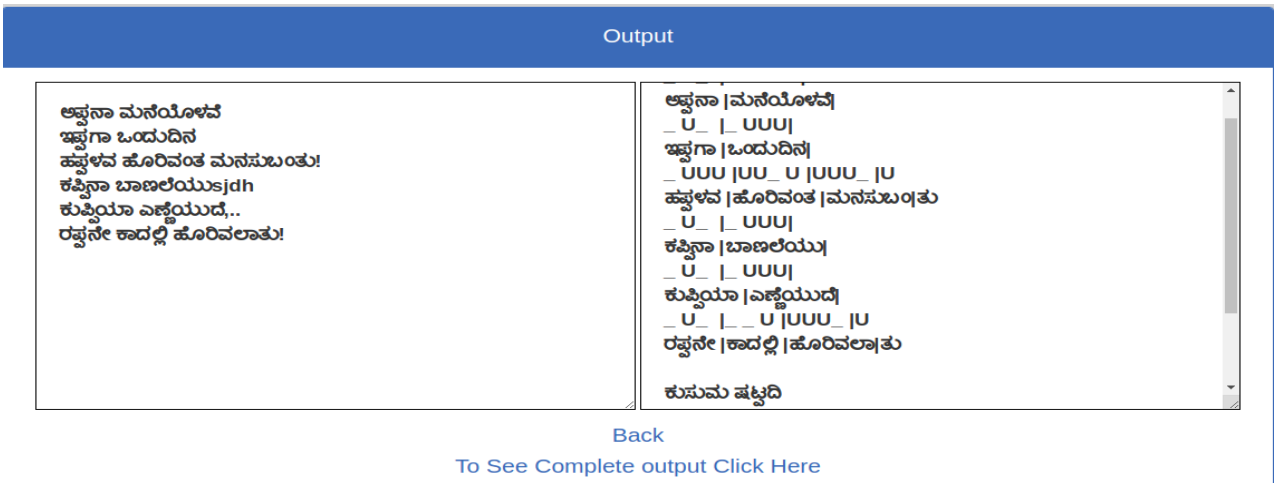

Fig 4: Output for Kusuma Shatpadhi verse.

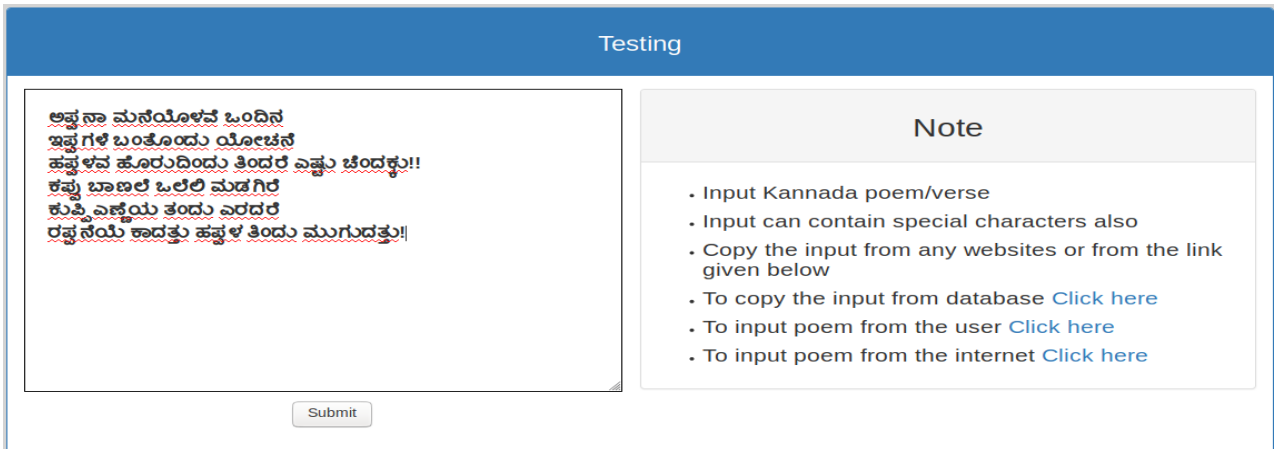

Fig 5: Testing input verse for Bhamini Shatpadhi.

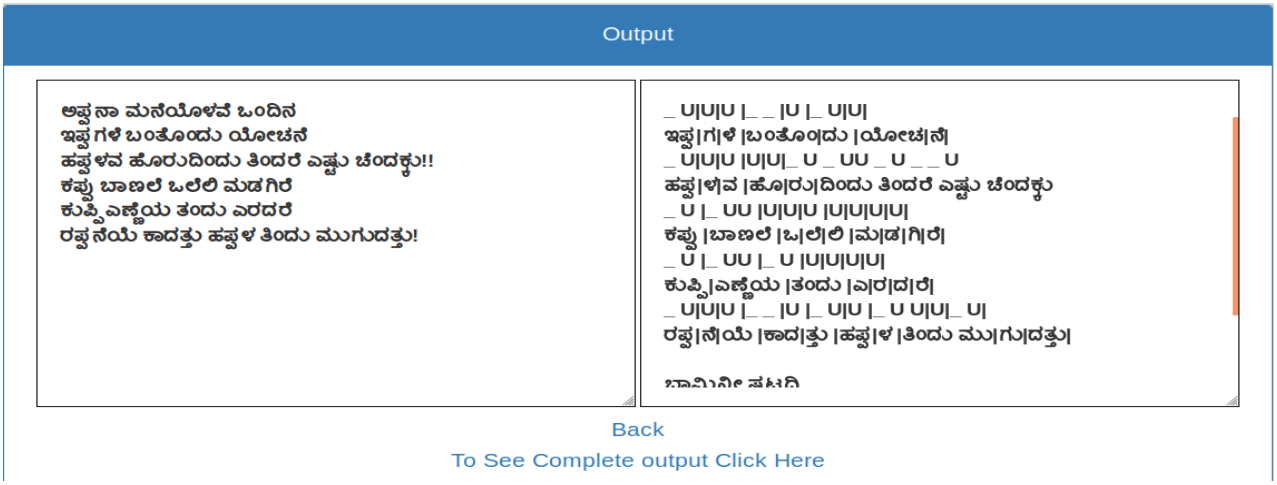

Fig 6: Output for Bhamini Shatpadhi verse.

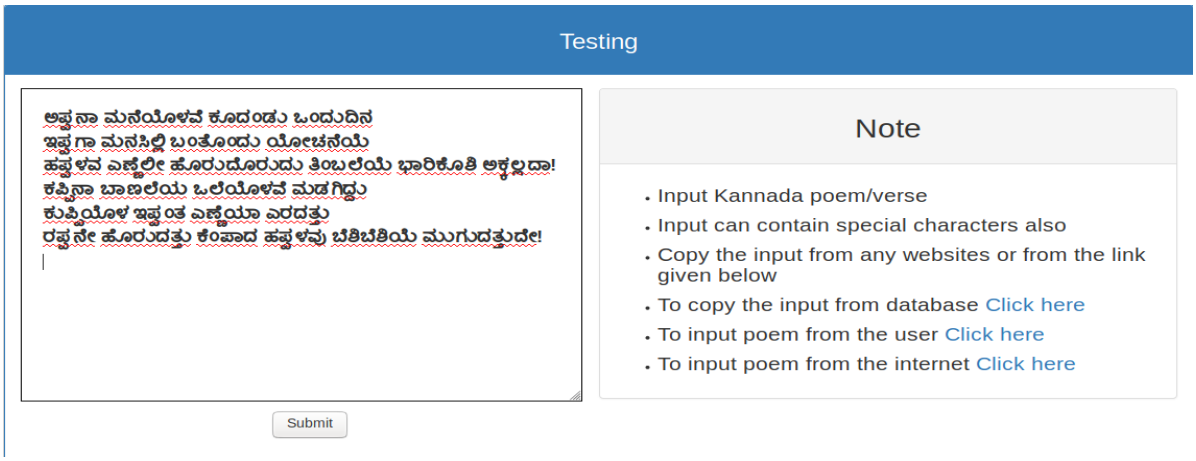

Fig 7: Testing input verse for Vardhika Shatpadi.

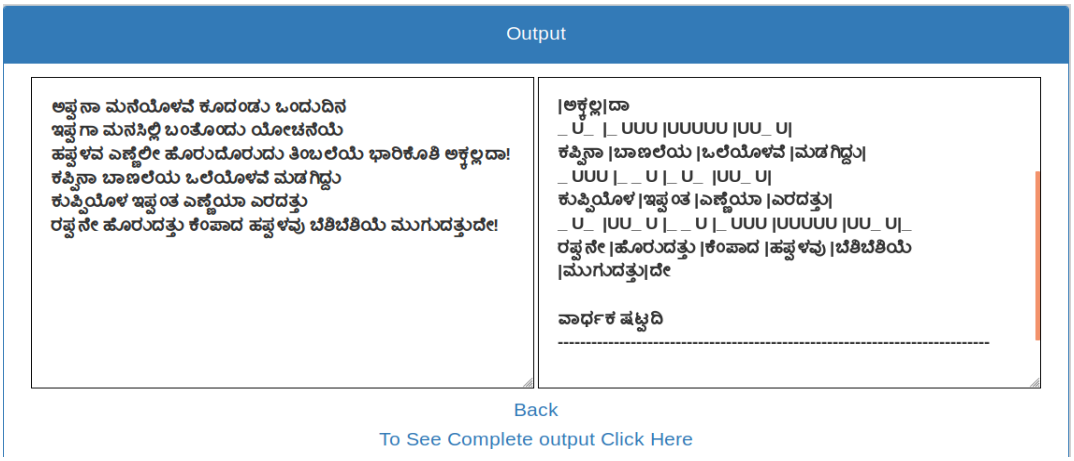

Fig 8: Output for Vardhika Shatpadhi verse.

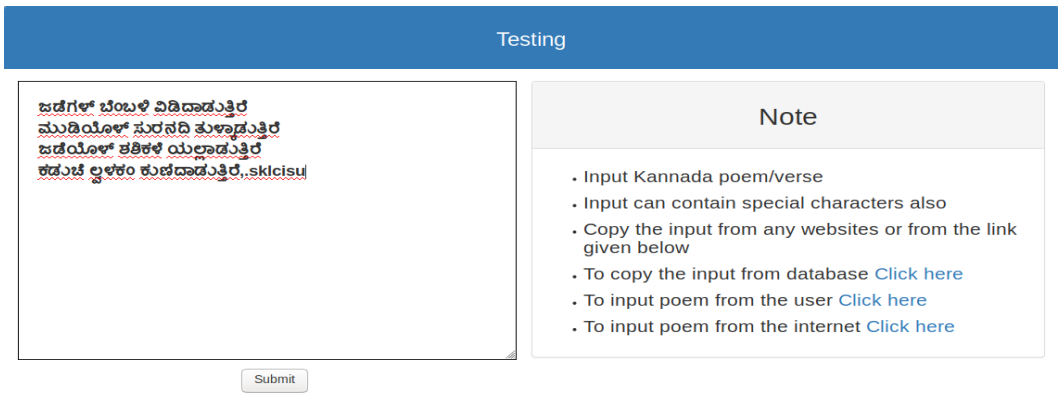

Fig 9: Testing input verse for Mandanila Ragale.

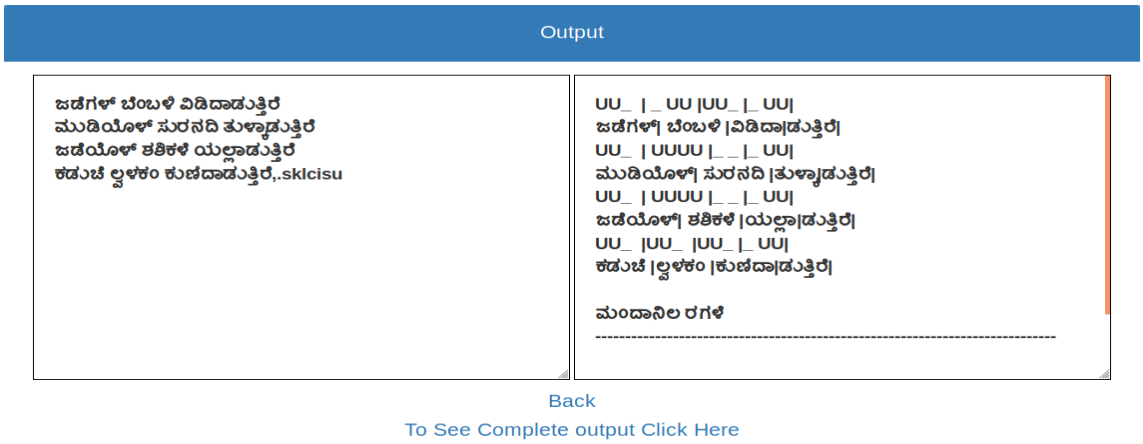

Fig 10: Output for Mandanila Ragale verse.

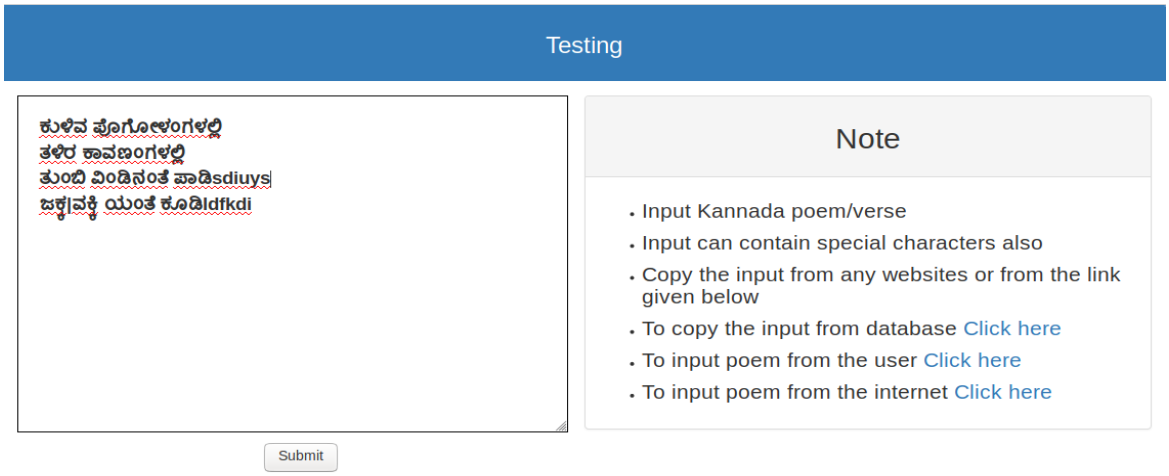

Fig 11: Testing input verse for Utsaha Ragale.

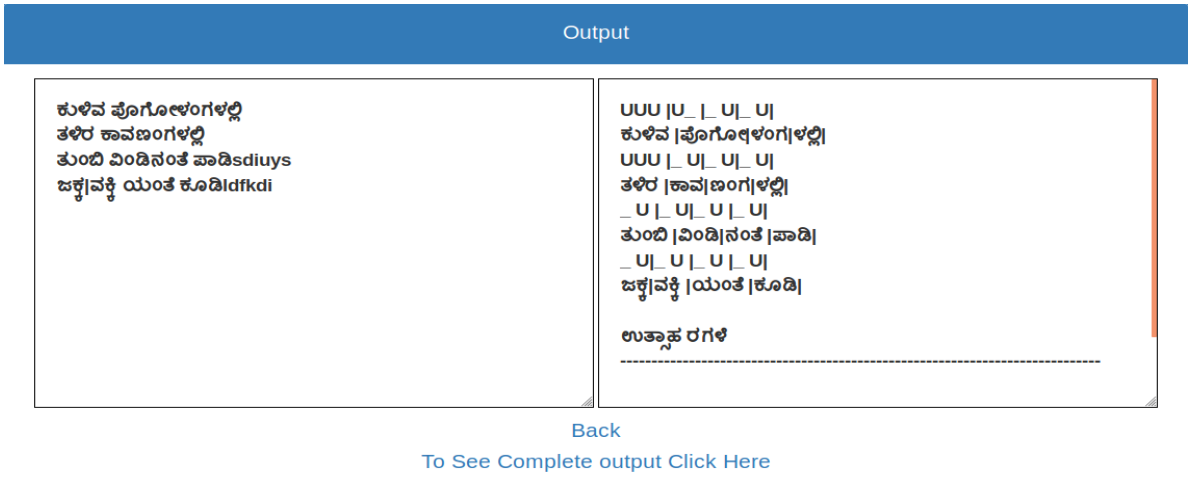

Fig 14: Output for Utsaha Ragale verse.

References

[1] Rama N and Meenakshi Lakshmanan," A Computational Algorithm for Metrical Classification of Verse", IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 2, No 1, March 2010, pp 46-53

[2] http://kannada.changathi.com/

[3] https://kannadadeevige.blogspot.com/2013/11/blog-post_4047.html

[4] http://padyapaana.com/

[5] http://www.nammakannadanaadu.com/vyakarana/chandassu.php

[6] Reddy Sekhar Reddy M.Humera Khanam," Chandhassu Recognizer for Telugu Poems",(IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 7 (3), 2016, 1425-1434