A COMPUTATIONAL LINGUISTIC APPROACH FOR METADATA GENERATION FOR HINDI POETRY

A Thesis submitted to Gujarat Technological University

for the Award of

Doctor of Philosophy

in

Computer Science

by

MILIND KUMAR AUDICHYA

159997431001

under supervision of

PROF. DR. JATINDERKUMAR RAMDASS SAINI

Professor and Director, Symbiosis Institute of Computer Studies and Research,
Symbiosis International (Deemed University), Pune, India

GUJARAT TECHNOLOGICAL UNIVERSITY
AHMEDABAD

September - 2021

A COMPUTATIONAL LINGUISTIC APPROACH FOR METADATA GENERATION FOR HINDI POETRY

A Thesis submitted to Gujarat Technological University

for the Award of

Doctor of Philosophy

in

Computer Science

by

MILIND KUMAR AUDICHYA

159997431001

under supervision of

PROF. DR. JATINDERKUMAR RAMDASS SAINI

Professor and Director, Symbiosis Institute of Computer Studies and Research,
Symbiosis International (Deemed University), Pune, India

GUJARAT TECHNOLOGICAL UNIVERSITY
AHMEDABAD

September - 2021

DECLARATION

I declare that the thesis entitled **A Computational Linguistic Approach for Metadata Generation for Hindi Poetry** carried out by me during the period from **October 2016** to **June 2021** under the supervision of **Dr. Jatinderkumar Ramdass Saini** and this has not formed the basis for the award of any degree, diploma, associateship, fellowship, titles in this or any other University or other institution of higher learning.

I further declare that the material obtained from other sources has been duly acknowledged in the thesis. I shall be solely responsible for any plagiarism or other irregularities, if noticed in the thesis.

Signature of the Research Scholar: …… Date: **16/09/2021**

Name of Research Scholar: **Milind Kumar Audichya**

Place: **Surat**

CERTIFICATE

I certify that the work incorporated in the thesis **A Computational Linguistic Approach for Metadata Generation for Hindi Poetry** submitted by Shri **Milind Kumar Audichya** was carried out by the candidate under my supervision/guidance. To the best of my knowledge:

(i) the candidate has not submitted the same research work to any other institution for any degree/diploma, Associateship, Fellowship or other similar titles

(ii) the thesis submitted is a record of original research work done by the Research Scholar during the period of study under my supervision, and

(iii) the thesis represents independent research work on the part of the Research Scholar.

Signature of Supervisor: …… Date: **16/09/2021**

Name of Supervisor: **Dr. Jatinderkumar Ramdass Saini**

Place: **Pune**

Course-work Completion Certificate

This is to certify that Mr./Mrs./Ms. **MILIND KUMAR AUDICHYA** enrolment no. **159997431001** is a PhD scholar enrolled for PhD program in the branch **COMPUTER SCIENCE** of Gujarat Technological University, Ahmedabad

(Please tick the relevant option(s))

☐ He/She has been exempted from the course-work (successfully completed during M.Phil Course)

☐ He/She has been exempted from Research Methodology Course only (successfully completed during M.Phil Course)

☑ He/She has successfully completed the PhD course work for the partial requirement for the award of PhD Degree. His/ Her performance in the course work is as follows-

Grade Obtained in Research Methodology (PH001)	Grade Obtained in Self Study Course (Core Subject) (PH002)
BB	**AB**

Signature of Supervisor

Dr. Jatinderkumar Ramdass Saini

Originality Report Certificate

It is certified that PhD Thesis titled **A COMPUTATIONAL LINGUISTIC APPROACH FOR METADATA GENERATION FOR HINDI POETRY** by **MILIND KUMAR AUDICHYA** has been examined by us. We undertake the following:

a. Thesis has significant new work / knowledge as compared already published or are under consideration to be published elsewhere. No sentence, equation, diagram, table, paragraph or section has been copied verbatim from previous work unless it is placed under quotation marks and duly referenced.

b. The work presented is original and own work of the author (i.e. there is no plagiarism). No ideas, processes, results or words of others have been presented as Author own work.

c. There is no fabrication of data or results which have been compiled / analysed.

d. There is no falsification by manipulating research materials, equipment or processes, or changing or omitting data or results such that the research is not accurately represented in the research record.

e. The thesis has been checked using **Subscribed Version of Turnitin** (copy of originality report attached) and found within limits as per GTU Plagiarism Policy and instructions issued from time to time (i.e. permitted similarity index <10%).

Signature of the Research Scholar: …… Date: **16/09/2021**

Name of Research Scholar: **Milind Kumar Audichya**

Place: **Surat**

Signature of Supervisor: …… Date: **16/09/2021**

Name of Supervisor: **Dr. Jatinderkumar Ramdass Saini**

Place: **Pune**

Copy Originality Report

Copy of Originality Report from **Subscribed Version of Turnitin**:

Thesis

by Milind Audichya

Submission date: 02-Jun-2021 09:49AM (UTC+0530)
Submission ID: 1598810997
File name: CoreTheis.pdf (1.28M)
Word count: 21854
Character count: 106989

Thesis

ORIGINALITY REPORT

5%	2%	4%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Milind Kumar Audichya, Jatinderkumar R.. "Stanza Type Identification using Systematization of Versification System of Hindi Poetry", International Journal of Advanced Computer Science and Applications, 2021 Publication	1%
2	thesai.org Internet Source	1%
3	Milind Kumar Audichya, Jatinderkumar R. Saini. "Computational linguistic prosody rule-based unified technique for automatic metadata generation for Hindi poetry", 2019 1st International Conference on Advances in Information Technology (ICAIT), 2019 Publication	1%
4	docplayer.net Internet Source	<1%
5	Submitted to University of Sydney Student Paper	<1%

6	Submitted to Symbiosis International University Student Paper	<1 %
7	Submitted to University of Florida Student Paper	<1 %
8	Submitted to University of Johannsburg Student Paper	<1 %
9	eprints.lancs.ac.uk Internet Source	<1 %
10	bradscholars.brad.ac.uk Internet Source	<1 %
11	Submitted to University of Stellenbosch, South Africa Student Paper	<1 %
12	Submitted to Institute of Technology Blanchardstown Student Paper	<1 %
13	link.springer.com Internet Source	<1 %
14	ngct2017.ngct.org Internet Source	<1 %
15	Submitted to Fountainhead School Student Paper	<1 %
16	Milind Kumar Audichya, Jatinderkumar R.. "Towards Natural Language Processing with	<1 %

	Figures of Speech in Hindi Poetry", International Journal of Advanced Computer Science and Applications, 2021 Publication	
17	mdpi.com Internet Source	<1 %
18	www.omicsonline.org Internet Source	<1 %
19	Submitted to University of Sheffield Student Paper	<1 %
20	citeseerx.ist.psu.edu Internet Source	<1 %
21	ns1.shudo-u.ac.jp Internet Source	<1 %
22	Navin Sabharwal, Amit Agrawal. "Chapter 1 Introduction to Natural Language Processing", Springer Science and Business Media LLC, 2021 Publication	<1 %
23	www.indianmediastudies.com Internet Source	<1 %
24	archive.org Internet Source	<1 %
25	"Mining Intelligence and Knowledge Exploration", Springer Science and Business Media LLC, 2017	<1 %

	Publication	
26	www.interactions.com Internet Source	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

PhD THESIS Non-Exclusive License to GUJARAT TECHNOLOGICAL UNIVERSITY

In consideration of being a PhD Research Scholar at GTU and in the interests of the facilitation of research at GTU and elsewhere, I, **Milind Kumar Audichya** having (Enrollment No.) **159997431001** hereby grant a non-exclusive, royalty free and perpetual license to GTU on the following terms:

a) GTU is permitted to archive, reproduce and distribute my thesis, in whole or in part, and/or my abstract, in whole or in part (referred to collectively as the "Work") anywhere in the world, for non-commercial purposes, in all forms of media;

b) GTU is permitted to authorize, sub-lease, sub-contract or procure any of the acts mentioned in paragraph (a);

c) GTU is authorized to submit the Work at any National / International Library, under the authority of their "Thesis Non-Exclusive License";

d) The Universal Copyright Notice (©) shall appear on all copies made under the authority of this license;

e) I undertake to submit my thesis, through my University, to any Library and Archives.

Any abstract submitted with the thesis will be considered to form part of the thesis.

f) I represent that my thesis is my original work, does not infringe any rights of others, including privacy rights, and that I have the right to make the grant conferred by this non-exclusive license.

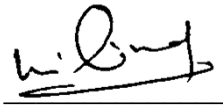
g) If third party copyrighted material was included in my thesis for which, under

the terms of the Copyright Act, written permission from the copyright owners is required, I have obtained such permission from the copyright owners to do the acts mentioned in paragraph (a) above for the full term of copyright protection.

h) I retain copyright ownership and moral rights in my thesis, and may deal with the copyright in my thesis, in any way consistent with rights granted by me to my University in this non-exclusive license.

i) I further promise to inform any person to whom I may hereafter assign or license my copyright in my thesis of the rights granted by me to my University in this non- exclusive license.

j) I am aware of and agree to accept the conditions and regulations of PhD including all policy matters related to authorship and plagiarism.

Signature of the Research Scholar: 

Name of Research Scholar: **Milind Kumar Audichya**

Date: **16/09/2021** Place: **Surat**

Signature of the Supervisor: 

Name of Supervisor: **Dr. Jatinderkumar Ramdass Saini**

Date: **16/09/2021** Place: **Pune**

Thesis Approval Form

The viva-voce of the PhD Thesis submitted by Shri. **Milind Kumar Audichya** (Enrollment No.: **159997431001**) entitled **A Computational Linguistic Approach for Metadata Generation for Hindi Poetry** was conducted on **Thursday, 16th September 2021** (day and date) at Gujarat Technological University.

(Please tick any one of the following option)

☑ The performance of the candidate was satisfactory. We recommend that he/she be awarded the PhD degree.

☐ Any further modifications in research work recommended by the panel after 3 months from the date of first viva-voce upon request of the Supervisor or request of Independent Research Scholar after which viva-voce can be re-conducted by the same panel again.

(briefly specify the modifications suggested by the panel)

☐ The performance of the candidate was unsatisfactory. We recommend that he/she should not be awarded the PhD degree.

(The panel must give justifications for rejecting the research work)

Name and Signature of Supervisor with Seal
Dr. Jatinderkumar R. Saini
Professor and Director, Symbiosis Institute of Computer Studies and Research, Pune, India.

---

1) (External Examiner 1) Name and Signature
Dr. Sonal Kanungo Sharma
Associate Professor, DPG Institute of Technology & Management, Gurugram, India.

---

2) (External Examiner 2) Name and Signature
Dr. Lalit Goyal
Associate Professor, DAV College, Jalandar, India.

---

3) (External Examiner 3) Name and Signature
Dr. Tabea De Wille
Lecturer & Director of Localisation Research Centre, Dept. Of Computer Science and Information Systems, Limerick, Ireand.

Abstract

Name: Milind Kumar Audichya

Enrollment: 159997431001

Branch: Computer Science

Title of the Thesis: A Computational Linguistic Approach For Metadata Generation For Hindi Poetry

Abstract

The research work presents the mechanism for the automatic metadata generation for Hindi poetry. The research work is carried out with the mixed approach of a combination of Natural Language Processing and Computational Linguistics as per the demand of the research problem. In this research work, the primary focus is on the Hindi Verses (‘Ch-hand’).

The hidden knowledge behind the writing mechanics of Hindi verses is coming from the long back times of the ‘Vedas’ and ‘Puranas’. Unfortunately, due to the lack of proper management and documentation, this is slowly getting extinct. This research work tried to generate the hierarchical structure for the Hindi verses and put the appropriate Hindi Verses in the relevant classes. After collecting, identifying, and manually validating the rules through the manual calculations of each rule found from the different sources, the structure is created to make it usable for research purposes.

Further, the construction rules of Hindi Verses are rule-based modeled, through which the detection and identification of the Hindi Verses take place. Along with that, stopwords filtering is also incorporated. With the integration of the wordnet, the meanings of the words and example sentences of the respective words are also included in the automatic metadata. Apart from this, examples of the detected Hindi verses are also populated to understand the detected Hindi Verse better. The automatically generated metadata concerning computational linguistics includes details about lines, characters, stanzas, symbolic representation, diacritics, type, subtype, Quantities Count, Sequence of Characters,

and much more metadata. The Hindi poetry corpus is also one of the problems which is required to be handle while working for Hindi poetry-based research works, this research work also contribute for the corpus creation of Hindi poetry.

The research work is not limited to Hindi Verses. It also opens up the new research stream of the Hindi 'Figure of Speech' for which the Hierarchical structure is constructed as similar to the Hindi Verses. This Ph.D. Thesis will help in automatic metadata generation for Hindi Poetry, it will improve existing keyword-based search results delivery with metadata, enhance the management of Hindi Poetic literature, and, most importantly, it will play a vital role in saving the extinct Hindi Verses.

List of Publications:

1. M. K. Audichya and J. R. Saini, "Computational linguistic prosody rule-based unified technique for automatic metadata generation for Hindi poetry," *2019 1st International Conference on Advances in Information Technology (ICAIT)*, Jul. 2019, doi: 10.1109/icait47043.2019.8987239.

2. M. K. Audichya and J. R. Saini, "Stanza Type Identification using Systematization of Versification System of Hindi Poetry," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 1, 2021, doi: 10.14569/ijacsa.2021.0120117.

3. M. K. Audichya and J. R. Saini, "Towards Natural Language Processing with Figures of Speech in Hindi Poetry," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 3, 2021, doi: 10.14569/ijacsa.2021.0120316.

Acknowledgment

Every good work starts with a good thought, the seeds of good thinking in me were sown long ago by my late grandfather and a great soul, Pujari Shree Bhanwarmata Ji, Chhoti Sadri / Mewar - Rajasthan, Shree Ramprasad Narayan Ji Audichya (Former lecturer). His heartfelt desire was that by completing all the dimensions of education, I would gain a good place in life and give my invaluable contribution to social service activities associated with spirituality.

If we talk about living people, then there was an opportunity to reconcile with many people in this PhD journey. Still, of all the people I have met, the best and the most influential person is none other than my mentor Prof. Dr. Jatinder R. Saini, Professor and Director, Symbiosis Institute of Computer Studies and Research, Symbiosis International Deemed University, Pune, India. His faith in me remained from the beginning of this research journey, which is why I have been able to do this research. I have often seen people say very big things to say, but the reality is something else, but there is no such thing at all here. A simple event can prove that when I started searching for my supervisor, I contacted many people. Still, out of those selected few who believed in me, only Dr. Saini was the one who believed the strongest and deeply and allowed me to get enrolled under his guidance on the last date of enrollment. He always taught me that 'Keep in mind that Ph.D. journey is a process of overall grooming of a person along with the intellect'. I feel fortunate enough that I got him as a research supervisor, and trust me this is the only best thing that happens to me in this entire research. I am especially thankful to him for sparing his time around the clock to answer my questions, reviewing my research papers and presentations, and keeping a good eye on my research progress.

Apart from the supervisor, I thank my Doctoral Progress Committee (DPC) members 1. Dr. Ravi M. Gulati, Associate Professor, Veer Narmad South Gujarat University, Surat and 2. Dr. Rustom D. Morena, Professor, Veer Narmad South Gujarat University, Surat for their continuous evolution throughout the Ph.D., Their insightful remarks and comments for the betterment of the research quality helped me improve the research standard. I am also thankful of the Gujarat Technological University Ph.D. Section Staff, to make the doctoral progress documentation related process smoothly and responding to

the queries I had over the long period of this research work.

From my perspective, I think a Ph.D. research life is full of ups and downs in which one has to deal with so many things. Family members are people with whom one has to live around with. I am thankful to every family member and friends for dealing with my enormous moods and behaviors during this research journey. My grandmother (SMT. Geeta Devi Audichya) can often be heard saying to people that "I could not complete my studies due to the lack of facilities in my time but, my grandson is doing it fully today". I am out of words to express my parent's (Advocate Arvind Kumar Audichya and Smt. Shobhana Audichya) endless support and love, which they showered on me irrespective of the situation. In this episode, how can I forget my brother (Advocate Sudhanshu Audichya), who always tried to make me feel lighter in my anxious times by disturbing me all the time. I express my gratitude toward all the friends and well-wishers who were there to check about my research progress.

Last but not least, I am grateful of all those supernatural energies, and almighty god and goddess surrounding to give me power, faith and strength to pursue this research.

Milind Kumar Audichya
(159997431001)

॥ॐ शिव ॐ॥

Dedicated to

The almighty God,
&
Late Shree Ramprasad
Narayan Ji Audichya

- पुजारी श्री भँवरमाता जी, देवनगरी
छोटीसादड़ी / मेवाड़, प्रतापगढ़, राजस्थान - भारत.

List of Abbreviations

IVR Interactive Voice Response

IWN IndoWordNet

KNN K-Nearest Neighbors Algorithm

LIWC Linguistic Inquiry and Word Count

ME Maximum Entropy

ML Machine Learning

MT Machine Translation

NB Naïve Bayesian Algorithm

NLG Natural Language Generation

NLP Natural Language Processing

NLTK Natural Language Toolkit

NLU Natural Language Understanding

POS Parts of Speech

PPM Preliminary Pragmatic Model

PSMT Phrase-Based Statistical Machine Translation

RBF Radial Basic Function

SMT Statistical Machine Translation

SVM Support Vector Machine

TF Term Frequency

TF-IDF Term Frequency Inverse Document Frequency

UTF Unicode Transformation Format

VSM Vector Space Model

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Natural Language Processing (NLP) is among the emerging technologies, which can be said as the combination of computer science (CS), computational linguistics (CL), and artificial intelligence (AI) [1]. The main goal of NLP is to make computers process natural languages to understand the language and accomplish meaningful tasks to make everyday life much easier. Some recent trending examples of the NLP of recent times are speech to real-time text typing, email reply sentence writing suggestions, sentence formation with spelling checking and grammatical rules, the suggestion to create reminders based on the dates mentioned in the event-based emails and meetings, searching online about any information. Life-changing products like Smart Voice Assistants, Interactive Voice Response (IVR) and automatic question and answering systems, etc., are becoming a part of our routine life based on the NLP. All these products are using natural language to communicate with the users.

Based on these remarkable examples, one can easily understand that in NLP, work is going on for understanding the real meaning of natural language first. The efforts are going on to use the same understanding in representing or responding according to the already understood meaning, but this is still a challenging aim to achieve. A natural language is a systematically managed system that is specially designed to express the conveyor's sense. The meaning can be conveyed in several forms, such as: Using sound or audio (Speech) [2, 3], With the help of some gestures (Sign Language) [4], or with the help of some symbols or image sequence (Writing) [5]. These are the several ways one can communicate, and it seems easy to interpret, right? Obviously, it looks easy but let us know what makes NLP hard. Several aspects make the NLP field harder to achieve its fundamental goals. It is complex to make computers learn and represent the natural language using linguistics, while the natural languages are full of plenty of ambiguities [6]. The natural languages used by humans require a lot of understanding of real-world situations, context-based knowledge, and some common sense.

Many significant improvements were observed in NLP in various stages such as text, speech, semantics, and syntax. Different tools for the parts-of-speech (POS), parsing, and entities are developed too. These are helping society with the question-answering systems, automatic chatbots, emotion detection, machine translation (MT), etc.

Natural Language Processing (NLP) is made up of two significant parts [7]:

1. Natural Language Understanding (NLU),
2. Natural Language Generation (NLG)

In initial NLP-based research works using various techniques, the unstructured data is converted into structured data, which is used to be analyzed with the NLU [8, 9]. Natural Language Understanding helps computers understand the meaning with the impact or intent of the phrases used in human's natural languages for communication. Sentiment detection, classification based on various aspects, entities detection is some of the actual applications of the NLU. In NLU, the machine needs to analyze, deal and interpret different kinds of unknown and ambiguous or erroneous expressions, making it more difficult.

Natural Language Generation (NLG) comes into the picture when something is produced by computers concerning the natural language, either in written or spoken form from a specific dataset [10, 11]. NLG aims to better communicate between machines and humans; it is also trying to simulate the human-to-human kind of conversations. Suppose one has to understand it from the research perspective. In that case, it can be said as NLG is using different mathematical information and formulas to analyze and extract the different patterns from the appropriate databases and try to make sense out of them in a form that humans can quickly understand. The application of the NLG can be understood with one of its examples. In Automatic content writing, the computer systems usually try to scrape the related keywords articles of the desired titled document from the different sources on the internet. After collecting the data, it tries to summarize everything quickly and develop a new piece of meaningful information. In NLG, the various ideas which are usually transformed into the response are generally known precisely.

Now it's time to learn a bit of the Computational Linguistics (CL) part of NLP, powerful complementary technology to NLP. It is a more often asked question that how both of these are different from each other? NLP and CL are both considered near-synonyms nowadays [12, 13]. In NLP, machines learn from repetitive examples and usually work very well when one has large datasets or plenty of data. Computational Linguistics, which is also a part of Artificial Intelligence (AI), is more about deconstructing any language, the syntax, and the composition of the words and sentences to know the actual meanings, And CL works very well even when one is not having an extensive data corpus or the

datasets are smaller in size. These technologies (NLP and CL) work together based on the need for hours to achieve the desired outcome. Both are interdisciplinary fields and work for the computational modeling of natural languages.

The importance of CL can be understood by Mark Steedman's following statements of 2007 in The Association for Computational Linguistics (ACL) Presidential Address, Mark is a cognitive scientist and computational linguist.

> "Human knowledge is expressed in language. So computational linguistics is very important [14]."

Two main types of processing approaches are there when it comes to processing natural languages with context to computational linguistics [12]:

1. Text Based Approach,
2. Speech Based Approach

It is known as per the name itself that the text-based approach is used to deal with the text-related processing works, and the speech-based method is used to deal with the audio or sound-based data processing of the natural languages.

As per the Ethnologue [15], 7139 languages are spoken around the world as of 2021. Out of these languages, only 23 languages are spoken by more than half of the world's population. Research works are going on for many languages which are widely known and used across the world. Good growth concerning NLP and CL can be observed for the languages such as Arabic, Chinese, English, Persian, etc., which will be discussed in-depth in Chapter 2. Literature Review. Hindi is one of the top 10 most spoken languages in the world in 2021 [16]. In India Hindi is a constitutionally accepted official language along with English [17].

Every language requires a script for written representation. Hindi also requires the same and is written through the Devanagari script. With the help of Devanagari script, more than 120 languages are written [18]. Nowadays, Hindi can be typed easily with Unicode's help. The Devanagari script is a part of the Unicode Standard, and its Unicode range is 0900-097F [19], which includes almost all the alphabets of the Hindi language. Nearly all the Hindi alphabets for which appropriate rules were found are included in this research work. Now the question that comes here is: How many alphabets/letters are there in Hindi?

According to Wikipedia [17], there are 44 primary characters, but there is a difference of opinion on this number and there are many different opinions [20–23]. For this

research work 52 alphabets are considered. The segregation of the Hindi Alphabets is as follow:
The vowels (‘स्वर’) : 11
(‘अ’, ‘आ’, ‘इ’, ‘ई’, ‘उ’, ‘ऊ’, ‘ऋ’, ‘ए’, ‘ऐ’, ‘ओ’, ‘औ’)

The consonants (‘व्यंजन’) – 33
(‘क’, ‘ख’, ‘ग’, ‘घ’, ‘ड’, ‘च’, ‘छ’, ‘ज’, ‘झ’, ‘ञ’, ‘ट’, ‘ठ’, ‘ड’, ‘ढ’, ‘ण’, ‘त’, ‘थ’, ‘द’, ‘ध’, ‘न’, ‘प’, ‘फ’, ‘ब’, ‘भ’, ‘म’, ‘य’, ‘र’, ‘ल’, ‘व’, ‘श’, ‘ष’, ‘स’, ‘ह’)

Combined consonants (‘संयुक्त व्यंजन’) – 4
(‘क्ष’, ‘त्र’, ‘ज्ञ’, ‘श्र’)

Anusvar, Anunasic or Chandrapindu (‘अनुस्वार’, ‘अनुनासिक या चंद्रबिंदु’) – 1
(‘ं’) or (‘ँ’)

Visarg (‘विसर्ग’) - 1
(‘ः’)

Binary consonant (‘द्विगुण व्यंजन’) – 2
(‘ड़’, ‘ढ़’)

There are fifty-two (11+33+4+1+2=52) total alphabets/letters in Hindi as discussed, through which the words and sentences are formed, which is the initial entity of any literature.

The prevalence of any language can be known from its literature. Literature is a word from Latin that means “writing formed with letters”, in general, it is also meant as togetherness [24]. The Hindi language is having a lot of literature in its heritage. There are mainly two forms of literature which are: Prose and Poetry. A well-known French poet and critic named Paul Valéry compared the prose to walking and poetry to dancing [25].

The word “Prose” is coming from Latin language, which means’ Straight’ or ‘Direct’ [26]. The prose is written in paragraphs which is a combination of the sentences. In short words, it can be expressed as that prose is direct and straightforward writing practice in which the writer tries to convey the feelings and thoughts as precisely and transparently as possible. As prose is compared with walking, it means prose is functional and provides some information. In prose, the message which includes information is essential. Prose can be paraphrased or summarized.

The word “Poetry” is derived from Greek language, which means ‘Making’ [27]. Poetry is generally written in verse, and verses are made up of stanzas. Poetry usually leaves plenty of things unsaid for its reader’s imagination, which is often known as “reading between the lines”. As poetry is compared with dancing, it aims to delight. Usually, in a poem, the importance conveys the experience rather than any information or meaning. One can paraphrase or summarize poetry, but that paraphrase cannot be said a poem as it will eventually lose its charm of poetry stanzas.

Literature, prose, poetry, and language-related discussions are added here due to a purpose. Here the author wants to contrast the concept of prose and poetry because it is essential to know this difference from the point of view of NLP. Thus, the language processing treatment given to the prose is different, and the treatment given to poetry is also different. It will further differ in the case of different types of languages and their respective scripts. From the language processing view, prose processing is more straightforward than the processing of poetry as the construction of prose is simple, and the structure of poetry itself is complex.

In Hindi literature, there are several styles or ways of writing. Like: Satire, Playwriting, Poetry, Essay Writing. Hindi Kavita or Hindi Poetry is one of the very popular among these ways of writing, and This is the only reason there are plenty of poems written in the Hindi language by many writers. To name a few, some great names such as Tulsidas, Surdas, Kabirdas, Rahimdas, Suryakant Tripathi ‘Nirala’, Harivanshrai Bachchan, Mahadevi Verma, Makhanlal Chaturvedi, Maithili Sharan Gupt, Sumitranandan Pant, and Ramdhari Singh ‘Dinkar’ can be considered. Still, the list is not limited to these only. There are plenty of writers who contributed to the Hindi poetry segment of Hindi Literature. In this research work, efforts were given to include the poems from as many writers as possible. There is a vast repository of poems in Hindi literature in physical form. A significant portion of poetry is available in the form of physical books, magazines, and papers. Physical Hindi poetry will be digitized today or tomorrow as research works to digitize the offline physical data at a good pace using Optical Character Recognition (OCR), which helps store data through the higher quality of document imaging. Some advanced OCR-based research works [28–31] even claiming to extract text out of it .

Poems have been written in the Hindi language for so many decades. One of Hindi Poetry’s best examples is ‘Hanuman Chalisa’(‘हनुमान चालीसा’). Poet Goswami Tulsidas wrote ‘Hanuman Chalisa’(‘हनुमान चालीसा’) in the 16th Century. The poem is a combination of ‘Doha’(‘दोहा’) and ‘Chaupai’(‘चौपाई’), which are the core types of Hindi Verses [32].

Here is the concluding ‘Doha’(‘दोहा’) from ‘Hanuman Chalisa’(‘हनुमान चालीसा’).

‘पवनतनय संकट हरन, मंगल मूरति रूप।
राम लखन सीता सहित, हृदय बसहु सुर भूप॥’

Hindi poetry is made up of the three major components, which are as follows:

- ‘Chhand’ (Verse or Metre – ‘छंद’) [33]
- ‘Alankar’ (Figure of Speech – ‘अलंकार’) [34]
- ‘Ras’ (Sentiment – ‘रस’)[35]

All three components develop the magical essence in Hindi poetry, giving readers or listeners the feeling of amusement, entertainment, or mesmerizing glide. These three have their own respectively associated rules which need to be followed while constructing any poem. Each of these is having different classes or types based on the various criteria. This research work revolves around the Verse or Metre, known as ‘Chhand / Chhands’ in Hindi. For better explanation and understanding, these terms are used throughout this thesis. The research work also tries to touch the ‘Alankar’ component of Hindi poetry known as ‘Figure of Speech’ in English. It was not the initial part of the main objective and scope of this research work. Later on, it was included as it will give a fresh start to a separate wing of research in Hindi poetry as distinguished Hindi Metre/Verse or Hindi sentiment-related research.

Let us understand the Hindi Verse and The Hindi Figure of Speech more one by one.

- Hindi Verse,
- Hindi Figure of Speech

1.1 Hindi Verse

Hindi Verses (‘Chhands’- ‘छंद’) or Metre are coming from the Sanskrit language originally [33, 36]. Hindi Verses (‘Chhands’- ‘छंद’) can be said as one of its highest ancient development because it inherits many refinements and the original tongue and verbal flexibility. To understand this research better, one needs to know about the complexity and construction rules of Hindi Verse first. Plenty of type of Verses (‘Chhands’- ‘छंद’) are written in the Hindi language; one thing is to be noted here because each type has its own unique rule of formation and that only makes it special or different from other ‘Chhand’ types.

In this research work, efforts were made to include as many Hindi Verses(‘Chhands’- ‘छंद’) as possible, and it is designed in such a way that if any type is left, then that can be added with no or minor change. To accomplish the same initial information about Hindi Verses (‘Chhands’- ‘छंद’) were needed, the information collection phase was tough and challenging because no standard or well-managed information was found which can be used directly for research purpose. To include every penny information, all the trustworthy sources like recently published books, old ancient books, internet blogs [37, 38], portals [39–41], websites [42], handwritten notes, etc. were traced as per the validation by Hindi academic expert’s views on different contents with the better understanding of the construction rules relevant to the Hindi Verses (‘Chhands’ - ‘छंद’).

This research attempts to give a systematic view of Hindi Verses (‘Chhands’- ‘छंद’) from a computational linguistics perspective and is mainly related to Hindi Verses only. Verses are called as ‘Chhand’ in Hindi and holds an imperative spot in the poetry division of Hindi literature. As per the ancient times books, it was found that the Hindi Verses (‘Chhand’ - ‘छंद’) is derived from the long back times of Vedas and Puranas, which are the uttermost part of the Indian knowledge management and Hinduism – The oldest religion in the world. Being a ‘Vedangs’(‘वेदांग’) [43], ‘Chhand’ (‘छंद’) is one of the six ancient times auxiliary disciplines which are required for the study of the ancient Vedic ‘Vedas’. An old saint and renowned mathematician ‘Pingal’(‘पिंगल’) [44], who is also identified as ‘Pingalacharya’(‘पिंगलाचार्य’) and ‘Sheshaavtar’(‘शेषावतार’), created a ‘Chhandshastra’(‘छंदशास्त्र’) [45] used to teach and provide the knowledge of Hindi Verse is also discussed and referenced in ‘Vedas’(‘वेद’) and ‘Puranas’(‘पुराण’) [46].

The poetic composition is known as Verse or Metre or ‘Chhand’(‘छंद’), and the composition which is not poetical is usually called Prose.

1.2 Core 'Chhand' Types

'Chhands' are bifurcated into two core types as given below [45]:

1. 'Vedic Chhands' ('वैदिक छंद')
2. 'Laukik Chhands'('लौकिक छंद')

1.2.1 'Vedic Chhands' ('वैदिक छंद')

'Vedic Chhands' ('वैदिक छंद') is from the ancient time of 'Vedas' ('वेद') and is primarily written in the Sanskrit language. The Vedic Sanskrit Mantras are based on the 'Vedic Chhands' ('वैदिक छंद'), that can be observed and traced in different Mantras. To name a few some well known 'Vedic Chhands' ('वैदिक छंद') are: 'Atyashthi' ('अत्यष्टि'), 'Anushtup' ('अनुष्टुप'), 'Ashti' ('अष्टि'), 'Atijagti' ('अतिजगती'), 'Atishakkari' ('अतिशक्वरी'), 'Bruhati' ('बृहती'), 'Dhruti' ('धृति'), 'Dwipada Virat' ('द्विपदा विराट'), 'Ekpada Virat' ('एकपदा विराट'), 'Gayatri' ('गायत्री'), 'Jagati' ('जगती'), 'Mahabruhati' ('महाबृहती'), 'Pankti' ('पंक्ति'), 'Pragath' ('प्रगाथ'), 'Prastar Pankti' ('प्रस्तार पंक्ति'), 'Shakkari' ('शक्वरी'), 'Trishthup' ('त्रिष्टुप'), 'Ushnik' ('उष्णिक्'), 'Virat' ('विराट') [47].

One of the prevalent and familiar Vedic Mantra from – *Rigveda 3.62.10*, named Gayatri Mantra, is written in Gayatri Chhand [48].

'ॐ भूर्भुवः स्वः तत्सवितुर्वरेण्यं भर्गो देवस्य धीमहि धियो यो नः प्रचोदयात् ॥'

('om bhur bhuvah svah tatsavitur varenyam bhargo devasya dhimahi dhiyo yo nah prachodayat' – Let's pray in front of God, who is the producer of this beautiful world. May God enlightens us by divine spirituality.)

1.2.2 ‘Laukik Chhands’(‘लौकिक छंद’)

‘Laukik Chhands’ (‘लौकिक छंद’) are not from Vedas, and these are the verses that the people create. These ‘Chhands’ (‘छंद’) are written in Sanskrit as well as Hindi. In this research work, only Hindi ‘Laukik Chhands’ are focused and included. ‘Laukik Chhands’ (‘लौकिक छंद’) are organized into the three groups as per the characteristics of the construction practices of these verses as follow [45]:

1. ‘Matrik Chhands’ (‘मात्रिक छंद’)
2. ‘Varnik Chhands / Vrutts’ (‘वर्णिक छंद / वृत्त’)
3. ‘Mukt / Muktak Chhand’ (‘मुक्तक/मुक्त छंद’)

Further sub-classes of these classes are also there, based on the nature and rules of construction.

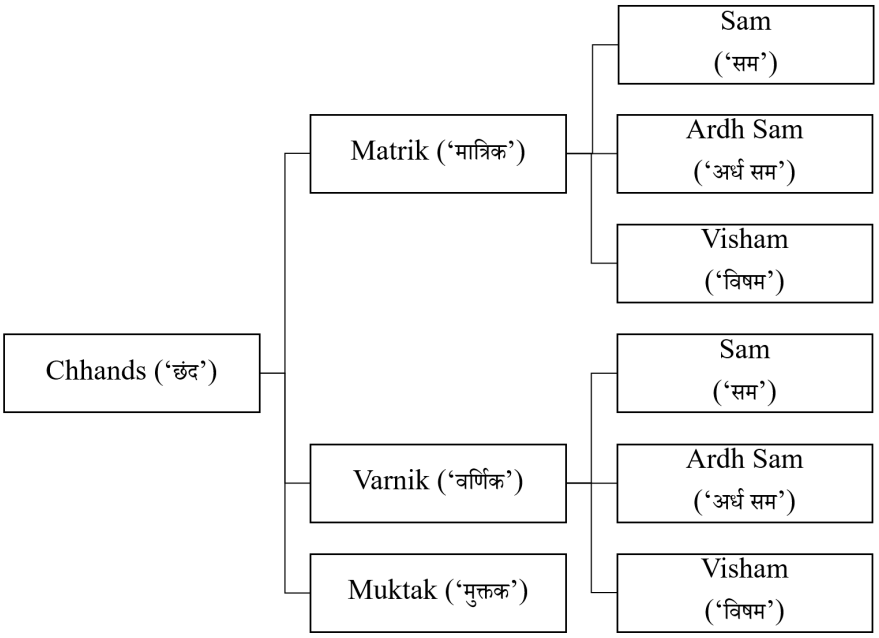


Figure 1.1: Classes and Subclasses of ‘Laukik Chhand’

Figure 1.1 represents about Classes and Subclasses of ‘Laukik Chhand’. There are three different types of verses concerning even-odd stanzas [49–52].

Even Verses (‘सम’)

Whose all four stanzas have the same symptoms [49, 52].

Half-Even Verses (‘अर्धसम’)

Whose odd-odd, i.e., first and the third stanza are of the same symptoms, and even-even, the second and fourth stanzas are of the same symptoms [49, 52]. The verses that are written in two lines are called a ‘Dal’(‘दल’) [51].

Odd Verses (‘विषम’)

Which are neither Even nor Half-Even are considered Odd. The composition of verses with more than four phases is also considered as Odd [49, 52].

There are further two types of Even Verses also.

In Matrik Verses, if the number of quantities (‘मात्रा’) is up to 32, then these verses are considered as General Verses (‘साधारण छंद’). If the number of quantities (‘मात्रा’) is more than 32, then these verses are considered as ‘Dandak’ (‘दंडक’) Verses [49, 52].

In Varnik Verses/Vrutt, if the number of Varna is up to 26, then these verses are considered as General Verses/Vrutt (‘साधारण छंद / वृत्त’). If the number of Varna is more than 26, then these verses are considered as ‘Dandak’ Verses/Vrutt (‘दंडक छंद / वृत्त’) [49, 52].

In this research work, the hierarchy is considered up to three levels only, in that also specially in ‘Varnik Verses/ Vrutt’(‘वर्णिक छंद / वृत्त’) the theoretical presence of Half-Even Verses (‘अर्धसम’) and Odd Verses (‘विषम’) were seen but appropriate satisfactory rules and examples were not found. It is believed that in Hindi Verses, no Matrik Odd Verses, Varnik Half-Even Verses, and Varnik Odd Verses are there [52]. Still, There are provisions to add these in case any adequate information is found in upcoming times. The detailed structure of these classes is included in Section 3.4 Hierarchical structure construction of Hindi Metre.

1.3 Core Classes of Hindi Verse

It is known as per the discussion in 'Laukik Chhands'('लौकिक छंद') and Figure 1.1 Classes and Subclasses of 'Laukik Chhand', that there are three core classes of Hindi Verse. Let us discuss something about each.

1.3.1 'Matrik Chhands' ('मात्रिक छंद')

The verses which are made up of the fixed number of quantities ('मात्रा') in each of the basic four stanzas ('Charan'-('चरण')) are known as 'Matrik Chhand' ('मात्रिक छंद') or Matrik Verse [49]. The sequence of various characters ('Varna' - 'वर्ण' such as ('Laghu'-('लघु') / 'Guru'-('गुरु')) is irrelevant in Matrik Verses.

Example:

'माखन-मिश्री हाथ में,
लेकर कान्हा भागा।
लीला जिसने देख ली,
भाग्य उसी का जागा।'

Here in this example of one of Matrik Verse named 'Muktamani' ('मुक्तामणि') Verse in which the quantities in each stanza are [13, 12, 13, 12]. One might wonder how these quantities are calculated; for the answer to the same, the quantity calculation is discussed in depth in Section 3.3 Simplified Quantity Calculation Rules.

Table 1.1: A few 'Sam Matrik Verses'('सममात्रिक छंद')

S. N.	Verse Name	Class	Sub Class	Matra Count
1	Ahir ('अहीर')	'Matrik Verse' ('मात्रिक छंद')	'Sam Matrik Verse' ('सममात्रिक छंद')	11
2	Tomar ('तोमर')	'Matrik Verse' ('मात्रिक छंद')	'Sam Matrik Verse' ('सममात्रिक छंद')	12
3	Chaupai ('चौपाई')	'Matrik Verse' ('मात्रिक छंद')	'Sam Matrik Verse' ('सममात्रिक छंद')	16
4	Rola ('रोला')	'Matrik Verse' ('मात्रिक छंद')	'Sam Matrik Verse' ('सममात्रिक छंद')	24
5	Ullala ('उल्लाला' ('सममात्रिक'))	'Matrik Verse' ('मात्रिक छंद')	'Sam Matrik Verse' ('सममात्रिक छंद')	26
6	Gitika ('गीतिका')	'Matrik Verse' ('मात्रिक छंद')	'Sam Matrik Verse' ('सममात्रिक छंद')	26
7	Harigitika ('हरिगीतिका')	'Matrik Verse' ('मात्रिक छंद')	'Sam Matrik Verse' ('सममात्रिक छंद')	28
8	Tribhangi ('त्रिभंगी')	'Matrik Verse' ('मात्रिक छंद')	'Sam Matrik Verse' ('सममात्रिक छंद')	32

Table 1.1. is representing A few 'Sam Matrik Verses'('सममात्रिक छंद') along with their Matra Count or Quantity. In 'Sam Matrik' Verse, all stanzas have the same Quantity or Matra Count.

Different tables mentioned here (Table 1.1,1.2,1.3,1.4) are derived from the discussion and pieces of information collected from multiple sources such as recently published

books [50], old ancient books [49, 46, 51], internet blogs [37, 38], portals [39–41], websites [42, 52], after proper mannual verification.

Table 1.2: Some 'Ardh Sam Matrik Verses'('अर्द्धसममात्रिक छंद')

S. N.	Verse Name	Class	Sub Class	Even Matra Count	Odd Matra Count
1	Barvai ('बरवै')	'Matrik Verse' ('मात्रिक छंद')	'Ardh Sam Matrik Verse' ('अर्द्धसममात्रिक छंद')	7	13
2	Doha ('दोहा')	'Matrik Verse' ('मात्रिक छंद')	'Ardh Sam Matrik Verse' ('अर्द्धसममात्रिक छंद')	11	11
3	Sortha ('सोरठा')	'Matrik Verse' ('मात्रिक छंद')	'Ardh Sam Matrik Verse' ('अर्द्धसममात्रिक छंद')	13	13
4	Roopmala ('रूपमाला')	'Matrik Verse' ('मात्रिक छंद')	'Ardh Sam Matrik Verse' ('अर्द्धसममात्रिक छंद')	14	12
5	Ullala ('उल्लाला' ('अर्द्धसममात्रिक'))	'Matrik Verse' ('मात्रिक छंद')	'Ardh Sam Matrik Verse' ('अर्द्धसममात्रिक छंद')	15	10
6	Tantak ('तांटक')	'Matrik Verse' ('मात्रिक छंद')	'Ardh Sam Matrik Verse' ('अर्द्धसममात्रिक छंद')	16	14
7	Kukubh ('कुकुभ')	'Matrik Verse' ('मात्रिक छंद')	'Ardh Sam Matrik Verse' ('अर्द्धसममात्रिक छंद')	16	14
8	Veer ('वीर')	'Matrik Verse' ('मात्रिक छंद')	'Ardh Sam Matrik Verse' ('अर्द्धसममात्रिक छंद')	16	15

Table 1.2 is representing Some 'Ardh Sam Matrik Verses'('अर्द्धसममात्रिक छंद') along with their Matra Count or Quantity. In 'Ardh Sam Matrik' Verse, all even stanzas have the same Quantity or Matra Count, and Odd stanzas have the same Quantity or Matra Count.

Table 1.3: 'Visham Matrik Verses'('विषम मात्रिक छंद')

S. N.	Verse Name	Class	Sub Class	Details
1	Kundliya ('कुंडलिया')	'Matrik Verse' ('मात्रिक छंद')	'Visham Matrik Verse' ('विषम मात्रिक छंद')	'Doha' + 'Rola' ('दोहा' + 'रोला')
2	Chappay ('छप्पय')	'Matrik Verse' ('मात्रिक छंद')	'Visham Matrik Verse' ('विषम मात्रिक छंद')	'Rola' + 'Ullala' ('रोला' + 'उल्लाला')

Table 1.3 is showing about 'Visham Matrik Verses'('विषम मात्रिक छंद') which are made up of the combination of 'Sam Matrik' Verse and 'Ardh Sam Matrik' Verse.

1.3.2 'Varnik Chhands / Vrutts' ('वर्णिक छंद / वृत्त')

The verses based upon the computation of characters ('Varnas' - 'वर्ण') using the Predefined Sequenced Characters ('Ganas'-'गण') and mainly based on the sequence of characters ('Varnas'-'वर्ण') are called 'Varnik Chhands / Vrutts' ('वर्णिक छंद / वृत्त') or Varnik Verse [49]. The verses which are based on the 'Ganas'('गण'), and having the fixed sequence of characters ('Varnas'- 'वर्ण') ('Laghu'('लघु') / 'Guru'('गुरु')) in all four stanzas are named as 'Varnik Vrutt' ('वर्णिक वृत्त') or Even Verses ('सम छंद').
Example:

'प्रिय पति वह मेरा प्राण प्यारा कहाँ है।
दुख-जलधि निमग्ना का सहारा कहाँ है।।
अब तक जिसको मैं देख के जी सकी हूँ।
वह हृदय हमारा नेत्र तारा कहाँ है।।'

Here is the example of one of the Varnik Verse named 'Malini' ('मालिनी'), which is made up of the combination sequence of 'Ganas' in each stanza like : ('NaGana + NaGana + MaGana + YaGana + YaGana') for which symbolic arrangement is ('। । । । । । ऽ ऽ ऽ । ऽ ऽ । ऽ ऽ'). Quantity calculation and Symbolic rules are discussed in depth in 'Ganas' are discussed in Section 3.2 Components of Hindi Verse or 'Chhand' and Section 3.3 Simplified Quantity Calculation Rules.

Table 1.4: Some 'Varnik Verses'('वर्णिक छंद')

S. N.	Verse Name	Class	Sub Class	Varna-Count	Sequence	Symbolic Representation
1	Indravajra ('इन्द्रवज्रा')	'Varnik' ('वर्णिक')	'Sam Varnik' ('समवर्णिक')	11	'मगण, तगण, तगण, गुरु, गुरु'	'SSSSSISSISS'
2	Upendravajra ('उपेन्द्रवज्रा')	'Varnik' ('वर्णिक')	'Sam Varnik' ('समवर्णिक')	11	'जगण,तगण, जगण, गुरु, गुरु'	'ISISSIISISS'
3	Vasanttilka ('वसंततिलका')	'Varnik' ('वर्णिक')	'Sam Varnik' ('समवर्णिक')	14	'तगण, भगण, जगण, जगण, गुरु,गुरु'	'SSISIIISIISISS'
4	Malini ('मालिनी'')	'Varnik' ('वर्णिक')	'Sam Varnik' ('समवर्णिक')	15	'नगण, नगण, मगण, यगण, यगण'	'IIIIIISSSISSISS'
5	Madakranta ('मन्दाक्रान्ता'')	'Varnik' ('वर्णिक')	'Sam Varnik' ('समवर्णिक')	17	'मगण, भगण, नगण, तगण, तगण, गुरु,गुरु'	'SSSSIIIIISSISSISS'

Table 1.4 is representing Some 'Varnik Verses'('वर्णिक छंद') with their respective 'Chhand' Name, Type, Sub Type, Character ('Varna') Count, Predefined Sequence of 'Ganas', and the Symbolic Representation, which is required to full fill the construction rules criteria of respective 'Chhands'.

1.3.3 'Mukt / Muktak Chhand' ('मुक्तक/मुक्त छंद')

The verses that do not follow any specific predefined rules are known as 'Mukt / Muktak Chhand' ('मुक्तक/मुक्त छंद'). In other simple words, it can be said that this is a free form class or category in which all the poetic creation are included which are not included in either 'Matrik Chhands' ('मात्रिक छंद') or 'Varnik Chhands / Vrutt' ('वर्णिक छंद / वृत्त') [49]. Example:

'विजन-वन-वल्लरी पर
सोती थी सुहाग भरी स्नेह स्वप्न मग्न
अमल-कोमल-तनु तरुणी जुही की कली,
दृग बन्द किये, शिथिल पत्रांक में,
वासन्ती निशा थी;
विरह विधुर प्रिया संग छोड़
किसी दूर देश में था पवन
जिसे कहते हैं मलयानिल।'

The above example of 'Muktak' taken from 'Juhi Ki Kali' penned by Suryakant Tripathi 'Nirala', as it is not following any rules of Matrik or Varnik Verses, so it is considered as Mukt / Muktak Verse.

There are some basic rules and components of 'Chhands' used in the construction of the Hindi Verses collected from various sources [33, 49, 50] also verified and validated manually through calculations.

1.4 Hindi Figure of Speech (‘अलंकार’)

The Hindi Figure of Speech is called ‘Alankar’(‘अलंकार’) in the Hindi language. The appearance of any ‘Alankar’ is sufficient to add magic in any poetry. Similar to the Hindi ‘Chhands’ efforts were made to collect as much information as possible to systematically manage every penny detailed information of the ‘Alankars’. Different meaningful offline and online sources (Physical printed educational books and articles, Websites and Portals [53–55]) were explored for the rules and meaningful information. Figure 1.2 is derived based on such meaning pieces of information only.

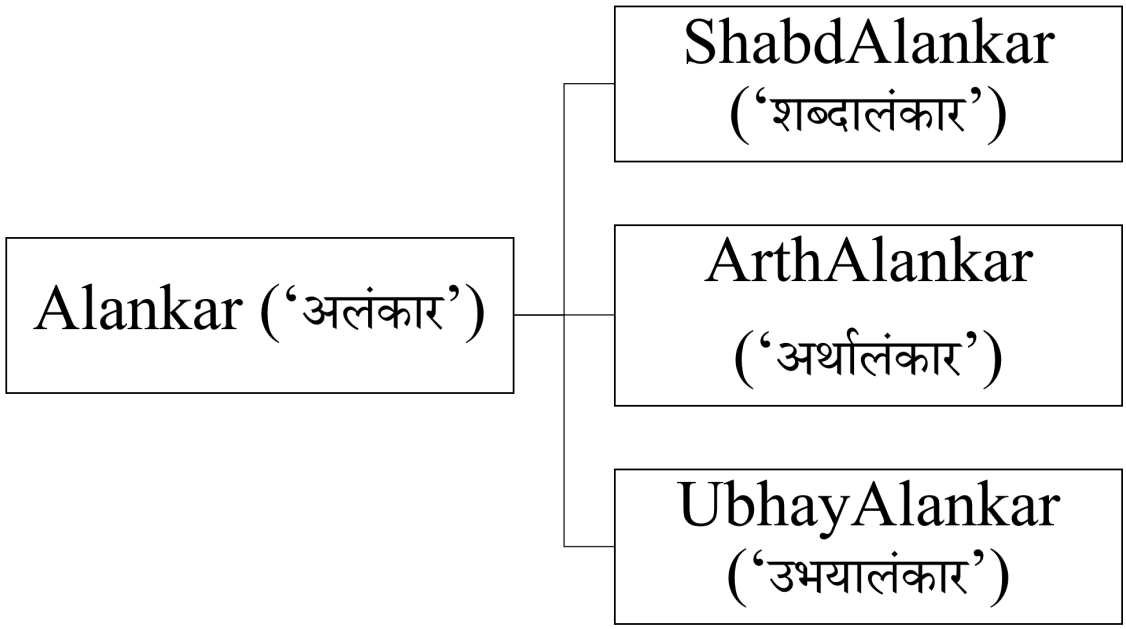


Figure 1.2: Core Classes of Hindi ‘Alankar’ (‘अलंकार’)

Figure 1.2 is showing the Core Classes of Hindi ‘Alankar’ (‘अलंकार’). There are three major classes of the Hindi ‘Alankars’, as given below:

1. ‘ShabdAlankar’ (‘शब्दालंकार’)
2. ‘ArthAlankar’ (‘अर्थालंकार’)
3. ‘UbhayAlankar’ (‘उभयालंकार’)

Let us understand about each of these classes.

1.4.1 'ShabdAlankar' ('शब्दालंकार')

The Hindi figure of speech ('अलंकार') in which the uses of specific words generate the miracle in the poetry, and the miracle disappears when synonyms of these words are used is known as 'ShabdAlankar' ('शब्दालंकार') [54].
Example:

'चारु चन्द्र की चंचल किरणें,
खेल रहीं थीं जल-थल में।'

In the example, the presences of the characters and words are more impactful instead of meaningful.

1.4.2 'ArthAlankar' ('अर्थालंकार')

The Hindi figure of speech ('अलंकार') in which the meaning of the specific words used in poetry generate the miracle, is known as 'ArthAlankar' ('अर्थालंकार')[55].
Example:

'हरि पद कोमल कमल ।'

The primary focus is on the meaning of the words and the meaning's context in the example.

1.4.3 'UbhayAlankar' ('उभयालंकार')

The Hindi figure of speech('अलंकार') in which the presence of specific words and meaning of the specific words jointly generate the miracle, is known as 'UbhayAlankar'('उभयालंकार')[56].
Example:

'कजरारी अंखियन में कजरारी न लखाय ।'

In the given example both the words and meanings are showing their impact.

Detailed hierarchical structure and classification are discussed in Section 3.4 Hierarchical structure construction of Hindi Metre. In this present research work, efforts were made for following things with context to 'Alankar':

- In-depth exploration of the Hindi Figure of Speech (Alankar-'अलंकार')
- The hierarchical structure standardization of the taxonomic classification of the various type of 'Alankar'
- Initiate the approach to identify 'Alankar' with the identification of three popular 'Alankar'

Here comes the need for the hour; there is a need for a particular automatic system to generate the metadata about these poems individually to manage and digitize the Hindi poetry systematically. In various research of languages by natural language processing (NLP), Hindi has an important place. However, the verses are still untouched in this context. Computational linguistics (CL) is a field that deals with the computational modeling of natural language [12]. From the point of view of computational linguistics (CL), knowledge of the extinct verses can be saved. This research is the best combination of the NLP and CL, wherein NLP one can do wonders using human language, and in CL, computers are used to understand the languages. Because as it is already known that nowadays, these two are considered near-synonyms.

In this aspect, the problem is developing a rule-based automatic metadata generator that can identify the poetry based on the existing rules and capable enough to incorporate the new practices in upcoming times with no or minimal effort.

Objectives and Scope

This research work is trying to aim the following objectives:

- To come out with a systematic hierarchical structure of Hindi Verses.
- To model the metadata generator through computational linguistics perspective rule-based modeling of Hindi Verses construction rules for Hindi poetry.
- To develop a corpus for Hindi poetry

Chapter 2

Literature Review

The in-depth review of literature helps the researcher understand the research domain more and helps find the research scope and the need of the research in a particular research domain. However, the literature review itself was challenging in this research work because the author tried to find the research works associated with metadata generation for Hindi Poetry automatically based on computational linguistics and related explicitly to Hindi Verses. Unfortunately, such research works are still too little or nearly none. Nevertheless, the author divided the research literature review into several parts. Several factors are considered of different related works as no direct research work was seen. The division of the several factors are based on the other aspect like Natural Language Processing(NLP), Computational Linguistics(CL) based research studies, Indian languages poetry-based research works, Foreign languages poetry-based research works, Hindi related researches, the works related to metadata generation, and many more related linked research works are considered for this literature review.

As it is known now that research work directly related to this research work was not seen that only made this research work more unique and challenging. There were only two slightly nearby research works by Kushwah and Joshi that were somewhat related but can't be genuinely relevant when the different aspects of the studies are analyzed. Kushwah and Joshi [56] researched one of the 'Equi-Matrik Chhands' of Hindi poems named 'Rola'. In that research work, they achieved 89.83% accuracy through automatic and manual detection of their algorithm. In another research work, Joshi and Kushwah [57] researched the automatic detection of one of the Hindi 'Chhands', namely 'Chaupai' in which they achieved 95.03% of accuracy. These research studies [56, 57] confirm that Hindi Verses can be detected if the rules of different verses are systematically organized and modeled, which is the core of the current research problem. Even though these studies deal with the single type of Hindi Verse, multiple Hindi Verses can be managed with a broader approach. Therefore, this research work will be covering numerous types of Hindi Verses.

Pursuing the literature review further, many research studies were explored. Only related works such as text classification, tokenization or token extraction, emotion detection, sentiment analysis, and much other associated research are added in this research literature review.

Kaur and Saini [58] worked with the help of different machine learning algorithms for the Punjabi poetry classification. They found that with 60% accuracy Naive Bayes performed the best. Kaur and Saini [59] researched on Punjabi poetry classification and performed tests on 240 different poetries through the 10 types of the machine learning algorithms. They achieved respective accuracy of 50.63% (Hyperpipes), 52.92% (K-nearest neighbor), 52.75% (Naive Bayes), and 58.79% (Support Vector Machine). Kaur and Saini [60] worked for automatic Punjabi poetry classification using poetic features on 2034 poetries and achieved 71.98% accuracy with the Support Vector Machine (SVM) algorithm. Kaur and Saini [61, 62] also attempted to develop a Punjabi poetry classifier using logistics features and weighting in different research works and tested using different algorithms with various textual features.

Kaur and Saini [58–62] have done excellent and progressive work to classify Punjabi poems in their various research works. Through these research papers, the author found that they worked for four significant classes based on the different themes of Punjabi poems based on the uses of words in respective categories. They considered NAFE (Nature and Festival), LIPA (Logistic and Patriotic), RORE (Romantic and Relation), PHSP (Philosophical and Spiritual) as four classes of classification with the help of machine learning to solve the classification problem with the use of different algorithms.In these research works, authors deal with the Punjabi language's poetry classification with the help of several machine learning algorithms. For writing, the Punjabi language script is Gurumukhi, and Hindi is written using Devanagari script, so it is evident that there will be some differences. Still, to know the conceptual idea of dealing with the classification of poetry, these research work were explored. Classification of Punjabi Poetry by Kaur and Saini was based on the theme (e.g. nature, romance, philosophy, etc.) of the poetry rather than the structure of the poetry as is proposed in the present research work.

In classification-related problems, several research works were found. Hansen [63] worked for the solution of the classification problems with the use of automatic programming.The current research work is trying to generate the metadata automatically through the core programming. To know how to handle the things automatically, the research work was explored. Abbas and Asif [64] researched to compute prosody for Punjabi Ghazal to detect the Arud Metre. Arud is understood as the study of poetic meters. They worked phonetically and phonologically for their research work and developed an au-

tomatic process that was satisfactory. The current research work is dealing with metre through a text-based approach still, to know about how they managed to deal with prosody poetically, the research work was studied.

Pandian et al. [65] researched for Author Identification of Hindi Poetry, they worked on 3 separate writer's 100 Hindi poems. The features were extracted from their poems and provided to the classifiers based on various classification algorithms. LogitBoost achieved 75.6757% maximum accuracy on the given corpus. Author identification is different, yet the work was related to the classification of Hindi poetry. Although current work also deals with Hindi poetry to explore the similarities, the research work was studied.

Bafna and Saini [66] classify Hindi Verses such as Bal geet, Bhajan, Desh Bhakti, and Updesh Geet using several algorithms of machine learning. In various studies, they also created the class predictor for the Hindi Verse using concept learning algorithms and classification of the poetries in Hindi language with the use of the eager supervised machine learning(ML) algorithms [67, 68]. Bafna and Saini [69] further worked for poems and stories of Hindi and Marathi languages token extraction applications using Zipf's law. They used 820 poems, 710 stories for Hindi and 505 poems, and 610 stories for Marathi. Bafna and Saini [70] additionally worked with their own technique, "BaSa" for the context-based standard tokens identification of for Hindi proses and verses. Total 710 verses and 820 proses were used. Bafna and Saini [66–70], in their different research works, worked for the Hindi language-based research, which is common with the current research work. They also worked to identify the Hindi Verses based on the different themes (Balgeet, Bhajans, DeshBhakti, and UpdeshGeets) where the present work is trying to identify the Hindi verses through their structure of construction.

In Hindi-related more works, Joshi and Kushwah [71, 72] worked for one of the Hindi grammatical concepts called 'Sandhi' and 'Sandhi-Vichchhed'. In a rule-based way of word formation in Hindi, they tested their algorithm on 887 different compound words on which 'Sandhi-Vichchhed' (Splitting of the compound words) can be performed. Joshi and Kushwah [71, 72] managed to deal with Hindi words with Unicode (UTF-8) characters [19], and the current research work also has a significant role of the Hindi Unicode (UTF-8) characters in different processes.

In another research study Kaur and Saini [73] researched the document classification through the text classification of the different Indian languages (Urdu, Bangla, Punjabi, Telugu, Tamil, Kannada, Assamese) content-based researches. In conclusion, they found that Indian content must be explored much more for text classification as very few works

were found during their study.Kaur and Saini [74] studied and analyzed eight languages (Hindi, Bengali, Punjabi, Oriya, Urdu, Marathi, Telugu, and Manipuri) from three different languages families, namely Indo-Aryan, Dravidian, and Tibeto-Burman. In this study, they worked for opinion mining in comparison to the English language. They resulted in found that higher performance in English language text in contrast to the Indian Languages. In these research works, Kaur and Saini [73, 74] worked with the text of different languages, including Hindi, and the current research work also deals with the Hindi text-based classification approach.

While finding the research works, some Sanskrit language-based Computational Linguistics related work was also seen and observed. They worked on Panini's Astadhyayi structure and parsing related issues and machine translation [75]. Current research work deals with Hindi Poetry Identification with context to the structure where these research works [75] are related to and dealing with Panini's Astadhyayi, written in the Sanskrit Language. Pennebaker et al. [76] introduced multiple computational linguistics-based aspects for the different linguistics dimensions in their research context-based program Linguistic Inquiry and Word Count (LIWC). The current work is also dealing with word count-related operations along with the linguistics aspects.

In some other translation-related works, Pallavi and Mojibur [77] developed an initial pragmatic model for the poetry translation evolution. Yadav et al. [78] worked on translating the couplets from the English language to Hindi language using statistical machine translation (SMT). Chakrawarti et al. [79] worked for the Phrase-based statistical machine translation (PSMT) to translate the Hindi language poems into the English language. These translation-related research works [77–79] are not directly related to the proposed research work, yet included here as they deal with the text. Current research work also deals with text to get ideas about the different ways of dealing with text such works were referred.

Emotion detection and sentiment analysis-related works were also seen and explored. Kaur and Saini [80] also worked for the emotion detection in Punjabi poetry on a manually annotated corpus. Furthermore, the Support Vector Machine (SVM) achieved best accuracy with 70.02% . Pal and Patel [81] developed a model for the Hindi language based classification of poems on the basis of nine categories of the Ras, 55 poems were used, and they found that the Support Vector Machine algorithm performed better than the Naïve Bayes algorithm. Jha et al. [82] explored the sentiment analysis for Hindi movie reviews. They used Support Vector Machine (SVM), Naive Bayes Classifier, and Maximum Entropy (ME) technique of Machine Learning and Lexicon Based Classification Technique to detect document polarity. In the emotion analysis domain, Kumar et al. [83] created a Hindi annotated corpus named 'BHAAV' from Hindi stories for emotion

analysis. Twenty thousand three hundred four sentences were collected from 18 different genres of 230 different stories. Barros et al. [84] worked for the automatic emotion detection based classification of Quevedo's Poetry. Emotion detection-related research works can be considered the nearby domain of the proposed research problem. So Hindi and Poetry related works are analyzed, but all the found works [80–84] are working based on the specific set of words where the proposed work is trying to works upon the overall structure.

Furthermore, some international languages-based research works were explored too. Out of those works, a few were added here as per the relevance. Tizhoosh and Dara [85] did an initial in-depth study intending to initiate the research in recognizing poems with several algorithms. Kumar and Minz [86] worked for the classification of English language poems utilizing machine learning. They worked with Support Vector Machine(SVM), Naïve Bayesian(NB), and K-Nearest Neighbor(KNN). They found that SVM was most accurate with 93.25% of accuracy. Jamal et al. [87] performed experimental research work for the Malay poems classification using Support Vector Machine (SVM) and achieved 58.44% accuracy for classifying thematic poetry and 100% accuracy for distinguishing poetry from prose. Alsharif et al. [88] did extensive research work for Arabic poetry classification based on the sentiment analysis and categories of Arabic poetry (Fakhr, Ghazal, Heja, and Retha). Hamidi et al. [89] worked to classify the metre in Persian poetries automatically. He et al. [90] worked with SVM-based classification methods for Chinese poetry styles. Abbasi et al. [91] did sentiment analysis in multiple languages. Manurung et al. [92] tried to work for the development of the model of computational poetry generation of their preliminary work. International language-based poetry research works [85–92] were observed to understand how things are going with the other language's poetry and what adopted approaches. This helped to decide the text-based approach for the current research based on nature of the problem.

As the research work also focuses on automatic metadata generation, metadata-related research works are also explored. Han et al. [93] worked for the rule-based text classification clustering to extract metadata from different documents. In another research work, Yu et al. [94] did research to make a structured syllabus repository with the help of freely available syllabi on the web. They worked for the information recognition to segmentation and classification using automatically extracted metadata. Sagri and Tiscornia [95] researched the metadata for the description of the content in the legal information. They tried to clear the views among the common language and technical legal terminologies. Klavans et al. [96] worked of the image metadata, with the text mining to help identifying, categorizing, and terms disambiguation related to the subject automatically. As the metadata generation will be the outcome of the present research work, metadata-related

research works [93–96] were analyzed from different segments to decide the appropriate set of metadata for the current research.

The research work will generate as much necessary metadata it can, so the Wordnet will play a vital role. Some wordnet-related works and challenges which are still faced, especially by Hindi wordnet, are explored too. Panjwani et al. [97] developed a python programming language-based Application Programming Interface (API) named ‘pyiwn’ to use the Indian Language Wordnets. So far, the work done is recommendable and best concerning the Wordnets for different Indian languages based on the Indo-wordnet. The Hindi wordnet ’pyiwn’ by this research work is adopted in the current research work to know the meanings of words and example uses of those specific words. The meaning of words is still a challenge with the context, and the Indo-wordnet faces that. Also, a similar problem is discussed by Rajendran and Arulmozi [98] in their research work. Polysemy and Homonymy [99] are also conceptual and practical issues for the ambiguities and word sense disambiguation. Bakliwal et al. [100] worked for the betterment of the Hindi wordnet with the synonym and antonym relation, and they developed an annotated corpus of the product reviews of the Hindi language. While working with the Natural Language Processing (NLP) and Computational Linguistics stop words also have their importance, Jha et al. [101] created a hybrid list of the Hindi stop words the comparison of three popular research works after the finalization of the necessary corrections, The hybrid list of Hindi stopwords by this research work is considered in the current research work for Stop Word Filtering.

After the all-research publication, the author tried to find some more relevant aspects in the ancient Indian Knowledge system and other related articles. Meghani [102] discussed the ‘Chhand’ and ‘Duha’ in ‘Charani’ lore from the Saurashtra’s Charan community. Howladar [43] analyzed the importance of the six ‘Vedangas’, ‘Chhand’ is one of those ‘Vedangas’. The verses are also mentioned in Rudrashtadhyayi [103]. ‘Rudrashtadhyayi’ is a part of ‘Yajurveda’[104]. The topic of Vedic verses has been discussed in the second, third, and fourth mantras of the first chapter of ‘Rudrashtadhyayi’ [105]. Knowledge associated with ‘Chhands’ was also found in ‘Agni Purana’(‘अग्नि पुराण’) [46], from Page 328 to 347. Some ancient books published almost before 50 years were found in the archive of the web in which information regarding ‘Chhands’ was observed, And it was found that all the books were coming from a base book named “Chhand Prabhakar” written by Jagannath Prasad [49–51]. These ancient books played a vital role in present research work. In other words, it can be said that the information in the context of Hindi Verses obtained by these books is the heart of this research work. Significant Research works are listed in the 2.1 Literature Review as per the relevance sequence and the respective contribution and conclusion.

Table 2.1: Literature Review Matrix

Sr. No.	Authors	Year	Contribution	Conclusion
1	Kushwah K. K. and Joshi B. K. [56]	2017	Developed Rola Chhand Detection Algorithm	Accuracy 89.83 %, Threshold exceeds the issue
2	Joshi B. K. and Kushwah K. K. [57]	2018	Developed Chaupai Chhand Detection Algorithm, Java	Accuracy 95.03 %, Increase/decrease Matras issue
3	Kaur J. and Saini J. R. [74]	2014	3 Language Family, 8 Indian Language-Based, Opinion Mining	Comparative Study with the English Language, Computational linguistic-based classification is difficult in Indian Languages
4	Kaur J. and Saini J. R. [58]	2017	k-KNN, NB, SVM, and hyperpipes based training and testing	Naïve Bayes best Performing, Hyperpipes least performing
5	Kaur J. and Saini J. R. [59]	2017	10 ML Algorithms tested	SVM Performed best with 58.79% accuracy
6	Kaur J. and Saini J. R. [60]	2018	TF-IDF with k-KNN, NB, SVM, and hyperpipes, Weka toolset	Highest accuracy (71.98%) by SVM
7	Kaur J. and Saini J. R. [62]	2020	Poetry Classifier, TF-IDF with k-KNN, NB, SVM, and hyperpipes, Weka toolset	Highest accuracy (76.02%) by SVM

Table 2.1 continued from previous page

Sr. No.	Authors	Year	Contribution	Conclusion
8	Kaur J. and Saini J. R. [73]	2015	Text Classification Study for Indian Languages	Supervised algorithms perform better. (SVM, NB, ANN and N-gram)
9	Saini J. R. and Kaur J. [80]	2020	Emotion Detection based on 'Navrasa', NB, and SVM	SVM Performed better with 70.02% accuracy
10	Pal K. and Patel B. V. [81]	2020	Developed Model for Poems Classification based on Ras, SVM, and NB	SVM Performs better than NB for Hindi Poems
11	Alsharif et al. [88]	2013	Sentiment Analysis for Arabic Poetry, Naïve Bayes, SVMs, VFI, and Hyperpipes.	Hyperpipes performed best with 79%
12	Bafna P. B. and Saini J. R. [69]	2020	BaSa, Zipf's Law, Prose and Verse, Hindi and Marathi Languages, TF-IDF,	BaSa served better than Zipf's Law
13	Bafna P. B. and Saini J. R. [66]	2020	Hindi Verses Classification, TF-IDF, SVM, Decision tree, Neural network, Naïve Byes	SVM Performed Best can be used for the classification in similar scenarios
14	Bafna P. B. and Saini J. R. [67]	2020	Hindi Verse Class Predictor, Linear Regression, K-nearest neighbor	K-nearest neighbor is better in performance.
15	Bafna P. B. and Saini J. R. [70]	2020	Hindi Verse and Prose, Token Extraction, TF-IDF, BaSa	Common Token extracted from 1.5 million tokens from Hindi Verse and Prose

Table 2.1 continued from previous page

Sr. No.	Authors	Year	Contribution	Conclusion
16	Bafna P. B. and Saini J. R. [68]	2020	Hindi Poetry Classification, Supervised Machine Learning Algorithms, Random forest, naïve byes	Evaluation of the classifiers. Use of TF-IDF
17	Hamidi et al. [89]	2009	Automatic meter classification of Parsian Poetries, SVM	91% of accuracy in three top meters, new approach to know about poems and literature theoretically and practically
18	He et al. [90]	2007	Vector Space Model, SVM-based classification for the style of poetry	88.6% average accuracy in Chinese Poetry classification
19	Kaur J. and Saini J. R. [61]	2017	PuPoCI, Punjabi Poetry Classifier, Linguistics Features and Weighting, TF, TF-IDF, SVM, HP, NB, KNN	SVM is most efficient concerning the time and accuracy
20	Han et al. [93]	2005	Rule-based, Metadata Extraction	89.9% accuracy in bibliographic field extraction
21	Yu et al. [94]	2007	Metadata Extraction, Structured Syllabus	Automatically annotate free syllabus for educational benefits from the internet
22	Sagri M.-T. and Tiscornia D. [95]	2003	Legal Information, Metadata	Content Description based on the legal terminologies

Table 2.1 continued from previous page

Sr. No.	Authors	Year	Contribution	Conclusion
23	Klavans J. L. et al. [96]	2009	Computational Linguistics, Metadata Building, Text Mining, Image Metadata, CLiMB	Thesaurus of terms, names, and geographics. Metadata for images of painting, vernacular, sculpture, and landscape architecture.
24	Abbasi A. et al. [91]	2008	Sentiment Analysis, Text Analysis, Feature Selection, EWGA, SVM	EWGA with SVM delivered 91% accuracy
25	Panjwani R. et al. [97]	2018	Wordnet, Indian Languages, Python, NLTK, IWN	API for IWN through Python for Indian Languages
26	Jha V. et al. [82]	2016	Hindi, Sentiment Analysis, Naive Bayes Classifier, Support Vector Machin,e and Maximum Entropy techniques	Bollywood Hindi movies review opinion mining, 93.59% accuracy through Lexicon Based Classifier
27	Kulkarni A. and Huet G. [75]	2009	Sanskrit, Computational Linguistics, Panini's Astadhyayi	A well-managed way to deal with Sanskrit Computational Linguistics based researches on Panini's Astadhyayi
28	Rajendran S. and Arulmozi S. [98]	2010	Indo-wordnet, Context	Context-based Indo-wordnet proposal
29	Dash N. S. [98]	2010	Polysemy, Homonymy	Highlights ideas about polysemy and homonymy, Discussed some related unsolved problems.

Table 2.1 continued from previous page

Sr. No.	Authors	Year	Contribution	Conclusion
30	Abbas M. R. and Asif K. H. [64]	2020	Computing Prosody, Punjabi, Ghazal, Arud Meter	Automatic process for arud meter detection
31	Pandian et al. [65]	2020	Author identification, Hindi, Poetry, Weka, Feature Selection	Logicboost achieved 75.6757% accuracy
32	Pallavi K. and Mojibur R. [77]	2018	Poetry Translation, PPM, Pragmatic Evolution	Pragmatic technique based model for quality translation of poetry
33	Yadav et al. [78]	2020	Couplets, English to Hindi, Machine Translation, Statistical MT, SMT	English couplets to Hindi statistical machine translation. Also helping to overcome translation of ambiguous sentences
34	Chakrawarti et al. [79]	2020	SMT, Hindi Poetries, Phrase-Based SMT, Translation, PSMT	Translating Hindi Poetries into English using PSMT
35	Joshi B. K. and Kushwah K. K. [71]	2016	Sandhi, Hindi, Rule-based, Word formation, Java	A rule-based approach for Hindi words formulation, Meaning based words formulation not included
36	Bakliwal et al. [100]	2012	Hindi, Product Reviews, Sentiment Analysis, Hindi Subjective Lexicon	Lexicon of adjectives and adverbs using Hindi Wordnet, Corpus of Hindi Product Reviews
37	Kumar V. and Minz S. [86]	2014	Poem Classification, Machine Learning, KNN, NB, SVM	Maximum accuracy 93.25% through SVM

Table 2.1 continued from previous page

Sr. No.	Authors	Year	Contribution	Conclusion
38	Jamal et al. [87]	2012	Malay Poetry, Poetry Classification, Text Classification, SVM, RBF, Linear kernel	Linear kernel performed best with 58.44% accuracy
39	Tizhoosh H. R. and Dara R. A. [85]	2006	Poem Recognition, Fuzzy Logic, Bayesian Approach, Decision Trees	Challenges and Case Studies are discussed. NB performed best with 96.50% accuracy
40	Manurung et al. [92]	2000	Poetry Generation, Preliminary work	Poetry generation is different from normal information generation. Semantics, syntax, and lexis matters more
41	Barros et al. [84]	2013	Automatic Classification, Study on Quevedo's Poetry, Emotion Detection	Study automatic classifier based on emotion detection, different algorithms and approaches used and discussed
42	Hansen S. [63]	2007	Classification Problems, Automatic Programming, ADATE	Automatic Programming using ADATE for the direct need of algebraic data types, auxiliary functions, and recursion for different problems
43	Kumar et al. [83]	2019	Text Corpus, Emotions Analysis, Hindi Stories	20,304 Hindi sentences annotated corpus named BHAAV

Table 2.1 continued from previous page

Sr. No.	Authors	Year	Contribution	Conclusion
44	Chung C. K. and Pennebaker J. W. [76]	2013	Linguistics, Word Count, LIWC	Linguistics based operations on words and sentences using LIWC
45	Gupta P. and Goyal V. [72]	2009	Sandhi, Sandhi-Vicheda, Hindi, Rule-based, Compound Hindi Words	60-80% accuracy based on the different number of rules integration of SANDHI-VICHEDA
46	Meghani J. [102]	2000	Chhand, Duha, Charani Lore	Ancient Information about the Charan, Chhand, and Duha concerning Charani Lore
47	Howladar M. [43]	2016	Vedangs, Chhanda, Shiksha, Kalpa, Vyakaran, Nirukta	Discussed the different Vedangs
48	Henderson J. [13]	2020	Computational Linguistics, Deep Learning	Discussed the growth of CL with deep learning researches in recent times.
49	Gehrmann et al. [11]	2021	Natural Language Generation, NLG	About NLG, different aspects and growth, datasets and tool, with varying tasks over the time
50	Bowman S. R. and Dahl G. E. [9]	2021	Natural Language Understanding, NLU	Discussed the different aspects of the benchmarking in NLU

Table 2.1 continued from previous page

Sr. No.	Authors	Year	Contribution	Conclusion
51	Bender E. M. and Koller A. [8]	2020	Natural Language Understanding, NLG	Discussions about NLU, its different forms, and understanding of data in the current era

After a careful literature review, it was further noted that Hindi has a rich heritage of poetry. Countless verses are written in the Hindi language. The composition of poetry in the Hindi language is from old eras. The wisdom protected following the customs of the formulation of verses is genuinely inspiring. However, not much research work was seen in this area, and this research gap sparked us to carry out this research work.

The simultaneously significant capacity of the scope of applicability and aim, this research work will save the ancient knowledge of extinct Hindi Verses and improve the existing keyword-based searching with multiple aspects of filtered results with the metadata generated using the outcome of this research work. There can be much more applications of automatically generated metadata which can help in innumerous ways such as Digital or e-Libraries, Better Encyclopedia Management, Knowledge Management Systems, and many more.

Especially for the Hindi language and Hindi Verse, this research can be used as better and systematic administration of the Hindi databases through the metadata. In addition, such metadata can help to filter results in numerous ways, like better search results, better management of Hindi Verses-based poems, articles, books, and any other form of digitally well-managed literature.

Chapter 3

Research Methodology

This research work revolves around the Hindi Verses, and that is why the initial work started with the construction rules of 'Chhands'. The composition of the verses has been going on for a long time. There are specific types of rules for the composition of the verses. Due to the lack of composing the verses from long intervals and not compiling their knowledge, there were also contradictions in the rules of creating the verses. And for this reason, no clear classification structure of the verses that are useful from the point of view of quality research could not be found.

The report on the present research or methodology has several parts and subparts. This chapter is divided into several sub-chapters based on the different levels and sequence of the research works accomplished. Initially, a short Section 3.1 Introduction enlightens the research concept, followed by the Section 3.2 Components of Hindi Verse or 'Chhand' discussing different components of Hindi Verse and simplified quantity calculation rules are explained in Section 3.3 Simplified Quantity Calculation Rules. Furthermore, Section 3.4 Hierarchical structure construction of Hindi Metre will cover the hierarchical structure construction part of Hindi Metre. After that, using the systematically constructed and collected rules, the basic core idea of the Metadata Generator's modeling is discussed in Section 3.5 Basic Core Idea of the Metadata Generator Modelling. In the next Section 3.6 Advancement in Core Metadata Generator, the advancement in the core metadata generator is explained, making this automatic metadata generator more effective and reliable with some additional advanced features.

The last Section 3.7 An approach to identify and detect 'Alankars', which is consists of something that was not an earlier part of this research work, but later on, it was decided as an additional part of research and incorporated with the best research efforts.

Different parts of the methodology are entitled as follows:

1. Introduction
2. Components of Hindi Verse or ‘Chhand’
3. Simplified Quantity Calculation Rules - ‘Matra Ganana Niyam’ – (‘मात्रा गणना नियम’)
4. Hierarchical structure construction of Hindi Metre
5. Basic Core Idea of the Metadata Generator Modelling
6. Advancement in Core Metadata Generator
7. An approach to identify and detect ‘Alankars’
8. Summary

Let’s understand each one by one in-depth.

3.1 Introduction

This part reveals how much of the research work-related information is explored so far and how things will be managed further. As observed in the Introduction and Literature review section, it is pretty much clear now that this research spins around Hindi verses. Yet, it is also known now that only Hindi Laukik Verses are going to be considered. Hindi Core Verses are having three major classes 'Matrik Chhands', 'Varnik Chhands' and 'Mukt / Muktak Chhand'. Further sub-classes are there for 'Matrik Chhands' and 'Varnik Chhands' known as 'Even Verses', 'Half-Even Verses', and 'Odd Verses' as discussed in Section 1.2 Core 'Chhand' Types.

During the research findings, the author also realized that in Hindi Verses, no Matrik Odd Verses, Varnik Half-Even Verses, and Varnik Odd Verses are there [52]. However, as no such examples or literature discussed much of it, there is still a provision to integrate if it will be found in upcoming times.

So based on all of this core information and knowledge discussed in the previous sections, different components, rules of calculation, special rules, automatic metadata generator modeling, and its various aspects and features integration will be discussed in the upcoming sections of the research methodology.

3.2 Components of Hindi Verse or ‘Chhand’

Let us know about the components or parts of the ‘Chhand’ [33, 40]:

1. Stanza (‘चरण या पाद’ - ‘Charan’ or ‘Pad’)
2. Characters and Quantity (‘वर्ण और मात्रा’ – ‘Varna’ and ‘Matra’)
3. Flow (‘गति’ - ‘Gati’)
4. Pause (‘यति’- ‘Yati’)
5. End of Charan / Stanza (‘तुक’ - ‘Tuk’)
6. Predefined Sequence of Varnas / Characters (‘गण’ - ‘Ganas’)

3.2.1 Stanza (‘चरण या पाद’ - ‘Charan’ or ‘Pad’)

A Hindi Verse is usually having four Stanzas or ‘Charans’. It is usually one-fourth part of the ‘Chhand’. Each Hindi Verse consists of a fixed number of ‘Matras’ and ‘Varnas’ (Diacritics Count and Character Count). Some Hindi Verses can have more than four stanzas also [49].

There are two main types of the ‘Charan’ [41]:

Even Stanzas (‘सम चरण’ - ‘Sam Charan’)

2^{nd} and 4^{th} stanzas are known as Even Stanzas or ‘Sam Charan’. In case of more than four stanzas all even stanzas are considered.

Odd Stanzas (‘विषम चरण’ - ‘Visham Charan’)

1^{st} and 3^{rd} stanzas are known as Odd Stanzas or ‘Visham Charan’. In the case of more than four stanzas, all Odd stanzas are considered.

Here is an example Hindi Verse (‘Chhand’) named ‘Doha’(‘दोहा’):

‘पढ़े पढ़ाई पारखी, खेल खेलते नैन ।
पंडित जी की लाकड़ी, नटखट मन बेचैन ।।’

Stanzas of the example are as follows:

1st: ‘पढ़े पढ़ाई पारखी,’

2nd: खेल खेलते नैन ।’

3rd: पंडित जी की लाकड़ी,’

4th: नटखट मन बेचैन ।।’

Here 1^{st} and 3^{rd} are considered as Odd Stanzas (‘विषम चरण’ - ‘Visham Charan’), and 2^{nd} and 4^{th} are treated as Even Stanzas (‘सम चरण’ - ‘Sam Charan’).

3.2.2 Characters and Quantity (‘वर्ण और मात्रा’ – ‘Varna’ and ‘Matra’)

The time duration required to pronounce any character (‘वर्ण’ - ‘Varna’) is known as Quantity (‘मात्रा’ - ‘Matra’). Each Hindi Verse is made of the words which are the combination of the characters (‘वर्ण’ - ‘Varna’). There are two primary type of characters (‘वर्ण’ - ‘Varna’) [41]:

‘Laghu Varna’ (‘लघु वर्ण या ह्रस्व वर्ण’)

These are the characters denoted by the symbol (‘।’) while counting quantities counted as one quantity. The reason these are called ‘Laghu’(‘लघु’) is, these characters take a little time to pronounce. The characters, which are known as ‘Laghu Varna’(‘लघु वर्ण’), are as follows:

‘अ’, ‘इ’ (‘ि’), ‘उ’ (‘ु’), ‘ऋ’, (‘ृ’)

‘Guru Varna’ (‘गुरु वर्ण या दीर्घ वर्ण’)

These are the characters denoted by the symbol (‘ऽ’) and counted as two quantities while measuring quantities. Such characters take an additional amount of duration to pronounce hence termed ‘Guru Varna’(‘गुरु वर्ण’). ‘Guru Varna’(‘गुरु वर्ण’) Characters are as follows:

‘आ’ (‘ा’), ‘ई’ (‘ी’), ‘ऊ’ (‘ू’), ‘ए’ (‘े’), ‘ऐ’ (‘ै’), ‘ओ’ (‘ो’), ‘औ’ (‘ौ’), ‘अं’ (‘ं’), ‘अः’ (‘ः’)

‘Plut Varna’ (‘प्लुत वर्ण’)

This is the third kind of character which takes a considerably longer time to pronounce than ‘Guru Varna’(‘गुरु वर्ण’). It is used in musical composition only. It is treated as three quantities in quantities calculation.

As this research work is not dealing with the musical composition related stuff, the third kind of character ‘Plut Varna’(‘प्लुत वर्ण’), which is not used by any of the construction rules of the textual Hindi Verse hence not included in this research work due to irrelevance in the text-based research approach. Apart from this, there are several instances where some aspects, rules, and regulations, and some exceptions where ‘Laghu Varna’(‘लघु वर्ण’) becomes ‘Guru Varna’(‘गुरु वर्ण’) and ‘Guru Varna’(‘गुरु वर्ण’) are treated as ‘Laghu Varna’(‘लघु वर्ण’) too. Such rules will be discussed in Section 3.3 Simplified Quantity Calculation Rules.

3.2.3 Flow (‘गति’ - ‘Gati’)

While reciting any poetry or Hindi Verse, the reciter experiences a flow or rhythm known as Flow (‘गति’ - ‘Gati’) or speed [49].

3.2.4 Pause (‘यति’- ‘Yati’)

While reciting any poetry or Hindi Verse, wherever the reciter takes a small break or stop is known as Pause (‘यति’- ‘Yati’). Some fixed symbols for ‘Yati’ are as follows [41]:

‘ , ’, ‘ | ’, ‘ | | ’, ‘ ? ’, ‘ ! ’

3.2.5 End of Charan / Stanza (‘तुक’ - ‘Tuk’)

The frequencies of the characters at the end of the charan / stanzas (‘चरण या पाद’ - ‘Charan’ or ‘Pad’) is known as ‘Tuk’(‘तुक’ - ‘Tuk’) [40].

3.2.6 Predefined Sequence of Varnas / Characters (‘गण’ - ‘Ganas’)

The ‘Gana’ is usually a sequence of three characters. There are eight types of ‘Gana’. The key to remember the ‘Ganas’ is known as ‘Gana-Sutra’, which is (‘यमाताराजभानसलगा’) [40]. To understand the particular ‘Gana’, choose a character of the initial eight characters of the ‘Gana-Sutra’ sequence. To distinguish particular ‘Gana’, choose three continuous characters, starting from the desired ‘Ganas’ first character.

Table 3.1: ‘Gana-Sutra’(‘गण-सूत्र’)

य	मा	ता	रा	ज	भा	न	स	ल	गा
।	ऽ	ऽ	ऽ	।	ऽ	।	।	।	ऽ

Examples:

Actual ‘Gana-Sutra’ string sequence pairs as shown in Table 3.1 ‘Gana-Sutra’(‘गण-सूत्र’):

य मा ता रा ज भा न स ल गा

। ऽ ऽ ऽ । ऽ । । । ऽ

Example 1:

य मा ता रा ज भा न स ल गा

। ऽ ऽ ऽ । ऽ । । । ऽ

भ + गण = भानस = भगण (ऽ । ।)

Example 2:

य मा ता रा ज भा न स ल गा

। ऽ ऽ ऽ । ऽ । । । ऽ

म + गण = मातारा = मगण (ऽ ऽ ऽ)

As shown in the examples, any ‘Gana’ related information can be retrieved with the help of ‘Gana-Sutra’ string sequence pair of key and symbols.

Table 3.2 Different ‘Ganas’ Symbol with Example represents the uses of the ‘Gana-Sutra’ string and the eight ‘Ganas’ symbols and examples.

In ancient texts, the symptoms of Matrik verses were used to be found somewhere through the Matrik Ganas which were used to be of (‘Matrik Ganas’ - (‘टगण’, ‘ठगण’,

Table 3.2: Different ‘Ganas’ Symbol with Example

S. N.	Ganas	Keys	Symbols	Examples
1	YaGana (‘यगण’)	(‘य’)	‘। ऽ ऽ’	‘जमाना’
2	MaGana (‘मगण’)	(‘मा’)	‘ऽ ऽ ऽ’	‘काकाजी’
3	TaGana (‘तगण’)	(‘ता’)	‘ऽ ऽ ।’	‘पाषाण’
4	RaGana (‘रगण’)	(‘रा’)	‘ऽ । ऽ’	‘माननाा’
5	JaGana (‘जगण’)	(‘ज’)	‘। ऽ ।’	‘जमीन’
6	BhaGana (‘भगण’)	(‘भा’)	‘ऽ । ।’	‘चाकर’
7	NaGana (‘नगण’)	(‘न’)	‘। । ।’	‘मनन’
8	SaGana (‘सगण’)	(‘स’)	‘। । ऽ’	‘कहना’

‘ङगण’, ‘ढगण’, ‘णगण’) – Respective Matras (2,3,4,5,6)) [49]. But nowadays, the poets do not find that much importance of ‘Matrik Ganas’ instead, and they use Matra count and special words. Nowadays, in ‘Matrik Chhands’ if somewhere required, then the normal eight types ‘Ganas’ are only used instead of ‘Matrik Ganas’, which are majorly used for maintaining the sequence of characters in ‘Varnik Chhands.’ calculation.

3.3 'Matra Ganana Niyam' – ('मात्रा गणना नियम')

Many Hindi poetry rules were tested and validated through manual calculations to authenticate the appropriate construction rules of Hindi Verse. The manual analysis was done for the fact check and to figure out the correctness of the various regulations collected from the different online and offline sources [33, 49, 50].

The rules including Primary Rules, Some Exceptional Rules and Highly Impactful Exceptions mentioned in this section and subsection of this section, are derived from the discussion and pieces of information collected from multiple sources such as recently published books [50], old ancient books [46, 49, 51], internet blogs [37, 38], portals [39–41], websites [42, 52] and and expert's contributions.The rules found from these sources vary, or it can be said that they are contradictory many times. For example, for a Hindi verse named 'Bhujangi', creation rules were different from different sources [41, 106]. After proper manual verification, modeling was done according to the ease of the computational operations.

The research work is designed in such a systematic manner that in case of any rules was not found until now, and later on, if it will be found, it can also be incorporated very quickly with minimal or no efforts. The rules that will be discussed here are only included and considered for the calculation in the automatic metadata generator designing. Possibly in upcoming times, certain new or old rules can be discovered or developed which also can be incorporated easily.

While dealing with the various task related to the Hindi Verse, one must know the basic common rules. It helps one understand the Hindi Verse's formation from the core depth and the metadata generation process efficiently. Initially, one should know about the simplified quantity count practices, which is also known as ('Matra Ganana Niyam' – 'मात्रा गणना नियम'). Before proceeding further, one point to be noted here is that these 'Chhand' rules were searched out, validated, and molded for the first time with the computational linguistics perspective in the NLP domain. It demands a massive amount of time, along with a lot of effort to scrutinize and model something from scratch. Additionally, several special rules were also modeled and organized.

Different rules and situations of quantity calculation are managed in the following groups.

1. Primary Rules
2. Some Exceptional Special Rules
3. Highly Impactful Exceptions

3.3.1 Primary Rules

- 'Hrasva Vowels' ('ह्रस्व स्वर') ('अ', 'इ', 'उ', 'ऋ') are treated 'Laghu Varna' ('लघु वर्ण या ह्रस्व वर्ण') counted as one quantity only.
- 'Dirgh Vowels' ('दीर्घ स्वर') ('आ', 'ई', 'ऊ', 'ए', 'ऐ', 'ओ', 'औ', 'अं') are treated 'Guru Varna' ('गुरु वर्ण या दीर्घ वर्ण') counted as two quantities only.
- Consonants ('क, 'ख', 'ग', 'घ', 'ड़', 'च', 'छ', 'ज', 'झ', 'ञ', 'ट', 'ठ', 'ड', 'ढ', 'ण', 'त', 'थ', 'द', 'ध', 'न', 'प', 'फ', 'ब', 'भ', 'म', 'य', 'र', 'ल', 'व', 'श', 'ष', 'स', 'ह') are treated like 'Laghu Varna' ('लघु वर्ण या ह्रस्व वर्ण') and considered as one quantity only.
  Examples:

  'कमल'=111, 'चमन'=111, 'करतब'=1111
- Suppose 'Hrasva Vowel Diacritic' (' ह्रस्व स्वर मात्राएँ') ('ि', 'ु', 'ृ') are applied on any of the consonants. In that case, it does not affect quantity count and considers as 'Laghu Varna' ('लघु वर्ण या ह्रस्व वर्ण') so calculated as one quantity.

  Examples:
  'मिलन'=111, 'दिवस'=111, 'सुन'=11, 'झुनझुन'=1111, 'ऋषि'=11
- If 'Dirgh Vowel Diacritic' ('दीर्घ स्वर मात्राएँ') ('ा', 'ी', 'ू', 'े', 'ै', 'ो', 'ौ', 'ं', 'ः') are applied on any of the consonants than it is considered as 'Guru Varna' ('गुरु वर्ण या दीर्घ वर्ण') and counted as two quantities instead of one quantity of consonant.
  Examples:

  'माला'=22, 'सजाना'=122, 'किला'=12, 'नीला'=22, 'मोर'=21, 'वंदन'=211, 'पहना'=112

3.3.2 Some Exceptional Special Rules

- There is a precise rule for 'Anunasik'('अनुनासिक') or 'Ardh Chandra-Bindu'('अर्ध चंद्र-बिंदु')('ँ'). If any of these applied with any consonant which is usually considered as 'Laghu Varna' ('लघु वर्ण या ह्रस्व वर्ण') and if not used with any 'Dirgh Vowel Diacritic' ('दीर्घ स्वर मात्राएँ'), then it is treated as 'Laghu Varna' ('लघु वर्ण या ह्रस्व वर्ण') only and quantity is considered one. If used with the 'Dirgh Vowel Diacritic' ('दीर्घ स्वर मात्राएँ') than it is treated as 'Guru Varna' ('गुरु वर्ण या दीर्घ वर्ण') and counted as two quantities.
  Examples:

  'चाँद'=21, 'माँ'=2, 'हँसी'=12

- Joint Characters or Ligatures at the starting of the word is treated as 'Laghu Varna' ('लघु वर्ण या ह्रस्व वर्ण') and one quantity is considered.
  Examples:

  'न्याय'=21, 'प्रयास'=121, 'ज्वर'=11

- Joint Characters or Ligatures at the initial of the word accompanying with 'Dirgh Vowel Diacritic' ('दीर्घ स्वर मात्राएँ') is considered as 'Guru Varna' ('गुरु वर्ण या दीर्घ वर्ण'), Half Character's worth becomes zero. Hence, the total calculated value's worth becomes two quantities only.
  Examples:

  'ध्यान'=21, 'ज्ञान'=21, 'भ्राता'=22

- If any 'Laghu Varna' ('लघु वर्ण या ह्रस्व वर्ण') occurred before ligature, then it is considered as 'Guru Varna' ('गुरु वर्ण या दीर्घ वर्ण') and counted as two quantities instead of one.
  Examples:

  'चक्षु'=21, 'सत्य'=21, 'वृक्ष'=21, 'गर्भ'=21, 'विनम्र'=221, 'अध्यक्ष'=221

- If any 'Guru Varna' ('गुरु वर्ण या दीर्घ वर्ण') occurred before ligature, then it is considered as 'Guru Varna' ('गुरु वर्ण या दीर्घ वर्ण') and counted as two quantities.
  Examples:

  'भास्कराचार्य'=21221, 'प्राप्तांक'=221, 'आत्मा'=22

- If 'ह' is the following letter after the ligature including 'Dirgh Vowel's Diacritics' ('दीर्घ स्वर मात्राएँ') is applied, then half character's value becomes zero.
  Examples:

  'तुम्हारे'=122, 'मल्हार'=121, 'कन्हैया'=121

- If ‘ह’ is the following letter after the ligature including ‘Hrasva Vowel Diacritic’ (‘ह्रस्व स्वर मात्राएँ’) is applied, then quantity calculation rules will be changed.
  Examples: ‘अल्हड़’=211, ‘दुल्हन’=211

3.3.3 Highly Impactful Exceptions

- Several times in the last of the stanzas, if 'Laghu Varna' ('लघु वर्ण या ह्रस्व वर्ण') is present, It is considered as 'Guru Varna' ('गुरु वर्ण या दीर्घ वर्ण') and vice versa on the pronunciation's basis.
- Some writers add or remove the diacritics to sustain the flow as per their own decision and not as per the Hindi Verse rules.
- Some creators take references to existing Hindi Verse's construction rules but do not follow the practice entirely.
- Badly formatted and Inclusion of Junk letters added by evading Hindi Verse construction rules. Like using irrelevant special characters, emojis, characters from other languages etc.

3.4 Hierarchical structure construction of Hindi Metre

Whenever research work is carried out, there is a need for systematic information related to that research work. In this research work, efforts were made to find some similar systematic details and knowledge with relevant aspects. While collecting the various information, information was collected from with the help of recently published books [50], old ancient books [49, 46, 51], internet blogs [37, 38], portals [39–41], websites [42, 52], handwritten notes, etc, but it was either contradictory or incomplete or was not enough to be used. When the actual research work was started, there was no source of quality information or no completely structured information regarding all the found Hindi Verse hierarchy that could be used in research work in a very systematic way.

To systemize the already found rules and example-based knowledge and information, continuing the standard approach based on the Hindi Metre is divided into several classes. The core structure coming from the ages is the same, and different levels are incorporated in the same for the better hierarchical structure. Let's see the core structure first, so it can be understood how the other Hindi Verses were added in the respective classes.

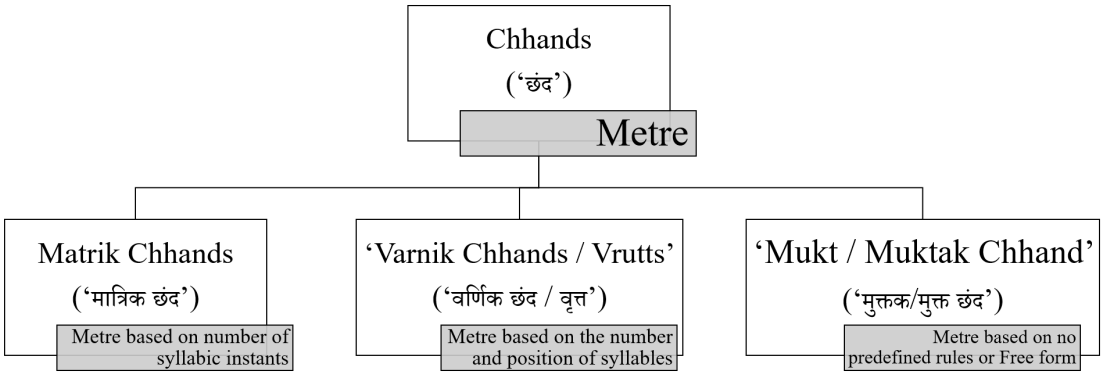


Figure 3.1: Core Hindi Metre Classification

Figure 3.1 is derived from the discussions of the different sources [33, 45, 49–51], represents the Core Hindi Metre Classification which incudes primary classes of the Hindi 'Chhands', also known as Metre. 'Chhands' are divided into three introductory classes, which are:

1. ‘Matrik Chhands’ (‘मात्रिक छंद’)
2. ‘Varnik Chhands / Vrutts’ (‘वर्णिक छंद / वृत्त’)
3. ‘Mukt / Muktak Chhand’ (‘मुक्तक/मुक्त छंद’)

Metre based on the number of syllabic instants is known as the ‘Matrik Chhand’, Metre, based on the number and position of the syllables known as ‘Varnik Vrutt / Chhand’. The remaining Metre for which no specific rules are predefined or, in other words, the Metre wrote in the free form is known as that ‘Mukt / Muktak Chhand’.

These classes are further divided into several parts, which are based on the different construction rules. Classes divided into subclasses are divided to mapping in upcoming research related to other processes and data-related operations. Figure 3.2 showing the subclasses of the introductory classes. It can be observed that the ‘Matrik Chhand’ and ‘Varnik Vrutt’ are having subclasses, but ‘Muktak / Mukt Chhand’ is not having further bifurcation. These are the level which is predefined and coming up from the long back. Still, the problem arises after this level. As for research study and implementation, the core ‘Chhand’ types need to manage in these specific classes, but that was not readily available.

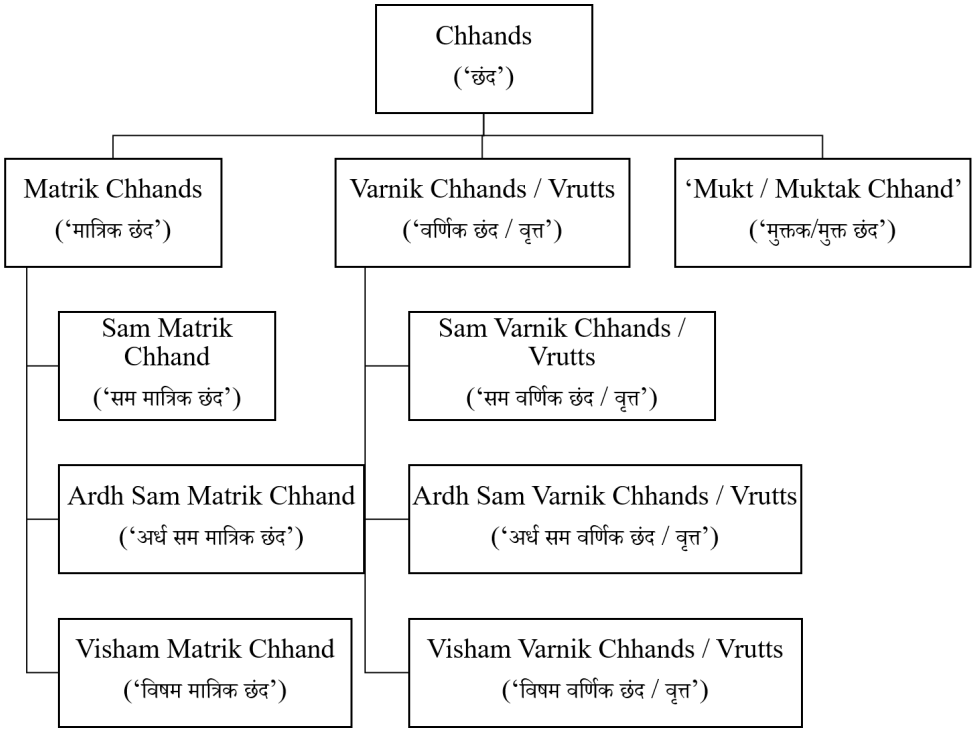


Figure 3.2: Core Sub Classes of Primary Classes of Hindi Metre

Figure 3.2 which is made up based on the explanations and discussions at several sources [33, 45, 49–51], showing the core sub classes of primary classes of Hindi Metre.

Different ‘Chhand’ examples of the each class mentioned in Figure 3.2 were manually analyzed and identified by the author based on the respective rules to construct the hierarchical structure of Hindi Metre. The manual analysis of the examples was required to check that the particular example fits in the specific class/subclass or not based on the different ’Chhand’ rules. If the example got fit into any rules, that was considered part of that class/subclass. Else, it was considered as ‘Mukt / Muktak Chhand’. Another reason behind the manual analysis is that no such tool or way is available to perform this task automatically.

The hierarchically structured list is representing the different ‘Chhand’ included in the various categories, is generated based on such meaning pieces of information collected from different sources such as recently published books [50], old ancient books [49, 46, 51], internet blogs [37, 38, 107], portals [39–41], websites [42, 52].

1. **Matrik Chhand (‘मात्रिक छंद’)**

    1.1. *Sam Matrik Chhand (‘सम मात्रिक छंद’)*

        1.1.1. Rola (‘रोला’)

        1.1.2. Harigitika / Hargitika (‘हरिगीतिका / हरगीतिका’)

        1.1.3. Chaupai (‘चौपाई’)

        1.1.4. Ullala (Sam Matrik) (‘उल्लाला(सममात्रिक)’)

        1.1.5. Ahir (‘अहीर’)

        1.1.6. Tomar (‘तोमर’)

        1.1.7. Tribhangi (‘त्रिभंगी’)

        1.1.8. Gitika (Chanchari / Charchari) (‘गीतिका (चंचरी / चर्चरी)’)

        1.1.9. Shakti (‘शक्ति’)

        1.1.10. Manoram (‘मनोरम’)

        1.1.11. Piyush Varsh (‘पीयूष वर्ष’)

        1.1.12. Sagun (‘सगुण’)

        1.1.13. Sindhu (‘सिन्धु’)

        1.1.14. Bihari (‘बिहारी’)

        1.1.15. Digpal (‘दिगपाल’)

        1.1.16. Shuddh Geeta (‘शुद्ध गीता’)

        1.1.17. Gagnagna (‘गगनांगना’)

        1.1.18. Shankar (‘शंकर’)

        1.1.19. Nishachal (‘निश्चल’)

        1.1.20. Sar (‘सार’)

1.1.21. Lavni (‘लावणी’)

1.1.22. Madhumalti (‘मधुमालती’)

1.1.23. Vijat (‘विजात’)

1.1.24. Aansu (‘आँसू’)

1.1.25. Kamroop (‘कामरूप’)

1.1.26. Chancharik / Haripriya (‘चंचरीक/हरिप्रिया’)

1.1.27. Chaupaiya (‘चौपइया’)

1.1.28. Nidhi (‘निधि’)

1.1.29. Marhatha (‘मरहठा’)

1.1.30. Ras (‘रास’)

1.1.31. Vidhata (‘विधाता’)

1.2. *Ardh Sam Matrik Chhand (‘*अर्धसम मात्रिक छंद’*)*

1.2.1. Doha (‘दोहा’)

1.2.2. Sortha (‘सोरठा’)

1.2.3. Ullala (Ardh Sam Matrik) (‘उल्लाला(अर्द्ध सममात्रिक)’)

1.2.4. Barvai (‘बरवै’)

1.2.5. Roopmala / Madan (‘रूपमाला / मदन’)

1.2.6. Tantak (‘ताटंक’)

1.2.7. Kukubh (‘कुकुभ’)

1.2.8. Veer / Alha / Matrik Savaiya (‘वीर / आल्हा / मात्रिक सवैया’)

1.2.9. Muktamani (‘मुक्तामणि’)

1.2.10. Sarsi / Kabir / Samundar (‘सरसी / कबीर / समुंदर’)

1.2.11. Udiyana (‘उड़ियाना’)

1.2.12. Janak (‘जनक’)

1.3. *Visham Matrik Chhand (‘*विषम मात्रिक छंद’*)*

1.3.1. Kundliya (‘कुंडलिया’)

1.3.2. Chhappay (‘छप्पय’)

2. **‘Varnik Chhands / Vrutts’ (‘**वर्णिक छंद / वृत्त’**)**

2.1. *‘Sam Varnik Chhands / Vrutts’ (‘*सम वर्णिक छंद / वृत्त*’)*

2.1.1. Indravajra (‘इन्द्रवज्रा’)

2.1.2. Upendravajra (‘उपेन्द्रवज्रा’)

2.1.3. Vasanttilka (‘वसंततिलका’)

2.1.4. Malini (‘मालिनी’)

2.1.5. Mandakranta (‘मन्दाक्रान्ता’)
2.1.6. Vanshasth (‘वंशस्थ’)
2.1.7. Drutvilambit (‘द्रुतविलम्बित’)
2.1.8. Shikhrini (‘शिखरिणी’)
2.1.9. Totak / Trotak (‘तोटक/त्रोटक’)
2.1.10. Matgayand Savaiya (‘मत्तगयन्द सवैया’)
2.1.11. Shardul Vikridit (‘शार्दुल विक्रीडित’)
2.1.12. Pramanika (‘प्रमाणिका’)
2.1.13. Swagata (‘स्वागता’)
2.1.14. Bhujangi (‘भुजंगी’)
2.1.15. Dodhak (‘दोधक’)
2.1.16. Chanchala (‘चंचला’)
2.1.17. Shalini (‘शालिनी’)
2.1.18. Bhujangprayag (‘भुजन्गाप्रयाग’)
2.1.19. Panchchamar (‘पंचचामर’)
2.1.20. Madira Savaiya (‘मदिरा सवैया’)
2.1.21. Sumukhi Savaiya (‘सुमुखी सवैया’)
2.1.22. Sundari / Madhvi Savaiya (‘सुंदरी / माधवी सवैया’)
2.1.23. Kirit Savaiya (‘किरीट सवैया’)
2.1.24. Chakor Savaiya (‘चकोर सवैया’)
2.1.25. Sukhi Savaiya (‘सुखी सवैया’)
2.1.26. Arsat Savaiya (‘अरसात सवैया’)
2.1.27. Arvind Savaiya (‘अरविंद सवैया’)
2.1.28. Durmil Savaiya (‘दुर्मिल सवैया’)
2.1.29. Lavnglata Savaiya (‘लवंगलता सवैया’)
2.1.30. Mukhara Savaiya (‘मुक्तहरा सवैया’)
2.1.31. Vam Savaiya (‘वाम सवैया’)
2.1.32. Mod Savaiya (‘मोद सवैया’)
2.1.33. Krupan Dhanakshari (‘कृपाण घनाक्षरी’)
2.1.34. Sur Dhanakshari (‘सूर घनाक्षरी’)
2.1.35. Damru Dhanakshari (‘डमरू घनाक्षरी’)
2.1.36. Manharan Dhanakshari (‘मनहरण घनाक्षरी’)
2.1.37. Janharan Dhanakshari (‘जनहरण घनाक्षरी’)
2.1.38. Dev Dhanakshari (‘देव घनाक्षरी’)
2.1.39. Vijya Dhanakshari (Kamini) (‘विजया घनाक्षरी (कामिनी)’)

2.1.40. Jalharan Dhanakshari (‘जलहरण घनाक्षरी’)

2.1.41. Asbandha (‘असबंधा’)

2.1.42. Indira (‘इंदिरा’)

2.1.43. Kanthi (‘कण्ठी’)

2.1.44. Kanakmajari (‘कनक मंजरी’)

2.1.45. Kusumasamudita (‘कुसुमसमुदिता’)

2.1.46. Gath (‘गाथ’)

2.1.47. Giridhari (‘गिरिधारी’)

2.1.48. Ghanshyam (‘घनश्याम’)

2.1.49. Chandrika (‘चन्द्रिका’)

2.1.50. Tilka (‘तिलका’)

2.1.51. Dhar (‘धार’)

2.1.52. Dhuni (‘धुनी’)

2.1.53. Neel (‘नील’)

2.1.54. Panktika (‘पंक्तिका’)

2.1.55. Padhymala (‘पद्यमाला’)

2.1.56. Pavan (‘पवन’)

2.1.57. Paavan (‘पावन’)

2.2. *‘Ardh Sam Varnik Chhands / Vrutts’ (‘*अर्धसम वर्णिक छंद / वृत्त*’)*

2.3. *‘Visham Varnik Chhands / Vrutts’ (‘*विषम वर्णिक छंद / वृत्त*’)*

3. **Mukt / Muktak Chhand (‘**मुक्तक/मुक्त छंद**’)**

So, this is the final systematic structure on which the research work was carried out. In ‘Ardh Sam Varnik Vrutt / Chhand’ and ‘Visham Varnik Vrutt / Chhand’, not much information, types, or examples were found, but the conceptual presence related to these was seen. So, keeping in mind that in the future, in case of any further development, if anything is found, then it will be easier for integration with this balanced approach in this structure. ‘Mukt / Muktak Chhand’ is a free form class so that no further subclasses or types exist.

3.5 Basic Core Idea of the Metadata Generator Modelling

This is the central core part of this research work. After creating the well-managed hierarchical structure of Hindi' Chhands' based on the different rules and examples, it's time to model each of the rules associated with Hindi Metre's construction. Let's understand it with a clear flow chart first.

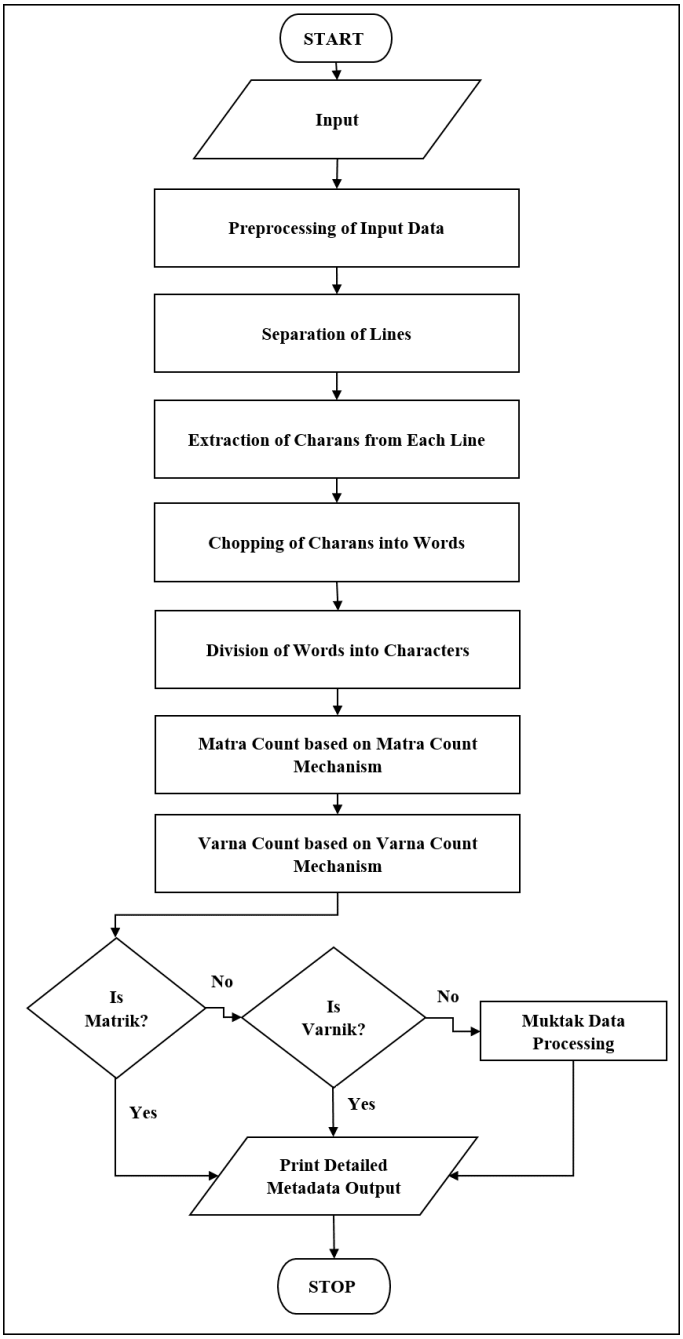


Figure 3.3: Core Flow of the Metadata Generator

The steps and their order represented in Figure 3.3 and basic algorithm is generated based on the discussion presented in Section 3.3 Simplified Quantity Calculation Rules and Section 3.4 Hierarchical structure construction of Hindi Metre.

Figure 3.3 representing the Core Flow of the Metadata Generator for Hindi Poetry. Here it can be seen that initially input gets processed through the cleaning and preprocessing operation. For preprocessing and cleaning, the basic trimming operations were performed on the provided input. Unnecessary extra spaces, comma, full stop, double full stop, and signs were removed because they are not required in further stages and can impact the processing of upcoming stages if not performed properly. After that various separation operation takes place in which line, charan, words (Tokenization), characters get separated. Furthermore, based on the 'Matra Calculation' and 'Varna Count' mechanism, the different calculations required for the upcoming process identification and detection occur.

From the NLP perspective, the tokenization is taking place in the stage where 'Charans' are getting chopped into words as it was required. Therefore, stemming and Lemmatization were not needed based on the nature of the problem and the upcoming steps.The separation of the words into the characters is required because the different rules are based on the character level calculations discussed in Section 3.3 Simplified Quantity Calculation Rules. Also, to decide the classes or subclasses, the count of characters is required.

There were so many challenges needs to be taken care of while dealing with characters. The character-related operation is complex when you deal with UTF-8 based Devanagari script. Hindi has diacritics and ligatures, which require special or additional efforts while performing different functions. Along with that, different primary and exceptional rules mentioned in Section 3.3 Simplified Quantity Calculation Rules were also handled. Human errors like spell mistakes and signs at the wrong place also affect the respective counts. Due to such errors, faulty input gets identified into the 'Mukt / Muktak Chand' class instead of its original class.

The calculated quantities and sum are stored in the memory for further measures. In the next conditional check, all the data based on the different calculations are passed to the various methods to identify and detect the specific 'Chhand' that is pre-modeled with its own unique rules. It is a bottom-up approach that helps in the quicker mapping of parent and grand-parent classes of the detected 'Chhand'.The significant benefit of using the bottom-up approach here is that classification will be correct and easy compared to the top-down approach in which it becomes more complex. Therefore, due to the ease and accuracy, the bottom-up approach was chosen over the top-down approach.

To understand it better, let us go through the classification of a vehicle. For example, suppose a cycle needs to be classified, which can be done in two ways.

- The first way or a top-down approach could be, detect it as a Vehicle initially, then it gets identified as a Two-Wheeler. Then, it gets recognized as a Non-Motor Vehicle, and at last, it gets seen as a cycle.
- In a second way or a bottom-up approach, the Cycle gets detected first, which instantly goes into a Non-Motor Vehicle class. Then, further, It is mapped into a Two-Wheeler class, and at last, it gets the final Vehicle classification.

So ultimately, the comparatively second or bottom-up approach seems to be more straightforward as per the nature of the problem because once cycle get detected first. It gets easier to map with the classes for which information is already known. Hence the same thing goes with the detection of the different verses too. The root-level Hindi Metre gets detected first, and once it gets caught, the respective parent and grand-parent types are already stored with a unique mapping mechanism through which all the three-level gets identified.

Metadata with all the necessary information such as ‘Chhand’ Class, ‘Chhand’ Parent Class, and ‘Chhand’ Grand Parent class and that the statical and quantitative metadata for the computational linguistics also added in the metadata. In the additional data, it includes Line Count, Charan/Staza Count, Character (Including Spaces) Count, Character (After Cleaning and Excluding Spaces) Count, Diacritics Count, Guru Character Count, Laghu Character Count, Half Character Count, Symbolic Representation, Charan-wise Matra Count, Charan-wise Matra Sum, Charan-wise Varna, Charan-wise Varna Sum. One can understand it with the basic algorithm consisting of several steps.

Let us understand the Basic Algorithm of Automatic Metadata Generator.

Basic Algorithm of Automatic Metadata Generator

Step-1: Start

Step-2: Take Input UTF-8 Based Text Data (Poem, Verse, Stanza, etc.)

Step-3: Cleaning including Preprocessing of Data from Step-2

Step-4: Lines Splitting (Split each line)

Step-5: 'Charans' disjoining of every split line from Step-4

Step-6: Disjointing words from every 'Charan' from Step-5

Step-7: Chopping Characters of all words (Tokenization).

Step-8: Store 'Charan Wise Matra' Calculations and Sum of 'Matras' based on the predefined rules of Matra Calculation ('Matra Gadana')

Step-9: Store the order of characters ('Varna'), Character ('Varna') count based on 'Matra' and 'Varna' Count mechanism.

Step-10: Momentarily, Process the systematic data through the different set of rules for the specific group of 'Matrik Verses' while comparing if any verse's set of regulations matched exactly, then store the output with detected verse's type subtype and verse related details. Move to Step-13

Step-11: If not found in 'Matrik Verses', then recheck the systematic data with all the distinct particular rules of 'Varnik Verses', while comparing if any verse's set of regulations matched exactly, then store output with detected verse's type, subtype, and verse related details. Move to Step-13

Step-12: If not detected in any of the 'Matrik Verses' or 'Varnik Verses' set of rules, set output with the type of 'Mukt / Muktak Verses'

Step-13: Represent the final output with the suitable form, including every additional metadata information associated with Inputted Data

Step-14: Stop

That is the way automatic metadata generator works, and the metadata generation takes place. To understand it with a more clear aspect, let us know with several example inputs and outputs. The information provided to the automatic metadata generator for the processing is as follows.

3.5.1 'Matrik Chhand' Detection

'ऐसी वाणी बोलिए, मन का आपा खोय ।
औरन को शीतल करे, आपहु शीतल होय ।।'

This is one of the very popular 'Dohe'('दोहे') written by a well-known poet named Kabir Das Ji. While providing this Doha as an input to the automatic metadata generator, it is first inputted in UTF-8 standard encoded text. Various separations take place in different processes after the initial cleaning process. Let's see each operation one by one:

Separation of Lines: (2 Lines)

Line 1: 'ऐसी वाणी बोलिए, मन का आपा खोय ।'

Line 2: 'औरन को शीतल करे, आपहु शीतल होय ।।'

Separation of Stanza / 'Charan': (4 Stanzas)

Stanza 1: 'ऐसी वाणी बोलिए'

Stanza 2: ' मन का आपा खोय'

Stanza 3: 'औरन को शीतल करे'

Stanza 4: 'आपहु शीतल होय'

Separation of Words: (14 Words)

1. 'ऐसी', 2. 'वाणी', 3. 'बोलिए', 4. 'मन', 5. 'का', 6. 'आपा', 7. 'खोय'

8. 'औरन', 9. 'को', 10. 'शीतल', 11. 'करे', 12. 'आपहु', 13. 'शीतल', 14. 'होय'

Separation of Words (Stanza Wise):

(i) 1. 'ऐसी', 2. 'वाणी', 3. 'बोलिए'

(ii) 1. 'मन', 2. 'का', 3. 'आपा', 4. 'खोय'

(iii) 1. 'औरन', 2. 'को', 3. 'शीतल', 4. 'करे'

(iv) 1. 'आपहु', 2. 'शीतल', 3. 'होय'

Character Wise Separation: (45 Characters)

‘ऐ’, ‘स’, ‘◌ी’, ‘व’, ‘◌ा’, ‘ण’, ‘◌ी’, ‘ब’, ‘◌ो’, ‘ल’, ‘◌ि’, ‘ए’, ‘म’, ‘न’, ‘क’, ‘◌ा’, ‘आ’, ‘प’, ‘◌ा’, ‘ख’, ‘◌ो’, ‘य’,‘औ’, ‘र’, ‘न’, ‘क’, ‘◌ो’, ‘श’, ‘◌ी’, ‘त’, ‘ल’, ‘क’, ‘र’, ‘◌े’, ‘आ’, ‘प’, ‘ह’, ‘◌ु’, ‘श’, ‘◌ी’, ‘त’, ‘ल’, ‘ह’, ‘◌ो’, ‘य’

These separated parts are processed through several methods for the ‘Matra Calculation’ and ‘Varna Count’ after these separations. These calculations are based on the predefined set of rules. Let’s understand the stanza wise ‘Matra’ allocation and calculation.

Table 3.3: Stanza 1 and 2 Matra Calculation for Matrik Verses

Stanza 1			**Stanza 2**			
‘ऐसी’	‘वाणी’	‘बोलिए’	‘मन’	‘का’	‘आपा’	‘खोय’
22	22	212	11	2	22	21
2+2	2+2	2+1+2	1+1	2	2+2	2+1
4	4	5	2	2	4	3
4+4+5 = 13			**2+2+4+3 = 11**			

Based on the rules of ‘Matra Calculation’, the respective allocation for the individual character is decided computationally, and summation of the allocated quantities and the allocation is stored for the different decision-making process. At the same time, identification and classification of the ‘Chhand’ take place. Now stanza-wise quantity sum is available, which is [13,11,13,11] as shown in Table 3.3 and 3.4. This sum now gets processed through the core identification methods that are already modeled with the predefined set of rules associated with the specific ‘Chhand’. The exact criteria need to be fulfilled for the respective ‘Chhand’ to get detected.

The ‘Chhand’ is already having a mapping with the Parent and Grand-Parent classification. Once the root ‘Chhand’ gets detected, through the mapping of the detected ‘Chhand’, the Parent’s information and Grand-Parent classes can be fetched quickly.

Table 3.4: Stanza 3 and 4 Matra Calculation for Matrik Verses

Stanza 3				**Stanza 4**		
'औरन'	'को'	'शीतल'	'करे'	'आपहु'	'शीतल'	'होय'
211	2	211	12	211	211	21
2+1+1	2	2+1+1	1+2	2+1+1	2+1+1	2+1
4	2	4	3	4	4	3
4+2+4+3 = 13				**4+4+3 = 11**		

For example, as soon as the 'Chhand' named 'Doha' gets detected as a rule for the 'Doha' says the 'Matra' cunt of odd stanza should be 13 and for even it should be 11, and even stanza needs to finish with the 'Laghu' character which is denoted by 1. All the criteria are getting full filled in the given input, so 'Doha' gets detected. Now 'Doha' is mapped with its parent category, 'Ardh Sam Matrik Chhand', and the Grand-Parent, 'Matrik Chhand'. It is easier to detect Parent and Grand-Parent because of the 'Systematic Hierarchical Structure' constructed initially and discussed in Section 3.4 Hierarchical structure construction of Hindi Metre. These separated parts are processed through several methods for the 'Matra Calculation' and 'Varna Count' after these separations. These calculations are based on a predefined set of rules. Let us understand the stanza-wise 'Matra' allocation and calculation.

3.5.2 'Varnik Chhand / Vrutt' Detection

'जो मैं नया ग्रंथ विलोकता हूँ, भाता मुझे सो नव मित्र सा है ।
देखूँ उसे मैं नित सार वाला, मानो मिला मित्र मुझे पुराना ।।'

The above example is a 'Varnik Chhand' named 'Indravajra Chhand' from a poet Giridhar Sharma. 'Varnik Chhands' is based on the characters ('Varna') and the characters' sequence, while the 'Matrik Chhands' was based on the Quantity ('Matra'). So, the detection of the 'Varnik Chhands' slightly differs from the 'Matrik Chhand'. Initial processes are the same, but there are some significant changes in later stages where actual detection occurs because of the formation rule and regulations of 'Varnik Chhands'.

Like the 'Matrik Chhands', the separation process will be similar for 'Varnik Chhands'. In fact, for any 'Chhand', it's going to be the same. And all the basic operations for all three primary types, 'Matrik', 'Varnik', and 'Mukt / Muktak', occur as soon as the input data is processed. So it becomes easier while processing the data in these classes if the 'Chhand' doesn't get detected in 'Matrik', it can quickly process with the 'Varnik' and 'Matrik'. Based on the provided input, let us quickly go through the cleaning and splitting operation.

Separation of Lines: (2 Lines)

Line 1: जो मैं नया ग्रंथ विलोकता हूँ, भाता मुझे सो नव मित्र सा है ।'

Line 2: 'देखूँ उसे मैं नित सार वाला, मानो मिला मित्र मुझे पुराना ।।'

Separation of Stanza / 'Charan': (4 Stanzas)

Stanza 1: 'जो मैं नया ग्रंथ विलोकता हूँ'

Stanza 2: 'भाता मुझे सो नव मित्र सा है'

Stanza 3: 'देखूँ उसे मैं नित सार वाला'

Stanza 4: 'मानो मिला मित्र मुझे पुराना'

Separation of Words: (24 Words)

1. 'जो', 2. 'मैं', 3. 'नया', 4. 'ग्रंथ', 5. 'विलोकता', 6. 'हूँ', 7.'भाता', 8. 'मुझे', 9. 'सो', 10. 'नव', 11. 'मित्र', 12. 'सा', 13. 'है',

14. ‘देखूँ’, 15. ‘उसे’, 16. ‘मैं’, 17. ‘नित’, 18. ‘सार’, 19. ‘वाला’, 20. ‘मानो’, 21. ‘मिला’, 22. ‘मित्र’, 23. ‘मुझे’, 24. ‘पुराना’

Separation of Words (Stanza Wise):

(i) 1. ‘जो’, 2. ‘मैं’, 3. ‘नया’, 4. ‘ग्रंथ’, 5. ‘विलोकता’, 6. ‘हूँ’,

(ii) 1. ‘भाता’, 2. ‘मुझे’, 3. ‘सो’, 4. ‘नव’, 5. ‘मित्र’, 6. ‘सा’, 7. ‘है’,

(iii) 1. ‘देखूँ’, 2. ‘उसे’, 3. ‘मैं’, 4. ‘नित’, 5. ‘सार’, 6. ‘वाला’,

(iv) 1. ‘मानो’, 2. ‘मिला’, 3. ‘मित्र’, 4. ‘मुझे’, 5. ‘पुराना’

Character Wise Separation: (88 Characters)

‘ज’, ‘ो’, ‘म’, ‘ै’, ‘ं’, ‘न’, ‘य’, ‘ा’, ‘ग’, ‘्’, ‘र’, ‘ं’, ‘थ’, ‘व’, ‘ि’, ‘ल’, ‘ो’, ‘क’, ‘त’, ‘ा’, ‘ह’, ‘ू’, ‘ँ’, ‘भ’, ‘ा’, ‘त’, ‘ा’, ‘म’, ‘ु’, ‘झ’, ‘े’, ‘स’, ‘ो’, ‘न’, ‘व’, ‘म’, ‘ि’, ‘त’, ‘्’, ‘र’, ‘स’, ‘ा’, ‘ह’, ‘ै’, ‘द’, ‘े’, ‘ख’, ‘ू’, ‘ँ’, ‘उ’, ‘स’, ‘े’, ‘म’, ‘ै’, ‘ं’, ‘न’, ‘ि’, ‘त’, ‘स’, ‘ा’, ‘र’, ‘व’, ‘ा’, ‘ल’, ‘ा’, ‘म’, ‘ा’, ‘न’, ‘ो’, ‘म’, ‘ि’, ‘ल’, ‘ा’, ‘म’, ‘ि’, ‘त’, ‘्’, ‘र’, ‘म’, ‘ु’, ‘झ’, ‘े’, ‘प’, ‘ु’, ‘र’, ‘ा’, ‘न’, ‘ा’,

These separated parts are processed through several methods for the ‘Matra Calculation’ and ‘Varna Count’ after these separations. These calculations are based on the predefined set of rules. Let’s understand the stanza-wise ‘Matra’ allocation and calculation, but the focus will be on the ‘Varna Count’ and ‘Varna Sequence’ more because that only matters in ‘Varnik Chhand’ detection.

Table 3.5: Stanza 1 Matra Calculation and Varna Sequence for Varnik Verses

Stanza 1					
‘जो’	‘मैं’	‘नया’	‘ग्रंथ’	‘विलोकता’	‘हूँ’
2	2	12	21	1212	2
2	2	1+2	2+1	1+2+1+2	2
2	2	3	3	6	2
Matra Count :2+2+3+3+6+2= 18					
Varna Sequence : [[2], [2], [1, 2], [2, 1], [1, 2, 1, 2], [2]] = [22122112122]					
Varna Sequence Length : 11					

After the ‘Matra Calculation’ and ‘Varna Count’ and Sequence generation processing, one can now observe that all the three stanzas are having similar ‘Varna Sequence’

[22122112122], which is based on the 'Ganas' and the length of the sequence is 11 for all the stanza as shown in Table 3.5, 3.6, 3.7 and 3.8.

Table 3.6: Stanza 2 Matra Calculation and Varna Sequence for Varnik Verses

Stanza 2						
'भाता'	'मुझे'	'सो'	'नव'	'मित्र'	'सा'	'है'
22	12	2	11	21	2	2
2+2	1+2	2	1+1	2+1	2	2
4	3	2	2	3	2	2
Matra Count : 4+3+2+2+3+2+2= 18						
Varna Sequence : [[2, 2], [1, 2], [2], [1, 1], [2, 1], [2], [2]] = [22122112122]						
Varna Sequence Length : 11						

These all-calculated values pass through the various methods modeled using the specific rule for every unique 'Chhand'. The formation rule of 'Indravajra' says that the 'Varna Sequence' should be 'TaGana', 'TaGana', 'JaGana', 'Guru', and 'Guru', which is consist of the sequence length of 11.
[22122112122] = 11
[221- 'TaGana', 221-'TaGana', '121'-JaGana, 2-'Guru', 2-'Guru'] = 11

Table 3.7: Stanza 3 Matra Calculation and Varna Sequence for Varnik Verses

Stanza 3					
'देखूँ'	'उसे'	'मैं'	'नित'	'सार'	'वाला'
22	12	2	11	21	22
2+2	1+2	2	1+1	2+1	2+2
4	3	2	2	3	4
Matra Count :4+3+2+2+3+4= 18					
Varna Sequence : [[2, 2], [1, 2], [2], [1, 1], [2, 1], [2, 2]] = [22122112122]					
Varna Sequence Length : 11					

The provided input fulfills all the criteria mentioned in the rule of 'Indravajra'. So it gets detected, and as the mapping with the Parent and Grand-Parent is already pre-structured, the Parent 'Sam Varnik Vrutt / Chhand' and Grand-Parent class 'Varnik Vrutt / Chhand' can be mapped quickly. This is how 'Varnik Chhands' gets detected based on the predefined unique rule modeled methods as similar as 'Matrik Chhands'.

Table 3.8: Stanza 4 Matra Calculation and Varna Sequence for Varnik Verses

Stanza 4				
'मानो'	'मिला'	'मित्र'	'मुझे'	'पुराना'
22	12	21	12	122
2+2	1+2	2+1	1+2	1+2+2
4	3	3	3	5
Matra Count :4+3+3+3+5= 18				
Varna Sequence : [[2, 2], [1, 2], [2, 1], [1, 2], [1, 2, 2]] = [22122112122]				
Varna Sequence Length : 11				

This is how 'Varnik Chhand / Vrutt' Detection works.

3.5.3 'Mukt / Muktak Chhand' Detection

It is known how 'Matrik Chhands' and 'Varnik Chhands' get identified and detected. It's time to identify the third primary class, which is 'Mukt/Muktak Chhand'.

'Mukt / Muktak Chhand' are also processed as same as the 'Matrik' and 'Varnik,' but things get different in later parts. So, the initial part of the splitting process and calculation remains the same. Let's understand this with an example quickly. Poet Dr. Kumar Vishwas pen this 'Muktak Chhand' example.

इस उड़ान पर अब शर्मिंदा, में भी हूँ और तू भी है।
आसमान से गिरा परिंदा, में भी हूँ और तू भी है।।
छुट गयी रस्ते में, जीने मरने की सारी कसमें।
अपने-अपने हाल में जिंदा, में भी हूँ और तू भी है।।'

The initial splitting and cleaning data operation will occur as similar to earlier discussed 'Matrik' and 'Varnik' examples. Let's go through this also very quickly.

Separation of Lines: (4 Lines)

Line 1: इस उड़ान पर अब शर्मिंदा, में भी हूँ और तू भी है'

Line 2: 'आसमान से गिरा परिंदा, में भी हूँ और तू भी है'

Line 3: 'छुट गयी रस्ते में, जीने मरने की सारी कसमें'

Line 4: 'अपने-अपने हाल में जिंदा, में भी हूँ और तू भी है'

Separation of Stanza / 'Charan': (8 Stanzas)

Stanza 1: इस उड़ान पर अब शर्मिंदा'

Stanza 2: में भी हूँ और तू भी है'

Stanza 3: 'आसमान से गिरा परिंदा'

Stanza 4: 'में भी हूँ और तू भी है'

Stanza 5: 'छुट गयी रस्ते में'

Stanza 6: 'जीने मरने की सारी कसमें'

Stanza 7: 'अपने-अपने हाल में जिंदा'

Stanza 8: 'में भी हूँ और तू भी है'

Separation of Words: (24 Words)

1. ‘इस’, 2. ‘उड़ान’, 3. ‘पर’, 4. ‘अब’, 5. ‘शर्मिंदा’, 6. ‘में’, 7. ‘भी’, 8. ‘हूँ’, 9. ‘और’, 10. ‘तू’, 11. ‘भी’, 12. ‘है’,

13. ‘आसमान’, 14. ‘से’, 15. ‘गिरा’, 16. ‘परिंदा’, 17. ‘में’, 18. ‘भी’, 19. ‘हूँ’, 20. ‘और’, 21. ‘तू’, 22. ‘भी’, 23. ‘है’,

24. ‘छुट’, 25. ‘गयी’, 26. ‘रस्ते’, 27. ‘में’, 28. ‘जीने’, 29. ‘मरने’, 30. ‘की’, 31. ‘सारी’, 32. ‘कसमें’,

33. ‘अपने-अपने’, 34. ‘हाल’, 35. ‘में’, 36. ‘जिंदा’, 37. ‘में’, 38. ‘भी’, 39. ‘हूँ’, 40. ‘और’, 41. ‘तू’, 42. ‘भी’, 43. ‘है’

Separation of Words (Stanza Wise):

(i) 1. ‘इस’, 2. ‘उड़ान’, 3. ‘पर’, 4. ‘अब’, 5. ‘शर्मिंदा’,

(ii) 1. ‘में’, 2. ‘भी’, 3. ‘हूँ’, 4. ‘और’, 5. ‘तू’, 6. ‘भी’, 7. ‘है’,

(iii) 1. ‘आसमान’, 2. ‘से’, 3. ‘गिरा’, 4. ‘परिंदा’,

(iv) 1. ‘में’, 2. ‘भी’, 3. ‘हूँ’, 4. ‘और’, 5. ‘तू’, 6. ‘भी’, 7. ‘है’,

(v) 1. ‘छुट’, 2. ‘गयी’, 3. ‘रस्ते’, 4. ‘में’,

(vi) 1. ‘जीने’, 2. ‘मरने’, 3. ‘की’, 4. ‘सारी’, 5. ‘कसमें’,

(vii) 1. ‘अपने-अपने’, 2. ‘हाल’, 3. ‘में’, 4. ‘जिंदा’

(viii) 1. ‘में’, 2. ‘भी’, 3. ‘हूँ’, 4. ‘और’, 5. ‘तू’, 6. ‘भी’, 7. ‘है’,

Character Wise Separation: (137 Characters)

‘इ’, ‘स’, ‘उ’, ‘ड’, ‘ड़’, ‘ा’, ‘न’, ‘प’, ‘र’, ‘अ’, ‘ब’, ‘श’, ‘र’, ‘्’, ‘म’, ‘ि’, ‘ं’, ‘द’, ‘ा’, ‘म’, ‘े’, ‘ं’, ‘भ’, ‘ी’, ‘ह’, ‘ू’, ‘ँ’, ‘औ’, ‘र’, ‘त’, ‘ू’, ‘भ’, ‘ी’, ‘ह’, ‘ै’, ‘आ’, ‘स’, ‘म’, ‘ा’, ‘न’, ‘स’, ‘े’, ‘ग’, ‘ि’, ‘र’, ‘ा’, ‘प’, ‘र’, ‘ि’, ‘ं’, ‘द’, ‘ा’, ‘म’, ‘े’, ‘ं’, ‘भ’, ‘ी’, ‘ह’, ‘ू’, ‘ँ’, ‘औ’, ‘र’, ‘त’, ‘ू’, ‘भ’, ‘ी’, ‘ह’, ‘ै’, ‘छ’, ‘ु’, ‘ट’, ‘ग’, ‘य’, ‘ी’, ‘र’, ‘स’, ‘्’, ‘त’, ‘े’, ‘म’, ‘े’, ‘ं’, ‘ज’, ‘ी’, ‘न’, ‘े’, ‘म’, ‘र’, ‘न’, ‘े’, ‘क’, ‘ी’, ‘स’, ‘ा’, ‘र’, ‘ी’, ‘क’, ‘स’, ‘म’, ‘े’, ‘ं’, ‘अ’, ‘प’, ‘न’, ‘े’, ‘-’, ‘अ’, ‘प’, ‘न’, ‘े’, ‘ह’, ‘ा’, ‘ल’, ‘म’, ‘े’, ‘ं’, ‘ज’, ‘ि’, ‘ं’, ‘द’, ‘ा’, ‘म’, ‘े’, ‘ं’, ‘भ’, ‘ी’, ‘ह’, ‘ू’, ‘ँ’, ‘औ’, ‘र’, ‘त’, ‘ू’, ‘भ’, ‘ी’, ‘ह’, ‘ै’

After the separation and initial cleaning operation, the separated data go through all the predefined rule-based modeled methods of all ‘Matrik Chhands’ and ‘Varnik Chhands’. But as ‘Mukt / Muktak Chhand’ does not follow any specific rules and always written as free form writing, none of the modeled ‘Chhands’ gets detected, then it is considered ‘Mukt / Muktak Chhand’. There are some advancements regarding the ‘Mukt / Muktak Chhand’ after detection, which will be discussed in the next Section 3.3 Simplified Quantity Calculation Rules. The initial splitting and cleaning data operation will occur as similar to earlier discussed ‘Matrik’ and ‘Varnik’ examples. Let’s go through this also very quickly.

3.6 Advancement in Core Metadata Generator

After developing the Core Metadata Generator, it was felt that it could be straightened more by incorporating some latest advancements to understand and learn about Hindi poetry in-depth. The improvements incorporated are divided into several points for better understanding.

1. Advance ‘Mukt / Muktak’ Identification and Detection
2. Stop Words Filtering
3. Populating Meaning and Example of Word with Wordnet Integration
4. Suggesting Examples of the identified ‘Chhand’
5. Additional Several Utilities

Let us go through all of these one by one.

3.6.1 Advance ‘Mukt / Muktak’ Identification and Detection

Advance ‘Mukt / Muktak’ Identification and Detection comes when any input gets detected into the ‘Mukt / Muktak Chhand’ class at the identification and detection time. Now the thing is that as it is already known that the ‘Mukt / Muktak Chhand’ is written in free form, and usually no rules such as ‘Matrik Chhand’ or ‘Varnik Chhand’ are followed. To understand the ‘Mukt / Muktak Chhands’ more, the research work is trying to figure out that if any writer tried writing the write-up using any combination of any ‘Chhand’ rules. In Advanced ‘Mukt / Muktak’ identification and detection, the actual input gets divided into several parts in different combinations of lines and stanzas. These split lines and stanzas are treated as input until the combination gets processed through the automatic metadata generator.

If any of these combinations get detected through any pre-defined rule-based modeled methods of ‘Matrik Chhand’ and ‘Varnik Chhand’. They are added to the metadata. The exact process goes on till the last combination gets checked. Finally, all the detected ‘Chhands’ list is displayed in the metadata even though the primary input got detected as a ‘Mukt / Muktak Chhand’. But after processing with the advanced ‘Mukt / Muktak Chhand’ detection mechanism, the metadata generator found traces of the uses of some ‘Chhands’ as a part of the given input, and all those found ‘Chhands’ are added to final metadata.

Let us understand with an example. To test and understand the concept, the provided input will be consisting of a combination of the two different ‘Chhand’ named ‘Doha’ and ‘Sortha’:

‘बड़ा हुआ तो क्या हुआ, जैसे पेड़ खजूर ।
पंछी को छाया नहीं, फल लागै अति दूर ।।
कुंद इंदु सम देह, उमा रमन करुनायतन ।
जाहि दीन पर नेह, करहु कृपा मर्दन मयन ॥’

In this example, when it is processed through the automatic metadata generator, it will get detected as ‘Mukt / Matrik Chhand’. Still, once it is reprocessed using the Advanced ‘Mukt / Muktak Chhand’ detection and identification technique. It will try to find the uses of construction rules of the various ‘Chhand’ already modeled in the automatic metadata generator. After chopping in different combinations, the input parts combination will be processed to detect the ‘Doha’ and ‘Sortha’, which will be detected eventually and added into the final metadata. The metadata for the same will be as following:

Type : मुक्तक/मुक्त छंद

मेटाडाटा जनरेटर के पुन: प्रयासो से निम्नलिखित आंतरिक छंदो का समावेश आपके द्वारा दिये गए इनपुट मे किया गया है :

[['दोहा', 'मात्रिक छंद', 'अर्ध सममात्रिक छंद'], ['सोरठा', 'मात्रिक छंद', 'अर्ध सममात्रिक छंद']]

3.6.2 Stop Words Filtering

Stop Words filtering is one of the significant advancements of this automatic metadata generator. It is a considerable advancement because it plays a vital role in upcoming processing and the metadata generator's overall efficiency. In Stop Words filtering, the Stop Words are filtered using an existing hybrid research-based list of the Hindi Stop Words through which the stopwords of a given input is filtered. This filtering is essential because in the following process, while populating meaning and examples of different words, through the wordnet integration. Only words that are left after stop word filtering are processed, which usually takes less time and improves the execution time and the overall efficiency of the metadata generator. Stop Word filtering saves a lot of time and provides improvement to the automatic metadata generator.

Stopwords filtering or removal is not only about saving time. It also improves the accuracy of the system because stopwords are irrelevant for the upcoming processing. Therefore, even though this research work is not dealing with time-related things, still removing stopwords saves time, then it should be filtered out. Another reason why it is required is that in the next step, meanings and example's suggestions will be generated for the words with the help of wordnet. And in that process, stopwords are not required to be processed again.

Let us quickly understand this from the following example:

'बड़ा हुआ तो क्या हुआ, जैसे पेड़ खजूर ।
पंछी को छाया नहीं, फल लागै अति दूर ।।'

For the given example input, the stop words will be filtered, and results will be as follows:

Found Stop Words:

['हुआ', 'तो', 'क्या', 'हुआ', 'जैसे', 'को', 'दूर']

3.6.3 Populating Meaning and Example of Word with Wordnet Integration

After filtering the Stop Words, the need for a meaning of words was required to accomplish the same the Hindi Wordnet was integrated. With the integration of the Hindi Wordnet, the remaining words list after the filtering of Stop Words is processed, as the Hindi Poem consists of so many words, and one might do not aware of the meaning of many words. So, the word meaning and an instance of the usage of that word is populated with Hindi WordNet's help.

In Wordnet's integration, some words meaning was found, and some word's meaning was not found. The reason why several words meaning was not found can be either the word not included yet in the Wordnet, or the words are coming from the local native language, which not coming from directly Hindi.

Let us understand the same with an example of 'Kanthi Chhand':

'हुआ सवेरा।
मिटा अँधेरा।।
सुषुप्त जागो।
खुमार त्यागो।।'

Here are the word meanings with an example which are found in wordnet:
Words Meaning and Examples:
सवेरा (Meaning) : दिन निकलने का समय
Example: सुबह होते ही किसान खेत की ओर चल दिया ।
अँधेरा (Meaning) : प्रकाश का अभाव
Example: सूर्य डूबते ही चारों ओर अंधकार हो जाता है ।
खुमार (Meaning) : वह मानसिक अवस्था जो शराब, भाँग आदि मादक पदार्थों के सेवन से होती है
Example: शराब के नशे में चूर सिपाही ने निर्दोष रवि को बहुत पीटा । Words Meaning Not Found: ['सुषुप्त', 'जागो', 'मिटा', 'त्यागो']

This is how the word's meaning and the example uses of the words or similar sentences are generated, which makes it easier for readers to understand Hindi poetry. There can be multiple meanings or usefulness of the words, but here, the most common meanings and examples are populated as the context-based meaning for words is still a challenging task for Hindi wordnet development itself, for Hindi the context-based meaning is currently not possible might be that will be in possible upcoming future. With a minor change in the automatic metadata generator, that can also be incorporated.

3.6.4 Suggesting Examples of the identified 'Chhand'

This is another valuable feature of the automatic metadata generator through which similar examples are suggested for the identified verses. For example, when a user generates metadata of a particular type of verse through an automatic metadata generator, the user may be interested in more examples of Hindi verses of the same type.The different examples are delivered through the JavaScript Object Notation (JSON) based pre-managed file of examples. In this mechanism, new examples can be added systematically in key-value pairs. Verse types are used as keys, and their respective examples are stored as values. Random examples get populated for the detected and identified Verse type from the stored examples. Like the following provided input was detected as 'Doha':

'ऐसी वाणी बोलिए, मन का आपा खोय ।
औरन को शीतल करे, आपहु शीतल होय ।।'

As soon as it is detected as a 'Doha', the automatic metadata generator looks for the examples from the already managed example JSON file and populated any random example from the respective 'Chhand' examples. For this, another example populated was as following:
Chhand Example:

'बड़ा हुआ तो क्या हुआ, जैसे पेड़ खजूर ।
पंछी को छाया नहीं, फल लागै अति दूर ।।'

So, this is how example suggestion for the detected 'Chhand' are populated through which whoever is willing to learn more about the appropriate 'Chhand' can explore more and learn more about the construction of the 'Chhand', which is very helpful in understanding the concept behind the structure of the 'Chhand'.

3.6.5 Additional Several Utilities

As part of the research, while carrying out this research, the standard utilities needs were felt, which did not exist but were required. Data collection utility was created for the collection of the data to develop a systematic data corpus. Later on, a bulk collection utility for the data collection was also developed through which bulk data can be processed and stored via text and comma-separated value (CSV) files. To the research progress status, a utility for the research stats generation was also designed. These all utilities were developed for systematic management and to reduce the manual work efforts.

3.7 An approach to identify and detect 'Alankars'

'Alankar', also known as the figure of speech which was not the initial objective of this research, was also explored in depth during this research study. The 'Chhand' and the 'Alankar' part in the Hindi language is truly untouched with the research perspective that was observed while carrying out the extensive literature review. Like the 'Chhand', no systematic structure was found for 'Alankar', which can be directly used for the research purpose. Massive efforts were made to structure and classify the different 'Alankar' into specific classes to create a proper hierarchical structure with the continuation of the traditional ways of 'Alankar' class rules and examples.

The hierarchically structured list is representing the different 'Alankars' included in the various categories, is generated based on such meaning pieces of information collected from different sources [53–55].

1. **'ShabdAlankar' ('शब्दालंकार')**
    - 1.1. *Alliteration ('अनुप्रास अलंकार')*
        - 1.1.1. 'Chekanupras' ('छेकानुप्रास अलंकार')
        - 1.1.2. 'Vrutyanupras' ('वत्यानप्रास अलंकार')
        - 1.1.3. 'Latanupras' ('लाटानुप्रास अलंकार')
        - 1.1.4. 'Antyanpras' ('अन्तत्यानप्रास अलंकार')
        - 1.1.5. 'Shrtyanpras' ('श्रत्यानप्रास अलंकार')
    - 1.2. *'Yamak' ('यमक अलंकार')*
    - 1.3. *'Punrukti' ('पुनरुक्ति अलंकार')*
    - 1.4. *'Vipsa' ('विप्सा अलंकार')*
    - 1.5. *'Vakrokti' ('वक्रोक्ति अलंकार')*
        - 1.5.1. 'Kaku Vakrokti' ('काकु वक्रोक्ति अलंकार')
        - 1.5.2. 'Shelsh Vakrokti' ' ('श्लेष वक्रोक्ति अलंकार')
    - 1.6. *Pun or Irony ('श्लेष अलंकार')*
        - 1.6.1. 'Abhang Shlesh' ('अभंग श्लेष अलंकार')
        - 1.6.2. 'Sabhang Shlesh' ('सभंग श्लेष अलंकार')
2. **'ArthAlankar' ('अर्थालंकार')**
    - 2.1. *Simile ('उपमा अलंकार')*
        - 2.1.1. 'Purnopama' ('पूर्णोपमा अलंकार')

2.1.2. 'Luptopama' ('लुप्तोपमा अलंकार')

2.2. *Metaphor (*'रूपक अलंकार'*)*

2.2.1. 'Sam Rupak' ('सम रूपक अलंकार')

2.2.2. 'Adhik Rupak' ('अधिक रूपक अलंकार')

2.2.3. 'Nyun Rupak' ('न्यून रूपक अलंकार')

2.3. *Poetic Fancy (*'उत्प्रेक्षा अलंकार'*)*

2.3.1. 'Vastupreksha' ('वस्तुप्रेक्षा अलंकार')

2.3.2. 'Hetupreksha' ('हेतुप्रेक्षा अलंकार')

2.3.3. 'Falotpreksha' ('फलोत्प्रेक्षा अलंकार')

2.4. *Exemplification (*'द्रष्टान्ति अलंकार/दृष्टान्त'*)*

2.5. *Doubt (*'संदेह अलंकार'*)*

2.6. *Hyperbole (*'अतिश्योक्ति अलंकार'*)*

2.7. *'Upmeyopma' (*'उपमेयोपमा अलंकार'*)*

2.8. *Converse (*'प्रतीप अलंकार'*)*

2.9. *Self Comparison (*'अनन्वय अलंकार'*)*

2.10. *Error (*'भ्रांतिमान अलंकार'*)*

2.11. *Illuminator(*'दीपक अलंकार'*)*

2.12. *Concealment (*'अपह्नति अलंकार'*)*

2.13. *'Vyatirek' (*'व्यतिरेक अलंकार'*)*

2.14. *Peculiar Causation (*'विभावना अलंकार'*)*

2.15. *Peculiar Allegation (*'विशेषोक्ति अलंकार'*)*

2.16. *Corroboration (*'अर्थान्तरन्यास अलंकार'*)*

2.17. *'Ullekh' (*'उल्लेख अलंकार'*)*

2.18. *Contradiction (*'विरोधाभास अलंकार'*)*

2.19. *Disconnection (*'असंगति अलंकार'*)*

2.20. *Personification(*'मानवीकरण अलंकार'*)*

2.21. *'Anantyokti' (*'अन्तयोक्ति अलंकार'*)*

2.22. *Poetical Reason (*'काव्यलिंग अलंकार'*)*

2.23. *Natural Description (*'स्वभावोती अलंकार'*)*

2.24. *Typical Comparison (*'प्रतिवस्तूपमा'*)*

2.25. *Chain of Similes (*'मालोपमा'*)*

2.26. *Equal Pairing (*'तुल्योगिता'*)*

2.27. *Illustration (*'निदर्शना'*)*

2.28. *Speech of Brevity (*'समासोक्ति'*)*

2.29. *Indirect Dissection (*'अप्रस्तुतप्रशंसा'*)*

2.30. *Special Mention (*'परिसंख्या'*)*

3. **'UbhayAlankar' (**'उभयालंकार'**)**

3.1. *Combination of Figures of Speech (*'संसृष्टि'*)*

3.2. *The fusion of Figures of Speech (*'संकर'*)*

Based on the diverse collection of practices of 'Alankars', the 'Alankars' listed in this hierarchical managed list was used for additional research purposes in this research work. Also, it will help the upcoming research works in the same segment in the future.

'Alankars' were also identified and structured. A total of 58 classification types of Alankars were found, out of which three classes can be identified using this metadata generator ('Shabdalankar'('शब्दालंकार'), 'Anupras'('अनुप्रास'), 'Punrukti'('पुनरुक्ति')).

Apart from these, efforts were made to classify four more classes named ('ArthAlankar'('अर्थालंकार'), 'Yamak'('यमक'), 'Utpeksha'('उत्प्रेक्षा'), 'Upma'('उपमा')). All four were also implemented, but as the context-based meaning of the word is required, which is still a challenge in Hindi NLP, these classes cannot be said or claimed to be genuinely classified with this automatic metadata generator.

3.8 Summary

This section is the last subsection of the Research Methodology chapter of this thesis. A detailed description and summarised discussion of the metadata generator's modeling is included to accommodate the research work explained in the earlier Section 3.1 to 3.7 of this chapter. In Section 3.1, the Hindi Verse's classes and subclasses and the different findings during the Literature Review are enlightening. The very next Section 3.2, discusses the various components of Hindi Verse and appropriate examples for better clarity.

Quantity calculation primary rules and some special exceptional rules are incorporated in Section 3.3, with suitable cases to understand correctly. Section 3.4 Hierarchical structure construction of Hindi Metre is based on the different classes discussed in Section 3.1, components from section 3.2, sets of rules of Section 3.3, and individual unique rules of particular types, which are manually identified and validated using testing through different examples.

Further, in Section 3.5, the basic core idea of modeling of metadata generator is represented. Modeling is based on each point discussed from Section 3.1 to Section 3.4 and the unique rules of the individual Hindi Verses. Modeling is described using a core flow chart and simple steps through the algorithm. Based on the modeled metadata generator, the three different scenarios of 'Matrik Chhand' detection, 'Varnik Chhand' detection, and 'Mukt / Muktak Chhand' detection is explained, which includes the proper step by step explanation of each calculation also.

Advanced 'Mukt / Muktak' identification and detection are added to advance the metadata generator in Section 3.6. It works on the reanalysis by separating the input detected as the 'Mukt / Muktak Chhand' class earlier to check if any part of it uses any existing modeled Hindi Verses rules through reprocessing the separated parts of the actual input. Furthermore, Stopword filtering, wordnet integration for the meanings of the words, and examples of those words are integrated. A similar example generation for the detected Hindi Verse type of the given input is also one of the metadata generator's beneficial features discussed in the second last subsection of Section 3.6. The final subsection of Section 3.6 talks about the different utilities developed during research work for corpus creation.

Section 3.7 representing an approach to identify and detect which was not the earlier part and objective of this research work 'Alankars'. Here everything about the research methodology of this research work ends in brief. The results are discussed in the Chapter 4 Results and Discussions.

Chapter 4

Results and Discussions

Results are the essential part of the research. To simply and understand the research result in this kind of research work, the best way to represent the result is to know with the actual example results first, and later the overall result can be discussed.

The results discussed here are widely tested, and the details about the data on which testing was done are represented in the Table 4.1 Overall Results. Here the classification was done automatically through the automatic metadata generator, and the validation was done manually. To understand the results better, let us check out the results for each primary 'Chhand' class. For a better understanding, the same examples will be used, which were used in the Methodology section.

Different parts of the Results and Discussions are entitled as follows:

1. Implementation Specifications
2. 'Matrik Chhand' Result
3. 'Varnik Chhand' Result
4. 'Mukt / Muktak Chhand' Result
5. Advance 'Mukt / Muktak Chhand' Result
6. Overall Results
7. Discussions

Before understanding the results, let us go through the implementation specifications first.

4.1 Implementation Specifications

Following are the specifications on which implementation and the test execution of the automatic metadata generator took place.

- **System:** MacBook Air (13-inch, 2017)
- **Operating System:** macOS Big Sur Version 11.4
- **Processor:** 1.8GHz dual-core Intel Core i5
- **Storage:** 128GB PCIe-based flash storage
- **Memory:** 8GB of 1600MHz DDR3
- **Graphics:** Intel HD Graphics 6000
- **Programming Langauge:** Python Version 3.9
- **Editors:** PyCharm (Community Edition) Version 2019.3, Visual Studio Code Version 1.57.1
- **External APIs / Libraries:** pyiwn (Python-based API for IndoWordNet) [97] and Hindi Language Stop Words List [101]

Let us understand why Python was chosen for implementation.

4.2 Why Python?

Python is the best suitable programming language based on the nature of the research problem. The current problem is a rule-based modeling-related problem. Therefore, a dynamic programming language that focuses on code readability is required, and Python is the best fit.

Python is open-source, easy to learn and implement, and has a rich set of libraries. Due to that, it is becoming the first choice for data or data science-related problems. According to Google Trends and GitHub, till 2020, Python is still the most popular programming language as shown in Figure 4.1 [108].

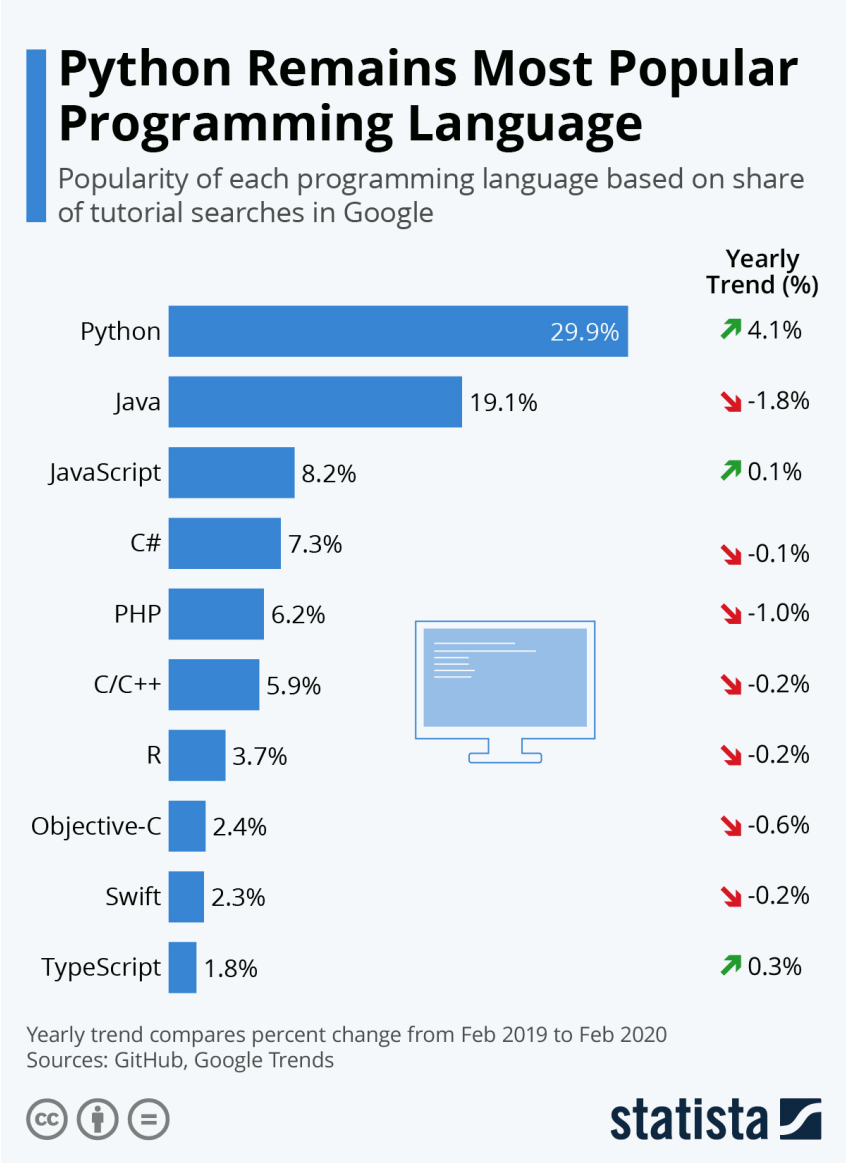


Figure 4.1: Popularity of Python

Let us understand the implementation through results now.

4.3 ‘Matrik Chhand’ Result

While processing the ‘Matrik Chhand’, the result produced by the automatic metadata generator is as follow:

‘ऐसी वाणी बोलिए, मन का आपा खोय ।
औरन को शीतल करे, आपहु शीतल होय ।।’

Let us provide this as an input to the automatic metadata generator and check what metadata is generated as output results.

Metadata Generator For Hindi Poetry
Content of file :
ऐसी वाणी बोलिए, मन का आपा खोय ।
औरन को शीतल करे, आपहु शीतल होय ॥
Number of characters (Including Space) : 66
Number of characters (After Cleaning and Without Spaces) : 45
Number of Total Diacritics : 14
Number of Total Guru Matra : 12
Number of Total Laghu Matra : 2
Number of Total Half : 0
Number of Words : 14
Number of Lines : 2
Number of Charans : 4
Charan Wise Matra (QTY) Count :
[[[2, 2], [2, 2], [2, 1, 2]], [[1, 1], [2], [2, 2], [2, 1]], [[2, 1, 1], [2], [2, 1, 1], [1, 2]], [[2, 1, 1], [2, 1, 1], [2, 1]]]
Charan Wise Matra (QTY) Sum : [13, 11, 13, 11]
Charan Wise Varna : ['2222212', '1122221', '211221112', '21121121']
Charan Wise Varna Sum : [7, 7, 9, 8]
Symbolic Rule Representation (For Varnik) : ['ऽऽऽऽऽ।ऽ', '।।ऽऽऽऽ।', 'ऽ।।ऽऽ।।ऽ', 'ऽ।।ऽ।।ऽ।']
Type : मात्रिक छंद
Sub Type : अर्ध सममात्रिक छंद
Chhand : दोहा
Chhand Example :
बड़ा हुआ तो क्या हुआ, जैसे पेड़ खजूर
पंछी को छाया नहीं, फल लागै अति दूर
Found Stop Words: ['का', 'को']
Words Meaning & Examples:
वाणी (Meaning) : मनुष्य के मुख से निकलने वाला सार्थक शब्द
Examples : ऐसा वचन बोलें जो दूसरों को अच्छा लगे ।
मन (Meaning) : प्राणियों में अनुभव, संकल्प-विकल्प, इच्छा, विचार आदि करने वाली शक्ति
Examples : मन की चंचलता को दूर करना कठिन कार्य है ।,दूसरे के मन की बात कौन जान सकता है ।
आपा (Meaning) : वह बहन जो उम्र में बड़ी हो
Examples : मेरी बड़ी बहन अध्यापिका हैं ।
Words Meaning Not Found ['शीतल', 'होय', 'औरन', 'बोलिए', 'करे', 'ऐसी', 'आपहु', 'खोय']
Alankar Detection in Original Input:
Type : शब्दालंकार
Sub Type : अनुप्रास अलंकार

Figure 4.2: Matrik Verses Result

The automatic metadata generator has generated the complete metadata based on the given input for the ‘Matrik Chhand’, which consists of so much information helpful with Computational Linguistics’s perspective as shown in Figure 4.2. The type, subtype, and ‘Chhand’ are based on the rule-based modeling of plenty of the combinations and complex rules of ‘Matrik Chhand’.

4.4 ‘Varnik Chhand’ Result

While processing the ‘Varnik Chhand’, the result produced by the automatic metadata generator is as follow:

‘जो मैं नया ग्रंथ विलोकता हूँ, भाता मुझे सो नव मित्र सा है ।
देखूँ उसे मैं नित सार वाला, मानो मिला मित्र मुझे पुराना ।।’

Let us provide this as an input to the automatic metadata generator and check what metadata is generated as output results.

Metadata Generator For Hindi Poetry

Content of file :
जो मैं नया ग्रंथ विलोकता हूँ, भाता मुझे सो नव मित्र सा है ।
देखूँ उसे मैं नित सार वाला, मानो मिला मित्र मुझे पुराना ॥
Number of characters (Including Space) : 119
Number of characters (After Cleaning and Without Spaces) : 88
Number of Total Diacritics : 41
Number of Total Guru Matra : 28
Number of Total Laghu Matra : 10
Number of Total Half : 3
Number of Words : 24
Number of Lines : 2
Number of Charans : 4
Charan Wise Matra (QTY) Count : [[[2], [2], [1, 2], [2, 1], [1, 2, 1, 2], [2]], [[2, 2], [1, 2], [2], [1, 1], [2, 1], [2], [2]], [[2, 2], [1, 2], [2], [1, 1], [2, 1], [2, 2]], [[2, 2], [1, 2], [2, 1], [1, 2], [1, 2, 2]]]
Charan Wise Matra (QTY) Sum : [18, 18, 18, 18]
Charan Wise Varna : ['22122112122', '22122112122', '22122112122', '22122112122']
Charan Wise Varna Sum : [11, 11, 11, 11]
Symbolic Rule Representation (For Varnik) : ['ऽऽ।ऽऽ।ऽ।ऽऽ', 'ऽऽ।ऽऽ।ऽ।ऽऽ', 'ऽऽ।ऽऽ।ऽ।ऽऽ', 'ऽऽ।ऽऽ।ऽ।ऽऽ']
Type : वर्णिक वृत्त / छंद
Sub Type : सम वर्णिक वृत्त / छंद
Chhand : इन्द्रवज्रा
Chhand Example :
संसार का है वह वीर आला। था ज्ञान एवं तलवार भाला।
था तुंग जैसे वह वीर सोला। गंभीर रत्नाकर धीर भोला।
Found Stop Words ['जो', 'मैं', 'हूँ', 'सो', 'है', 'उसे', 'मैं', 'मानो']
Words Meaning & Examples
ग्रंथ (Meaning) : मोटी पुस्तक
Examples : रामायण, पुराण, बाइबिल आदि ग्रंथ हैं ।
नव (Meaning) : आठ और एक के योग से प्राप्त संख्या
Examples : पाँच और चार नौ होता है ।
मित्र (Meaning) : प्रायः समान अवस्था का वह व्यक्ति जिससे स्नेहपूर्ण संबंध हो तथा जो सब बातों में सहायक और शुभचिन्तक हो
Examples : सच्चे मित्र की परीक्षा आपत्ति-काल में होती है ।
सार (Meaning) : किसी विचार या अनुभव का सबसे आवश्यक या सबसे महत्वपूर्ण हिस्सा
Examples : एक घंटे की कड़ी मेहनत के बाद ही हम इस लेख के निष्कर्ष तक पहुँच पाए ।
Words Meaning Not Found ['पुराना', 'मिला', 'वाला', 'विलोकता', 'भाता', 'देखूँ', 'नित', 'नया', 'मुझे', 'सा']
Alankar Detection in Original Input
Type : शब्दालंकार
Sub Type : अनुप्रास अलंकार

Figure 4.3: Varnik Verses Result

Similar to the ‘Matrik Chhand’ in ‘Varnik Chhand’ also the Type, Sub Type, and ‘Chhand’ get identified and detected based on the already modeled rules and so much relevant metadata information the metadata generator generates the metadata automatically as shown in Figure 4.3.

4.5 ‘Mukt / Muktak Chhand’ Result

While processing the ‘Mukt / Muktak Chhand’, the result produced by the automatic metadata generator is as follow:

इस उड़ान पर अब शर्मिंदा, में भी हूँ और तू भी है।
आसमान से गिरा परिंदा, में भी हूँ और तू भी है।।
छुट गयी रस्ते में, जीने मरने की सारी कसमें।
अपने-अपने हाल में जिंदा, में भी हूँ और तू भी है।।’

Metadata Generator For Hindi Poetry

Content of file :
इस उड़ान पर अब शर्मिंदा, में भी हूँ और तू भी है
आसमान से गिरा परिंदा, में भी हूँ और तू भी है
छुट गयी रस्ते में, जीने मरने की सारी कसमें
अपने-अपने हाल में जिंदा, में भी हूँ और तू भी है
Number of characters (Including Space) : 184
Number of characters (After Cleaning and Without Spaces) : 137
Number of Total Diacritics : 58
Number of Total Guru Matra : 48
Number of Total Laghu Matra : 8
Number of Total Half : 2
Number of Words : 43
Number of Lines : 4
Number of Charans : 8
Charan Wise Matra (QTY) Count : [[[1, 1], [1, 2, 1], [1, 1], [1, 1], [2, 2, 2]], [[2], [2], [2], [2, 1], [2], [2], [2]], [[2, 1, 2, 1], [2], [1, 2], [1, 2, 2]], [[2], [2], [2], [2, 1], [2], [2], [2]], [[1, 1], [1, 2], [2, 2], [2]], [[2, 2], [1, 1, 2], [2], [2, 2], [1, 1, 2]], [[1, 1, 2, 1, 1, 2], [2, 1], [2], [2, 2]], [[2], [2], [2], [2, 1], [2], [2], [2]]]
Charan Wise Matra (QTY) Sum : [16, 15, 16, 15, 11, 18, 17, 15]
Charan Wise Varna : ['111211111222', '22221222', '2121212122', '22221222', '1112222', '22112222112', '11211221222', '22221222']
Charan Wise Varna Sum : [12, 8, 10, 8, 7, 11, 11, 8]
Symbolic Rule Representation (For Varnik) : ['।।ऽ।।।।ऽऽऽ', 'ऽऽऽऽ।ऽऽऽ', 'ऽ।ऽ।ऽ।ऽ।ऽऽ', 'ऽऽऽऽ।ऽऽऽ', '।।।ऽऽऽऽ', 'ऽऽ।।ऽऽऽऽ।।ऽ', '।।ऽ।।ऽऽ।ऽऽऽ', 'ऽऽऽऽ।ऽऽऽ']
Type : मुक्तक/मुक्त छंद
मेटाडाटा जनरेटर के पुनः प्रयासो से पश्चात भी आंतरिक छंदो का समावेश आपके द्वारा दिये गए इनपुट मे नहीं पाया गया है!
Found Stop Words : ['इस', 'पर', 'अब', 'में', 'भी', 'हूँ', 'और', 'भी', 'है', 'से', 'में', 'भी', 'हूँ', 'और', 'भी', 'है', 'गयी', 'में', 'की', 'में', 'में', 'भी', 'हूँ', 'और', 'भी', 'है']
Words Meaning & Examples
उड़ान (Meaning) : हाथ का वह भाग जहाँ हथेली का जोड़ रहता है
Examples : राम ने मेरी कलाई पकड़ ली ।
……
जिंदा (Meaning) : वह प्राणी जो मरा न हो या जिसमें प्राण हो
Examples : जीवितों पर संस्मरण लिखना साहस का काम है ।
Words Meaning Not Found ['तू', 'जीने', 'मरने', 'अपने-अपने', 'शर्मिंदा', 'रस्ते', 'कसमें', 'छुट']
Alankar Detection in Original Input
Type : शब्दालंकार
Sub Type : अनुप्रास अलंकार

Figure 4.4: Mukt / Muktak Verses Result

Figure 4.4 is representing the Mukt / Muktak Verses Result. After understanding the ‘Matrik Chhand’ and ‘Varnik Chhand’, if the input is not identified or detected, then that will fall under the ‘Mukt / Muktak’ category. Still, after that, as per the advanced ‘Mukt / Muktak’ detection mechanism, efforts were made to find any ‘Chhand’ type used in stanzas, but none was found.

4.6 Advance ‘Mukt / Muktak Chhand’ Result

While Advance processing the ‘Mukt / Muktak Chhand’, the result produced by the automatic metadata generator is as follow:

‘जो मैं नया ग्रंथ विलोकता हूँ, भाता मुझे सो नव मित्र सा है ।
देखूँ उसे मैं नित सार वाला, मानो मिला मित्र मुझे पुराना ।।
ऐसी वाणी बोलिए, मन का आपा खोय ।
औरन को शीतल करे, आपहु शीतल होय ।।’

Metadata Generator For Hindi Poetry
Content of file :
जो मैं नया ग्रंथ विलोकता हूँ, भाता मुझे सो नव मित्र सा है ।
देखूँ उसे मैं नित सार वाला, मानो मिला मित्र मुझे पुराना ॥
ऐसी वाणी बोलिए, मन का आपा खोय ।
औरन को शीतल करे, आपहु शीतल होय ॥
Number of characters (Including Space) : 185
Number of characters (After Cleaning and Without Spaces) : 133
Number of Total Diacritics : 55
Number of Total Guru Matra : 40
Number of Total Laghu Matra : 12
Number of Total Half : 3
Number of Words : 38
Number of Lines : 4
Number of Charans : 8
Charan Wise Matra (QTY) Count : [[[2], [2], [1, 2], [2, 1], [1, 2, 1, 2], [2]], [[2, 2], [1, 2], [2], [1, 1], [2, 1], [2], [2]], [[2, 2], [1, 2], [2], [1, 1], [2, 1], [2, 2]], [[2, 2], [1, 2], [2, 1], [1, 2], [1, 2, 2]], [[2, 2], [2, 2], [2, 1, 2]], [[1, 1], [2], [2, 2], [2, 1]], [[2, 1, 1], [2], [2, 1, 1], [1, 2]], [[2, 1, 1], [2, 1, 1], [2, 1]]]
Charan Wise Matra (QTY) Sum : [18, 18, 18, 18, 13, 11, 13, 11]
Charan Wise Varna : ['22122112122', '22122112122', '22122112122', '22122112122', '2222212', '1122221', '211221112', '21121121']
Charan Wise Varna Sum : [11, 11, 11, 11, 7, 7, 9, 8]
Symbolic Rule Reperenesntation (For Varnik) : ['ऽऽ।ऽऽ।ऽ।ऽऽ', 'ऽऽ।ऽऽ।ऽ।ऽऽ', 'ऽऽ।ऽऽ।ऽ।ऽऽ', 'ऽऽ।ऽऽ॥ऽ।ऽऽ', 'ऽऽऽऽ।ऽ', '॥ऽऽऽ।', 'ऽ॥ऽऽ॥।ऽ', 'ऽ॥ऽ।ऽ।']
Type : मुक्तक/मुक्त छंद
मेटाडाटा जनरेटर के पुनः प्रयासो से निम्नलिखित आंतरिक छंदो का समावेश आपके द्वारा दिये गए इनपुट मे किया गया है :
[['इन्द्रवज्रा', 'वर्णिक वृत्त / छंद', 'सम वर्णिक वृत्त / छंद'], ['दोहा', 'मात्रिक छंद', 'अर्ध सममात्रिक छंद']]
2021-08-14:13:50:23,711 INFO [iwn.py:43] Loading hindi language synsets...
Found Stopwords ['जो', 'मैं', 'हूँ', 'सो', 'है', 'उसे', 'मैं', 'मानो', 'का', 'को']
Words Meaning & Examples
ग्रंथ (Meaning) : मोटी पुस्तक
Examples : रामायण, पुराण, बाइबिल आदि ग्रंथ हैं ।
नव (Meaning) : आठ और एक के योग से प्राप्त संख्या
Examples : पाँच और चार नौ होता है ।
मित्र (Meaning) : प्रायः समान अवस्था का वह व्यक्ति जिससे स्नेहपूर्ण संबंध हो तथा जो सब बातों में सहायक और शुभचिन्तक हो
Examples : सच्चे मित्र की परीक्षा आपत्ति-काल में होती है ।
सार (Meaning) : किसी विचार या अनुभव का सबसे आवश्यक या सबसे महत्वपूर्ण हिस्सा
Examples : एक घंटे की कड़ी मेहनत के बाद ही हम इस लेख के निष्कर्ष तक पहुँच पाए ।
वाणी (Meaning) : मनुष्य के मुख से निकलने वाला सार्थक शब्द
Examples : ऐसा वचन बोलें जो दूसरों को अच्छा लगे ।
मन (Meaning) : प्राणियों में अनुभव, संकल्प-विकल्प, इच्छा, विचार आदि करने वाली शक्ति
Examples : मन की चंचलता को दूर करना कठिन कार्य है ।,दूसरे के मन की बात कौन जान सकता है ।

Words Meaning Not Found ['मिला', 'बोलिए', 'औरन', 'सा', 'करे', 'देखूँ', 'ऐसी', 'पुराना', 'आपहु', 'वाला', 'नया', 'नित', 'होय', 'भाता', 'शीतल', 'खोय', 'विलोकता', 'मुझे']

Alankar Detection in Original Input
Type : शब्दालंकार
Sub Type : अनुप्रास अलंकार

Figure 4.5: Advance Mukt / Muktak Verses Result

Figure 4.5 is representing the Advance Mukt / Muktak Verses Result. After understanding the ‘Matrik Chhand’, ‘Varnik Chhand’ and ‘Mukt / Muktak Chhand’, if the input is not identified or detected, then that will go for the Advance ‘Mukt / Muktak’ category as discussed in 3.6.1 Advance ‘Mukt / Muktak’ Identification and Detection. As per the advanced ‘Mukt / Muktak’ detection mechanism, efforts were made to find any ‘Chhand’ type used in stanzas, and ‘Indravajra’ and ‘Doha’ were found.

4.7 Overall Results

Results are always an essential part of any research study. Before moving on to the results, the thing that highlights something here is that not much has been found in the current work in this field, so it is impossible to compare the research work and the results. No similar work has been found relating to the classification and identification of Hindi Verses, which makes this work unique and novel.Just two nearby research papers were observed. One was related to the identification of 'Chaupai' ('चौपाई') with 95.03% accuracy, while 97.09% accuracy is achieved by the proposed research. Similarly, the second research paper relevant to 'Rola'('रोला') by 89.83% accuracy, where 95.24% accuracy is provided by the current research work.

The modeled automatic metadata generator relies on the varying rules of the one hundred and eleven (111) Hindi Verses classification types, subtypes, and subsubtypes. The fifty-three (53) distinct types of Hindi Verses data collection was carried out for the test and validation purpose. Every class had a minimum of twenty (20) and a maximum of three hundred and ten (310) records as per the data availability. From various sources Total of 3330 records were found and tested, out of which 3195 records were detected successfully.

The final accuracy rate on the basis of the results is 94.99%, and the inaccuracy rate is 05.01%. Figure 4.6 is consists of a graphical representation of the 'Chhand' data found, detected, and not found, including the proper accuracy and failure or inaccuracy rates of various 'Chhands'. It is also revealing that the several 'Chhands' accuracy rate ranges are between 84%-100%, and the failure or inaccuracy rate is between 0%-16%.

Table 4.1 is representing the Overall Results of Automatic Hindi Verse Detection, which includes Hindi Verses 'Chhand' Name, Found, Detected, Not Detected including success% and failure% of respective Hindi Verses detection by the automatic metadata generator.' Found' means the number of examples collected. 'Detected' represents the number of detected Verses out of 'Found' when processed through an automatic metadata generator. The example collection process was entirely manual, and the collected examples can be validated with the classification only, which was done automatically using the metadata generator. There is no such automatic tool/process available for verification, so verifying the identified Hindi Verses was manual. 'Not Detected' are the ones that are not detected as Hindi Verse in the appropriate type. 'Success %' and 'Failure%' are the accuracy and failure percentage based on 'Detect', 'Not Detected', and 'Found'. If we talk about labeling, for 'Found' labeling was manual and processed 'Detected' data, labeling was automatic through the automatic metadata generator.

Figure 4.6: Chhand Detection Results

Table 4.1: Overall Results of Automatic Hindi Verse Detection

Sr. No.	Chhand	Found	Detected	Undetected	Success %	Failure %
1	Doha (‘दोहा’)	277	265	12	95.67	4.33
2	Sortha (‘सोरठा’)	52	51	1	98.08	1.92
3	Rola (‘रोला’)	147	140	7	95.24	4.76
4	Harigitika (‘हरिगीतिका" / "हरगीतिका’)	83	80	3	96.39	3.61
5	Chaupai (‘चौपाई’)	310	301	9	97.09	2.91
6	Barvai (‘बरवै’)	36	35	1	97.22	2.78
7	Kundlia (‘कुंडलिया’)	35	33	2	94.29	5.71
8	Chhappay (‘छप्पय’)	25	21	4	84.00	16.00
9	Ullala (‘उल्लाला (सममात्रिक)’)	31	30	1	96.77	3.23
10	Ahir (‘अहीर’)	46	43	3	93.48	6.52
11	Tomar (‘तोमर’)	45	41	4	91.11	8.89
12	Tribhangi (‘त्रिभंगी’)	27	25	2	92.59	7.41
13	Roopmala (‘रूपमाला / मदन’)	20	19	1	95.00	5.00

Table 4.1 continued from previous page

14	VasantTilka (‘वसंततिलका’)	238	222	16	93.28	6.72
15	Malini (‘मालिनी’)	105	103	2	98.10	1.90
16	Mandakranta (‘मन्दाक्रान्ता’)	153	153	0	100.00	0.00
17	Vanshasth (‘वंशस्थ’)	109	108	1	99.08	0.92
18	Drutvilambit (‘द्रुतविलम्बित’)	162	161	1	99.38	0.62
19	Trotak (‘तोटक / त्रोटक’)	52	51	1	98.08	1.92
20	Shardul Vikridit (‘शार्दुल विक्रीडित’)	32	29	3	90.63	9.38
21	Pramanika (‘प्रमाणिका’)	39	38	1	97.44	2.56
22	Swagta (‘स्वागता’)	25	23	2	92.00	8.00
23	Bhujangi (‘भुजंगी’)	78	77	1	98.72	1.28
24	Dodhak (‘दोधक’)	47	45	2	95.74	4.26
25	Bhujangprayag (‘भुजन्गप्रयाग’)	42	42	0	100.00	0.00
26	Panchchamar (‘पंचचामर’)	43	41	2	95.35	4.65
27	Tantak (‘तांटक’)	30	27	3	90.00	10.00

Table 4.1 continued from previous page

28	Kukubh (‘कुकुभ’)	35	33	2		94.29	5.71
29	Madira Savaiya (‘मदिरा सवैया’)	60	56	4		93.33	6.67
30	Durmil Savaiya (‘दुर्मिल सवैया’)	58	56	2		96.55	3.45
31	Manoram (‘मनोरम’)	48	46	2		95.83	4.17
32	Piyushvarsh (‘पीयूष वर्ष’)	50	49	1		98.00	2.00
33	Sagun (‘सगुण’)	24	22	2		91.67	8.33
34	Sindhu(‘सिन्धु’)	34	32	2		94.12	5.88
35	Digpal (‘दिग्पाल’)	40	37	3		92.50	7.50
36	Shankar (‘शंकर’)	30	26	4		86.67	13.33
37	Nishchal (‘निश्चल’)	30	28	2		93.33	6.67
38	Sar (‘सार’)	28	25	3		89.29	10.71
39	Sarsi (‘सरसी छंद / कबीर / समुंदर छंद’)	38	36	2		94.74	5.26
40	Krupan Dhanakshari (‘कृपाण घनाक्षरी’)	29	27	2		93.10	6.90
41	Sur Dhanakshari (‘सूर घनाक्षरी’)	38	37	1		97.37	2.63

Table 4.1 continued from previous page

42	Manharan Dhanakshari (‘मनहरण घनाक्षरी’)	43	40	3	93.02	6.98
43	Aansu (‘आँसू’)	30	30	0	100.00	0.00
44	Chopaiya (‘चौपइया’)	20	17	3	85.00	15.00
45	Janak (‘जनक’)	36	36	0	100.00	0.00
46	Nidhi (‘निधि’)	75	72	3	96.00	4.00
47	Marhatha (‘मरहठा’)	25	23	2	92.00	8.00
48	Ras (‘रास’)	33	31	2	93.94	6.06
49	Chandrika (‘चन्द्रिका’)	34	33	1	97.06	2.94
50	Tilka (‘तिलका’)	54	54	0	100.00	0.00
51	Dhar (‘धार’)	35	34	1	97.14	2.86
52	Vidhata (‘विधाता’)	56	53	3	94.64	5.36
53	Muktak (‘मुक्तक’)	58	58	0	100.00	0.00
Total		**3330.00**	**3195.00**	**135.00**	**5034.29**	**265.70**
Maximum		**310.00**	**301.00**	**16.00**	**100.00**	**16.00**

Table 4.1 continued from previous page

Minimum	**20.00**	**17.00**	**0.00**	**84.00**	**0.00**
Average	**62.83**	**60.28**	**2.54**	**94.99**	**5.01**

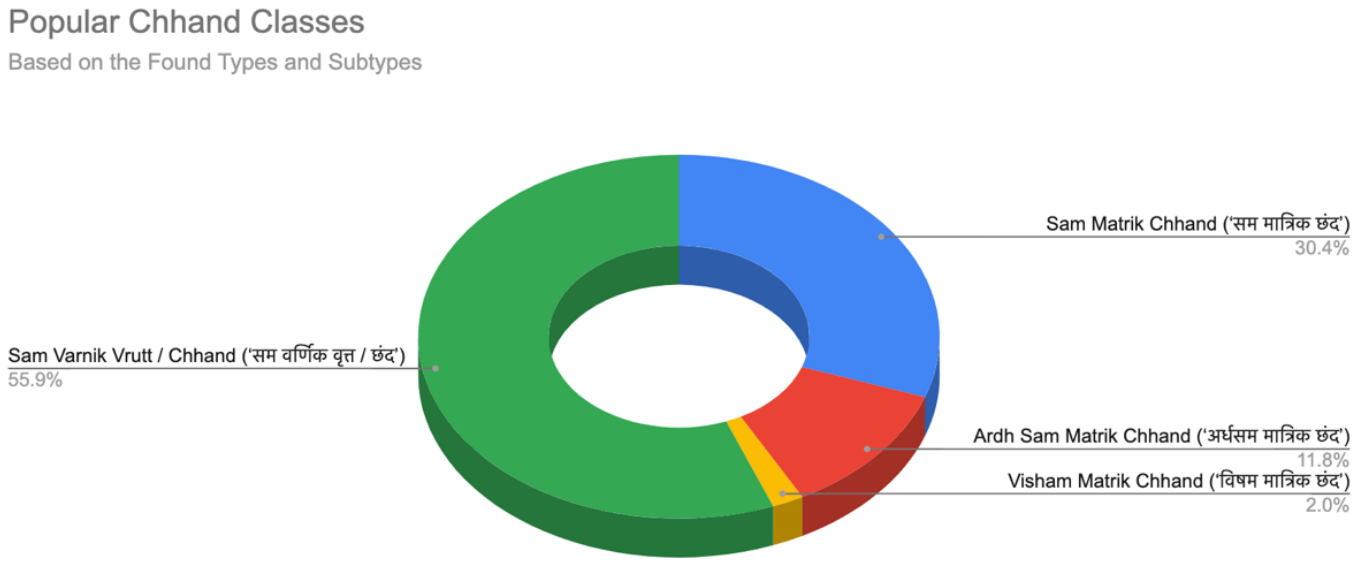


Figure 4.7: Popular Chhand Classes Based on the Found Types and Subtypes

95

Out of all the fifty three (53) various kinds of Hindi Verses data, only five (5) Hindi Verses (‘Chhands’- ‘छंद’) (‘Chhappay’- ‘छप्पय’, ‘Chopaiya’- ‘चौपैया’, ‘Shankar’- ‘शंकर’, ‘Sar’- ‘सार’, ‘Tantak’- ‘तांतक’) were consist below 90% of accuracy rate, and the remaining forty eight (48) were within 90-100%, and further out of these twenty nine (29) (‘Roopmala’- ‘रूपमाला’, ‘Rola’- ‘रोला’, ‘Panchchamar’- ‘पंचचामर’, ‘Doha’- ‘दोहा’, ‘Dodhak’- ‘दोधक’, ‘Manoram’ - ‘मनोरम’, ‘Nidhi’- ‘निधि’, ‘Harigitika’- ‘हरिगितिका’, ‘Durmil Savaiya’- ‘दुर्मिल सवैया’, ‘Ullala (Sam Matrik)’- ‘उल्लाला (सम मात्रिक)’, ‘Chandrika’- ‘चंद्रिका’, ‘Dhar’- ‘धार’, ‘Barvai’- ‘बरवै’, ‘Sur Dhanakshari’- ‘सुर धनाक्षरी’, ‘Pramanika’- ‘प्रमाणिका’, ‘Chaupai’- ‘चौपाई’, ‘Piyushvarsh’- ‘पीयूषवर्ष’, ‘Sortha’- ‘सोरठा’, ‘Trotak’- ‘त्रोटक’, ‘Malini’- ‘मालिनी’, ‘Bhujangi’- ‘भुजंगी’, ‘Vanshasth’ - ‘वंशस्थ’, ‘Drutvilambit’- ‘द्रुतविलंबित’, ‘Mandakranta’- ‘मंदाक्रांता’, ‘Bhujangprayag’ - ‘भुजंगप्रयाग’, ‘Aansu’- ‘आंसू’, ‘Janak’- ‘जनक’, ‘Tilka’- ‘तिलका’, ‘Muktak’- ‘मुक्तक’) were owning 95% or greaterin rage of 95-100% of accurary. The least accuracy, 84% was detected for ‘Chhapay’- ‘छप्पय’, that is formed of a combination of a pair of Hindi Verses (‘Chhands’- ‘छंद’), which usually causes the creation and classification complex, and because of that the problems occur extra. The most excellent performing Hindi Verses (‘Chhands’- ‘छंद’) amidst 100% accuracy rate was ‘Aansu’- (‘आंसू’, ‘Bhujangprayag’ - ‘भुजंगप्रयाग’, ‘Janak’ - ‘जनक’, ‘Mandakranta’ - ‘मंदाक्रांता’, ‘Muktak’- ‘मुक्तक’ and ‘Tilka’ - ‘तिलका’).

The research work is not limited to the already mentioned Hindi Verses (‘Chhands’- ‘छंद’) 227 records of fifty (50) more Hindi Verses (‘Chhands’- ‘छंद’) types were modeled and tested based on these records, for which the data for the same were between 1 to 15 records of each. Out of fifty (50 Hindi Verses (‘Chhands’- ‘छंद’), thirty four (34) Hindi Verses (‘Chhands’- ‘छंद’) (‘Gitika / Chanchri / Charchari’- ‘गीतिका/चंचरी/चर्चरी सवैया’, ‘Sundari / Madhavi Savaiya’- ‘सुंदरी/माधवी सवैया’, ‘Chakor Savaiya’- ‘चकोर सवैया’, ‘Sukhi Savaiya’- ‘सुखी सवैया’, ‘Arsat Savaiya’- ‘अरसात सवैया’, ‘Lavanglata Savaiya’- ‘लवाँगलता सवैया’, ‘Mukthara Savaiya’- ‘मुक्तहरा सवैया’, ‘Vam Savaiya’- ‘वाम सवैया’, ‘Mod Savaiya’- ‘मोद सवैया’, ‘Shuddh Gita’- ‘शुद्ध गीता’, ‘Gaganangana’- ‘गगनंगना’, ‘Lavni’- ‘लावणी’, ‘Madhumalti’- ‘मधुमालती’, ‘Vijat’- ‘विजात’, ‘Janharan Dhanakshari’- ‘जनहरण धनाक्षरी’, ‘Dev Dhanakshari’- ‘देव धनाक्षरी’, ‘Vijya Dhanakshari / Kamini’- ‘विजया धनाक्षरी / कामिनी’, ‘Jalharan Dhanakshari’- ‘जलहरण धनाक्षरी’, ‘Kanak Manjari’- ‘कनक मंजरी’, ‘Giridhari’- ‘गिरिधारी’, ‘Panktika’- ‘पंक्तिका’, ‘Mattgayand Savaiya’- ‘मत्तगयंद सवैया’, ‘Sumukhi Savaiya’- ‘सुमुखी सवैया’, ‘Bihari’- ‘बिहारी’, ‘Nil’- ‘नील’, ‘Ullala (Ardh Sam Matrik)’, ‘Shikhrini’- ‘शिखरिनी’, ‘Arvind Savaiya’- ‘अरविंद सवैया’, ‘Muktamani’ - ‘मुक्तामणि’, ‘Indira’- ‘इंदिरा’, ‘Gath’- ‘गाथ’, ‘Damru Dhanakshari’- ‘डमरू धनाक्षरी’, ‘Asabandha’- ‘असबँधा’, ‘Padhyamala’) were having 1 to 5 examples for each, nine (9) ‘Chhands’ (‘Kusumasamudita’ - ‘कुसुमुदिता’,‘Dhuni’- ‘धूनी’,‘Pavan’- ‘पवन’, ‘Chanchala’ - ‘चंचला’, ‘Udiyana’ - ‘उड़ियाना’,‘Kaamrup’- ‘कामरूप’,‘Chanchrik/Haripriya’- ‘चँचरिक/हरिप्रिया’, ‘Kanthi’- ‘कंठी’, ‘Veer / Aalha / Matrik Savaiya’- ‘वीर/आल्हा/मात्रिक सवैया’) were having 6 to 10 examples each, the remaining seven (‘Shalini’- ‘शालिनी’,‘Kirit Savaiya’- ‘किरिट

सवैया',‘Shakti’- ‘शक्ति’, ‘Indravajra’- ‘इंद्रवज्रा’, ‘Ghanshyam’ - ‘घनश्याम’, ‘Upendravajra’ - ‘उपेंद्रवज्रा’, ‘Pavan’- ‘पवन’) were having 11-15 records for each type. All the records were detected successfully to their associated Hindi Verses ‘Chhand’ types.

This research work is the novel and first of its kind of research work in the world, as no such similar research work was done earlier, so no benchmarking of the accuracy of results can be done here.

The primary three classes(‘Matrik Chhand’- ‘मात्रिक छंद’, ‘Varnik Chhand’- ‘वर्णिक छंद’, ‘Muktak / Mukt Chhand’- ‘मुक्तक/मुक्त छंद’) were incorporated, along with their associative further six subclasses (‘Sam Matrik’- ‘सम मात्रिक’, ‘Ardh-Sam Matrik’- ‘अर्धसम मात्रिक’, ‘Visham Matrik’- ‘विषम मात्रिक’, ‘Sam Varnik’- ‘सम वर्णिक’, ‘Ardh Sam Varnik’- ‘अर्धसम वर्णिक’, ‘Visham Varnik’- ‘विषम वर्णिक’) were also included. However, any ‘Chhand’ type rules or examples under two of the subclasses (‘Ardh Sam Varnik’- ‘अर्धसम वर्णिक’, ‘Visham Varnik’- ‘विषम वर्णिक’) were not found, to maintain the hierarchical structure and with a vision of upcoming times, these two were also added so in upcoming times if any ‘Chhand’ is found in these subclasses than that can be included quickly.

Figure 4.7 is representing the percentage-wise popularity based on the found types and subtypes of the Hindi Verses (‘Chhands’-‘छंद’). Maximum 55.9% found Hindi Verses belongs to ‘Sam Varnik Vrutt / Chhand’, which falls under the ‘Varnik Vrutt / Chhand’ class. Rest 30.4%, 11.8%, and 2.0% were from ‘Sam Matrik Chhand’ (‘सम मात्रिक छंद’), ‘Ardh Sam Matrik Chhand’ (‘अर्द्धसम मात्रिक छंद’) and ‘Visham Matrik Chhand’ (‘विषम मात्रिक छंद’) respectively which is a total of 44.1% and falls under ‘Matrik Chhand’ main class. ‘Mukt or Muktak Chhands’ is not having any other type or subtype classes hence not considered in this representation. Also no ‘Chhands’ were found under the ‘Ardh Sam Varnik Chhand / Vrutt’(‘अर्द्धसम वर्णिक छंद / वृत्त’) and ‘Visham Varnik Chhand / Vrutt’(‘विषम वर्णिक छंद / वृत्त’).

After analyzing all the results, it was found out that the lengthy data inputs needed extra time. Hindi Verses (‘Chhands’-‘छंद’), consist of a few numbers of lines, words, or characters, gets recognized faster compare to ‘Hindi Verses (‘Chhands’-‘छंद’) made up of a more number of lines, words, and characters.

Some Hindi Verses have higher accuracy than others as their rule is simple, so it is easier to construct those verses, and detection is also easier. But, on the other hand, wherever some Hindi Verses have the least accuracy because the construction rule of such Hindi Verses is more complex, the detection is tricky, and the errors are more in creation and detection both.

4.8 Discussions

Till now, it is known and understood that it is a complex job to make people know about the Hindi Verses (‘Chhands’) while this research work tried to cover most out of it, which can help both humans and computers understand the Hindi Verses (‘Chhand’). The initial challenge for this research work was collecting and analyzing the information and validating the same. Information found from the different sources was full of new aspects and knowledge but conflicting and incomplete also. Sometimes it was much difficult to treat something as ‘Chhand’ too. And the best example for that is ‘Ardhali’ (‘अर्धाली’) which is not a Hindi Verse but a half part of the popular Hindi Verse named ‘Chaupai’ (‘चौपाई’), which is consist of two stanzas only. At the same time, ‘Chaupai’ (‘चौपाई’) is the Hindi Verse that is made up of four stanzas and considered as actual Hindi Verse. Such issues are there because the people who really know and understand is less and people usually write more Muktak Verses intend of the learning the rule and construction the rules-based ‘Chhands’ like Matrik and Varnik Verses due to that only it is becoming difficult and the actual Hindi Verse are disappearing. A long-time duration (December 2017 to January 2021) was spent on the automatic metadata generator’s data collection and rule-based modeling.

There are many Hindi Verses (‘Chhands’) types, subtypes, and can be further sub-sub types. For this research work, only the first three levels were considered. The complexity in itself was that much that it can be understood with several aspects. Each Verse type and subtype is made up of a different set of rules. In that rules also, if it is coming from the Matrik Verses stream, then rules are different, and if it’s coming from the Varnik Verses stream, then again, rules are different.

For Matrik Verses, the Quantity calculation or ‘Matra Gadna’ is unique, and for the Varnik Verses, the mapping of sequences using the eight types of ‘Gana’ is again different. Apart from these various rules, while modeling the basic rules if any changes occurred, it needs to thought of the impact of the same on all the other modeled rules of the construction of Hindi Verses used to identify and detect.

Dealing with the Hindi Verses (‘Chhands’), which are made up of combining the rules of some existing Hindi Verses (‘Chhands’). This is another level of challenge because whenever the metadata generator check, it will found more than one Hindi Verses and end up saying that two or any number of Hindi Verses were found. Still, this metadata generator is designed by overcoming all such issues and detected the single specific name of Hindi Verse (‘Chhand’) which is made up of the combination of existing Hindi Verses.

In Muktak Verses, as it is known that if any Hindi Verse not gets detected under the Matrik Verses or Varnik Verses, it goes into the Matrik Verses. But one might have used the multiple types of verses references while writing the Muktak Verses. Then all the reference verses used get detected and listed while populating the automatic metadata. This is achieved through the different separation and the combination of those separations of stanzas. That is also a tedious thing to accomplish with context to Hindi Verses.

To make the Hindi Verses easier to understand, the stop words were filtered and better understanding the meanings as the examples of the words were produced with the help of the Hindi wordnet.

As same as the Hindi Verse the efforts were made for the collection of the rules and taxonomic structure creation so it can be used for the research, similar issues were faced while performing this also. But Here issues are different because there are several people who know about the ‘Alankars’ and use also but the issue is that all the ‘Alankars’ were not managed as per the research standards. So, efforts were required for the systematic taxonomic structure creation. Also, to open up a new stream of research the automatic identification of ‘Alankars’ is introduced which was not the part of this research work in initial objectives.

Chapter 5

Conclusions, Major Contributions, and Scope of Further Work

5.1 Conclusions

In conclusion, it can be said that to start this research work, In the beginning, the Hindi Verses were arranged with their hierarchy in order. Also, by keeping the various aspects of computational linguistics-based research work in mind, the rules of stanza have been identified, verified, and arranged adequately by this research work.

In this research work, Hindi Verses are classified into standard classes for better and easier hierarchical management. During the research work, it was felt that the detection of Hindi Verses based on various rules is a complex process. It takes longer to find Hindi Verses made up of complex regulations.Special exception rules slow down execution time because it takes longer to moderate and check data. The Hindi Verses made up of one or more stanza rules also take longer because they have to pass more than once during meta-data generation processing to detect them.

Additionally, this research work can filter out an existing list of stop words and remove the stop words. The meaning and example usage of the words can be suggested with Wordnet integration, which helps understand the poems better. Words that were not still included in Wordnet are also filtered. After doing this research work, it can be powerfully expressed that the systematic Hindi Verses rules and the concept of automatic metadata generation for Hindi poetry can automatically generate meaningful metadata. The research work done so far is enough for the incoming researchers to think about some other relevant aspects and open a new way to contribute to the natural language processing and computational linguistics research domain.

Several aspects which can be concluded here is that dealing with Hindi text is tedious due to the diacritics, ligatures, and complex word-formation. Such things are not there in other languages such as English. Moreover, if Poetry and Prose are considered, dealing with poetry is challenging in Hindi with context to Hindi Verses because Hindi Verses are made up of so many complex and exceptional rules. From the perspective of NLP, dealing with the context of words in poetry is way more complicated than managing in prose.

From the perspective of the Indian Knowledge Management Systems, Hindi Verse is coming from long back from ancient times. However, it was observed that they are getting extinct. This research work is an effort to save such precious knowledge, which is an asset in the true sense from the ancestors. People who know about the Hindi Verses are less. Even if they exist more in numbers, this research is helpful for everybody as it is tough to remember all the rules and regulations for all the types, subtypes, and subsubtypes of the Hindi Verses.

The major-specific contribution of this research work are included in the Section 5.2 Major Contributions.

5.2 Major Contributions

- A hierarchical structure of Hindi Verse types is constructed.
- The construction rules of Hindi Verses were identified and validated manually.
- An automatic metadata generator based on Hindi Verse's rule-based modeling with a computational linguistics perspective is developed.
- A Hindi poetry corpus based on the Hindi Verses is ready, and the utility to collect data systematically for corpus creation for Hindi Poetry is developed.
- Apart from the core objectives, To explore the 'Alankar' detection, a separate module for 'Alankar' detection was also developed. Similar to 'Chhands', the hierarchical structure was constructed.

5.3 Limitation

- The third type of character called ‘Plut’ used in Musical Composition is not considered in this research because it only deals with the text-based approach.
- The metadata generator can also work well with other Indian regional languages with minor or no changes. Still, especially for Sanskrit, some significant change will be required as the writing style, and formation of words are much more complex in Sanskrit.
- Most affecting exceptions may impact the results positively or negatively if the corpus is increased or decreased.
- Sufficient information and examples of ‘Ardh Sam Varnik’ and ‘Visham Varnik’ were not found yet included in the automatic metadata generator with a provision that it may be found then it can be implemented with ease in upcoming times.

5.4 Scope of Further Work

- Continuous evolution and adaptation of the newly known rule and new ‘Chhands’.
- Large data corpus creation for upcoming technologies such as machine learning (ML).
- More ‘Alankars’ can be integrated will some different approach.
- ‘Ras’ detection based on emotion detection.
- The complete work is based on a computational linguistics text-based approach so that a speech-based approach can be explored

References

[1] W. Contributors, “Natural Language Processing - Wikipedia,” 2021. [Online]. Available: https://en.wikipedia.org/wiki/Natural_language_processing

[2] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition [draft]*, third edit ed., 2020. [Online]. Available: https://web.stanford.edu/$\sim$jurafsky/slp3/ed3book_dec302020.pdf

[3] J. Glass, “A brief introduction to automatic speech recognition,” 2018. [Online]. Available: http://www1.cs.columbia.edu/~mcollins/6864/slides/asr.pdf

[4] H. Cooper, B. Holt, and R. Bowden, “Sign Language Recognition,” in *Visual Analysis of Humans*. London: Springer London, 2011, pp. 539–562. [Online]. Available: http://link.springer.com/10.1007/978-0-85729-997-0_27

[5] A. Kao and S. R. Poteet, *Natural Language Processing and Text Mining*, A. Kao and S. R. Poteet, Eds. London: Springer London, 2007, vol. 98, no. 2. [Online]. Available: https://link.springer.com/book/10.1007/978-1-84628-754-1http://link.springer.com/10.1007/978-1-84628-754-1

[6] K. Chowdhary, “Natural language processing,” *Fundamentals of artificial intelligence*, pp. 603–649, 2020. [Online]. Available: http://krchowdhary.com/me-nlp12/nlp-01.pdf

[7] E. D. Liddy, “Natural Language Processing,” in *Encyclopedia of Library and Information Science*, 2nd ed. Taylor & Francis, 2003, pp. 2126–2136. [Online]. Available: https://books.google.co.in/books?id=Sqr-%5C_3FBYiYC

[8] E. M. Bender and A. Koller, “Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, no. 2. Online: Association for Computational Linguistics, jul 2020, pp. 5185–5198. [Online]. Available: https://aclanthology.org/2020.acl-main.463

[9] S. R. Bowman and G. E. Dahl, “What Will it Take to Fix Benchmarking in Natural Language Understanding?” *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4843–4855, apr 2021. [Online]. Available: http://arxiv.org/abs/2104.02145

[10] J. Bateman and M. Zock, *Natural Language Generation*, R. Mitkov, Ed. Oxford University Press, sep 2012, vol. 1, no. April 2018. [Online]. Available: http://oxfordhandbooks.com/view/10.1093/oxfordhb/9780199276349.001.0001/oxfordhb-9780199276349-e-15

[11] S. Gehrmann, T. Adewumi *et al.*, “The GEM Benchmark: Natural Language Generation, its Evaluation and Metrics,” feb 2021. [Online]. Available: http://arxiv.org/abs/2102.01672

[12] W. Contributors, “Computational Linguistics - Wikipedia,” 2021. [Online]. Available: https://en.wikipedia.org/wiki/Computational_linguistics

[13] J. Henderson, “The Unstoppable Rise of Computational Linguistics in Deep Learning,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, may 2020, pp. 6294–6306. [Online]. Available: https://www.aclweb.org/anthology/2020.acl-main.561

[14] L. Schubert, “Computational Linguistics,” 2014. [Online]. Available: https://plato.stanford.edu/archives/spr2020/entries/computational-linguistics/

[15] Ethnologue.com, “Languages of the world - Ethnologue,” 2021. [Online]. Available: https://www.ethnologue.com/guides/how-many-languages

[16] Ethnologue.com, “What are the top 200 most spoken languages? - Ethnologue,” 2021. [Online]. Available: https://www.ethnologue.com/guides/ethnologue200

[17] W. Contributors, “Hindi - Wikipedia,” 2021. [Online]. Available: https://en.wikipedia.org/wiki/Hindi

[18] W. Contributors, “Devanagari - Wikipedia,” 2021. [Online]. Available: https://en.wikipedia.org/wiki/Devanagari

[19] I. Unicode, “The Unicode Standard, Version 13.0.0.” 2020. [Online]. Available: http://www.unicode.org/versions/Unicode13.0.0/

[20] Alphabetsin Hindi, “How Many Letters Are There In Hindi Varnamala.” [Online]. Available: https://alphabetsinhindi.com/how-many-letters-are-there-in-hindi-varnamala/

[21] Q. Authors, “What are the alphabets in the Hindi language?” [Online]. Available: https://www.quora.com/What-are-the-alphabets-in-the-Hindi-language

[22] BBC, “A Guide to Hindi - The Hindi alphabet,” 2021. [Online]. Available: https://www.bbc.co.uk/languages/other/hindi/guide/alphabet.shtml

[23] Omniglot, “Devanāgarī alphabet,” 2021. [Online]. Available: https://www.omniglot.com/writing/devanagari.htm

[24] W. Contributors, “Literature - Wikipedia,” 2021. [Online]. Available: https://en.wikipedia.org/wiki/Literature

[25] C. Davis, “Paul Valéry and the Truth of Prose and Poetry,” *Orbis Litterarum*, vol. 43, no. 3, pp. 260–269, 1988.

[26] W. Contributors, “Prose - Wikipedia,” 2021. [Online]. Available: https://simple.wikipedia.org/wiki/Prose

[27] W. Contributors, “Poetry - Wikipedia,” 2021. [Online]. Available: https://en.wikipedia.org/wiki/Poetry

[28] R. Mittal and A. Garg, “Text extraction using OCR: A Systematic Review,” *Proceedings of the 2nd International Conference on Inventive Research in Computing Applications, ICIRCA 2020*, pp. 357–362, 2020.

[29] K. M. Yindumathi, S. S. Chaudhari, and R. Aparna, “Analysis of Image Classification for Text Extraction from Bills and Invoices,” *2020 11th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2020*, 2020.

[30] M. K. Audichya and J. R. Saini, “A Study to Recognize Printed Gujarati Characters Using Tesseract OCR,” *International Journal for Research in Applied Science and Engineering Technology*, vol. V, no. IX, pp. 1505–1510, sep 2017.

[31] M. K. Audichya and J. R. Saini, “An Overview of Optical Character Recognition for Gujarati Typed and Handwritten Characters,” in *The Journey of Indian Languages: Perpectives on Culture and Society*, 2019, pp. 144–151. [Online]. Available: http://14.139.122.13:8080/jspui/bitstream/123456789/479/1/EnglishVolume-2-144-151.pdf

[32] W. Contributors, “Hanuman Chalisa - Wikipedia,” 2021. [Online]. Available: https://en.wikipedia.org/wiki/Hanuman_Chalisa

[33] W. Contributors, “Chhand - Wikipedia,” 2021. [Online]. Available: https://hi.wikipedia.org/s/a5j

[34] W. Contributors, “Alankar - Wikipedia,” 2021. [Online]. Available: https://hi.wikipedia.org/s/9ll

[35] W. Contributors, “Ras - Wikipedia,” 2021. [Online]. Available: https://hi.wikipedia.org/s/a4j

[36] M. Howladar, “An Introduction to Sanskrit Chanda,” *IJRAR- International Journal of Research and Analytical Reviews*, vol. 5, no. 3, pp. 1893–1898, 2018. [Online]. Available: http://ijrar.com/upload_issue/ijrar_issue_1564.pdf

[37] G. Laxminarayan, “Kavyakala,” 2021. [Online]. Available: https://kavyakala.blogspot.com/

[38] S. Rahul, “Swasthya Hindi Samaj,” 2021. [Online]. Available: http://hhindisamaj.blogspot.com/

[39] “Bhartiya Chhand Vidhan - Open Books Online,” 2021. [Online]. Available: http://openbooksonline.com/group/chhand

[40] “Chhand - Bharatkosh Gyan Ka Hindi Mahasagar,” 2021. [Online]. Available: https://bharatdiscovery.org/india/%E0%A4%9B%E0%A4%A8%E0%A5%8D%E0%A4%A6#gsc.tab=0

[41] “Chhand in Hindi - Hindi Meaning,” 2021. [Online]. Available: https://www.hindimeaning.com/2018/02/chhand-in-hindi.html

[42] “Hindi Sahitya,” 2021. [Online]. Available: https://www.hindisahitya.org/

[43] M. Howladar, “Importance of the Vedangas : An Analysis,” vol. 7969, no. 77, pp. 77–85, 2016.

[44] W. Contributors, “Pingala - Wikipedia,” 2021. [Online]. Available: https://en.wikipedia.org/wiki/Pingala

[45] W. Contributors, “Bhartiya Chhandshastra - Wikipedia,” 2021. [Online]. Available: https://hi.wikipedia.org/s/ztf

[46] *Agni Puran*. Gita Press, Gorakhpur. [Online]. Available: https://book.gitapress.org/product-style-5/gita-press-570/

[47] W. Contributors, “Vedic Chhand - Wikipedia,” 2021. [Online]. Available: https://hi.wikipedia.org/s/7zbf

[48] W. Contributors, “Gayatri Mantra - Wikipedia,” 2021. [Online]. Available: https://en.wikipedia.org/wiki/Gayatri_Mantra

[49] J. Prasad, *Chhand Prabhakar*. Jagannath Press, Bilas Pur, 1935.

[50] N. Das, *Hindi Chhandolakshan*. Vani Prakashan, 2000.

[51] J. Prasad, *Chhand Sarawali*. Jagannath Press Bilas Pur, 1917. [Online]. Available: https://archive.org/details/in.ernet.dli.2015.478241

[52] e. Repository, "Chhand evm Uske Bhed - I," pp. 1–14, 2021. [Online]. Available: http://egyankosh.ac.in/bitstream/123456789/27894/1/Unit-35.pdf

[53] W. Contributors, "Bhartiya Kavyshastra - Alankar - Wikibooks," 2021. [Online]. Available: https://hi.wikibooks.org/w/index.php?title=%E0%A4%AD%E0%A4%BE%E0%A4%B0%E0%A4%A4%E0%A5%80%E0%A4%AF_%E0%A4%95%E0%A4%BE%E0%A4%B5%E0%A5%8D%E0%A4%AF%E0%A4%B6%E0%A4%BE%E0%A4%B8%E0%A5%8D%E0%A4%A4%E0%A5%8D%E0%A4%B0_(%E0%A4%A6%E0%A4%BF%E0%A4%B5%E0%A4%BF)/%E0%A4%85%E0%A4%B2%E0%A4%82%E0%A4%95%E0%A4%BE%E0%A4%B0&oldid=55920

[54] "Alankaar(Figure of speech) - Hindi Grammar," 2021. [Online]. Available: http://hindigrammar.in/alankar.html

[55] "Alankar Definition Type and Examples." [Online]. Available: https://www.mycoaching.in/2018/09/alankar.html

[56] K. K. Kushwah and B. K. Joshi, "Rola: An Equi-Matrik Chhand of Hindi Poems," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 15, no. 3, pp. 362–364, 2017. [Online]. Available: https://sites.google.com/site/ijcsis/

[57] B. Joshi and K. Kushwah, "A Novel Approach to Automatic Detection of Chaupai Chhand in Hindi Poems," in *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*. IEEE, 2018, pp. 223–228. [Online]. Available: https://ieeexplore.ieee.org/document/8675052

[58] J. Kaur and J. R. Saini, "Automatic Punjabi poetry classification using machine learning algorithms with reduced feature set," *International Journal of Artificial Intelligence and Soft Computing*, vol. 5, no. 4, p. 311, 2016. [Online]. Available: http://www.inderscience.com/link.php?id=81353https://dl.acm.org/doi/10.1504/IJAISC.2016.081353

[59] J. Kaur and J. R. Saini, "Punjabi Poetry Classification: The Test of 10 Machine Learning Algorithms," in *Proceedings of the 9th International Conference on Machine Learning and Computing*, vol. Part F1283. New

York, NY, USA: ACM, feb 2017, pp. 1–5. [Online]. Available: https://dl.acm.org/doi/10.1145/3055635.3056589

[60] J. Kaur and J. R. Saini, "Automatic classification of Punjabi poetries using poetic features," *International Journal of Computational Intelligence Studies*, vol. 7, no. 2, p. 124, 2018. [Online]. Available: http://www.inderscience.com/link.php?id=10016073https://dl.acm.org/doi/10.5555/3282660.3282663

[61] J. Kaur and J. R. Saini, "PuPoCl : Development of Punjabi Poetry Classifier Using Linguistic Features and Weighting," *Infocomp*, vol. 16, no. 1–2, pp. 1–7, 2017. [Online]. Available: https://infocomp.dcc.ufla.br/index.php/infocomp/article/view/546/491

[62] J. Kaur and J. R. Saini, "Designing punjabi poetry classifiers using machine learning and different textual features," *International Arab Journal of Information Technology*, vol. 17, no. 1, pp. 38–44, 2020.

[63] S.-e. Hansen, "Solving Classification Problems through Automatic Programming," Ph.D. dissertation, 2007.

[64] M. R. Abbas and K. H. Asif, "Computing prosody to detect the Arud meter in Punjabi Ghazal," *Sadhana - Academy Proceedings in Engineering Sciences*, vol. 45, no. 1, pp. 1–20, 2020. [Online]. Available: https://doi.org/10.1007/s12046-020-01458-3

[65] A. Pandian, P. Maurya, and N. Jaiswal, "Author identification of hindi poetry," *International Journal of Scientific and Technology Research*, vol. 9, no. 3, pp. 3791–3795, 2020.

[66] P. B. Bafna and J. R. Saini, "On Exhaustive Evaluation of Eager Machine Learning Algorithms for Classification of Hindi Verses," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 2, pp. 181–185, 2020. [Online]. Available: www.ijacsa.thesai.orghttp://thesai.org/Publications/ViewPaper?Volume=11&Issue=2&Code=IJACSA&SerialNo=24

[67] P. B. Bafna and J. R. Saini, "Hindi Verse Class Predictor using Concept Learning Algorithms," in *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*. IEEE, mar 2020, pp. 318–322. [Online]. Available: https://ieeexplore.ieee.org/document/9074850/

[68] P. Bafna and J. R. Saini, "Hindi Poetry Classification using Eager Supervised Machine Learning Algorithms," in *2020 International Conference on Emerging Smart Computing and Informatics (ESCI)*. IEEE, mar 2020, pp. 175–178. [Online]. Available: https://ieeexplore.ieee.org/document/9167632/

[69] P. B. Bafna and J. R. Saini, "An Application of Zipf's Law for Prose and Verse Corpora Neutrality for Hindi and Marathi Languages," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 3, pp. 261–265, 2020. [Online]. Available: www.ijacsa.thesai.orghttp://thesai.org/Publications/ViewPaper?Volume=11&Issue=3&Code=IJACSA&SerialNo=31

[70] P. B. Bafna and J. R. Saini, "BaSa: A Technique to Identify Context based Common Tokens for Hindi Verses and Proses," in *2020 International Conference for Emerging Technology (INCET)*. IEEE, jun 2020, pp. 1–4. [Online]. Available: https://ieeexplore.ieee.org/document/9154124/

[71] B. K. Joshi and K. K. Kushwah, "Sandhi : The Rule Based Word Formation in Hindi," vol. 14, no. 12, pp. 781–785, 2016.

[72] P. Gupta and V. Goyal, "Implementation of Rule Based Algorithm for Sandhi-Vicheda Of Compound Hindi Words," *International Journal of Computer Science Issues*, vol. 3, pp. 45–49, 2009. [Online]. Available: http://ijcsi.org

[73] J. Kaur and J. R. Saini, "A Study of Text Classification Natural Language Processing Algorithms for Indian Languages," *VNSGU JOURNAL OF SCIENCE AND TECHNOLOGY*, vol. 4, no. 1, pp. 162–167, 2015.

[74] J. Kaur and J. R. Saini, "A Study and Analysis of Opinion Mining Research in Indo-Aryan, Dravidian and Tibeto-Burman Language Families," *International Journal of Data Mining And Emerging Technologies*, vol. 4, no. 2, p. 53, 2014.

[75] A. Kulkarni and G. Huet, Eds., *Sanskrit Computational Linguistics*, ser. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, vol. 5406. [Online]. Available: http://link.springer.com/10.1007/978-3-540-93885-9

[76] C. K. Chung and J. W. Pennebaker, "Linguistic Inquiry and Word Count (LIWC)," *Applied Natural Language Processing*, no. January 1999, pp. 206–229, 2013.

[77] K. Pallavi and R. Mojibur, "A preliminary pragmatic model to evaluate poetry translation," *Babel. Revue internationale de la traduction / International Journal of TranslationBabel / Revue internationale de la traduction / International Journal of TranslationBabel*, vol. 64, no. 3, pp. 434–463, 2018.

[78] A. Yadav, R. K. Chakrawarti, and P. Bansal, "Couplets Translation from English to Hindi Language," in *Lecture Notes in Networks and Systems*, vol. 100, 2020, pp. 285–294.

[79] R. K. Chakrawarti, P. Bansal, and J. Bansal, "Phrase-Based Statistical Machine Translation of Hindi Poetries into English," *Smart Innovation, Systems and Technologies*, vol. 196, pp. 53–65, 2021.

[80] J. R. Saini and J. Kaur, "Kāvi: An Annotated Corpus of Punjabi Poetry with Emotion Detection Based on 'Navrasa'," in *Procedia Computer Science*, vol. 167. Elsevier B.V., 2020, pp. 1220–1229.

[81] K. Pal and B. V. Patel, "Model for Classification of Poems in Hindi Language Based on Ras," in *Smart Innovation, Systems and Technologies*, vol. 141. Springer, 2020, pp. 655–661.

[82] V. Jha, P. Deepa Shenoy, and V. K. R, "Sentiment Analysis in a Resource Scarce Language:Hindi," *International Journal of Scientific and Engineering Research*, vol. 7, no. 9, 2016. [Online]. Available: https://goo.gl/N8GXAUhttps://www.cse.iitb.ac.in/$\sim$pb/papers/gwc18-pyiwn.pdf

[83] Y. Kumar, A. Chugh, D. Mahata, R. Maheshwari, S. Aggarwal, and R. R. Shah, "BHAAV - A text corpus for emotion analysis from Hindi stories," *arXiv*, 2019.

[84] L. Barros, P. Rodriguez, and A. Ortigosa, "Automatic classification of literature pieces by emotion detection: A study on quevedo's poetry," in *Proceedings - 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction, ACII 2013*, 2013, pp. 141–146.

[85] H. R. Tizhoosh and R. A. Dara, "On poem recognition," *Pattern Analysis and Applications*, vol. 9, no. 4, pp. 325–338, 2006.

[86] V. Kumar and S. Minz, "Poem Classification Using Machine Learning Approach," *Advances in Intelligent Systems and Computing*, vol. 236, no. SocProS, pp. 1117–1126, 2014. [Online]. Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-84928041358&partnerID=tZOtx3y1

[87] N. Jamal, M. Mohd, and S. A. Noah, "Poetry classification using support vector machines," *Journal of Computer Science*, vol. 8, no. 9, pp. 1441–1446, 2012.

[88] O. Alsharif, D. Aleshamaa, and N. Ghneim, "Emotion Classification in Arabic Poetry Using Machine Learning," *Article in International Journal of Computer Applications*, vol. 65, no. 16, pp. 10–15, 2013. [Online]. Available: https://www.researchgate.net/publication/284326319_Emotion_Classification_in_Arabic_Poetry_Using_Machine_Learning

[89] S. Hamidi, F. Razzazi, and M. P. Ghaemmaghami, "Automatic meter classification in Persian poetries using support vector machines," in *2009 IEEE International*

Symposium on Signal Processing and Information Technology (ISSPIT). IEEE, dec 2009, pp. 563–567. [Online]. Available: http://ieeexplore.ieee.org/document/5407514/

[90] Z.-S. He, W.-T. Liang, L.-Y. Li, and Y.-F. Tian, "SVM-Based Classification Method for Poetry Style," in *Proceedings of the Sixth International Conference on Machine Learning and Cybernetics*. IEEE Xplore, 2007.

[91] A. Abbasi, H. Chen, and A. Salem, "Sentiment analysis in multiple languages: Feature selection for opinion classification in Web forums," *ACM Transactions on Information Systems*, vol. 26, no. 3, jun 2008.

[92] H. Manurung, G. Ritchie, and H. Thompson, "Towards A Computational Model of Poetry Generation," *Processing*, no. May, 2000.

[93] H. Han, E. Manavoglu, H. Zha, K. Tsioutsiouliklis, C. L. Giles, and X. Zhang, "Rule-based word clustering for document metadata extraction," in *Proceedings of the 2005 ACM symposium on Applied computing - SAC '05*. New York, New York, USA: ACM Press, 2005, p. 1049. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1066677.1066917

[94] X. Yu, M. Tungare, W. Fan, M. Pérez-Quiñones, E. A. Fox, W. Cameron, and L. Cassel, "Using automatic metadata extraction to build a structured syllabus repository," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4822 LNCS. Springer Verlag, 2007, pp. 337–346.

[95] M.-T. Sagri and D. Tiscornia, "Metadata for content description in legal information," in *14th International Workshop on Database and Expert Systems Applications, 2003. Proceedings.* IEEE Comput. Soc, 2003, pp. 745–749. [Online]. Available: http://ieeexplore.ieee.org/document/1232110/

[96] J. L. Klavans, C. Sheffield, E. Abels, J. Lin, R. Passonneau, T. Sidhu, and D. Soergel, "Computational linguistics for metadata building (CLiMB): Using text mining for the automatic identification, categorization, and disambiguation of subject terms for image metadata," *Multimedia Tools and Applications*, vol. 42, no. 1, pp. 115–138, mar 2009.

[97] R. Panjwani, D. Kanojia, and P. Bhattacharyya, "pyiwn: A Python-based API to access Indian Language WordNets," in *Proceedings of the Global WordNet Conference*, 2018, p. 2018. [Online]. Available: https://goo.gl/N8GXAUhttps://www.cse.iitb.ac.in/$\sim$pb/papers/gwc18-pyiwn.pdf

[98] S. Rajendran and S. Arulmozi, "Augmenting Indo-wordnet with Context," Tech. Rep. [Online]. Available: http://www.cfilt.iitb.ac.in/wordnet/webhwn/IndoWordnetPapers/13_iwn_AugmentingIndo-wordnetwithContext.pdf

[99] N. S. Dash, "Polysemy and Homonymy: A Conceptual Labyrinth," Tech. Rep. [Online]. Available: http://www.cfilt.iitb.ac.in/wordnet/webhwn/IndoWordnetPapers/08_iwn_PolysemyandHomonymy.pdf

[100] A. Bakliwal, P. Arora, and V. Varma, "Hindi subjective lexicon: A lexical resource for Hindi polarity classification," *Proceedings of the 8th International Conference on Language Resources and Evaluation, LREC 2012*, pp. 1189–1196, 2012.

[101] V. Jha, N. Manjunath, P. Deepa Shenoy, and V. K. R, "Hindi Language Stop Words List," *Mendeley Data*, vol. V1, 2018. [Online]. Available: https://data.mendeley.com/datasets/bsr3frvvjc/1

[102] J. Meghani, "Elegiac "Chhand" and "Duha" in Charani Lore," *Asian Folklore Studies*, vol. 59, no. 1, p. 41, 2000.

[103] W. Contributors, "Rudrashtadhyayi," 2021. [Online]. Available: https://hi.wikipedia.org/s/7eh9

[104] W. Contributors, "Yajurveda," 2021. [Online]. Available: https://en.wikipedia.org/wiki/Yajurveda

[105] G. Press, *Rudrashtadhyayi - Gita Press Gorakhpur*. Gita Press Gorakhpur. [Online]. Available: https://archive.org/details/RudrashtadhyayiGItaPressGorakhpur

[106] S. Rahul, "Swasth Hindi Samaj," 2018. [Online]. Available: http://hhindisamaj.blogspot.com/2018/05/blog-post_13.html

[107] "Chhand Definition Type and Examples." [Online]. Available: https://www.mycoaching.in/2018/09/chhand.html

[108] Statista, "Python Remains Most Popular Programming Language," 2020. [Online]. Available: https://www.statista.com/chart/21017/most-popular-programming-languages/

Publications

List of Publications

1. M. K. Audichya and J. R. Saini, "Computational linguistic prosody rule-based unified technique for automatic metadata generation for Hindi poetry," *2019 1st International Conference on Advances in Information Technology (ICAIT)*, Jul. 2019, doi: 10.1109/icait47043.2019.8987239.

2. M. K. Audichya and J. R. Saini, "Stanza Type Identification using Systematization of Versification System of Hindi Poetry," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 1, 2021, doi: 10.14569/ijacsa.2021.0120117.

3. M. K. Audichya and J. R. Saini, "Towards Natural Language Processing with Figures of Speech in Hindi Poetry," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 3, 2021, doi: 10.14569/ijacsa.2021.0120316.

Patents

NIL