

# Stanza Type Identification using Systematization of Versification System of Hindi Poetry

Milind Kumar Audichya<sup>1</sup>

Computer Science Department  
Gujarat Technological University  
Ahmedabad, India

Jatinderkumar R. Saini<sup>2\*</sup>

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

**Abstract**—Poetry covers a vast part of the literature of any language. Similarly, Hindi poetry is also having a massive portion in Hindi literature. In Hindi poetry construction, it is necessary to take care of various verse writing rules. This paper focuses on the automatic metadata generation from such poems by computational linguistics integrated advance and systematic, prosody rule-based modeling and detection procedures specially designed for Hindi poetry. The paper covers various challenges and the best possible solutions for those challenges, describing the methodology to generate automatic metadata for “Chhand” based on the poems’ stanzas. It also provides some advanced information and techniques for metadata generation for “Muktak Chhands”. Rules of the “Chhands” incorporated in this research were identified, verified, and modeled as per the computational linguistics perspective the very first time, which required a lot of effort and time. In this research work, 111 different “Chhand” rules were found. This paper presents rule-based modeling of all of the “Chhands”. Out of the all modeled “Chhands” the research work covers 53 “Chhands” for which at least 20 to 277 examples were found and used for automatic processing of the data for metadata generation. For this research work, the automatic metadata generator processed 3120 UTF-8 based inputs of 53 Hindi “Chhand” types, achieved 95.02% overall accuracy, and the overall failure rate was 4.98%. The minimum time taken for the processing of “Chhand” for metadata generation was 1.12 seconds, and the maximum was 91.79 seconds.

**Keywords**—Chhand; computational linguistics; Hindi; metadata; poetry; prosody; stanza; verse

## I. INTRODUCTION

Hindi (“हिंदी”) is known as a prevalent language. According to India’s 2011 census, there were 322 million native speakers with Hindi as their first language [1]. The script is required to write any language. For the Hindi language, the writing script is Devanagari (“देवनागरी”), which is fourth in the world when it comes to the most widely adopted writing systems [2]. With the help of the Devanagari script, more than 120 languages are written all over the world. As per The Unicode Standard, Version 13.0, the Devanagari Unicode range is 0900–097F [3].

Poetries hold an irreplaceable place in the world of literature in every language. Any poem’s creation usually follows some specific patterns or rules known as prosody or poetics. Based on the prosody rules, it can be detected and decided that what kind of poem or part of the poem is, but the patterns may differ from language to language, and even in the

same language, there can be plenty of prosody rules-based patterns [4].

There are two types of language processing approaches in the computational linguistics research domain: the text-based and speech-based approaches [5]. Both methods are required and play a vital role in research in the context of poetry. The text-based system is for working with the significant part of text-oriented rules, and for speech-related practices and patterns, the speech-based approach can be useful to fulfill the research demands. These approaches are adopted and followed based on the need and the nature of the research problem [6].

This research work revolves around Hindi poetry and its different prosody related rules. A significant part of the prosody rules is composed of the order of letters and their frequency. With the text-based approach of computational linguistics, the practices of the composition of stanza were initially systematically classified. Furthermore, the rules were used in the best possible way to carry out the rule-based modeling for the generation of the metadata automatically. In this research work, A proper classification structure will be introduced for Hindi verses. The research will also attempt to detect and identify the Hindi verse based on their appropriate formation rules.

Many verses are written in the Hindi language. This composition of verses in Hindi has been from ancient times. The knowledge hidden behind the rules of the creation of verses has inspired us to do this research work. The authors strongly believe that this research work will prove a milestone to preserve these verses’ composition and give a new direction in computational linguistics research.

The rest of the paper has five major sections: literature review, the knowledge base about the Hindi stanza, Methodology, Results, and Conclusion. One can get a better idea of current research need or gap by going through the literature review first. The next part explains the systematic classification of the stanza specially introduced by this research work. Further, to understand the methodology, formation, and calculation, details of the stanzas are discussed in depth. Based on the various test and experiments, the result section focuses on the outcomes. Finally, based on the complete research work, the overall conclusion is discussed in the last conclusion section, consisting of all significant findings, developments, and results during the research journey.

\*Corresponding Author

## II. LITERATURE REVIEW

A literature review is a fundamental part of any research work. For the same, best efforts had been given to find out the relevant research works. Research work directly related to the metadata generation, related to the Hindi Prosody, and precisely basis on the Computation Linguistics not found. Some nearby research work related to computational linguistics or metadata generation were seen, which are enlighten here.

Efforts were made to find Indian regional languages related research works to know the research's standing specifically in the Indian languages segment. Moreover, it was found that different research work focusing on the various aspects of problems in Indian language-based studies. Audichya and Saini [7] introduced a way through computational linguistics approach for the automatic metadata generation based on the unified rule-based technique for Hindi poetries. They achieved nearly 98% of correct results. Rest errors were due to some input and provided data-related issues. Joshi and Kushwah [8] did research studies that emphasize the detection of 'चौपाई' (Chaupai – A type of Hindi verses) and achieved more than 85% accuracy in detection. They found the issues because some poets usually increase or decrease the 'Matras' to maintain the rhythm or flow due to some different structured but similar sounding words in the end. They also did some research on another Hindi verse named 'रोला' (Rola – A type of Hindi verses) detection and were able to achieve around 89% of accuracy [9]. The remaining accuracy was not gained due to assumptions of the poets while creating and because of the higher sum value of 'मत्रा' (Matras – Quantity) than the expected threshold.

Bafna and Saini [10] tried to classify the Hindi verses based on the various Machine Learning algorithms. They did a comparative study of SVM, Decision Tree, Neural Network, and Naive Bayes on 697 poem classification. In another research work, they did Hindi poetry classification using Eager supervised machine learning algorithms and evaluated using the misclassification error [11]. Research work to predict the Hindi verse class using concept learning done by these researchers in which they found that K-nearest neighbors performed better [12].

Kaur and Saini [13] worked on Punjabi Poems' classification using ten different Machine Learning algorithms. In several other research works, they worked on various Punjabi poems using poetic features, linguistic Features, and Weighting [14-15]. They designed a content-based Punjabi poetry classifier using WEKA using different machine learning algorithms in another research work. They found that the Support Vector Machine algorithm was the best performing accuracy of 76.02% [16-17].

Pal and Patel [18] researched the development of a model based on the nine 'Ras' and tried to classify it using Machine Learning modeling. Saini and Kaur [19] also did emotion detection-based research focusing on 'Navrasa' using machine learning algorithm Naïve Bayes (NB) and Support Vector Machine (SVM), SVM performed better with 70.02% overall accuracy.

Bafna and Saini [20] also worked for the Hindi and Marathi language Prose and Verse application-based researches. Apart from that, research work from the same researchers introduced in which they have given a technique for Hindi Verses and Proses to identify context-based standard tokens [21].

Some research work in the Sanskrit language based on Computational Linguistics was also found [22]. Apart from that, other research works for automatic metadata generation [23], text-based document classification [24-25], and computational linguistics-based metadata building research (CLiMB) works were explored too [26].

The internationally well-known foreign language-based research was analyzed to see the research works, current trends, and progress related to the different language-based works. While reviewing, some excellent foreign language-based research works were found. In a research work of Arabic poetry, emotion classification using machine learning was carried out by Alsharif, AlShamaa, and Ghneim [27]. Hamidi, Razzazi, and Ghaemmaghami [28] researched using Support Vector Machines for the Persian automatic meter classification. In some other research works, researchers have explored Arabic, Bengali, Chinese, English, Hindi, Marathi, Oriya, Persian, Punjabi, and Urdu languages [29-31].

Based on this literature review, it was realized that there is still a lack of essential research for the Hindi poem's automatic metadata generation. Before that, a special effort to structure the combination of the universal unified Prosody ('Chhand') rules systematically is the need of the hour. Furthermore, a robust method based on those rules for metadata generation is the demand of this research wing.

## III. IDEA TO SYSTEMATIZE HINDI PROSODY RULES

To know the core idea to systematize the Hindi poetry rules, one needs to know about the Hindi poetry first so let us know about the same in brief. This research work is dealing with the various aspects of Hindi poetry. Let us first know what the essential elements of the construction of the Hindi poems are. There are three necessary parts which are: 'छंद' ('Chhand' - Prosody or Verse), 'अलंकार' ('Alankar' - Figure of speech) and 'स्त्र' ('Ras' - Sentiment). This research work covers the prosody or verse ('छंद') part in-depth, including the proper systematic collection of prosody rules and standardizing the classification for future research purposes. The term 'Chhand' or 'Chhands' is used to address the Hindi verse or stanza in this research paper.

To understand the 'Chhand' better, one must know about the parts of 'Chhands' that need to be taken care of while writing a poem. The six central components for various types of 'Chhand' writing are 1. Stanza ('चरण या पाद' - 'Charan ya Pad'), 2. Characters and Quantity ('वर्ण व मत्रा' - 'Varn Va Matra'), 3. Flow ('गति' - 'Gati'), 4. Pause ('यति' - 'Yati'), 5. End of Stanza ('तुक' - 'Tuk'), 6. Predefined Sequence of Characters ('गण' - 'Gana') [32-34].

These components play a vital role in different rules, and some are having other aspects as well. Let us see each one by one very quickly. Stanzas usually have a fixed number of

characters and quantity based on those characters. There are two types of characters 1. Short ('हस्त'-‘Hrasva’) and 2. Long ('दीर्घ'-‘Dirgha’). ‘Hrasva’ is denoted from the symbol (‘।’) while counting quantity is considered one (1). These (‘अ’, ‘इ’ (॒), ‘उ’ (॒॑), ‘ऋ’ (॑)) characters are regarded as ‘Hrasva’ while calculating quantity. ‘Dirgha’ holds two (2) quantities at the time of quantity calculation and is represented by the symbol (‘॥’). These (‘आ’ (००), ‘ई’ (००॑), ‘ऊ’ (०॒॑), ‘ए’ (०॒॑), ‘ओ’ (॒॑॑), ‘औ’ (॒॑॑॑), ‘अः’ (००ः), ‘अः’ (०॑॑ः)) are considered as ‘Dirgha’. For which knowledge of quantity calculation rules (‘मात्रा गाना नियम’) [35-37] is required, that is covered in section 3.1.1 of this paper. The predefined fixed sequence of characters group made up of three character’s (‘वर्ण’-‘Varna’) combination. There are eight such groups which are known as ‘Gana’ namely: 1. ‘YaGana’ (‘यगण’), 2. ‘MaGana’ (‘मगण’), 3. ‘TaGana’ (‘तगण’), 4. ‘RaGana’ (‘रगण’), 5. ‘JaGana’ (‘जगण’), 6. ‘BhaGana’ (‘भगण’), 7. ‘NaGana’ (‘नगण’), 8. ‘SaGana’ (‘सगण’). Let us see the quantity calculation’s different rules to understand the further analyses better and know more in-depth about the ‘Chhands’.

#### A. Quantity Calculation Rules Simplified

Plenty of poetry rules were tested and authenticated to found appropriate stanza formation rules. Some manual calculations were carried out to know the rules and figure out the facts out of those calculations results. To validate the identified and authenticated rules information from various sources such as advice from some experts, the notes provided by them, some ancient books [32,33,40], online articles, some books which contain some or very minimal parts of ‘Chhands’ were also considered [38-41]. The research work designed and carried out is a very smooth and systematic way to adopt the new or missing rules if some additional information or practice is found in the future that can also be incorporated easily. Here mentioning the only rules considered for the calculations, and there may or may not be the chance of some new or old rules to be found in the future.

To perform the research work as a researcher or become an influential ‘Chhands’ writer, one must know the rules for better results to know more about the poems and understand the ‘Chhands’ from the core. Let us see basic simplified rules of ‘Matra Ganna’ or quantity calculation for Hindi.

The ‘Chhand’ rules are identified, verified, and modeled in the computational linguistics aspect very first time in this research area. It requires a lot of effort and time to scrutinize and model something that not already standardize. Along with that, the special exceptional rules were incorporated.

##### 1) Common Rules

a) Each ‘Hrasva’ Vowel (‘हस्त स्वर’) which are (‘अ, इ, उ, ऋ’) is counted as one quantity and are called Small (‘लघु’ - ‘Laghu’).

b) Each ‘Dirgh’ Vowel (‘दीर्घ स्वर’) which are (‘आ, ई, ऊ, ए, ओ, औ, अं’) is counted as two quantity and known as Large (‘पूर्ण’- ‘Guru’).

c) Each Consonant is also counted as one at the time of quantity calculation and considered ‘Laghu’. The following

are the consonants: ‘क, ख, ग, घ, ङ, च, छ, ज, झ, ब, ठ, ड, ङ, त, थ, द, ध, न, प, फ, ब, भ, म, य, र, ल, व, श, ष, स, ह’. Example: चल = 11, नयन = 111, कलरब = 1111.

d) On applying diacritics of ‘Hrasva’ Vowel (‘हस्त स्वर’) which are (‘॒’, ‘॒॑’, ‘॑’) on any consonant does not change the quantity count and considered as ‘Laghu’ only so counted as one quantity only. Example: लिख=11, सिल=11, हिन=11, सुध=11, बुलबुल=1111, क्रषि=11.

e) On applying diacritics of ‘Dirgh’ Vowel (‘दीर्घ स्वर’) which are (‘०१’, ‘०१०’, ‘०॒॑’, ‘०॒॑॑’, ‘०॒॑॑॑’, ‘०॒॑॑॑॑’, ‘०॒॑॑॑॑॑’) on any consonant change the quantity count and considered as ‘Dirgh’ so counted as two quantity instead of 1. Example: काला=22, चलाना=122, जिला=12, टीला=22, भौंड=21, चंदन=211, रखवाना=1122.

##### 2) Special - Exceptional Rules:

a) A special rule of the ‘Anunasik’ or ‘Ardh Chandrabindu’ (‘ॐ’) is that if ‘Anunasik’ (‘ॐ’) is applied to consonants which are considered as ‘Laghu’ and are not applied with any ‘Harsva’ Diacritic than quantity is regarded as one only as it is treated as ‘Laghu’. If used on the ‘Harsva’, no changes in quantity and considered as two quantities. Example: हैंसी=12, चैंद=21.

b) Ligature or Joint Character at the starting of any word is considered ‘Laghu’ and counted as one quantity only. Example: ज्वर=11, प्रणाम=121, न्याय=21.

c) Ligature or Joint Character at the starting of any word is along with ‘Dirgh’ Vowel diacritic than considered ‘Guru’ and counted as two quantities, and Half Characters value becomes 0. Example: भ्राता=22, जान=21, ध्यान=21.

d) Before ligature, the ‘Laghu’ characters are considered ‘Guru’, and quantities become two instead of one. Example: सत्य=21, चक्षु=21, गर्भ=21, वृक्ष=21, अथवा=221, विनम्र=221.

e) Before the ligature, the ‘Guru’ characters are considered ‘Guru’, and quantity is counted as two quantities. Example: आत्मा=22, प्राप्तांक=221, भास्कराचार्य=21221.

f) If the next character after the ligature is ‘ह’ and ‘Dirgh’ Vowel Diacritic, then half the Characters value becomes zero. Example: कहैया=121, मत्तहार=121, तुम्हारे=122.

g) If the next character after the ligature is ‘ह’ with ‘Laghu’ Vowel Diacritic or Laghu Character, there will be no change in calculation rules. Example: दुल्हन=211, अल्हड=211.

3) Most affecting exceptions: Sometimes at the end of the stanza, if there is ‘Laghu’, it is considered ‘Guru’ as per the pronunciation. Also, some poets usually increase and decrease the diacritics to maintain the flow or their own choices. Some poet usually takes references from the existing ‘Chhand’ creation rules but does not follow those rules completely, which affects the automatic detection. Apart from the poorly formatted and junk characters added by avoiding the formation rules such as emojis, universal special characters,

special characters from other languages considered as the junk characters, and lower down the result accuracy.

After applying these rules, once one gets the allocated quantity, one needs to sum those quantities on a different basis as per the requirement. One example is here for the sum of the word-level quantity sum.

Example: सीताराम = 2221 = 2+2+2+1 = 7, सत्य = 21 = 2+1 = 3

4) ‘*Gana*’ Sequence: For ‘Varnik’ verses the ‘*Gana*’ is used in sequence-related rules, one needs to know more about the ‘*Gana*’, so Table I will show more about ‘*Gana*’. Table I is representing symbolic representation along with the example of all eight ‘*Gana*’.

That is all about the calculation-related rules and ways, which will help us understand the research work better and surely help future research works. These rules were needed to simplify, and a proper standardized flow by putting massive efforts and experiments were required. It was more challenging to manage everything because no such pertinent standard research-level articles or bases were found.

#### B. Structure Creation of Hindi’ Chhands’ from a Research Perspective

When the authors started finding the information related to the ‘Chhands’, it was found that only a tiny amount of properly arranged information is available [34], and whatever is available is also having some contradiction at different sources. For instance, Different rules and creation information of ‘Bujangi Chhand’ were found at various places [42-43]. It has been observed that the available data cannot be used with the research perspectives. With whatever information collected from the different sources which are mentioned in Section 3.1, manually analyzed, validated and once after proper authentication, this decision is made that whatever is available needs to be systemized first and for which an adequate structure creation is required, so this is an effort towards the same.

There are two mainstream types of ‘Chhands’, which are ‘Vedic Chhands’ and ‘Laukik Chhands’. ‘Vedic Chhands’ are coming from Vedas’ ancient times and mostly written in the Sanskrit language, as mentioned in the ‘Agni Purana’ [44]. A saint and mathematician named Pingala [45] have written ‘Chhandashastra’ based on the ‘Chhands’ only in ancient times. One of the best and trendy examples of ‘Vedic Chhand’ is a ‘Chhand’ named ‘Gayatri Chhand (गायत्री छंद)’. Gayatri Mantra, written in the Sanskrit language, is the best example of the ‘Gayatri Chhand’. ‘ॐ भूर्भुवः स्वः तस्मितुवैण्यं भर्गो देवस्यः धीमहि धियो यो नः प्रचोदयात्’ (‘om bhur bhuvah svah tat savitur varenyam bhargo devasya dhimahi dhiyo yo nah prachodayat’ – Let us pray to God who had produced this universe, may He enlighten us with spirituality) [46].

‘Laukik Chhands’ are the verses written by people and are not a part of Vedas. These are reported in both Sanskrit and the Hindi language. As the research work focuses on Hindi poetry, only Hindi ‘Chhands’ will be discussed. These verses can be classified into three classes based on the nature of rules of ‘Chhand’ writing, which are: 1. ‘Matrik Chhands (मात्रिक छंद)’, 2. ‘Varnik Vrutt/Chhands (वर्णिक वृत्त/छंद)’, 3. ‘Mukt/Muktak Chhands (मुक्त/मुक्तक छंद)’.

1) ‘*Matrik Chhands*’: Hindi verses written with the fixed number of quantities in all the four stanzas based on the different predefined ‘Chhand’ creation rules are known as ‘Matrik Chhand’. In this kind of ‘Chhands’, the sequence of characters discussed earlier, ‘Laghu Varna’ and ‘Guru Varna’ or ‘*Gana*’, does not matter. Some most popular ‘Matrik Chhands’ were managed and structured in systematic classes. There are several classes of ‘Matrik Chhands’: 1. ‘Sam Matrik Chhand’, 2. ‘Ardh Sam Matrik Chhand’, 3. ‘Visham Matrik Chhand’. Let us see in a managed and specific structured manner in Table II.

Table II shows the names of ‘Chhand’, classification type, and subtypes of ‘Chhand’ along with the ‘Matra Counts’ used for the detection at the time of rule-based modeling. Apart from these classifications and ‘Matra’ count, some more rules are associated with the specific verses, and these rules change for every verse. A few ‘Sam Matrik Chhand’, some ‘Ardh Sam Matrik Chhand’ and ‘Visham Matrik Chhand’ are shown in Tables III and IV.

Table III shows the information about the ‘Ardh Sam Matrik Chhands’ in which it can be seen that ‘Matra’ count of even and odd stanzas are different. Let’s see something about ‘Visham Matrik Chhands’ as well.

Table IV representing ‘Visham Matrik Chhand’, the remark of ‘Kundliya’ says that it can be authored by combining two verses named ‘Doha’ and ‘Rola’. This section was the most about the ‘Matrik Chhands’. Let us see the next primary class of ‘Chhands’.

TABLE I. ‘*GANA*’ WITH THEIR RESPECTIVE EXAMPLES

| Sr. No. | Gana | Symbolic Representation | Example |
|---------|------|-------------------------|---------|
| 1       | मण   | SSS                     | जागेली  |
| 2       | नगण  | III                     | शब्द    |
| 3       | भगण  | SII                     | पावन    |
| 4       | यगण  | ISS                     | हमारी   |
| 5       | जगण  | I SI                    | जयेश    |
| 6       | रगण  | SIS                     | राधिका  |
| 7       | सगण  | IIS                     | अपनी    |
| 8       | तगण  | SSI                     | दातार   |

TABLE II. SOME ‘SAM Matrik Chhands’ WITH THEIR RESPECTIVE MATRA COUNT

| Sr. No. | Chhand                  | Type        | Sub Type      | Matra Count |
|---------|-------------------------|-------------|---------------|-------------|
| 1       | रोला                    | मात्रिक छंद | सममात्रिक छंद | 24          |
| 2       | हरिगीतिका / हरगीतिका    | मात्रिक छंद | सममात्रिक छंद | 28          |
| 3       | चौपाई                   | मात्रिक छंद | सममात्रिक छंद | 16          |
| 4       | उल्लाला (सममात्रिक)     | मात्रिक छंद | सममात्रिक छंद | 26          |
| 5       | अहोर                    | मात्रिक छंद | सममात्रिक छंद | 11          |
| 6       | तोमर                    | मात्रिक छंद | सममात्रिक छंद | 12          |
| 7       | विभंगी                  | मात्रिक छंद | सममात्रिक छंद | 32          |
| 8       | गीतिका (चंचरी / चर्चरी) | मात्रिक छंद | सममात्रिक छंद | 26          |

TABLE III. SOME ‘ARDH SAM Matrik Chhands’ WITH THEIR RESPECTIVE MATRA COUNT

| Sr.No. | Chhand                    | Type        | Sub Type            | Even Matra Count | Odd Matra Count |
|--------|---------------------------|-------------|---------------------|------------------|-----------------|
| 1      | दोहा                      | मात्रिक छंद | अर्द्धसममात्रिक छंद | 11               | 13              |
| 2      | सोरठा                     | मात्रिक छंद | अर्द्धसममात्रिक छंद | 13               | 11              |
| 3      | उल्लाला (अर्द्धसममात्रिक) | मात्रिक छंद | अर्द्धसममात्रिक छंद | 15               | 13              |
| 4      | बरवै                      | मात्रिक छंद | अर्द्धसममात्रिक छंद | 7                | 12              |
| 5      | रूपमाला / मदन             | मात्रिक छंद | अर्द्धसममात्रिक छंद | 14               | 10              |
| 6      | तारंक                     | मात्रिक छंद | अर्द्धसममात्रिक छंद | 16               | 14              |
| 7      | कुकुभ                     | मात्रिक छंद | अर्द्धसममात्रिक छंद | 16               | 14              |
| 8      | वीर/आल्हा/मात्रिक सर्वेया | मात्रिक छंद | अर्द्धसममात्रिक छंद | 16               | 15              |

TABLE IV. SOME ‘VISHAM Matrik Chhands’

| Sr.No. | Chhand  | Type        | Sub Type         | Remark         |
|--------|---------|-------------|------------------|----------------|
| 1      | कुडलिया | मात्रिक छंद | विषम मात्रिक छंद | दोहा + रोला    |
| 2      | छप्पय   | मात्रिक छंद | विषम मात्रिक छंद | रोला + उल्लाला |

2) ‘Varnik Chhands’: In the creation or writing of ‘Varnik Chhands’, the ‘Gana’ plays a vital role as these verses are based on the characters or ‘Varnas’. The predefined sequences as per specific rules need to be maintained for each ‘Chhand’ according to the different arrangements of ‘Gana’, ‘Laghu’, and ‘Guru’ characters. Table V shows some of the ‘Varnik Chhands’ and the rules and the symbolic representation of the different regulations.

Here it can be seen that the characters’ sequence matters the most and is based on those eight ‘Gana’, ‘Guru’, and ‘Laghu’ characters. Here a few ‘Matrik’ and ‘Varnik’ verses are included only. Similarly, ‘Ardh Sam Varnik’ and ‘Visham Varnik’ information can be managed and organized.

There are plenty of verses available. So, telling how many ‘Chhands’ exist is impossible now, but that does not mean that the remaining possibilities should left, so the significant rules and classification-related information were added for many ‘Chhands’, including the ‘Chhands’ for which examples or information are less available. Fifty-three different verses information were added here, but to add all the verses is does

not seem to be appropriate as the list goes on and on and the remaining ‘Chhands’ are not having much information available. Managed information consisting of these verses how the relevant information of already written poems and if some new types follow specific rules, such new classes can be organized in systematic management. If research demands new or additional information blocks for each ‘Chhand’ can also be added, it was not possible until now due to cluttered and unmanaged raw information.

One more point to be noted here is that this information is just the core detailing of the verse creation structures. Along with that, there are still some more things are there which need to be managed and differs from one verse type to another verse type. The conceptual part will be discussed in the methodology section.

TABLE V. SOME ‘VARNIK CHHANDS’

| Sr. No. | Chhand        | Type   | Sub Type | Varna Count | Sequence                       | Symbolic Representation |
|---------|---------------|--------|----------|-------------|--------------------------------|-------------------------|
| 1       | इन्द्रवज्रा   | वर्णिक | समवर्णिक | 11          | मण, तगण, तगण, गुरु, गुरु       | SSSSSSSSSS              |
| 2       | उपेन्द्रवज्रा | वर्णिक | समवर्णिक | 11          | जगण, तगण, जगण, गुरु, गुरु      | TTTTTTTTTT              |
| 3       | वसंततिलका     | वर्णिक | समवर्णिक | 14          | तगण, भगण, जगण, जगण, गुरु, गुरु | TTTTTTTTTTTTTT          |
| 4       | मालिनी        | वर्णिक | समवर्णिक | 15          | नगण, नगण, मण, यगण, यगण         | SSSSSSSS                |
| 5       | मन्दाक्रान्ता | वर्णिक | समवर्णिक | 17          | मण, भगण, नगण, तगण, गुरु, गुरु  | SSSSSSSSSSSSSS          |

#### IV. METHODOLOGY

Based on the systemized rules, the best attempts were made to provide the best possible concrete concept for the automatic metadata generation for Hindi poetries by incorporating the rules-based unified modeling. This research work consists of multiple existing Hindi verses and trying to cover up every poetry that has been already written and having a systematic writing rule but is not found or missing until now and will be written in the future.

##### A. Data Pre-Processing

UTF-8 standard encoded data for Devanagari was chosen to work with as input for automatic metadata generator to use and integrate with the latest technologies will be more comfortable in the future. The automatic metadata generator expects the information in the form of the UTF-8 based complete poems or a few lines or part of poems [3]. These lines are further processed for the pre-processing of the input for passing it for the further calculations after some necessary trimming and cleaning operations. Once the pre-processed data is ready after the cleaning operation, the cleaned data can be passed for the next separation-related operations. There are several levels of the separation-related processes that occur as per the demand of this research work. The first stage of the separation of data is for the line-level break based on the new line character ‘\n’ as the standard delimiter for separating the lines. After separating

the lines, the lines need to be split further into the parts known as ‘Charans’ or stanzas of the verse. Separation consists of a few delimiters (‘;’, ‘||’, ‘|’). If there is any remaining delimiter that needs to be used, it can be used too easily. After the ‘Charan’ separation, the separate stanzas need to chop into the words by performing word-level separation. Each separated word further needs to be divided into the characters and diacritics.

### B. ‘Chhand’ Detection based on the Classification

One might wonder why this much separation is required, so the straight forward answer to the curiosity going on here is that these different separations are needed at different phases. Based on these separated data, the calculative operations can be performed efficiently and in an organized manner. Several kinds of separated data consisting of the line, stanza, words, and character level separation helps in getting so much meaningful information such as word count, character count, diacritic count, stanza count, the sequence of the words and characters, and these pieces of information help retrieve the more meaningful data while processing the data further.

Let us now understand this concept with an example for better conceptual clarity. Here is an example of one of the Hindi verse type ‘Doha’ from ‘Hanuman Chalisa’ by a well-known poet-saint Tulsidas [47]: ‘पवनतनय संकट हरन, मंगल मूरति रूप | राम लखन सीता सहित, ह्रदय बसहु सुर भूप ||’ (‘Pavan tanay sankat harana, mangala murati roop. Ram lakhana sita sahita, Hriday basahu soor bhoop - Oh! conqueror of the Wind, Destroyer of all miseries, you are a symbol of Auspiciousness. Along with Shri Ram, Lakshman and Sita, reside in my heart. Oh! King of Gods’).

Here if this text is inputted into metadata generation, the separation takes place in different forms after cleaning operations, which are as follow:

Line Separation: (2 Lines)

Line 1: 'पवनतनय संकट हरन, मंगल मूरति रूप',

Line 2: 'राम लखन सीता सहित, हृदय बसह सूर भूप'

### Stanza / ‘Charan’ Separation

### Stanza 1: 'पवनतनय संकट'

### Stanza 2: 'मंगल मूरति रूप'

**Stanza 3: 'राम लखन सीता सहि'**

Stanza 4: 'हृदय बसह सर भप'

#### **Words Separation: (14 Words)**

1. 'पवनतनय', 2. 'संकट', 3. 'हर्ष', 4. 'मंगल', 5. 'मूरति', 6. 'रूप', 7. 'राम',

8. 'लखन', 9. 'सीता', 10. 'सहित', 11. 'हृ

## Stanza Wise Words Sep

1. i. 'पवनतनय', ii. 'संकट', iii. 'हर्ष'

2. i. 'मंगल', ii. 'मूरति', iii. 'रूप',

3. i. 'राम', ii. 'लखन', iii. 'सीता', iv. 'सहित'

These are the type of different separations. Separation of the diacritic count, 'Guru' and 'Laghu' diacritic counts for the

diacritic count stats were done. To add how many times half characters were used the count of the ‘ং’ is used. After this, all the main parts come into the implementation, which is the calculation based on all the rules which can be seen under 3.1 Quantity Calculation Rules Simplified. Let us know the calculation mechanism for detecting the verse, type, subtype, and much more. Let us see stanza wise ‘Matra’ allocation and counting for the given input:

| पद्धतिय             | संकट  | हल,   | मात्र               | मूलि  | रूप।  |
|---------------------|-------|-------|---------------------|-------|-------|
| 111111              | 211   | 111   | 211                 | 211   | 21    |
| 1+1+1+1+1+1         | 2+1+1 | 1+1+1 | 2+1+1               | 2+1+1 | 2+1   |
| 6                   | 4     | 3     | 4                   | 4     | 3     |
| <b>6+4+3= 13</b>    |       |       | <b>4+4+3= 11</b>    |       |       |
| राम                 | लखन   | सीता  | सहित,               | हनुम  | बस्तु |
| 21                  | 111   | 22    | 111                 | 111   | 11    |
| 2+1                 | 1+1+1 | 2+2   | 1+1+1               | 1+1+1 | 1+1   |
| 3                   | 3     | 4     | 3                   | 3     | 2     |
| <b>3+3+4+3 = 13</b> |       |       | <b>3+3+2+3 = 11</b> |       |       |

The ‘Matra’ allocation and the ‘Matra’ Count will be used further to detect verse after the rule-based modeling of verse rules automatically. The ‘Matra’ allocation is also used after a few modifications and merging for character sequence mapping, specifically for ‘Varnik’ verses.

After this allocation and ‘Matra’ counting, the input passes through the different set of rule-based methods specifically designed for the specific verse-based rules. Each verse follows its own unique set of rules. The 53 verses were rule-based modeled, which can be detected in a bottom-up approach in which the verse is seen first. Later on, the verse type and subtype can be mapped with the already available and systematically managed list of types and subtypes relationships.

The provided input must follow all the associated rules of that particular verse to detect a verse automatically. Let us see about the given input after the different parts of the processing.

The automatic metadata generator still does not know which kind of verse it is. However, the metadata generator is modeled with rule-based modeling. The rules for the verse named ‘Doha’ are already available, so once it will process the data, it is capable enough to say that the provided input was ‘Doha’. Let us know more about how it can be said automatically. For ‘Doha’ writing, it is a rule that the odd stanzas’ Matra’ count must be 13, and even stanzas ‘Matra’ count must be 11.

Along with that, even stanzas should end with the ‘Laghu’ character. Now when the ‘Matra’ count operation is performed on the 1st and 3rd stanzas, ‘Matra’ counts are 13,13 and 2nd and 4th stanzas ‘Matra’ counts are 11, 11 respectively, also you can see that at the end of 2nd and 4th stanza there is ‘Laghu’ character. After this much modeling, the metadata generator is aware of the input and can say that it is a ‘Doha’. A systematically organized hierarchy from which ‘Doha’ can be mapped as ‘Matrik’ type and ‘Ardh Sam Martik’ subtype.

Similarly, for the ‘Varnik’ verses, there are different character sequence-based rules that work on the sequences related rules specified for the particular verses and are based on the ‘Gana’, ‘Guru’ and ‘Laghu’ sequences only. In ‘Varnik’ verses, each of the verses has its own unique rule and needs to

be managed separately. Similarly, after detecting each ‘Varnik’ verse, its type and subtype can be seen from the mapping with the systematically organized hierarchy of verse.

Even after this much processing, if any verses are not found or detected, they are considered the ‘Muktak’ verses.

#### C. Advance ‘Muktak’ Detection

Thought of advancement concerning research computationally means something which does not exist or available yet. During research, the need for this was felt the most, usually whenever the ‘Chhand’ detection takes place, and the input gets detected as ‘Muktak’. In that case, the input gets chopped into several parts again until it is possible to till ‘Charan’ level separation. Those separated parts are processed again from scratch as separate input and from which the results get recorded. At last, Metadata Generator can tell us that in ‘Muktak’ verses also it tried to find out if any part of the ‘Muktak’ verses is using any ‘Matrik’ or ‘Varnik’ verse rules than that will also be detected. That can be one or more than one ‘Chhand’ rules combination.

Apart from this, while processing the individual input, which uses only one type of ‘Chhand’ rules, some specific part was having some issues. Because of this mechanism, the input first goes into the ‘Muktak’ part. Except for the faulty/issue part, all remaining parts get detected as the specific ‘Chhand’, so metadata generator can generate data as the input is having this particular ‘Chhand’ and is the primary reason for the higher accuracy success rate of this metadata generator.

#### D. Stop Words Filtering

After this, to add a few more advancements into this metadata generator, detecting stop words from the given input is one of those advancements. Stop words are filtered through the list of already carried out hybrid research work [48], specifically on the stop words for Hindi’s stop words. Filtering the stop words is essential because in the next stage, when the terms are processed through wordnet, stop words should not be processed. This filtering gives the metadata generator ability not to process the stop words while getting meanings from the wordnet. It saves a lot of execution time as well as improve the overall efficiency of the metadata generator.

#### E. Wordnet Integration for Meaning and another Example of Words

Sometimes a user might want to know the meaning of the words used in verse, for which after removing the stop word, the Hindi wordnet named ‘pyiwn’ for the meaning of the words and the examples were integrated [49]. The wordnet integration makes the metadata generator more worthy because if a beginner wants to learn and understand Hindi Poetry should be aware of the meaning of the poem’s words. The user also gets one another similar example of that work, capable enough to make sure that how specific word should be used.

The wordnet integration helps the wordnet advancement too. As it separates, the list of the word does not exist in the wordnet still, or the meaning of the word is yet not incorporated in the wordnet. This list can be considered for the improvement of the ongoing wordnet research works as well.

#### F. Example Suggestions of Detected ‘Chhand’

This mechanism is integrated with keeping the scenario in mind that a user who wants metadata about the input might be interested in the same types ‘Chhand’ example of the detected ‘Chhand’ type. It gives the user more ability to know and understand the ‘Chhand’ formation better by comparing the input with the suggested example. The example sets were stored in key-value pair in the JavaScript Object Notation (JSON) file from which any random example gets populated whenever the specific ‘Chhand’ type gets detected.

#### G. Additional Several Utilities

Data collection issues were faced during the research work as the dataset for such research work is not available as of now directly. A systematic approach was required to avoid redundant data, to maintain and manage the collecting data in a decent form. A utility for data collection was also designed to store and check if the entered data already exists or not, and if not, it will add the inputted data. The data collection utility is capable enough to get the single and the multiple inputs from the Comma Separated Value (CSV) files or Text Files.

A utility to generate some random text-based ‘Chhand’ from the collection of words on the given parameters of the character set was also developed to test the metadata generator’s ability to ensure the capacity to handle the ‘Chhand’ types which do not exist currently or can come up in future. The metadata generator generates the data about the provided input by combining all the mentioned methodology parts, gives robust metadata, and strengthens the metadata performance.

## V. RESULTS

Hindi poems can be written using any ‘Chhand’ or any combination of ‘Chhands’. The research work is sufficient enough to help in the automatic generation of the metadata from the Hindi poetries by covering the majority of ‘Chhands’ already and having the capability to incorporate new kinds of ‘Chhands’ in a very systematic manner with ease. Along with the detection of ‘Chhands’, the metadata generator provides several meaningful and useful information. Information includes Word Count, Character Count, Diacritic Count, Quantity (‘Matra’) Count, Symbolical String Representation, stop words, Meaning of the words, along with an example of the use of the word for better understanding. So, the approach which is followed can be understood better as per this automatically generated metadata. The result is an example of metadata output generated by the metadata generator. The same example discussed in this paper since the beginning for a better understanding of the metadata generator’s working.

The metadata generator is modeled based on the rules of the 111 different ‘Chhands’ classification types, subtypes, and subtypes of subtypes. The 53 different types shown in Table VI, ‘Chhands’ data were collected for testing and validation. Each class has at least 20 and a maximum of 277 records based on the data’s availability, which may vary for different types. Total 3120 records were found from various sources and tested, from which 2992 records were detected successfully. One hundred twenty-eight records were not detected due to some reasons such as poorly formatted data, some grammar-

related issues, manipulated words from regional languages, unnecessary uses of the special symbols, and junk character.

The overall accuracy rate based on the results is 95.02%, and the failure rate is 04.98%. Fig. 1 represents a graph of the ‘Chhand’ data found, detected, and not found along with the individual accuracy and failure rates of different ‘Chhands’. Fig. 1 shows that the various ‘Chhands’ accuracy rate varies between 84% to 100%, and the failure rate lies between 0% and 16%. Out of all the 53 different types of ‘Chhands’ data,

only 5 ‘Chhands’ were having an accuracy rate below 90%, and the rest all were between 90 and 100%, and in that also 29 were having 95 or more than 95 and 100% accuracy rate. The lowest accuracy percentage is 84% of ‘Chhapay Chhand’, which is made up of two different ‘Chhands’, which makes its construction and detection complex, and due to that, only the error or issues occurs more. The six best performing ‘Chhands’ with 100% accuracy rate, were ‘Aansu’, ‘Bhujangprayag’, ‘Janak’, ‘Mandakranta’, ‘Muktak’ and ‘Tilka’.

TABLE VI. PROCESSED ‘CHHAND’ DATA DETAILS

| Chhand                  | Found | Detected | Not Detected | Success Rate | Failure Rate | Time (Min.) | Time (Avg.) | Time (Max.) |
|-------------------------|-------|----------|--------------|--------------|--------------|-------------|-------------|-------------|
| 1.Aansu                 | 30    | 30       | 0            | 100.00       | 0.00         | 1.35        | 1.53        | 1.80        |
| 2.Ahir                  | 46    | 43       | 3            | 93.48        | 6.52         | 1.23        | 2.32        | 16.37       |
| 3.Barvai                | 36    | 35       | 1            | 97.22        | 2.78         | 1.28        | 1.58        | 6.33        |
| 4.Bhujangi              | 78    | 77       | 1            | 98.72        | 1.28         | 1.36        | 6.50        | 17.29       |
| 5.Bhujngprayag          | 42    | 42       | 0            | 100.00       | 0.00         | 1.44        | 2.84        | 19.10       |
| 6.Chandrika             | 34    | 33       | 1            | 97.06        | 2.94         | 1.30        | 1.95        | 5.60        |
| 7.Chaupai               | 100   | 98       | 2            | 98.00        | 2.00         | 1.26        | 1.67        | 14.66       |
| 8.Chhappay              | 25    | 21       | 4            | 84.00        | 16.00        | 1.72        | 7.24        | 36.99       |
| 9.Chopaiya              | 20    | 17       | 3            | 85.00        | 15.00        | 1.32        | 2.20        | 5.26        |
| 10.Dhar                 | 35    | 34       | 1            | 97.14        | 2.86         | 1.22        | 4.57        | 14.65       |
| 11.Digpal               | 40    | 37       | 3            | 92.50        | 7.50         | 1.44        | 2.25        | 14.96       |
| 12.Dodhak               | 47    | 45       | 2            | 95.74        | 4.26         | 1.36        | 6.85        | 31.41       |
| 13.Doha                 | 277   | 265      | 12           | 95.67        | 4.33         | 1.21        | 1.62        | 6.40        |
| 14.Drutvilmbit          | 162   | 161      | 1            | 99.38        | 0.62         | 1.20        | 5.53        | 25.81       |
| 15.Durmil Savaiya       | 58    | 56       | 2            | 96.55        | 3.45         | 1.47        | 7.36        | 25.54       |
| 16.Harigitika           | 83    | 80       | 3            | 96.39        | 3.61         | 1.46        | 28.16       | 76.94       |
| 17.Janak                | 36    | 36       | 0            | 100.00       | 0.00         | 1.27        | 1.53        | 1.98        |
| 18.Krupan Dhanakshari   | 29    | 27       | 2            | 93.10        | 6.90         | 1.29        | 2.08        | 5.72        |
| 19.Kukubh               | 35    | 33       | 2            | 94.29        | 5.71         | 1.34        | 2.30        | 14.08       |
| 20.Kundlia              | 35    | 33       | 2            | 94.29        | 5.71         | 1.48        | 3.57        | 35.59       |
| 21.Madira Savaiya       | 60    | 56       | 4            | 93.33        | 6.67         | 1.45        | 13.48       | 91.79       |
| 22.Malini               | 105   | 103      | 2            | 98.10        | 1.90         | 1.31        | 6.45        | 22.10       |
| 23.Mandakrnta           | 153   | 153      | 0            | 100.00       | 0.00         | 1.31        | 10.60       | 68.96       |
| 24.Manharan Dhanakshari | 43    | 40       | 3            | 93.02        | 6.98         | 1.27        | 1.45        | 2.60        |
| 25.Manoram              | 48    | 46       | 2            | 95.83        | 4.17         | 1.40        | 2.85        | 13.83       |
| 26.Marhatha             | 25    | 23       | 2            | 92.00        | 8.00         | 1.40        | 2.01        | 5.45        |
| 27.Muktak               | 58    | 58       | 0            | 100.00       | 0.00         | 1.56        | 11.03       | 21.65       |
| 28.Nidhi                | 75    | 72       | 3            | 96.00        | 4.00         | 1.50        | 1.90        | 7.11        |
| 29.Nishchal             | 30    | 28       | 2            | 93.33        | 6.67         | 1.36        | 2.47        | 14.52       |
| 30.Panchamar            | 43    | 41       | 2            | 95.35        | 4.65         | 1.35        | 5.16        | 15.94       |
| 31.Piyushvarsh          | 50    | 49       | 1            | 98.00        | 2.00         | 1.29        | 1.70        | 13.75       |
| 32.Pramanika            | 39    | 38       | 1            | 97.44        | 2.56         | 1.27        | 7.29        | 18.65       |
| 33.Ras                  | 33    | 31       | 2            | 93.94        | 6.06         | 1.29        | 1.70        | 5.45        |
| 34.Rola                 | 147   | 140      | 7            | 95.24        | 4.76         | 1.31        | 3.68        | 38.87       |
| 35.Roopmala             | 20    | 19       | 1            | 95.00        | 5.00         | 1.59        | 7.58        | 15.55       |
| 36.Sagun                | 24    | 22       | 2            | 91.67        | 8.33         | 1.28        | 2.68        | 14.92       |
| 37.Sar                  | 28    | 25       | 3            | 89.29        | 10.71        | 1.28        | 5.76        | 2.08        |
| 38.Sarsi                | 38    | 36       | 2            | 94.74        | 5.26         | 1.30        | 1.68        | 5.61        |
| 39.Shankar              | 30    | 26       | 4            | 86.67        | 13.33        | 1.31        | 3.28        | 14.61       |
| 40.Shardul Vikridit     | 32    | 29       | 3            | 90.63        | 9.38         | 1.62        | 17.29       | 32.02       |
| 41.Sindhu               | 34    | 32       | 2            | 94.12        | 5.88         | 1.36        | 2.62        | 15.57       |
| 42.Sortha               | 52    | 51       | 1            | 98.08        | 1.92         | 1.26        | 1.49        | 5.22        |
| 43.Sur Dhanakshari      | 38    | 37       | 1            | 97.37        | 2.63         | 1.29        | 1.63        | 5.63        |
| 44.Swagta               | 25    | 23       | 2            | 92.00        | 8.00         | 1.42        | 8.32        | 17.01       |
| 45.Tantak               | 30    | 27       | 3            | 90.00        | 10.00        | 1.53        | 4.60        | 23.64       |
| 46.Tilka                | 54    | 54       | 0            | 100.00       | 0.00         | 1.22        | 3.65        | 15.19       |
| 47.Tomar                | 45    | 41       | 4            | 91.11        | 8.89         | 1.53        | 2.82        | 13.14       |
| 48.Tribhangi            | 27    | 25       | 2            | 92.59        | 7.41         | 1.23        | 4.07        | 16.86       |
| 49.Trotak               | 52    | 51       | 1            | 98.08        | 1.92         | 1.36        | 3.43        | 15.37       |

|                       |              |              |             |                |               |              |               |                |
|-----------------------|--------------|--------------|-------------|----------------|---------------|--------------|---------------|----------------|
| <b>50.Ullala</b>      | 31           | 30           | 1           | 96.77          | 3.23          | 1.28         | 1.82          | 7.45           |
| <b>51.Vanshasth</b>   | 109          | 108          | 1           | 99.08          | 0.92          | 1.28         | 12.66         | 55.10          |
| <b>52.VasantTilka</b> | 238          | 222          | 16          | 93.28          | 6.72          | 1.20         | 3.99          | 22.64          |
| <b>53.Vidhata</b>     | 56           | 53           | 3           | 94.64          | 5.36          | 1.25         | 1.67          | 5.34           |
| <b>Total</b>          | <b>3120</b>  | <b>2992</b>  | <b>128</b>  | <b>5035.21</b> | <b>264.79</b> | <b>71.69</b> | <b>256.45</b> | <b>1020.07</b> |
| <b>Maximum</b>        | <b>277</b>   | <b>265</b>   | <b>16</b>   | <b>100.00</b>  | <b>16.00</b>  | <b>1.72</b>  | <b>28.16</b>  | <b>91.79</b>   |
| <b>Minimum</b>        | <b>20</b>    | <b>17</b>    | <b>0</b>    | <b>84.00</b>   | <b>0.00</b>   | <b>1.20</b>  | <b>1.45</b>   | <b>1.80</b>    |
| <b>Average</b>        | <b>58.87</b> | <b>56.45</b> | <b>2.42</b> | <b>95.00</b>   | <b>5.00</b>   | <b>1.35</b>  | <b>4.84</b>   | <b>19.25</b>   |

### Chhand Detection Results

Detection Data, Success and Failure rates

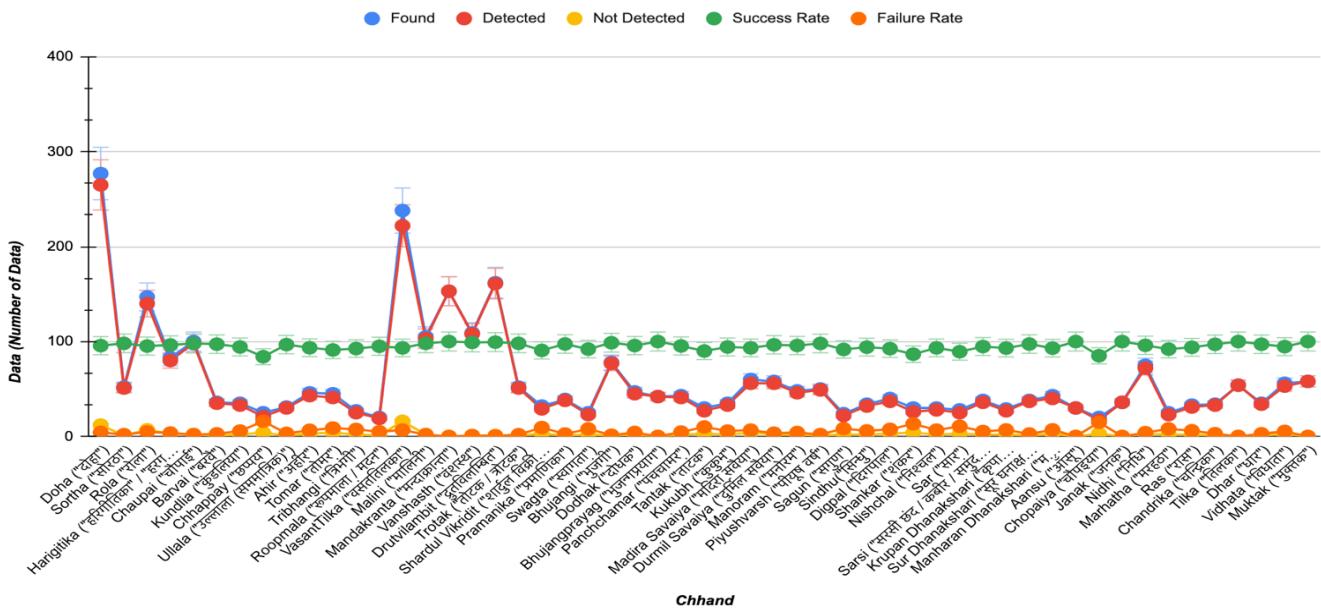


Fig. 1. Graph Showing 'Chhand' Detection Results.

### Execution Time Results

Metadata Generation Execution Time

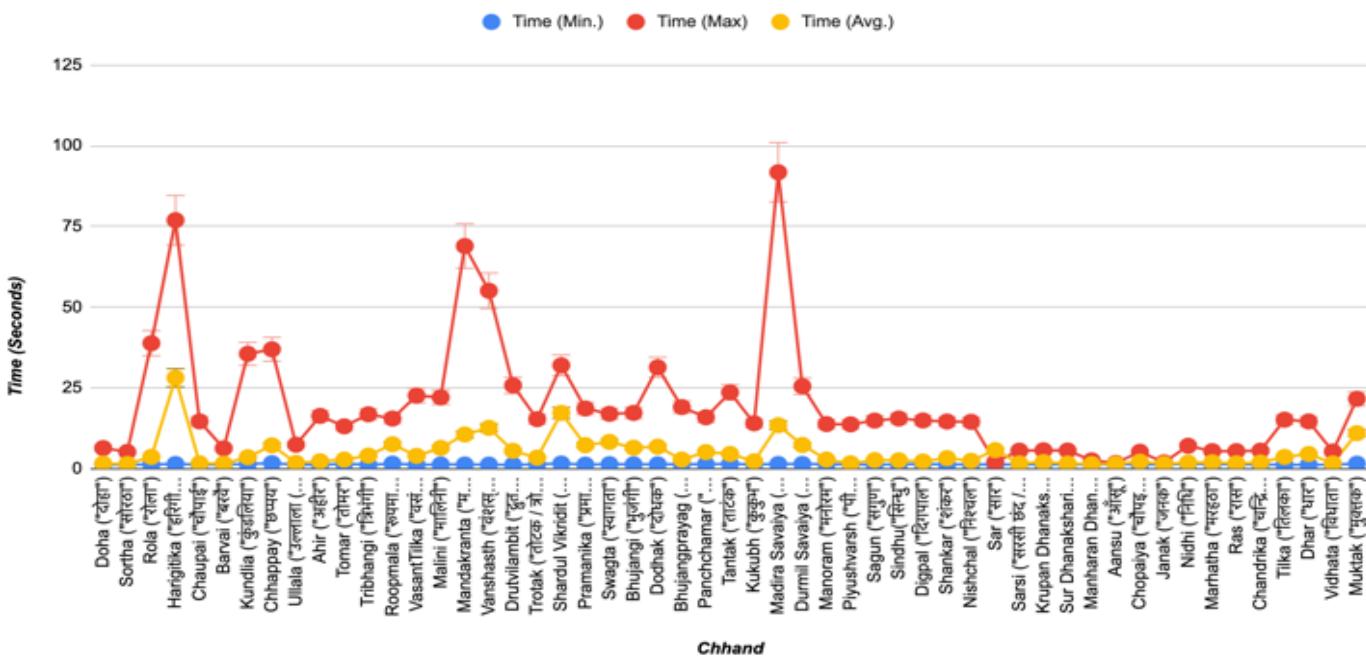


Fig. 2. Graph Showing Execution Time Results.

### Metadata Generator for Hindi Poetry

Content of file:

प्रवन्तनय संकट हस्त, मंगल मूरूति रूप |

राम लखन सीता सहित, हृदय बस्तु सुर भूष ||

Number of characters (Including Space): 74

Number of characters (After Cleaning and With Spaces): 53

Number of Total Diacritics: 13

Number of Total Guru Matra: 8

Number of Total Laghu Matra: 5

Number of Total Half: 0

Number of Words: 14

Number of Lines: 2

Number of Charans: 4

Charan Wise Matra (QTY) Count:

[[[1, 1, 1, 1, 1], [2, 1, 1], [1, 1, 1]], [[2, 1, 1], [2, 1, 1], [2, 1, 1]], [[2, 1], [1, 1, 1], [2, 2], [1, 1, 1]], [[1, 1, 1], [1, 1, 1], [1, 1, 1], [2, 1]]]]

Charan Wise Matra (QTY) Sum: [13, 11, 13, 11]

Charan Wise Varna: ['111111211111', '21121121', '2111122111', '111111121']

Charan Wise Varna Sum: [12, 8, 10, 10]

Found Stop Words: []

Words Meaning & Examples:

संकट (Meaning) : किसी अनियंत्रित पठन से उत्पन्न होने वाली ऐसी स्थिति जिसमें बड़ी हानि हो सकती हो

Examples: संकट में दियाग काम करना बंद कर देता है।

हस्त (Meaning) : छोने, लूटने या अनुचित रूप से बलपूर्वक ले लेने की क्रिया।

Examples: रावण ने सीता का हस्त किया था।

मंगल (Meaning) : किसी के द्वारा या अन्य किसी प्रकार से होने वाला कोई अच्छा काम

Examples: वही काम करें जिसमें सबका हित हो।

रूप (Meaning) : किसी विषय, वात या घटना की कोई विशेष स्थिति

Examples: ख्रोध की अवस्था में किया गया काम ठीक नहीं होता उसकी क्या गति हो गई है।

राम (Meaning) : राजा दशरथ के पुत्र जो भगवान विष्णु के अवतार माने जाते हैं

Examples: हिंदू श्री राम की पूजा करते हैं।

लखन (Meaning) : राजा दशरथ के पुत्र जो सुमित्रा के गर्भ से उत्पन्न हुए थे

Examples: लक्ष्मण शोशोबातर माने जाते हैं।

सीता (Meaning) : राजा जनक की पूरी तथा राम की पत्नी

Examples: सीता एक आदर्श पत्नी थीं, सीता को जगत जननी जगदम्बा का रूप माना जाता है।

हृदय (Meaning) : शरीर की वह अंतरिक अमूर्त सत्ता जिसमें भले-बूंदों का ठीक और स्पष्ट ज्ञान होता है।

Examples: अंतरात्मा से निकली आवाज सच होती है।

सु (Meaning) : स्वर्वा आदि में रहने वाले वे अमर प्राणी जो पूज्य माने जाते हैं।

Examples: इस मंदिर में कई देवताओं की मूर्तियाँ स्थापित हैं।

भूष (Meaning) : किसी देश का प्रधान शासक और स्वामी

Examples: त्रेतायुग में श्रीराम अयोध्या के राजा थे।

Words Meaning Not Found: ['सहित', 'बस्तु', 'भूति', 'प्रवन्तनय']

Type: मात्रिक छंद

Sub Type: अर्ध समात्रिक छंद

Chhand: तोहा

Chhand Example:

बड़ा हुआ तो क्या हुआ, जैसे पेड़ खजूर

पंछी को छाया नहीं, फल लाए अति दूर

Time Duration (In Seconds): 3.0534138679504395

The authors were unable to find much-existing work in this area, so comparing research work and the results is not possible. Any exactly similar work related to the Hindi verses classification identification was not found that only makes this work innovative through novelty. There were only two very nearby research works found. The first research work was based on identifying 'Chaupai' with 85% accuracy, while this research gives 98% accuracy. Similarly, another research work based on 'Rola' was seen with 89% accuracy, where this research provides 95.24% accuracy.

Execution time is also one of the significant aspects of any research work. Hence execution time was tracked. Fig. 2 represents the execution time graph for the tracked time of

average, minimum, and maximum time taken by each 'Chhand' for the metadata generated by the metadata generator. It can be seen that the Minimum time taken by different 'Chhands' is between 1.12 seconds and 1.72 seconds, and the average time taken for execution is between 1.45 seconds and 28.15 seconds. The maximum time is 1.80 seconds to 91.79 seconds. By analyzing all the time-based results, it was figured out that the inputs which were having more data took more time. 'Chhands', which have a smaller number of lines, words, or characters, gets detected quicker than the 'Chhands', which have a more significant number of lines, words, and characters.

### VI. CONCLUSIONS AND FUTURE WORK

The authors did a thorough literature review and found that no research study worked for the classification of Hindi verses out of all the research works, detection and identification is still missing. Even the main thing required initially for this type of research works, the taxonomical hierarchical structure of Hindi verses was absent.

To conclude, the authors taxonomically structured the Hindi verse hierarchy initially to begin the research, later the 'Chhand' rules were identified, verified, and appropriately structured with the aspect of the computational linguistics-based research works, which are managed very well by this research work. 'Chhands' were classified in the standard classes for better and smooth hierarchical management in this research work. It was experienced that 'Chhand' detection based on the rules is a complex process. 'Chhands' made up of complicated rules, takes more time to detect. Special exception rules slow down the execution time as it takes more time to process and check the data. 'Chhands' made up of the combination of the one or more 'Chhand' rules, also takes more time as they need to be gone through more than once while detection and metadata generation processing for the detection. Additionally, this research can extract stop words by filtering from the existing list of stop words. Wordnet was incorporated for the meaning and example uses of the phrase.

111 'Chhands' rules were found and modeled, out of which 53 'Chhands' were having at least 20 examples, so 53 'Chhands' example data was considered for this research study. It was concluded that 'Aansu', 'Bhujangprayag', 'Janak', 'Mandakranta', 'Muktak' and 'Tilka' were the six best performing 'Chhands' out of 53 'Chhands' with 100% accuracy. 'Chhapay Chhand' was the lowest performer with 84% accuracy. The examples of the detected 'Chhand' were populated, making it easier to understand the 'Chhand' creation rules with access to another example of the same type of the detected 'Chhand'. After carrying out this much research work, it can be powerfully conveyed that systematized Hindi Prosody Rules and the concept of automatic metadata generation for Hindi poetry are capable enough to automatically generate meaningful metadata. The research work done so far is sufficient to open a new path for upcoming researchers to think of some other relevant aspects and further contribute to Natural Language Processing and Computational linguistics research domain. For future works to continue this research work in more in-depth automatic grammatical correction, a speech-based approach for inputs can be explored and integrated.

REFERENCES

- [1] Wikipedia Contributors, "Hindi," *Wikipedia*, May 04, 2019. <https://en.wikipedia.org/wiki/Hindi> (accessed Sep. 18, 2020).
- [2] Wikipedia Contributors, "Devanagari," *Wikipedia*, Sep. 18, 2020. <https://en.wikipedia.org/wiki/Devanagari> (accessed Sep. 18, 2020).
- [3] "Unicode 13.0.0," [www.unicode.org](http://www.unicode.org), Sep. 18, 2020. <http://www.unicode.org/versions/Unicode13.0.0/> (accessed Sep. 18, 2020).
- [4] Wikipedia Contributors, "Poetry," *Wikipedia*, Sep. 18, 2020. <https://en.wikipedia.org/wiki/Poetry> (accessed Sep. 18, 2020).
- [5] Wikipedia Contributors, "Computational linguistics," *Wikipedia*, Sep. 18, 2020. [https://en.wikipedia.org/wiki/Computational\\_linguistics](https://en.wikipedia.org/wiki/Computational_linguistics) (accessed Sep. 18, 2020).
- [6] L. Schubert, "Computational Linguistics (Stanford Encyclopedia of Philosophy)," *Stanford.edu*, 2014. <https://plato.stanford.edu/entries/computational-linguistics/> (accessed Sep. 18, 2020).
- [7] 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.
- [8] B. K. Joshi and K. Kumar Kushwah, "A Novel Approach to Automatic Detection of Chaupai Chhand in Hindi Poems," *IEEE Xplore*, Sep. 01, 2018. <https://ieeexplore.ieee.org/document/8675052> (accessed Sep. 18, 2020).
- [9] K. K. Kushwah and B. K. Joshi, "Rola: An Equi-Matrik Chhand of Hindi Poems," *Journal of Computer Science Ijesis*, vol. 15, no. 3, 2017, Accessed: Sep. 18, 2020. [Online]. Available: [https://www.academia.edu/32482898/Rola\\_An\\_Equi-Matrik\\_Chhand\\_of\\_Hindi\\_Poems](https://www.academia.edu/32482898/Rola_An_Equi-Matrik_Chhand_of_Hindi_Poems).
- [10] P. B. Bafna and J. R., "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, 2020, doi: 10.14569/ijacsa.2020.0110224.
- [11] P. Bafna and J. R. Saini, "Hindi Poetry Classification using Eager Supervised Machine Learning Algorithms," *IEEE Xplore*, Mar. 01, 2020. <https://ieeexplore.ieee.org/abstract/document/9167632> (accessed Sep. 18, 2020).
- [12] P. B. Bafna and J. R. Saini, "Hindi Verse Class Predictor using Concept Learning Algorithms," *IEEE Xplore*, Mar. 01, 2020. <https://ieeexplore.ieee.org/document/9074850> (accessed Sep. 18, 2020).
- [13] J. Kaur and J. R. Saini, "Punjabi Poetry Classification," *Proceedings of the 9th International Conference on Machine Learning and Computing - ICMLC 2017*, 2017, doi: 10.1145/3055635.3056589.
- [14] 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, doi: 10.1504/ijaisc.2016.081353.
- [15] 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, doi: 10.1504/ijcistudies.2018.094901.
- [16] J. Kaur and J. R. Saini, "PuPoCl: Development of Punjabi Poetry Classifier Using Linguistic Features and Weighting," *INFOCOMP Journal of Computer Science*, vol. 16, no. 1-2, pp. 1-7, Dec. 2017, Accessed: Sep. 18, 2020. [Online]. Available: <https://infocomp.dcc.ufla.br/index.php/infocomp/article/view/546>.
- [17] J. Kaur and J. Saini, "Designing Punjabi Poetry Classifiers Using Machine Learning and Different Textual Features," *The International Arab Journal of Information Technology*, pp. 38-44, Jan. 2019, doi: 10.34028/iajat/17/1/5.
- [18] K. Pal and B. V. Patel, "Model for Classification of Poems in Hindi Language Based on Ras," *Smart Systems and IoT: Innovations in Computing*, pp. 655-661, Oct. 2019, doi: 10.1007/978-981-13-8406-6\_62.
- [19] J. R. Saini and J. Kaur, "Kāvi: An Annotated Corpus of Punjabi Poetry with Emotion Detection Based on 'Navrasa,'" *Procedia Computer Science*, vol. 167, pp. 1220-1229, Jan. 2020, doi: 10.1016/j.procs.2020.03.436.
- [20] P. B. Bafna and J. R., "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, 2020, doi: 10.14569/ijacsa.2020.0110331.
- [21] P. B. Bafna and J. R. Saini, "BaSa: A Technique to Identify Context based Common Tokens for Hindi Verses and Proses," *2020 International Conference for Emerging Technology (INCET)*, Jun. 2020, doi: 10.1109/incet49848.2020.9154124.
- [22] A. Kulkarni and G. Huet, "Google Books," [books.google.com/books/about/Sanskrit\\_Computational\\_Linguist.html](http://books.google.com/books/about/Sanskrit_Computational_Linguist.html) (accessed Sep. 18, 2020).
- [23] H. Han, E. Manavoglu, H. Zha, K. Tsoutsouliklis, C. L. Giles, and X. Zhang, "Rule-based word clustering for document metadata extraction," *Proceedings of the 2005 ACM symposium on Applied computing - SAC '05*, 2005, doi: 10.1145/1066677.1066917.
- [24] X. Yu *et al.*, "Using Automatic Metadata Extraction to Build a Structured Syllabus Repository," *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers*, pp. 337-346, 2007, doi: 10.1007/978-3-540-77094-7\_43.
- [25] M.-T. Sagri and D. Tiscornia, "Metadata for content description in legal information," *IEEE Xplore*, Sep. 01, 2003. <http://ieeexplore.ieee.org/document/1232110> (accessed Sep. 18, 2020).
- [26] J. L. Klavans *et al.*, "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, Nov. 2008, doi: 10.1007/s11042-008-0253-9.
- [27] N. Ghneim and O. Alsharif, "Emotion Classification in Arabic Poetry Using Machine Learning building a model for gender, dialect and age prediction from Arabic tweeps View project Interactive Arabic Dictionary View project Emotion Classification in Arabic Poetry using Machine Learning," *International Journal of Computer Applications*, vol. 65, no. 16, pp. 975-8887, 2013, Accessed: Sep. 18, 2020. [Online].
- [28] S. Hamidi, F. Razzazi, and M. P. Ghaemmaghami, "Automatic meter classification in Persian poetries using support vector machines," *IEEE Xplore*, Dec. 01, 2009. <https://ieeexplore.ieee.org/document/5407514> (accessed Sep. 18, 2020).
- [29] Z.-S. He, W.-T. Liang, L.-Y. Li, and Y.-F. Tian, "SVM-Based Classification Method for Poetry Style," *IEEE Xplore*, Aug. 01, 2007. <https://ieeexplore.ieee.org/document/4370650> (accessed Sep. 18, 2020).
- [30] A. Abbasi, H. Chen, and A. Salem, "Sentiment analysis in multiple languages," *ACM Transactions on Information Systems*, vol. 26, no. 3, pp. 1-34, Jun. 2008, doi: 10.1145/1361684.1361685.
- [31] 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, doi: 10.5958/2249-3220.2014.00002.0.
- [32] "छंद सारावली," <https://ia601900.us.archive.org/24/items/in.ernet.dli.2015.478241/2015.478241.Chhand-sarawali.pdf>.
- [33] J. Bhanu, "छंदःप्रभाकर," <https://ia801605.us.archive.org/15/items/in.ernet.dli.2015.322488/2015.322488.9999990253040.pdf>, 1935. .
- [34] Contributors to Wikimedia projects, "छंद," *Wikipedia.org*, Apr. 16, 2008. <https://hi.wikipedia.org/wiki/%E0%A4%9B%E0%A4%82%E0%A4%A6> (accessed Sep. 18, 2020).
- [35] C. by Admin and V. Groups, "भारतीय छंद विद्यन," [www.openbooksonline.com/groups/group/show?groupUrl=chhand&%2F](http://www.openbooksonline.com/groups/group/show?groupUrl=chhand&%2F) (accessed Sep. 18, 2020).
- [36] "छंद - भारतकोश, ज्ञान का हिंदी महासामान," <http://bharatdiscovery.org/india/%E0%A4%9B%E0%A4%A8%E0%A4%8D%E0%A4%A6..>
- [37] "kavyakala," *Kavyakala*. <https://kavyakala.blogspot.com/> (accessed Sep. 18, 2020).
- [38] "स्वस्थ हिंदी समाज," *hhindisamaj.blogspot.com*. <http://hhindisamaj.blogspot.com/> (accessed Sep. 18, 2020).
- [39] "हिंदी साहित्य काव्य संकलन – हिंदी की कविता, लघुगीत, बालगीत, फिल्म गीत, गजल, शायरी, धार्मिक लोकगीत का विशाल संकलन," हिंदी साहित्य. <http://www.hindisahitya.org/> (accessed Sep. 18, 2020).
- [40] *Hindi Chhandolakshan*. Vani Prakashan.
- [41] Contributors to Wikimedia projects, "बैदिक छंद," *Wikipedia.org*, Dec. 24, 2014. <https://hi.wikipedia.org/wiki/%E0%A4%B5%E0%A5%88%E0%A0>

- A4% A6% E0% A4% BF% E0% A4% 95\_% E0% A4% 9B% E0% A4% 82% E  
0% A4% A6 (accessed Sep. 18, 2020).
- [42] “Chhand In Hindi-छन्द की परिभाषा, छन्द और उदाहरण,” *hindimeaning.com*, Feb. 06, 2018. <https://www.hindimeaning.com/2018/02/chhand-in-hindi.html> (accessed Sep. 18, 2020).
- [43] “स्वर्ण हिन्दी समाज: भुजगी छन्द (साहित),” स्वर्ण हिन्दी समाज, May 13, 2018. [http://hhindisamaj.blogspot.com/2018/05/blog-post\\_13.html](http://hhindisamaj.blogspot.com/2018/05/blog-post_13.html) (accessed Sep. 18, 2020).
- [44] “सं.अनि पुण | गीतप्रेस, गोरखपुर.” <https://book.gitapress.org/product-style-5/gitapress-570/> (accessed Sep. 18, 2020).
- [45] “Pingala,” *Wikipedia*, Aug. 26, 2020. <https://en.wikipedia.org/wiki/Pingala> (accessed Sep. 18, 2020).
- [46] “Gayatri Mantra,” *Wikipedia*, Sep. 16, 2020. [https://en.wikipedia.org/wiki/Gayatri\\_Mantra](https://en.wikipedia.org/wiki/Gayatri_Mantra) (accessed Sep. 18, 2020).
- [47] “Hanuman Chalisa,” *Wikipedia*, Aug. 02, 2020. [https://en.wikipedia.org/wiki/Hanuman\\_Chalisa](https://en.wikipedia.org/wiki/Hanuman_Chalisa).
- [48] V. Jha, M. N, P. D. Shenoy, and V. K R, “Hindi Language Stop Words List,” *data.mendeley.com*, vol. 1, Apr. 2018, doi: 10.17632/bsr3frvvcj.1.
- [49] R. Panjwani and P. Bhattacharyya, “pyiwn: A Python-based API to access Indian Language WordNets.” Accessed: Sep. 18, 2020. [Online]