Computational Analysis of Figurative Language in Sanskrit with a Focus on *Alaṅkāras* and *Kāvyaguṇas*

A dissertation submitted during 2025 to the University of Hyderabad

in partial fulfillment of the award of the degree of

Doctor of Philosophy

in

Sanskrit Studies

by

Barbadikar Amruta Vilas

20HSPH02



Department of Sanskrit Studies

School of Humanities

University of Hyderabad

Hyderabad

2025



Certificate

This is to certify that the thesis entitled **Computational Analysis of Figurative Language in Sanskrit with a Focus on *Alaṅkāras* and *Kāvyaguṇas*** Submitted by **Barbadikar Amruta Vilas** bearing registration number **20HSPH02** in partial fulfilment of the requirements for the award of **Doctor of Philosophy** in the **School of Humanities** is a bonafide work carried out by her under my supervision and guidance.

The thesis is free from plagiarism and has not been submitted previously in part or in full to this or any other University or Institution for award of any degree or diploma.

Further, the student has the following journal publications before submission of the thesis for adjudication and has produced evidence for the same in the form of acceptance letter or the reprint in the relevant area of research:

1. **Yamaka Identifier and Classifier: A computational tool for the analysis of Sanskrit figure of sound.** In 'Annals of Bhandarkar Oriental Research Institute'. volume 102, pages 29-45 (2024), ISSN Number: 0378-1143.
   Chapter of dissertation where this publication appears: chapter 2.

2. **Symmetry in Bharata's Yamaka: A mathematical analysis**, Indian Journal of History of Science, Volume 59, pages 341–350, (2024) ISSN Number: 0019-5235.
   `https://doi.org/10.1007/s43539-024-00140-5`.
   Chapter of dissertation where this publication appears: chapter 5.

and has made presentations in the following conferences, the presented papers being published in the conference proceedings,

1. **Automatic Sanskrit Poetry Classification Based on Kāvyaguṇa.** In Proceedings of the 21st International Conference on Natural Language Processing (ICON), pages 109–119, AU-KBC Research Centre, Chennai, India. NLP Association of India (NLPAI), (2024). (This article contributes predominantly in chapter 4 of the thesis.)

2. **Anuprāsa Identifier and Classifier: A computational tool to analyze Sanskrit figure of sound.** In Proceedings of the 7th International Sanskrit Computational Linguistics Symposium, pages 102–112, Auroville, Puducherry, India. Association for Computational Linguistics, (2024). (This article contributes predominntly in chapter 2 of the thesis.)

3. **Issues in the Computational Processing of Upamā Alaṅkāra.** In *Proceedings of the 20th International Conference on Natural Language Processing (ICON)*, Goa University, Goa, India, Pages 180–190 (2023). NLP Association of India (NLPAI).

4. **Aesthetics of Sanskrit Poetry from the Perspective of Computational Linguistics: A Case Study Analysis on Śikṣāṣṭaka.** In *Computational Sanskrit and Digital Humanities – World Sanskrit Conference 2025*, Kathmandu, Nepal, Pages 15–36 (2025). Association for Computational Linguistics.

5. **Compound Type Identification in Sanskrit.** In *Computational Sanskrit and Digital Humanities – World Sanskrit Conference 2025*, Kathmandu, Nepal, Pages 90–108 (2025). Association for Computational Linguistics.

Further, the student has passed the following courses towards fulfillment of the course-work requirement for Ph.D.

No.	Course-Code	Course Title	Credits	Pass/Fail
1.	SK801	Research Methodology	4.00	Pass
2.	SK812	Topics in Natural Language Processing	4.00	Pass
3.	EN875	Research and Publication Ethics	2.00	Pass
4.	SK830	Dissertation Related Readings	4.00	Pass

Supervisor

Department of Sanskrit Studies

Supervisor
Department of Sanskrit Studies
School of Humanities
University of Hyderabad
P.O. Central University
Hyderabad-500 046.TS.

Head of the Department

Department of Sanskrit Studies

HEAD
Department of Sanskrit Studies
School of Humanities
University of Hyderabad
Central University P.O.
HYDERABAD-500 046

Dean of School

School of Humanities

DEAN
SCHOOL OF HUMANITIES
UNIVERSITY OF HYDERABAD
HYDERABAD-500 046. T.S. INDIA

Declaration

I, **Barbadikar Amruta Vilas**, hereby declare that the work embodied in this dissertation entitled "**Computational Analysis of Figurative Language in Sanskrit with a Focus on *Alaṅkāras* and *Kāvyaguṇas***" is submitted by me under the guidance and supervision of **Prof. Amba Kulkarni**, Sr. Professor, Department of Sanskrit Studies, University of Hyderabad, Hyderabad, is a bonafied research work. I also declare that it has not been submitted previously in part or full to this University or any other University or Institution for the the award of any degree or diploma. I hereby agree that my thesis can be deposited in Shodhganga/INFLIBNET.

A report on plagiarism statistics from the University Librarian is enclosed.The justification for the similarity index > 10% approved and signed by the supervisor is also enclosed.

Barbadikar Amruta Vilas

20HSPH02

Date:

Place: Hyderabad

Signature of the Supervisor

Supervisor
Department of Sanskrit Studies
School of Humanities
University of Hyderabad
P.O. Central University
Hyderabad-500 046.TS.

Computational Analysis of Figurative Language in Sanskrit With A Focus on *Alaṅkāras* and *Kāvyaguṇas*

by

Barbadikar Amruta Vilas

20HSPH02

under the guidance of

Amba Kulkarni

Sr. Professor, Department of Sanskrit Studies

Department of Sanskrit Studies

School of Humanities

University of Hyderabad

Hyderabad

Relevance of the dissertation to the United Nations Sustainable Development Goals (SDGs)

This dissertation in *Sanskrit Computational Linguistics* engages with the analytical study of poetic features such as *Yamaka*, *Anuprāsa*, *Upamā* and *Kāvyaguṇas*, combining traditional literary theory with modern computational methods. This work aligns with several factors of United Nations Sustainable Development Goals (SDGs). It primarily contributes to the **Goal 4: Quality Education**, as it facilitates the preservation and dissemination of India's classical knowledge systems through accessible digital frameworks. By integrating Sanskrit poetics with computational tools, the research enhances educational resources and supports innovative pedagogy for language and literature studies.

Further, the development of linguistic models for Sanskrit contributes to **Goal 9: Industry, Innovation and Infrastructure**. It promotes digital innovation and the creation of linguistic infrastructure for underrepresented languages, advancing the field of natural language processing within the Indian knowledge context. The work also resonates with **Goal 11: Sustainable Cities and Communities**, by digitally conserving intangible cultural her-

itage embedded in classical literature. It sustains linguistic and aesthetic traditions, ensuring their relevance for contemporary and future learners.

Finally, the interdisciplinary nature of this research embodies **Goal 17: Partnerships for the Goals**, encouraging collaboration between computational linguistics, traditional scholarship, and digital humanities. Through these connections, the study supports a broader vision of cultural and intellectual sustainability. Thus, my thesis not only contributes to Sanskrit studies and language technology but also advances global goals of inclusive education, innovation, cultural preservation, and collaborative knowledge creation.

Acknowledgments

I am deeply thankful to everyone whose guidance, encouragement, and support have made this research journey possible.

First and foremost, I would like to express my profound gratitude to my supervisor, *Prof. Amba Kulkarni* for her invaluable guidance, constant encouragement, and unwavering support throughout the course of my doctoral research. I have had the privilege of learning so much from her, both professionally and personally. Her confidence in my abilities has been a source of motivation and has inspired me to explore new ideas and pursue this work with dedication.

I gratefully acknowledge the financial support provided by the University Grants Commission (UGC) through the award of the Junior Research Fellowship (JRF). This fellowship has been instrumental in enabling me to conduct and complete my doctoral research.

I am thankful to the members of my Research Advisory Committee, *Prof. K. Rajyarama*, *Dr. Parameshwari Krishnamuthy*, and *Prof. S. Arulmozi* for their valuable suggestions and constructive feedback during the course of my research.

I extend my sincere thanks to *Prof. J. S. R. Prasad* for his encouragement and valuable suggestions throughout this journey.

I would like to express my heartfelt thanks to my fellow researchers and friends, *Dr. Sriram Krishnan*, *Sanal Vikram*, *Malay Maity*, *Amruta Malavade* and *Saee Vaze* for their support, warmth, and memorable moments that made this journey more enjoyable. I am also grateful to *Dr. Shailaja Nakkawar*, *Girija Gole*, *Himanshu Ayachit*, and *Vasudha Nilamana* for their constant encouragement and companionship. I would further like to acknowledge *Anilkumar Sanmukh*, *Dr. Samar Kumar*, *Dr. Abirlal*, *Dr. Brijesh*, *Subhashri*, and *Dr. Remya*.

I am thankful to the office staff of the Department of Sanskrit Studies, *Narasimha ji*, *Ranjith ji*, and *Mahalakshmi ji*, for their kind assistance and cooperation. I also wish to express my gratitude to the IGM library and various Administrative department staff for their constant support and help.

I extend my sincere thanks to the Svarupa team *Ramana ji*, *Veera ji* and *Sujata ji*, and other colleagues for their cooperation and support.

I express my sincere gratitude to all my teachers who have been a constant source of inspiration in pursuing my studies and research in Sanskrit. I am deeply indebted to my school teacher, *Akhilesh Warudkar Sir* recognised my potential and encouraged me to pursue Sanskrit for higher studies. I am also grateful to *Dr. Smita Hotey*, who made complex concepts easy to understand and strengthened my foundational knowledge of Sanskrit grammar and literature, for which I shall always remain thankful.

During my Master's programme, I consider myself fortunate to have had the opportunity to learn from *Prof. Srinivasa Varakhedi*, who, in a short period, imparted valuable insights into the essence of research and scholarly study. I had the privilege of learning from *Prof. Nanda Puri*, *Prof. Kavita Holey*, and all the professors of Kavikulaguru Kalidasa Sanskrit University, especially in the Department of Literature. I express my heartfelt thanks to them for motivating me and giving me the confidence to pursue research further.

My special thanks are due to *Dr. Saurabh Dwivedi* for helping me understand various concepts and for always being available to clarify my doubts.

I am thankful to my dear friends *Mrunal*, *Prachi*, *Veetrag*, *Pareekshit*, *Bhakti*, and *Jinendra* for always being there whenever I needed quick breaks and cheerful moments of relaxation.

To my grandfather, Late *Shri. Shridhar Barbadikar*, who gave me the first nourishment of the *Bhagavadgītā*, I shall always remain deeply indebted. I hope he would be proud to see me fulfilling this dream and earning a Ph.D. in Sanskrit.

To my mother, *Sau. Shraddha*, who taught me various ślokas and stories from the Epics and Purāṇas, nurturing in me a deep curiosity to read and understand them in their original Sanskrit form. My father, *Shri. Vilas Barbadikar*, has been my constant source of emotional strength and has always encouraged me to move forward with confidence. I am deeply grateful to my parents for believing in me and giving me the freedom to choose my own path.

To my brother and sister-in-law, *Dr. Makarand* and *Dr. Kalyani*, I owe my deepest gratitude for their immense love, care, and constant support during my stay in Hyderabad. Their affection and encouragement made my Ph.D. journey fulfilling and inspired me to pursue this research with greater dedication. Their little daughters, *Aadya* and *Kavya*, have been my greatest stress-busters during challenging times. Their innocent affection and joyful pres-

ence brought warmth and happiness to my days. This journey would not have been possible without their love and support.

My heartfelt thanks to my sister, *Ambika*, and my brother, *Omkar*, who have always been my pillars of strength. Their unwavering belief in me has kept me going through every challenge, and I can never thank them enough for their encouragement and care.

I heartily thank my brother-in-law, *Gopikrishna*, and sister-in-law, *Ranjani*, for their constant encouragement and support in pursuing my career after marriage.

To my companion, best friend, and husband, *Prasanna Venkatesh* — I can never thank you enough for your endless support, patience, and faith in me. Your thoughtful suggestions and our long discussions, especially during the final stage of my thesis, helped me think clearly and stay motivated. You have taught me to believe in myself and to face challenges with calm and confidence. Your love and understanding have been my greatest strength.

Lastly, I bow down in deep reverence to my *Sadguru, Shri Brahmachaitanya Gondavalekar Maharaj*, my eternal source of inspiration and contentment. Without His divine blessings, none of this would have been possible. He has always been my guiding light, and I dedicate this work at His feet.

Contents

Acknowledgments **i**

List of Figures **vii**

List of Tables **viii**

Acronyms **x**

Glossary **xi**

Articles Published During the Course **xiv**

1 Introduction **1**
1.1 Aim of the Research . . . 1
1.2 Motivation . . . 1
1.3 Significance . . . 2
1.4 Review of Previous Work . . . 3
1.4.1 Poetry Generation/Synthesis . . . 3
1.4.2 Figurative Speech Identification . . . 3
1.4.3 Poetry Analysis . . . 4
1.4.4 Poetry Classification . . . 4
1.4.5 Style Analysis . . . 4
1.4.6 Paraphrasing . . . 5
1.4.7 Meter analysis for Sanskrit . . . 5
1.5 Tradition of Poetics in Sanskrit . . . 5
1.6 Challenges in Sanskrit FLP . . . 7
1.6.1 Ambiguity in Interpretation . . . 7
1.6.2 Scarcity of Annotated Datasets . . . 7
1.6.3 Context Sensitivity and Cultural Dependence . . . 8
1.6.4 Morphological and Syntactic Complexity . . . 8
1.6.5 Multiple Levels of Meaning in Poetics . . . 8
1.6.6 Detecting Oblique Expressions (*Vakrokti*) . . . 9
1.7 Leveraging Alaṅkāraśāstra for FLP . . . 9
1.7.1 Figurative Language Tagging and Benchmarking . . . 9
1.7.2 *Dhvani* Theory and Implicit Meaning Detection . . . 9
1.7.3 *Vakrokti* and Creative Expression Recognition . . . 10
1.7.4 *Rasa* Theory for Emotion and Sentiment Analysis . . . 10
1.7.5 *Alaṅkāras* for Figurative Language Identification . . . 10
1.8 Objectives . . . 11
1.9 Contribution . . . 11
1.10 Thesis Outline . . . 12

2 Śabdālaṅkāra Identifier and Classifier **14**
2.1 Introduction . . . 14
2.2 Yamaka Identifier and Classifier . . . 15
2.2.1 *Yamaka* in Sanskrit Poetics . . . 15
2.2.2 Implementation: Yamaka Identifier and Classifier . . . 24
2.2.3 Performance Evaluation . . . 29
2.3 Anuprāsa Identifier and Classifier . . . 29
2.3.1 *Anuprāsa* in Sanskrit Poetics . . . 30
2.3.2 *Viśvanātha's* Classification . . . 33
2.3.3 Implementation: Anuprāsa Identifier and Classifier . . . 35
2.3.4 Performance Evaluation . . . 37
2.4 Interface: Śabdālaṅkār Identifier and Classifier . . . 38
2.5 Conclusion . . . 40

3 Mathematical Analysis of the Symmetries in Bharata's Yamaka **43**
3.1 Introduction . . . 43
3.2 Group . . . 44
3.3 Yamaka . . . 46
3.4 Symmetry analysis of various types of *Yamaka* . . . 48
3.5 Discussion on missing symmetries . . . 63
3.5.1 Repetition within the same *pāda* . . . 64
3.5.2 Repetition across the *pāda*s . . . 64
3.6 Conclusion: . . . 66

4 Poetry Classifier Based on *Kāvyaguṇas* **67**
4.1 Introduction . . . 67
4.2 Background . . . 67
4.3 Implementation and Algorithm . . . 71
4.3.1 Dataset . . . 71
4.3.2 Vectorization . . . 72
4.3.3 Learning Models . . . 75
4.4 Performance Evaluation . . . 76
4.4.1 Validation of Word2vec . . . 76
4.4.2 Validation of Feature Engineering-Based Models . . . 76
4.5 Error Analysis . . . 79
4.6 Comparison of Feature Engineering with Word2vec . . . 81
4.7 Conclusion . . . 82

5 Sentential Parsing of Upamā **83**
5.1 Introduction . . . 83
5.2 Upamā Alaṅkāra . . . 85
5.3 Types of *Upamā* . . . 86
5.4 Tagging guidelines for *Upamā* . . . 88
5.4.1 About the Notation and Convention . . . 88
5.4.2 Annotating *Upamā*: Guidelines . . . 90
5.4.3 Handling of special nuances from literature . . . 110
5.5 Conclusion and future direction . . . 115

6 Conclusion **117**
6.1 Limitations 117
6.2 Future work 118

Bibliography **119**

List of Figures

2.1 Śabdālaṅkāra Identifier and Classifier: Interface . . . 38
2.2 Śabdālaṅkāra Identifier and Classifier: Yamaka input . . . 39
2.3 Śabdālaṅkāra Identifier and Classifier: Yamaka output . . . 39
2.4 Śabdālaṅkāra Identifier and Classifier: Anuprāsa input . . . 40
2.5 Śabdālaṅkāra Identifier and Classifier: Anuprāsa output . . . 41

5.1 Example parse tree with *Upamā* specific tags . . . 89
5.2 Representation of X->Y . . . 90
5.3 Parse Tree for *'rāmaḥ gacchati|'* . . . 90
5.4 *Vākyagā Śrautī Pūrṇopamā* . . . 91
5.5 *Vākyagā Śrautī Dharmaluptā* . . . 92
5.6 *Vākyagā Ārthī Pūrṇopamā: type I* . . . 93
5.7 *Vākyagā Ārthī Pūrṇopamā: type II* . . . 94
5.8 *Vākyagā Ārthī Pūrṇopamā:type III* . . . 94
5.9 *Vākyagā Ārthī Dharmaluptā* . . . 95
5.10 *Vākyagā Ārthī Upamānaluptā* . . . 96
5.11 *Vākyagā Ārthī Dharma-upamānaluptā* . . . 97
5.12 *Samāsagā Śrautī Pūrṇopamā* . . . 98
5.13 *Samāsagā Śrautī Dharmaluptā* . . . 98
5.14 *Samāsagā Śrautī Vācakaluptā* . . . 99
5.15 *Samāsagā Śrautī Dharma-vācakaluptā* . . . 100
5.16 *Samāsagā Ārthī Pūrṇopamā* . . . 100
5.17 Samāsagā Ārthī Dharmaluptā . . . 101
5.18 Samśagā Ārthī Upamānaluptā . . . 102
5.19 Samāsagā Ārthī Dharma-upamānaluptā . . . 102
5.20 *Samāsagā Ārthī Dharma-upamāna-vācaka-luptā* . . . 103
5.21 *Pratyayagā Śrautī Pūrṇopamā* . . . 104
5.22 *Pratyayagā Ārthī Pūrṇopamā* . . . 105
5.23 *Pratyayagā Śrautī Vācakaluptā karma - kyac* . . . 105
5.24 *Pratyayagā Śrautī Vācakaluptā ādhāra - kyac* . . . 106
5.25 *Pratyayagā Śrautī Vācakaluptā kartā - kyaṅ* . . . 106
5.26 *Pratyayagā Śrautī Vācakaluptā karma - ṇamul* . . . 107
5.27 *Pratyayagā Śrautī Vācakaluptā kartā - ṇamul* . . . 108
5.28 *Pratyayagā Śrautī Vācaka-upameyaluptā : kyac* . . . 108
5.29 *Pratyayagā Śrautī Dharma-vācakaluptā - kvip* . . . 109
5.30 *Pratyayagā Ārthī Dharmaluptā* . . . 110
5.31 Representation of more than one *upamāna*s and *upameya*s . . . 111
5.32 Example parse tree for *Iva* not connected with the *upamāna* . . . 113
5.33 Representation of *Ananvayā* . . . 114
5.34 Representation of *Upameyopamā* . . . 114
5.35 Representation of *Utprekṣā* . . . 115

List of Tables

2.1 *Samasta-pādāvṛtti-yamaka* with examples. . . . 22
2.2 Differentiation of the various types of *Anuprāsa* from the implementation point of view. . . . 36

3.1 Symmetry patterns in *Bharata*'s classification with Examples . . . 47
3.2 Representation of *Yamaka* using variables: *Pādādi* and *Pādānta* types . . . 48
3.3 Representation of vertical reflection (V) using variables and verse: *Pādādi* type 49
3.4 Representation of Vertical reflection (V) using variables and verse: *Pādānta* type . . . 49
3.5 Representation of of Horizontal reflection (H) using variables and verse: *Pādādi* type . . . 49
3.6 Representation of of Horizontal reflection (H) using variables and verse: *Pādānta* type . . . 50
3.7 Representation of Rotational transformation (R) using variables and verse: *Pādādi* type . . . 50
3.8 Representation of Rotational transformation (R) using variables and verse: *Pādānta* type . . . 50
3.9 Representation of Identity operation (I) using variables and verse: *Pādādi* type 51
3.10 Representation of Identity operation (I) using variables and verse: *Pādānta* type 51
3.11 Composition matrix : *Pādādi / Pādānta Yamaka* . . . 52
3.12 Representation of *Yamaka* using variables: *Sandaṣṭaka* and *Pādāntāmreḍita* types . . 54
3.13 Composition matrix : *Vikrānta Yamaka* . . . 60
3.14 Repetition in the same foot . . . 64
3.15 One foot repeated . . . 65
3.16 Three feet repeated . . . 65
3.17 Two feet repeated . . . 66

4.1 Phonetic and Syntactic Features of *guṇa*s . . . 70
4.2 Distribution of syllable lengths. . . . 74
4.3 Performance Metrics in K-fold Validation of Different Models using `Word2vec` 76
4.4 Accuracy and Standard deviation scores of different models according to 10-fold cross-validation . . . 76
4.5 Performance Comparison of Machine Learning Models According to Holdout Validation . . . 77
4.6 Random Forest . . . 78
4.7 Gradient Boosting . . . 78
4.8 Support Vector Machine . . . 78
4.9 Multi-layer Perceptron . . . 78
4.10 Extra Trees . . . 78
4.11 XGBoost . . . 78
4.12 Average value for each variable for categories M, P and O . . . 79
4.13 Misclassified examples with their feature values and predicted labels. . . . 79

5.1 Classification of *Pūrṇopamā* 87
5.2 Classification of *Luptopamā* 88

Acronyms

ET Extra Trees.

FLP Figurative Language Processing.

GB Gradient Boosting.

HK Harvard-Kyoto encoding.

IAST International Alphabet of Sanskrit Transliteration.

LLM Large Language Model.

LRL Low-Resource Language.

ML Machine Learning.

MLP Multi-layer Perceptron.

NLP Natural Language Processing.

RF Random Forest.

SLP Sanskrit Library Phonetic encoding.

SVM Support Vector Machine.

XGB Extreme Gradient Boosting.

Glossary

Śabdālaṅkāra Figures of sound; beautiful, rhythmic, and repetitive sound patterns.

Abelian Group A group that is commutative under its binary operation, meaning the order in which elements are combined does not alter the result.

Alaṅkāraśāstra The traditional Sanskrit theory of aesthetics, poetics, and literary ornamentation.

Anuprāsa A figure of sound similar to alliteration, involving the repetition of consonants in close proximity.

Arthālaṅkāra Figures of sense; stylistic patterns that give rise to pleasure due to the underlying meaning.

Binary Operation A rule (denoted usually by $*$ or $+$) that operates on two elements of a set to produce a single output from that same set.

Citrabandha A type of literary construction that creates visual, geometric, or symmetrical patterns from the specific arrangement of syllables.

Confusion Matrix A table layout used in multi-class classification that provides a detailed breakdown of a model's performance by mapping predicted classes against true labels.

Cycle Decomposition A compact mathematical notation used to represent the cyclical rearrangement of elements in a permutation.

Dhvani The implied or suggested meaning in poetry.

Feature Engineering The process of manually extracting text features based on domain-specific clues (such as Sanskrit poetics) rather than relying entirely on automated neural networks.

Group Theory A rigorous mathematical framework used to quantify and describe the structural properties of symmetries in objects or literary patterns.

Holdout Validation A testing technique where a dataset is split into two distinct subsets: one for training the model and the other for testing its performance on unseen data.

Isomorphic Having a structure-preserving mapping between two sets, indicating that they share the exact same algebraic structure.

K-fold Cross-Validation A validation technique used to evaluate the precision, standard deviation, and overall stability of machine learning models.

Kāvya Literary language or poetry, distinguished from ordinary speech by its aesthetic and multi-layered meaning.

Kāvyaguña Syntactic and phonemic qualities or features of a literary text that enhance its overall aesthetic experience.

Klein Four-Group A specific Abelian group (denoted as K_4 or $\mathbb{Z}_2 \times \mathbb{Z}_2$) containing four elements where each element is its own inverse.

Luptopamā An elliptical simile where one or more components of the Upamā are omitted.

Mādhurya The poetic quality of sweetness, characterized by soft consonant combinations and short or no compound words.

n-gram A contiguous sequence of n items (such as syllables or letters) from a given sample of text.

Oja The poetic quality of floridity or grandeur, characterized by harsh consonant combinations and lengthy compounds.

Pāda A foot or one-quarter line of a poetic stanza.

Pūrñopamā A complete simile where all four components of Upamā are explicitly present.

Permutation Group A subgroup containing elements that represent the various symmetric transformations (such as reflections or rotations) of a structure.

Prasāda The poetic quality of lucidity or perspicuity, ensuring the meaning is easily comprehensible.

Rīti Poetic style or diction, primarily categorized as Vaidarbhī, Gauḍī, and Pāñcālī.

Rasa The aesthetic essence or emotional experience evoked in the reader or audience.

Sādhāraña-dharma The shared quality, property, or event linking the Upameya and Upamāna.

Samāsa Compound word formation in Sanskrit grammar.

Sandhi The phonetic joining and morphological fusion of adjacent words or syllables.

Upamā A figure of speech denoting a simile, expressing similarity between two distinct entities.

Upamā-dyotaka The comparative marker or comparator (e.g., *iva*, *yathā*, *tulya*).

Upamāna The object of comparison in an Upamā construction.

Upameya The subject of comparison in an Upamā construction.

Vakrokti Oblique or indirect expression; artistic deviation from ordinary speech.

Vectorization The conversion of text into numerical representations or vectors based on specific criteria so that machines can process the data.

Word Embedding A representation generated by models like Word2Vec where words with similar semantic meanings are mapped to nearby points in a continuous vector space.

Word2Vec A neural network-based vectorization method that generates vector representations of words based on semantic meanings and relationships.

WX Notation A transliteration scheme providing a one-to-one mapping between Sanskrit phonemes and Roman letters, ideal for computational processing.

Yamaka A figure of sound involving the repetition of a sequence of syllables, typically possessing different meanings at each occurrence.

Articles Published During the Course

1. Barbadikar, A. and Kulkarni, A. (2024). *Yamaka Identifier and Classifier: A Computational Tool for the Analysis of the Sanskrit Figure of Sound. Annals of the Bhandarkar Oriental Research Institute*, 102, 29–45.

2. Kulkarni, A., Barbadikar, A. and Bhat, B. (2024). *Symmetry in Bharata's Yamaka: A Mathematical Analysis. Indian Journal of History of Science*, 59, 341–350.

3. Barbadikar, A. and Kulkarni, A. (2024). *Automatic Sanskrit Poetry Classification Based on Kāvyaguṇa.* In *Proceedings of the 21st International Conference on Natural Language Processing (ICON)*, pp. 109–119. AU-KBC Research Centre, Chennai, India: NLP Association of India (NLPAI).

4. Barbadikar, A. and Kulkarni, A. (2024). *Anuprāsa Identifier and Classifier: A Computational Tool to Analyze the Sanskrit Figure of Sound.* In *Proceedings of the 7th International Sanskrit Computational Linguistics Symposium*, pp. 102–112. Auroville, Puducherry, India: Association for Computational Linguistics.

5. Jadhav, B., Barbadikar, A., Kulkarni, A., and Kulkarni, M. (2023). *Issues in the Computational Processing of Upamā Alaṅkāra.* In *Proceedings of the 20th International Conference on Natural Language Processing (ICON)*, pp. 180–190, Goa University, Goa, India: NLP Association of India (NLPAI).

6. Sandhan, J., Barbadikar, A., Maity, M., Satuluri, P., Sandhan, T., Gupta, R. M., Goyal, P., and Behera, L. (2025). *Aesthetics of Sanskrit Poetry from the Perspective of Computational Linguistics: A Case Study Analysis on Śikṣāṣṭaka.* In *Computational Sanskrit and Digital Humanities – World Sanskrit Conference 2025*, pp. 15–36, Kathmandu, Nepal: Association for Computational Linguistics.

7. Krishnan, S., Satuluri, P., Barbadikar, A., Venkatesh, T. S. P., and Kulkarni, A. (2025). *Compound Type Identification in Sanskrit.* In *Computational Sanskrit and Digital Hu-*

manities – World Sanskrit Conference 2025, pp. 90–108, Kathmandu, Nepal: Association for Computational Linguistics.

Chapter 1

Introduction

1.1 Aim of the Research

In this research, we essentially attempt to apply the theory of aesthetics and poetics (*Alaṅkāraśāstra*) established in the knowledge tradition of Sanskrit to build various computational applications for Figurative Language Processing (FLP) in Sanskrit. We also aim to relate the ancient theory of aesthetics to the contemporary theory of symmetry.

1.2 Motivation

The poetic literature of a language reflects its intellectual and cultural depth. Sanskrit, with its origins in the Vedic era, has a long-standing tradition of sophisticated aesthetic expression. Classical Sanskrit poetry employs intricate rhetorical devices, requiring an advanced level of understanding for accurate interpretation. *Ānandavardhana* highlights the necessity of deeper comprehension, stating that poetry is not fully appreciated through mere knowledge of words and meanings but requires an awareness of its underlying essence.[1] For machines, which lack human intuition and world knowledge, this poses a complex challenge. Without advanced computational methods to handle implied meanings, processing ornate poetic language remains difficult.

Natural Language Processing (NLP) is an interdisciplinary field combining computer science, artificial intelligence, and computational linguistics to enable machines to process human languages [Jurafsky and Martin (2009)]. While significant advancements have been made in data-driven NLP for various languages, Low-resource languages still face challenges such as data scarcity, complex morphology, orthographic and historical variations, and limited digital

[1] *śabdārthaśāsanajñānamātreṇaiva na vedyate |*
vedyate sa tu kāvyārthatattvajñaireva kevalam || 1.7, *Dhvanyāloka.*

resources. These issues, coupled with the inadequacy of models designed for other languages, make it difficult to develop robust NLP solutions for Sanskrit. However, despite traditionally being categorized as a Low-resource language, Sanskrit NLP has recently gained momentum due to its well-structured linguistic theories and the integration of modern computational methods. Several computational tools are available for analyzing the syntactic aspects of Sanskrit. (Goyal and Huet, 2013; Satuluri and Kulkarni, 2013; Kulkarni and Shukl, 2009; Kulkarni and Kumar, 2013, 2011; Goyal et al., 2009; Kulkarni et al., 2020; Kulkarni and Das, 2012; S, 2022)

In the realm of Sanskrit NLP, we can trace various works to process the grammar and syntactic factors of the Language. We do not find any computational work for Sanskrit that processes the poetic aspects of the literature. Our prime motivation behind this work is to initiate the Figurative Language Processing for Sanskrit starting with selected *Alaṅkāra*s and *Kāvyaguṇas*. Furthermore, to facilitate the readers with a computational framework, which can assist with understanding the figurative devices used to enhance the beauty of the literary language.

1.3 Significance

This research bridges the gap between traditional Sanskrit poetics and modern computing. Because of this interdisciplinary approach, the work is designed to be useful to several groups:

- **Computational Linguists and NLP Researchers:** Handling ornate and poetic language is a major challenge in Natural Language Processing. This work provides new computational frameworks, algorithms, and tagging schemes specifically designed for Figurative Language Processing (FLP) in a low-resource language like Sanskrit.

  To encourage further development in this field, we are making the annotated datasets used in this research publicly available. This will allow other scholars to test new models, improve upon our results, and collaborate on future NLP tools for Indian languages.

- **Sanskrit Scholars and Indologists:** The computational tools developed in this study, such as the classifiers for Yamaka and Anuprāsa, offer scholars a new way to analyze classical texts. These tools can automatically identify and categorize literary devices, making it easier to study the vast body of Sanskrit poetry at scale.

- **Educators and Students of Alaṅkāraśāstra:** For students learning how to identify complex poetic devices, these computational tools can serve as a practical guide to check their own understanding. For educators, the structured classifications, such as the tagging scheme for the 25 subcategories of Upamā, provide a clear, consistent way to teach the syntactic structures behind these poetic ornaments.

1.4 Review of Previous Work

Automatic processing of rhetorical language is assumed to be one of the toughest tasks in the area of Natural Language Processing, as it is heavily dependent on the word meaning. Despite this several kinds of works are found on the same. Some of the prime fields are mentioned below. Figurative Language Processing has gained significant attention in Natural Language Processing due to its importance in poetry and decorative language. Various research efforts have explored the generation, identification, analysis, classification, style analysis, and paraphrasing of figurative language.

1.4.1 Poetry Generation/Synthesis

Automatic generation of figurative language has been explored extensively. Chakrabarty et al. (2022) presents a systematic overview of developments in figurative language generation, focusing on metaphor, irony, and similes. Hämäläinen (2018) introduced 'Poem Machine' which produces poetry of its own and assists the user in authoring poems.

1.4.2 Figurative Speech Identification

The identification of figurative language elements is crucial for computational analysis. Liu et al. (2022) introduce FLUTE, a dataset that enhances figurative language understanding through textual explanations. They highlight the challenges faced by language models in recognizing non-literal expressions. Shutova (2011) presented a computational approach to process metaphors using statistical methods. Qadir et al. (2016) attempted the task of Automatic inference of implicit similarities in similes.

For Hindi, Audichya and Saini (2021) worked out the *alaṅkāras* in Hindi to present the hierarchical structure with a taxonomical listing of *alaṅkāras*. However, the computational implementation was not exercised.

1.4.3 Poetry Analysis

Analyzing poetry using NLP tools has been discussed by Bizzoni and Haider (2023), who survey automatic poetry analysis methods, emphasizing their applications in European languages. Computational linguistics tools such as SpaCy and ChatGPT-4 have been utilized to analyze poetry structures (Mishra et al., 2023).

Naaz and Singh (2022) was able to contribute by presenting three different tools for Hindi. 'Text2Mātrā' produces the laghu and guru mātras for the input, 'RPaGen' detects the rhyming quality of the poem and 'FoSCal' generates a score according to the quantity of *Anuprāsa* used over the poem.

1.4.4 Poetry Classification

Poetry classification is performed using various standards such as sentiment, poetic style, etc. Several researchers have explored the classification of figurative language in poetry. Kumar et al. (2023) proposes a method to categorize figures of speech in Devanagari-scripted languages using NLP techniques. Wujastyk (1978) worked out the classification for Sanskrit classical text of *Kumārasambhava* based on meter scansion for authorship criteria. Ahmad et al. (2020) classified poetry text into emotional states using deep learning techniques. Deshmukh et al. (2021) performed sentimental classification of Marathi poems using machine learning. Singh et al. (2023a) attempted classifying Hindi poetry on the phonemic features *komala* (soft) and *kathora* (harsh) with the use of learning algorithms using statistical data. Kaur and Saini (1970) developed Punjabi poetry classifiers using different textual features.

1.4.5 Style Analysis

Understanding stylistic elements in figurative language has been studied extensively. Veale and Hao (2018) compile various studies on figurative language processing, offering insights into computational modeling techniques. Englard (2013) used the rhetorical analysis of text

to predict the author. This study presents a novel approach based on ensemble learning, DistilBERT, and conventional machine learning techniques for authorship identification.

1.4.6 Paraphrasing

Paraphrasing figurative language remains a challenge in NLP. Chakrabarty et al. (2021) propose a knowledge-enhanced model to improve the interpretation of non-compositional figurative phrases.

1.4.7 Meter analysis for Sanskrit

Regarding the processing of aesthetic aspects, the automatic Meter Identification task has been worked out in a detailed manner, from various application perspectives, by different scholars for Sanskrit. Melnad et al. (2015), Rajagopalan (2018), Neil (2023), Terdalkar and Bhattacharya (2023) are some of the notable contributors to the available state-of-the-art Sanskrit Meter Identification systems.

However, no other focused computational work can be traced in this regard of Sanskrit FLP.

1.5 Tradition of Poetics in Sanskrit

The tradition of *Alaṅkāraśastra* (poetics) has evolved over centuries, emphasizing the aesthetic and structural ornamentation of poetry. The decorative components of a language, which enhance poetic beauty, can be traced extensively in Sanskrit literature. Even Vedic texts exhibit early examples of these rhetorical devices. Hence, Rajaśekhara says that other than the six disciplines viz. *śikṣā, kalpa, vyakaraṇa, nirukta, chandas* and *jyotiṣa* there exists a seventh discipline which is benevolent to Vedas which is the *Alaṅkāraśastra.* Without the understanding of the poetics the interpretation of the meaning of the Vedas is not possible.[2]

The tradition of Sanskrit poetics has developed over more than 2000 years, giving rise to six major schools, each presenting a unique perspective on literary beauty and poetic expres-

[2] *śikṣā, kalpo, vyākaraṇaṃ, niruktaṃ, chandovicitiḥ, jyautiṣaṃ ca ṣaḍaṅgāni ityācāryāḥ | upakārakatvādalaṅkāraḥ saptamamaṅgam iti yāyāvarīyaḥ | ṛte ca tatsarupaparijñānādvedārthā'navagatiḥ|*
Adhyāya-2, *Kāvyamīmāṁsā.*

sion. These schools emerged from differing scholarly views on what constitutes the soul of poetry. Various scholars identified distinct elements of poetic ornamentation as central to poetic excellence, enhancing the reader's aesthetic experience. Each school is founded on a particular aspect that it considers fundamental to the creation of poetry. A brief overview of these six schools is as follows.

1. **Rasa (Aesthetic Essence)** – Established by *Bharata* (1st century AD) in the *Nāṭyaśāstra*, the Rasa school focuses on the emotional experience evoked in the reader or audience. It identifies eight primary emotions. Later on, *Abhinavagupta* (10st century AD) added the ninth *rasa*, i.e. the *śānta rasa* in his commentary on the *Nāṭyaśāstra*[3]. The nine *rasa*s are: *śṛṅgāra* (love), *hāsya* (humour), *karuṇa* (pathos), *raudra* (fury), *vīra* (heroism), *bhayānaka* (fear), *bībhatsa* (disgust), *adbhuta* (wonder), and *śānta* (tranquillity), which a poet should aim to evoke through literary and dramatic works. *Bharata* outlines the step-by-step process of *rasa* elicitation in the form of the *rasasūtra.* The *rasa* is evoked by the combination of *vibhāva* (determinants), *anubhāva* (consequent expressions), and *vyabhicāri-bhāva* (transitory emotions).[4] The *rasa* perspective is respected even by the followers of the other schools.

2. **Alaṅkāra (Figures of Speech)** – The Alaṅkāra school, championed by *Bhāmaha* (7th century AD) and later *Daṇḍin* (8th century AD), emphasizes the role of rhetorical devices and embellishments (*alaṅkāras*) in poetry. These include figures of speech such as *Anuprāsa*, *Yamaka*, *Upamā*, *Rūpaka*, *Ananvaya*, etc., which enhance the aesthetic appeal of poetic compositions. The total number of *Alaṅkāra*s vary from scholar to scholar. In *Bharata*'s *Nāṭyaśāstra*, only 4 *Alaṅkāra*s are mentioned, whereas, in *Kuvalayānanda* of *Appaya dīkṣita* (16th century AD) 125 *Alaṅkāra*s are enlisted.

3. **Rīti (Poetic Style)** – Propounded by *Vāmana* (8th century AD), this school conceptualizes poetry in terms of stylistic elegance. According to this theory, the *rīti* (construct of expression) determines the impact of poetry, with well-structured compositions reflecting a refined and sophisticated style. There are three kinds of *rīti*s marking three different poetic styles, viz. *Vaidarbhī, Gauḍī* and *Pañcālī.*

[3] *mokṣānubhavajanitaṃ rasatvaṃ śāntasya lakṣaṇam | 6.31–32* (*Abhinavabhāratī* on *Nāṭyaśāstra*

[4] *vibhāva-anubhāva-vyabhicāri-saṃyogāt rasa-niṣpattiḥ ||* 6.31, *Nāṭyaśāstra.*

4. **Dhvani (Suggestion)** – Introduced by *Ānandavardhana* (9th century AD) in *Dhvanyāloka*, this school argues that poetry's true beauty lies in its suggestiveness rather than explicit meaning. The *Dhvani* theory emphasizes indirect expression (*vyañjanā*), where the implied meaning (*dhvani*) evokes deeper emotional and intellectual engagement, which is superior to the *abhidhā* (denotation) and *lakṣaṇā* (indication).

5. **Vakrokti (Oblique Expression)** – Formulated by *Kuntaka* (10th century AD), this school asserts that the essence of poetry lies in creative and unconventional linguistic expression. *Vakrokti* (literally, 'crooked speech') refers to artistic deviation from ordinary speech, where poetic excellence arises from unique and striking phraseology.

6. **Aucitya (Propriety)** – Championed by *Kṣemendra* (11th century AD), the *Aucitya* school emphasizes contextual appropriateness in poetry. It argues that every element of poetic composition—whether emotion, style, or figure of speech—must align harmoniously with the theme and situation to achieve literary excellence.

1.6 Challenges in Sanskrit FLP

1.6.1 Ambiguity in Interpretation

Figurative expressions convey meanings beyond their literal sense, posing challenges for computational models. For example, the phrase *'gṛham agnimiva praviṣṭaḥ'* (entered the house like fire) does not imply physical combustion but an overwhelming presence, requiring semantic understanding beyond word-level analysis. Machine learning models trained on literal text corpora struggle with such implicit meanings unless enriched with extensive contextual knowledge. Additionally, Sanskrit poetry often relies on multiple layers of meaning, making it harder to determine whether an expression should be interpreted literally or figuratively.

1.6.2 Scarcity of Annotated Datasets

A major challenge in Sanskrit FLP is the lack of large-scale annotated datasets. Unlike tasks such as part-of-speech tagging or dependency parsing, identifying figurative expressions like *alaṅkāras* (figures of speech) requires expert annotation, which is both labor-intensive and subjective. Further, figurative meanings often rely on subtle rhetorical features, making it

difficult for conventional machine learning models to capture indirect expressions such as *Dhvani* (suggestion) or *Vakrokti* (oblique speech). The absence of standardized resources limits model performance, making supervised learning approaches difficult to apply.

1.6.3 Context Sensitivity and Cultural Dependence

Figurative language is highly context-sensitive and culture-specific. Sarcasm, irony, and humor require an understanding of sentiment shifts and background knowledge that go beyond lexical patterns. Sanskrit poetry often encodes philosophical, mythological, or cultural references that are unfamiliar to modern NLP models. Many computational techniques developed for Western literary traditions fail to generalize to Sanskrit, where different figurative conventions and poetic traditions exist. This makes context modeling a critical yet unresolved issue in Sanskrit FLP.

1.6.4 Morphological and Syntactic Complexity

Sanskrit's inflectional morphology and free word order introduce significant challenges in processing figurative expressions. Word segmentation, part-of-speech tagging, and syntactic parsing are complex due to extensive sandhi rules and multiple morphological variations. Additionally, *Samāsa* (compound formation) further complicates interpretation, as compound words can encode entire phrases whose meaning depends on syntactic and semantic context. The figure of speech like *śleṣa* depends upon the different kinds of segmentations which lead to various interpretations of a single word or a sentence. In such cases, the potential figurative aspect should not be suppressed due to computational models' incapability.

1.6.5 Multiple Levels of Meaning in Poetics

Sanskrit poetics incorporates multi-layered meanings as described in theories such as *Śabd-abodha* (cognition through language). Unlike modern NLP approaches focusing on explicit textual patterns, Sanskrit poetry often relies on indirect suggestion. Computational models struggle to capture these nuanced interpretations, especially in cases where *Alaṅkāras* (figures of speech) introduce additional complexity. Recognizing rhetorical figures such as *Upamā*, *Rūpaka*, and *Śleṣa* requires deep semantic analysis and sophisticated linguistic modelling.

1.6.6 Detecting Oblique Expressions (*Vakrokti*)

A defining feature of Sanskrit poetry is its reliance on oblique expressions (*Vakrokti*), where meaning is deliberately indirect. Unlike prose, poetic language often deviates from conventional syntax and semantics, making sarcasm, irony, and satire difficult to recognize using rule-based or statistical methods. For instance, an expression like *'siṃha iva mṛduḥ'* (gentle like a lion) involves an intentional contrast, requiring a deep understanding of poetic conventions. Current NLP models lack the cognitive ability to interpret such deviations effectively.

1.7 Leveraging Alaṅkāraśāstra for FLP

Sanskrit poetics (*Alaṅkāraśāstra*) and cognitive theories offer valuable frameworks for addressing the challenges in FLP. Traditional NLP approaches primarily focus on syntactic structures and explicit semantics, often failing to capture the deeper, context-dependent meanings found in decorative language. However, Sanskrit literary theory has long recognized the multi-layered nature of meaning, and its systematic classification of poetic and cognitive phenomena provides crucial insights for computational modelling. Some of the possible applications of Sanskrit poetics are as follows.

1.7.1 Figurative Language Tagging and Benchmarking

These theories can be leveraged to construct Treebanks and tagging systems, serving as guidelines for annotation tasks. Such tagsets will be highly beneficial not only for Sanskrit but also for other Indian languages. Furthermore, the theoretical foundations of alaṅkāraśāstra can be utilized to evaluate, cross-check, and compare machine-generated annotations. They can also aid in resolving inter-annotator disagreements by providing an objective benchmark for linguistic annotation. By ensuring that annotated data is aligned with an established theoretical framework, we can significantly minimize human subjective biases. This results in high-quality datasets that yield reliable outcomes, even when the available data size is limited.

1.7.2 *Dhvani* Theory and Implicit Meaning Detection

The *Dhvani* (suggestion) theory, as elaborated by *Ānandavardhana*, asserts that the true essence of poetry lies in its implicit, suggestive meaning rather than in its direct expression. This con-

cept aligns with modern challenges in NLP, where models struggle with detecting indirect meanings, sarcasm, and irony. By integrating computational techniques that mimic the cognitive process of inference used in *Dhvani*, NLP models can be improved to recognize and interpret hidden meanings in texts. Transformer-based models, when fine-tuned with datasets annotated for implicit meanings, could significantly enhance the computational understanding of figurative expressions.

1.7.3 *Vakrokti* and Creative Expression Recognition

Kuntaka's *Vakrokti* (oblique expression) theory emphasizes artistic deviation from ordinary language, arguing that poetic beauty arises from unique and unconventional linguistic patterns. This principle is particularly useful for sarcasm detection, humour recognition, and creative text generation in NLP. Instead of relying solely on surface-level textual features, computational models can incorporate phonetic, syntactic, and semantic deviations as indicators of figurative usage. Developing feature engineering techniques inspired by *Vakrokti* would enable better identification of linguistic novelty and non-literal expressions in texts.

1.7.4 *Rasa* Theory for Emotion and Sentiment Analysis

Bharata's *Rasa* theory, foundational in Sanskrit poetics, provides a structured approach to emotional modelling in texts. The nine *Rasas* encapsulate the emotional spectrum of literature and drama. Unlike Western sentiment analysis, which primarily classifies texts as positive, negative, or neutral, *Rasa* theory enables a finer-grained classification of emotional states. Computational models can be trained to detect *Rasa* elements by leveraging annotated corpora of Sanskrit poetry and modern texts, enhancing emotion recognition in natural language processing.

1.7.5 *Alaṅkāras* for Figurative Language Identification

Sanskrit poetics categorizes a wide range of *Alaṅkāras* (figures of speech). The systematic classification of these rhetorical devices provides a structured basis for figurative language identification in computational models. Existing machine learning techniques, such as metaphor detection, can be improved by mapping these traditional Sanskrit classifications onto NLP fea-

ture extraction techniques. By incorporating the principles of *alaṅkāraśāstra*, computational models can generate well-formed poetry that adheres to classical stylistic norms.

By integrating these insights, we aim to bridge the gap between traditional Sanskrit poetics and modern NLP techniques, fostering advancements in the computational analysis of figurative language.

1.8 Objectives

The objectives of this research are as follows.

- To study the theory of Sanskrit poetics and compare different classifications and definitions of various aspects of the figurative language.
- Develop computational tools based on theory to analyze selected *alaṅkāra*s viz. *Yamaka, Anuprāsa*, and *Upamā* and *Kāvyaguṇa*s computationally.
- To study the symmetries considered in the types of *Yamaka* furnished by *Bharata* with the application of Group Theory.

1.9 Contribution

This research primarily contributes different computational models to analyze figurative aspects involved in Sanskrit poetry. In various chapters of the thesis, we first provide a brief discussion of the comparison of prominent definitions and classifications suggested by various scholars in the tradition of *alaṅkāra*, followed by the detailed algorithm on which the tools are built.

The independent tools which we present in this thesis are:

1. **Yamaka Identifier and Classifier.**
2. **Anuprāsa Identifier and Classifier.**
3. **Kāvyaguṇa based Poetry Classification.**

In this thesis, we also discuss the sentential parsing of the *Upamā* sentences based on the grammatical categorization propounded by *Mammaṭa*.

Furthermore, we mathematically analyze the symmetries present in the subtypes of *Yamaka* provided by *Bharata* using Group Theory, which provides a novel approach to study the aesthetics of Sanskrit poetry.

1.10 Thesis Outline

This study draws insights from *Alaṅkāraśāstra* to classify and analyze Sanskrit poetry using computational techniques. The various tasks covered in this research are presented systematically across different chapters. The organization of the thesis is as follows:

Chapter 1: Introduction

This chapter outlines the objectives and motivation behind the research. It highlights the gap in computational studies on figurative language processing (FLP) within the broader field of NLP. A review of previous attempts to generate and analyze rhetorical language is provided. Additionally, the chapter introduces six major schools of Sanskrit poetics, discusses the complexities of processing decorative language, and explores the applicability of Sanskrit poetic theories to modern FLP challenges.

Chapter 2: Śabdālaṅkāra Identifier and Classifier

This chapter describes the architecture of two distinct tools developed to automatically identify and classify the *Śabdālaṅkāras*, specifically *Yamaka* and *Anuprāsa*. The algorithmic approach is based on traditional definitions and examples provided in Sanskrit rhetoric. Various scholarly interpretations from the tradition are considered for implementing these tools. The user interface through which these tools are accessible is also presented.

Chapter 3: Mathematical Analysis of Yamaka

This chapter examines *Yamaka*, a *Śabdālaṅkāra* (figure of sound) from a mathematical perspective. The discussion begins with an explanation of *Yamaka* as propounded by Bharata, including its systematic definition and classification into 10 subtypes. The mathematical concept of symmetry, particularly Group Theory, is then introduced as a framework for quantifying the structural properties of *Yamaka*. Various symmetries observed in different types of

Yamaka are presented from the group theoretical perspective.

Chapter 4: Poetry Classifier based on Kāvyaguṇas

This chapter presents a novel approach to classifying poetry based on stylistic features such as Sweetness (*Mādhurya*), floridity (*Oja*) and lucidity (*Prasāda*) collectively known as *Kāvyaguṇas*. Various machine learning models are explored for this classification task. Additionally, the effectiveness of two different vectorization methods—`Word2Vec` and feature engineering, is analyzed and compared.

Chapter 5: Sentential Parsing of *Upamā*

This chapter discusses the computational parsing of the sentences of *Upamā alaṅkāra* by the Sanskrit sentential parser *Anusāraka*. *Upamā* is one of the popular *arthālaṅkāras* that describes the similarity between two distinct entities in a literary way. For processing purposes, we consider the classification provided by *Mammaṭā*, which is based on grammatical distinction. The main contributions of this chapter include the design of a detailed tagging scheme covering all 25 subcategories of *Upamā*, the proposal of representations for irregular constructions that are not considered in the grammatical categories, and the discussion of annotation strategies for other similarity-based *alaṅkāras*. Ultimately, we outline the challenges that need to be addressed in future work.

Chapter 6: Conclusion and Future Directions

This chapter summarizes the study's contributions, emphasizing its novelty in applying computational techniques to Sanskrit FLP. The limitations of the research are discussed along with potential future directions that can further advance computational approaches to Sanskrit poetics.

Chapter 2

Śabdālaṅkāra Identifier and Classifier

2.1 Introduction

The literal meaning of the term '*Alaṅkāra*' is an ornament or a decoration. In the context of literature, these ornaments are either related to the sounds of the phonemes in the words or to the meanings associated with the words or to both. The beautiful rhythmic patterns of sound are termed as *Śabdālaṅkāra*s (ornaments of sound) and the patterns that give rise to pleasure due to the meanings of the words are called *Arthālaṅkāra*s. There are constructions with rhythmic sound patterns that are also meaningful. These fall under the category of *Ubhayālaṅkāra*s. *Śabdālaṅkāra*s being governed by the rhythmic, repetitive sound patterns, are very easy to capture and can be identified with the pattern matching algorithms.

Repeated use of the same sequence of letters is one of the techniques poets use to enhance the beauty and charm of poetry. *Anuprāsa* (alliteration) and *Yamaka* are two popular *śabdālaṅkāra*s. Both use repetitive patterns. Among these two, in *Anuprāsa* typically a single letter or a syllable is repeated. For example, in the following phrase, the syllable 'ra' is repeated twice and is an instance of *Anuprāsa*.

'***rā***mo ***rā***jamaṇiḥ sadā vijayate
rāmaṁ ***ra***meśaṁ bhaje |'

Yamaka also shows the repetition of sound patterns. But these patterns are larger sequences of syllables, typically consisting of at least two syllables. For example, *"samara-samaraso'yam"*. Here, the sequence *'samara'* is repeated. The first occurrence is the left part of a compound and the second occurrence is a part of the word *'samarasaḥ'*.

In this chapter, we describe the automatic identification and classification task for the two distinct figures of sound.

2.2 Yamaka Identifier and Classifier

Background

The word '*Yamaka*' is derived from the nominal stem *'yama'* which means twin or coupled. Therefore, the literal sense of *Yamaka* is repetition. In the construction of a *Yamaka*, a group of phonemes is used more than once in a stanza to bring in a special rhythm. The meaning of these groups of phonemes at different occurrences may be the same or may be different. The effect it creates due to the sound patterns followed by the meaning, creates joy in the minds of the reader. The construction of *Yamaka* and understanding of the poems with *Yamaka* leads to a delightful experience both for the composer as well as the listener.

Notable theoretical work regarding poetics, especially *Yamaka*, can be traced in various studies done so far. Söhnen (1995) examined various definitions and classifications of Yamaka in five different treatises viz. *Nāṭyaśāstra, Bhaṭṭikāvya, Kāvyālaṅkāra, Kāvyādarśa* and *Agnipurāṇa.* The aim of this study was to find the influence of the definitions and classifications by the earlier scholars on the latter. The influence is presented in the form of a tree where the work at the child node inherits the properties from the work at the parent node. A Similar attempt has also been made by Hattori (1997) where he compared the three contemporary texts *Bhaṭṭikāvya, Kāvyālaṅkāra* and *Kāvyādarśa.* He mainly looked at the various patterns of repetition accepted by the three texts on rhetorics. Despite such theoretical works, we do not find any computational tool that analyzes *Yamaka* in the poetry.

2.2.1 *Yamaka* in Sanskrit Poetics

The use of various *alaṅkāra*s can be traced in the Sanskrit literature as early as *Vedas* and *Purāṇa*s. *Agnipurāṇa* and *Viṣṇudharmottarapurāṇa*[1] contains description of a *śabdālaṅkāra* called *Yamaka* in brief. The Sanskrit rock edicts of Aihole by the king *Pulakeśin II* (6th century AD) have many instances of *Yamaka.* Poets like *Kālidāsa, Bhāravi, Māgha* and *Daṇḍin* deployed *Yamaka* in their poetry.

[1] *ādau madhye tathaivānte pādasya tu tadiṣyate |*
sandaṣṭakasamudgākhyau tathaiva yamakau matau || 14.28*, Viṣṇudharmottarapurāṇa.*

The concept of *Yamaka* has evolved over a period of time ranging from *Bharata* (1st century AD)) to *Mammaṭa* (12th century AD). Below, we present a sketch of definitions and classification schemes by various scholars.

Before proceeding, a short note on the terminology is needed to ensure it is in order. A verse or stanza consists of four lines called a foot or *pāda* (We will use these terms alternately). In what follows, we consider only the repetitions within a verse. The *Yamaka*, due to consecutive similar verses being the same, is not considered here.

Bharata

For the first time, *Yamaka* as an *alaṅkāra* was presented by *Bharata* (1st century AD), the pioneer of Sanskrit poetics, in *Nāṭyaśāstra.* He has described four *alaṅkāras*[2] viz. *Yamaka*, *Rūpaka*, *Dīpaka* and *Upamā* out of which only *Yamaka* is a *śabdālaṅkāra.* According to *Bharata*, repetition of a word or a syllable is a characteristic of the figure of sound *Yamaka. Bharata* classified *Yamaka* into ten sub classes determined by the number of repetitions and the place of repetition in a line of a stanza. The names and patterns of these ten types are given below.

1. ***Pādānta*** - Once at the end of each foot (pāda). For example,

   *dinakṣayāt saṃhṛtaraśmi**maṇḍalaṃ***
   *divīva lagnaṃ tapanīya**maṇḍalam** |*
   *vibhāti tāmraṃ divi sūrya**maṇḍalaṃ***
   *yathā taruṇyāḥ stanabhāra**maṇḍalam** ||*

2. ***Kāñcī*** - At the end and the beginning of the same foot. For example,

   *yāmaṃ yāmaṃcan**dravatīnāṃ dravatīnāṃ***
   *vyaktāvyaktā sā**rajanīnāṃ rajanīnām** |*
   *phulle phulle sam**bhramare vābhramare vā***
   *rāmā rāmā vi**smayate ca smayate ca** ||*

3. ***Samudgaka*** - The first half is repeated in the second half (the half may refer to a stanza or a foot). For example,

[2] *upamā rūpakaṃ caiva dīpakaṃ yamakaṃ tathā |*
alaṅkārāstu vijñeyā catvāro nāṭakāśrayāḥ || 16.41*, Nāṭyaśāstra.*

ketakīkusumapāṇḍuradantaḥ
śobhate pravarakānanahastī |
ketakīkusumapāṇḍuradantaḥ
śobhate pravarakānanahastī ||

4. ***Vikrānta*** - Alternate feet are the same. For example,

sa pūrvaṃ vāraṇo bhūtvā
dviśruṅga iva parvataḥ|
abhavaddantavaikalyā-
dviśruṅga iva parvataḥ ||

5. ***Cakravāla*** - The pattern at the end of the preceding foot is repeated at the beginning of the next foot. Alternatively, if two adjacent feet are the same then also it is called *Cakravāla.* For example,

śailā yathā śatrubhirāhatā ***hatā***
hatā*śca bhūyastvanupuṅkhapuṅ****khagaiḥ |***
khagai*śca sarvairyudhi sañcitā****ścitā-***
ścitā*dhirūḍhā nihitāstalaistalaiḥ ||*

6. ***Sandaṣṭaka*** - The repetition is at the beginning of foot. For example,

paśya paśya *ramaṇasya me guṇān*
yena yena *vaśagāṃ karoti mām |*
yena yena *hi sameti darśanaṃ*
tena tena *vaśagāṃ karoti mām ||*

7. ***Pādādi*** - The pattern is repeated once at the beginning of at least two feet. For example,

viṣṇuḥ *srjati bhūtāni*
viṣṇuḥ *saṃharate prajāḥ |*
viṣṇuḥ *prasūte trailokyaṃ*
viṣṇu*rlokādhidaivatam ||*

8. ***Pādāntāmreḍita*** - The repetition is at the end of any foot. For example,

vijṛmbhitaṃ niḥśvasitaṃ ***muhurmuhuḥ***
kathaṃ vidheyaṃ smaraṇaṃ ***pade pade*** *|*
yathā ca te dhyānamidaṃ ***punaḥ puna-***
rdhruvaṃgatā te rajanī ***vinā vinā*** *||*

9. ***Caturvyavasita*** - All four feet of a stanza are the same. For example,

vāraṇānāmayameva kālo
vāraṇānāmayameva kālaḥ *|*
vāraṇānāmayameva kālo
vāraṇānāmayameva kālaḥ *||*

10. ***Mālā*** - Repetition of syllables like a garland. (This type is further elaborated by the later rhetoricians as *Anuprāsa alaṅkāra*.) For example,

asau hi ***rā****mā* ***ra****tivig****ra****hap****ri****yā*
ra*haḥp****ra****galbhā* ***ra****maṇaṃ manogatam |*
ra*tena* ***rā****t****ri****ṃ* ***ra****mayet pa****re****ṇa vā*
*na cedudeṣyatta****ru****ṇaḥ pa****ro*** ***ri****puḥ ||*

Bhāmaha

Bhāmaha (6th century AD) brought a valuable reform in the definition of *Yamaka*. While *Bharata* looked at only the sound patterns, *Bhāmaha* insisted on the patterns that are similar in sound but different in meaning.[3]

He divided the foot of a verse into three parts - *ādi* (beginning), *madhya* (middle) and *anta* (end) and classified *Yamaka* into five types as follows.

1. *Ādi* - at the beginning of a foot
2. *Madhyānta* - at the middle-end of a foot
3. *Pādābhyāsa* - Repetition of a foot in a verse
4. *Āvalī* - a series of repetitions without considering the position in the foot
5. *Samasta-pāda* - repetition of a sequence in all feet

[3] *tulyaśrutīnāṃ bhinnānāmabhidheyaiḥ parasparam |*
varṇānāṃ yaḥ punarvādo yamakaṃ tannigadyate || 2.17, *Bhāmaha-Kāvyālaṅkara.*

Bhāmaha believed that the types viz. *Sandaṣṭaka* (repetition of alternate feet), *Samudgaka* (repetition of adjoining feet), etc. of *Bharata* are nothing but the consequences of this fivefold classification. He directed that the repetition can occur either in the adjoining feet or the alternate feet or in all four feet.

Daṇḍin

Daṇḍin (7th century AD) furnished *Yamaka* with new insights. While defining the *Yamaka*, he referred only to the sound patterns. He looked at the repetitions from three different angles.

1. **Category - I (15 patterns):**
   Positions of the repeated patterns - within a *pāda* and across the *pāda*s. The patterns may repeat either in the same *pāda* or across the *pāda*s, as described below. The number in the parentheses indicates the possible patterns of that category in a verse.

   - Repetition in the same *pāda*. (4)
   - Pattern repeated in two different *pāda*s. (6)
   - Pattern repeated in three different *pāda*s. (4)
   - Pattern repeated in all four *pāda*s. (1)

   This results in $15(4 + 6 + 4 + 1)$ patterns.

2. **Category - II (3 patterns):**
   The second dimension that *Daṇḍin* brought in, refers to the intervention of other syllables in the repetition. This is more concerned with the repetition in the same *pāda*, rather than across the *pāda*s. If there is an intervention of other syllables between the two similar patterns, the pattern is labeled as '*Vyapeta*', and otherwise it is called '*Avyapeta*'. An example of *Vyapeta-yamaka*,

   kamale*ḥ samakeśānte*
   kamale*rṣyākaraṃ mukham |*
   kamale*khyaṃ karoṣi tvaṃ*
   kamale*vonmadiṣṇuṣu ||*

   And that of *Avypeta-yamaka* is,

***ramaṇī ramaṇī**yā me*
***pāṭalā pāṭalā**ṃśukā |*
***vāruṇīvāruṇī**bhūta-*
***saurabhā saurabhā**spadam ||*

Of course, one can have both intervened as well as un-intervened patterns in the same *pāda* and such an instance is called '*Avyapeta-vyapeta*'. For example,

***sālaṃ sālam**bakalikā*
***sālaṃ sālaṃ** na vīkṣitum |*
***nālī nālī**navakulā*
***nālī nālī**kinīrapi ||*

Thus, there are 3 possible variations due to the presence and/or the absence of intervention.

3. **Category - III (7 patterns):**
   A *pāda* is considered to be made up of three parts viz. beginning (*ādi*), middle (*madhya*) and end (*antya*). Since there are three parts, patterns may repeat at least twice and a maximum of thrice in the same *pāda*. Accordingly, there are seven patterns viz. -

   (a) Beginning (*Ādi*) - both the patterns are in the beginning.

   (b) Middle (*Madhya*) - the repetition is in the middle.

   (c) End (*Antya*) - both the patterns are at the end.

   (d) Beginning-middle (*Ādimadhya*) - once at the beginning and once in the middle.

   (e) Middle-end (*Madhyānta*) - once in the middle and once at the end.

   (f) Beginning-end (*Ādyanta*) - once at the beginning and once at the end.

   (g) Beginning-middle-end (*Ādimadhyānta*) - The pattern is found thrice in a *pāda* viz. beginning, middle and the end part.

This results into $15 \times 3 \times 7 = 315$ possible patterns of *Yamaka* if we combine all the three categories.

In addition to the *Yamaka* within a verse, *Daṇḍin* also provided examples of *Yamaka* across the verses. When two verses have the same sound patterns but are they having different meanings

then the *Yamaka* is termed '*Ślokābhyāsa-yamaka*', and if only a half of the verse is the same, then it is termed '*Śloka-ardha-abhyāsa-yamaka*'. Also, when there are two *pāda*s with the same sound pattern but have different meanings, this pattern is called '*Pādābhyāsa-yamaka*'. Another variety of *Yamaka* discussed by *Daṇḍin* is called '*Pratiloma-yamaka*'. In this variety, the repetition is in reverse order. The reversal may be within a *pāda* or between two *pāda*s or it may cover a complete stanza. Here is an example of '*Pratiloma-yamaka*' covering a *pāda*.

yāmatāśa kṛtāyāsā sā yātā kṛśatā mayā |
ramaṇārakatā testu stutetākaraṇāmara ||

We observe that the part

yā | ma | tā | śa | kṛ | tā | yā | sā

is repeated in the reverse order in the first line and the part

ra | ma | ṇā | ra | ka | tā | te | stu

is repeated in the reverse order in the second line.

Vāmana

Vāmana (8th century AD) underlined the 'different meaning' factor in the definition of *Yamaka* which was not explicitly mentioned in *Kāvyādarśa* of *Daṇḍin*. Also in the definition, he used the term *'Sthānaniyama'* which emphasizes that the words need to be repeated in some well-defined places only.[4] Thus, he insisted that there cannot be randomness in the repetition of patterns. Rather, the patterns are to be in specific positions in different *pāda*s.

Rudraṭa

Rudraṭa (9th century AD) has done significant work on *Yamaka* on the same grounds as *Daṇḍin*. In *Kāvyālaṅkāra*, *Rudraṭa* elaborately explained *Yamaka* and it's types. The *Pratiloma-yamaka* of *Daṇḍ*in is not accepted by *Rudraṭa*, since for him the order of syllables is also important.[5]

[4] *padamanekārthamakṣaraṃ cāvṛttaṃ sthānaniyame yamakam |*
pādāḥ pādasyaikasyānekasya ca ādimadhyāntabhāgāḥ sthānāni | 4.1-2 *, Kāvyālaṅkārasūtravṛtti.*

[5] *kramagrahaṇāt pratilomānulomasarvatobhadrānuprāsādināṃ yamakatvanirāsaḥ | nahi teṣu tulyaśrutisadbhāve'pi tulyakramo vidyate |* Commentary of Namisādhu on 3.1, *Kāvyālaṅkāra.*

Rudraṭa's main contribution is towards providing a multilevel classification with a systematic naming scheme. At the top level, he classified *Yamaka* into three categories depending on whether a complete verse is repeated or the repetition is only within a half-verse or the repetition occurs within one quarter of a verse. In the case of complete verse, there is no further sub-division. The other two patterns have further sub-classifications depending on the position of the repeated patterns. The repetitions can be either across the *pāda*s or within a *pāda*.

1. *Samasta-pādāvṛtti-yamaka* (Repetition across *pāda*s)

   In the case of repetitions across the *pāda*s, either the pattern is repeated in two or all the four *pāda*s or two different patterns are repeated in pairs of *pāda*s resulting in 10 different possible combinations which are shown in Table 2.1. Here, the 'p_i' denotes i^{th} *pāda*.

Type	Pattern	Example
Paṅkti	$p_1 = p_2 = p_3 = p_4$	**sabhājanenopari pūritāsau sabhājane noparipūritāsau** | **sabhā janeno'paripūritāsau sabhājane no'paripūritāsau** ||
Samudgaka	$p_1 = p_3, p_2 = p_4$	**nanāma loko vidamānavena** mahī na cāritramudāradhīram | **na nāmalo' kovidamānavena** mahīnacāritramudāradhīram ||
Parivṛtti	$p_1 = p_4, p_2 = p_3$	**mudā ratāsau ramaṇī yatā yāṃ** smarasyado'laṃ kurutena voḍhā | smarasyado'laṃkurute'navoḍhā **mudāratāsau ramaṇīyatāyām** ||
Yugmaka	$p_1 = p_2, p_3 = p_4$	vināyameno nayatā'sukhādinā vinā yamenonayatā sukhādinā | **mahājano'dīyata mānasādaraṃ mahājanodī yatamānasādaraṃ** ||
Mukha	$p_1 = p_2$	**cakraṃ dahatāraṃ cakranda hatāram** | khaḍgena tavājau rājannarinārī ||
Puccha	$p_3 = p_4$	uttuṅgamātaṅgakulākule yo vyajeṣṭa śatrūnsamare sadaiva | **sa sāramānīya mahāri cakraṃ sasāra mānī yamahāricakram** ||
Garbha	$p_2 = p_3$	yo rājyamāsāya bhavatyacintaḥ **samudratārambharataḥ sadaiva** | **samudratārambharataḥ sa daiva** pramāṇamārabhya payasyudāste ||
Sandaṃśa	$p_1 = p_3$	**sannārībharaṇomāya**-mārādhya vidhuśekharam | **sannārībharaṇo'māya**- statastvaṃ pṛthivīṃ jaya ||
Sandaṣṭaka	$p_2 = p_4$	idaṃ ca yena svayamātmabhogyatāṃ **samastakāñcīkamanīyatākulam** | nitambabimbaṃ kathamastu no nṛṇāṃ **sa mastakāñcī kamanīyatākulam** ||
Avṛtti	$p_1 = p_4$	**mudāratāḍī samarājirājitaḥ** pravṛddhatejāḥ prathamo dhanuṣmatām | bhavānbibhartīha nagadha medinī- **mudāratāḍīsamarājirājitaḥ** ||

Table 2.1: *Samasta-pādāvṛtti-yamaka* with examples.

2. *Ekadeśaja*[6]*-yamaka* (Repetition within a *pāda*).

   In the case of repetition within a *pāda*, he divides a quarter into two and three parts each and then considers the repetitions within and across each half or $\frac{1}{3}rd$. And finally, he divides the half part and $\frac{1}{3}rd$ part into two halves resulting into $\frac{1}{4}th$ and $\frac{1}{6}th$ of a *pāda* and names the repeating patterns.

[6] This is also known as '*pādabhāgāvṛtti*'

(a) *Pādārdhāvṛtti-yamaka* (Repetition in the half *pāda*)

The number in parentheses indicates the number of possible *Yamaka*s with that pattern.

i. *Pādādi-paṅkti-ādi* (10)

ii. *Pādānta-paṅkti-ādi* (10)

iii. *Ādyantika* (6)

iv. *Antādika* (6)

v. *Pādasamudgaka* (15)

vi. *Ardhaparivṛtti* (1)

vii. *Anya-bheda - Pādacaturthārdhāvṛtti-yamaka* (Repetition in the quarter of a pāda)

A. *Vaktra*

B. *Śikhā*

C. *Mālā*

D. *Madhya*

E. *Ādyanta*

F. *Kāñcī*

(b) *Pāda-tṛtīyārdhāvṛtti-yamaka* (Repetition in $\frac{1}{3}rd$ of a *pāda*)

i. *Pādādi-paṅkti-ādi* (10)

ii. *Pādamadhya-paṅkti-ādi* (10)

iii. *Pādānta-paṅkti-ādi* (10)

iv. *Ādyantika* (6)

v. *Antādika* (6)

vi. *Pādasamudgaka* (15)

vii. *Ardhaparivṛtti* (1)

viii. *Anya-bheda - Pādatṛtīyārdhāvṛtti* (Repetition in $\frac{1}{6}th$ of a *pāda*)

A. *Ādimadhya*

B. *Madhyānta*

C. *Ādyanta*

Rudraṭa admitted the possibility of innumerable permutations of these repetitions.[7] Thus, the combinations of the repetitive patterns, which do not follow any specific pattern regarding the position are labeled as '*Aniyata-pāda-bhāgāvṛtti yamaka*'.

Mammaṭa

Mammaṭa(12th century AD) brought forward a perfectly well-defined definition of *Yamaka*. He defined *Yamaka* as a repetition of similar sounding syllables, having different meanings when the sequence of letters is meaningful.[8] If both or any of the utterances of the Yamaka are meaningless, that also is considered to have different meanings. He accepted the same scheme as *Rudraṭa*'s for the repetition across the *pāda*s. But for the repetition within a *pāda*, he mentioned only the first two from 2a and the first three from 2b of *Rudraṭa*.

Post-*Mammaṭa*, no remarkable upgradation or proposal of any innovative classification scheme for *Yamaka* is observed.

2.2.2 Implementation: Yamaka Identifier and Classifier

Considered Classification Scheme

We have seen various definitions of *Yamaka* and various classification schemes provided by the Indian rhetorical tradition. The main difference in the variations of the definition is concerning the meaning of the repeated patterns. While *Mammaṭa* insisted on the repeated patterns that have different meanings, others are silent on it when they define *Yamaka*. From the computational implementation point of view, we decide to ignore the meaning, as it is heavily dependent on the complete context. Regarding the classification, we decide to consider three classifications viz. of *Bharata, Daṇḍin* and *Rudraṭa*.

We have chosen these three among the many for the following reasons. The three rhetoricians represent the development of the concept of *Yamaka* and the difference in the factors on which the classification is defined. *Bharata* is the foremost person providing a crude classification of *Yamaka*. *Daṇḍin* has provided a very elaborate classification scheme where he brings

[7] *yamakānāṃ gatireṣā deśāvayavāvapekṣamāṇānām |*
aniyatadeśāvayavaṃ tvaparamasaṃkhyaṃ sadevāsti || 3.56, *Kāvyālaṅkāra.*

[8] *arthe satyarthabhinnānāṃ*
varṇānāṃ sā punaḥśrutiḥ yamakam | 83.1, 9*th* chapter, *Kāvyaprakāśa.*

three dimensions, that is, position of the repeated pattern across or within a *pāda*, number of repetitions, and continuity. *Rudraṭa*, though followed *Daṇḍin* closely, provides a completely different classification following only the repetition across and within a *pāda*, considering the two-fold, three-fold, four-fold and six-fold symmetries within a *pāda*. Of course, the four-fold and six-fold symmetries make sense only if the *pāda* is sufficiently long.

Problems in pre-processing and solutions

Yamaka is a repetition of the same sequence of letters. Hence, extracting various sequences of n-letters called 'n-grams' and choosing the high-frequency longest matches should help in identifying the Yamaka and its classification. However, there are certain linguistic and prosodic structure related concerns we need to address. These concerns and the solutions we adopted for resolving them are stated below.

(i) Word Boundaries and Compound Formation: The repeated patterns in *Yamaka* are typically words. Though not every definition of *Yamaka* insists on the 'meaningfulness' of the repeated pattern, these patterns typically are meaningful and hence they are the words. However, these words can be part of a compound as well, and thus these can sometimes be bound morphemes and not isolated words. Sanskrit is very rich in compound formation and often in *Yamaka*, we see repeated patterns to be part of compounds. Therefore instead of taking the word as a unit, we consider the syllable as a unit for generating the n-grams. A syllable is zero or more consonants followed by a vowel. We ignore the vowel modifiers viz. *anusvāra* and *visarga*.

(ii) Handling Variations in Nasalization: Since the *Yamaka* is identified based on sound patterns, the variations in spelling need to be taken care of. Sanskrit allows some spelling variations with regard to nasalization. The nasal stops viz. *ṅ, ñ, ṇ, n* and *m* can be written as an *anusvāra (ṃ)* when it is followed by a homorganic stop. For example, *'ambuja'* versus *'aṁbuja'*, *'aṅka'* versus *'aṁka'*. Similarly, the nasal stop *'m'* at the end of a word is written as an *anusvāra* when it is followed by a word starting with a consonant. Since the *Yamaka* pat-

tern may cross the word boundaries, we decided to normalize all the nasal stops to *anusvāra*.

(iii) Treatment of *Avagraha* ('): A special character that needs special attention is the *avagraha* (represented as a single QUOTE (') in IAST[9]). The *avagraha* is a writing convention to indicate the elided *'a'* during the sandhi operation. Since for *Yamaka* identification, we look at the sandhied text only, we ignore the *avagraha* if it is present in the input text.

(iv) *Pāda* Boundaries and Word Spaces: Similarly, the *pāda* boundaries and the spaces between the words also do matter during pattern matching. In the oral tradition, the spaces between the words do not carry any significance. *Yamaka* deals with the sound patterns, and as such, we ignore the spaces between the words. However, we do retain the boundary between the two halves of the verse, since the pattern does not get carried over from one half of the verse into the other.

(v) Meaning Ambiguity in Repeated Words: From the very definition of *Yamaka*, it follows that the repeated words are ambiguous having more than one meaning and the meaning of such words gets decided only based on the context, the relation of the word with other words in the sentence and so on. In the absence of any computational tool that helps us in the disambiguation of such words, we decided to rely only on pattern matching, ignoring the meaning component. It is assumed that the input provided to the system is a real literary work by a human author and not a random piece of text generated by an automatic machine such as chat-GPT and as such, we can blindly assume that the repeated patterns do really have different meanings in different instances they are found.

(vi) Phonetic Variations Finally, a note about a small deviation due to the defects in pronunciation. Sounds such as *'r'* and *'d'* or *'l'* are pronounced interchangeably in different parts of the country. Similarly the sounds *'v'* and *'b'* or *'n'* and *'ṇ'*. This has been observed centuries

[9] https://en.wikipedia.org/wiki/International_Alphabet_of_Sanskrit_Transliteration

ago by the Alaṅkārins[10], and hence words with such differences are treated alike. Thus for example, even though *'abalaṃ vitatāravam'* and *'avalambitatāravam'* are two different words, for the purpose of *Yamaka*, these are treated alike. And hence the following two verses with these words fall under the category of *Yamaka alaṅkāra*.

sa tvārambharato'vaśyam-
*a**ba**laṃ **vi**tatāravam |*
sarvadā raṇamānaiṣīd-
avānalasamasthitaḥ ||

satvārambharatovaśyam-
*a**va**lam**bi**tatāravam |*
sarvadāraṇamānaiṣī
davānalasamasthitaḥ ||

In the current implementation, the tool does not handle this kind of example.

Algorithm for Yamaka identification and classification

Taking into consideration all the concerns and the solutions mentioned, we now present below the algorithm for the identification and classification of *Yamaka* following the three schemes- viz. that of *Bharata, Daṇḍin*, and *Rudraṭa*.

Input: A verse in any of the supported scripts or transliteration schemes (Devanagari, IAST, SLP, WX, HK, etc.).

Output: Identified *Yamaka* patterns and their classification based on *Bharata, Daṇḍin*, and *Rudraṭa*'s schemes.

Stepwise Algorithm

1. **Accept Input**

   The user provides the verse in one of the supported scripts (Devanagari, IAST, SLP, WX, HK)

[10] *ralayorḍalayostadvadvalayorbavayorapi |*
namayornaṇayoścānte savisargāvisargayoḥ || 7.6, *Ekāvalī*.

2. **Normalize Input**

   Convert all input text into WX notation for processing. WX notation is chosen because it provides a one-to-one mapping between Sanskrit phonemes and Roman letters, unlike Unicode Devanagari encoding. For example, क is a syllable composed of क्+अ, but it is denoted as a single character in Devanagari. For processing, we need a notation which denotes the consonant क् and the vowel अ separately. WX notation provides the one-to-one mapping of characters in Roman letters. Here, the Devanagari character क is transliterated as 'ka' in WX notation.

3. **Preprocessing**

   - Remove spaces between words, but retain the boundary between the two halves of the verse.
   - Normalize nasalization (convert all nasal stops to *anusvāra*).
   - Ignore *avagraha* (ऽ), as it does not affect *Yamaka* identification.

4. **Generate n-grams**

   - Extract syllable-based *n*-grams for each half of the verse separately.
   - Ignore unigrams even if they occur multiple times, as their repetition might be accidental.

5. **Filter n-grams**

   - Retain only those *n*-grams that occur at least twice.
   - Remove sub-patterns that are already part of longer *n*-grams.

6. **Identify Position of Repeated Patterns**

   - Determine the position of repeated patterns based on:
     - Beginning, middle, or end of the *pāda*.
     - Repetitions occurring within a *pāda* or across *pāda*s.
     - Whether repetition occurs with or without intervention.

7. **Classify *Yamaka***

   - Classify the identified *Yamaka* using three different schemes:

- ***Bharata*'s scheme**
- ***Daṇḍin*'s scheme**
- ***Rudraṭa*'s scheme**

8. **Output Results**

- Display the identified *Yamaka* patterns.
- Provide classification based on the three schemes.

2.2.3 Performance Evaluation

We have collected 185 different verses from *Nāṭyaśāstra, Bhāmaha*'s *Kāvyālaṅkāra, Daṇḍin*'s *Kāvyādarśa* and *Rudraṭa*'s *Kāvyālaṅkāra* along with the verses from *Raghuvaṃśa, Ghaṭakarpara, Śrimahābhāratatātparya* of *Ānandatīrtha*, etc. Out of which 163 were analyzed correctly. In 22 cases, the tool exhibited over-generation, by producing multiple solutions, only one of which was correct and the other solutions were incorrect. The extra solutions which were wrong were produced due to the following reasons. Classifying *Yamaka* from the verses constructed in *mātrāvṛttas* is currently beyond the scope of the classifier program. Although the *Yamaka* identification task is carried out flawlessly with such constructions. The verses with the patterns having interchangeable sounds mentioned before and the patterns ending with a consonant like *'d'* or *'t'* (Ex. *kamalaṃ kamalaṅkurvadalimad-dali matpriye*) get changed in their second repetition due to sandhi. In such cases, the tool identifies the repetition of *dalima* but fails to identify the repetition of *alimat*.

2.3 Anuprāsa Identifier and Classifier

Background

Anuprāsa is a *śabdālaṅkāra*. Essentially, it is the repetition of consonants. This repetition should be in proximity so that one can recall the prior instance.[11] The phenomenon of *Anuprāsa*

[11] *pūrvānubhavasaṃskārabodhinī yadi adūratā* ||1.55, *Kāvyādarśa*

is similar to Alliteration.[12]

Like *Yamaka*, *Anuprāsa* holds an important place in the *alaṅkāra*s. In the tradition of *alaṅkāraśāstra*, *Anuprāsa* was originally introduced as a subtype of *Yamaka* viz. *Mālā-yamaka.*[13] In *Mālā-yamaka* repetition of consonants is considered, which is similar to *Anuprāsa.* In *Yamaka* where repetition of the longer patterns are engaged in the poetry especially the metrical verses when employed in a more complex way, this might create a hindrance in the process of experiencing *rasa.*[14] Because the meaning is different in each repetition, the listener might find it difficult to understand the meaning of the complete verse and lose interest. But *Anuprāsa* is considered to be a contributor to the emergence of *rasa. Anuprāsa* can be traced in any text, suggesting any kind of *rasa.*

Anuprāsa is independent of the form of poetry, that is, it also is used in the prose format, like in *Kādambarī* of *Bāṇabhaṭṭa.* The classifications found are based on the variations in repetitions in terms of the categories of repeated consonants, number of repetitions, number of repeated consonants and mood emergence due to the combination of different consonants. Hence, we do not observe many variations in the definition and classification, but the number of subtypes considered varies.

2.3.1 *Anuprāsa* in Sanskrit Poetics

In this section, we present a brief overview of various types of *Anuprāsa* furnished by different scholars. *Bharata*, known as the first scholar of the tradition, has enlisted only four *alaṅkāra*s, namely *Yamaka, Upamā, Rūpaka* and *Dīpaka*, among which *Yamaka* was the only figure of sound. *Mālā-yamaka*, a subtype of *Yamaka* can be considered as the inspiration behind *Anuprāsa.* (See *Bharata*'s *Yamaka* Classification in section 2.2.1)

Other scholars, after *Bharata*, considered *Anuprāsa* as an individual *alaṅkāra* and classified it from different perspectives. *Bhāmaha* provided only two types, whereas *Bhoja* extended the

[12] In alliteration, consonant sounds in two or more neighbouring words or syllables are repeated. The repeated sounds are usually the first, or initial, sounds as in "seven sisters", but repetition of sounds in non-initial stressed, or accented, syllables is also common: "appear and report."-"Alliteration." `https://www.merriam-webster.com/dictionary/alliteration`. Merriam-Webster, 2023.

[13] *nānārūpaiḥ svarairyuktaṃ yatraikaṃ vyañjanaṃ bhavet |*
tanmālāyamakaṃ nāma vijñeyaṃ paṇḍitairyathā||16.84, Nāṭyaśāstra

[14] *tadetatkāvyāntargaḍubhītam|* in the vṛtti of 83rd kārikā, 9th chapter, *Kāvyaprakāśa*

count to 6 types. Some scholars, being excessively analytical, tried to increase the count even more.

Bhāmaha (6th century AD) for the first time put forward *Anuprāsa* as a separate *alaṅkāra* in his treatise *Kāvyālaṅkāra. Bhāmaha* declares the arrangement of similar letters as *Anuprāsa.*[15] Moreover, he provides two types of *Anuprāsa.* One is *Grāmyānuprāsa.* As the name suggests, the repetition of letters without any pattern or elegance is *Grāmyānuprāsa.* Learned people assume it as an ordinary repetition.[16] For example,

sa lolamālānīlālikulākula galo balaḥ | 2.6, *Kāvyālaṅkāra*

The another type is *Lāṭānuprāsa.* It is a repetition of a complete *pada* (word). But the meaning of *pada* does not change. For example,

dṛṣṭiṃ dṛṣṭisukhāṃ dhehi candraścandramukhoditaḥ | 2.8, *Kāvyālaṅkāra*

'Lāṭa' is a name of a geographical region. Poets belonging to the region 'Lāṭa' used to employ this kind of repetition in plenty. Most of the scholars in the tradition included *Lāṭānuprāsa* in the classification of *Anuprāsa. Lāṭānuprāsa* shows similarity with *Yamaka alaṅkāra.* The only difference is that, in *Yamaka* the repeated word or the sequence of sounds should possess different meanings in each repetition.

After *Bhāmaha, Daṇḍin* (7th century AD) added the clause of proximity to *Anuprāsa.* According to *Daṇḍin* the repetition of letters such that the listener remembers the previous occurrence of the repeated letter is called *Anuprāsa.*[17] He added that the repetition in *Anuprāsa* nourishes the *rasa.*

Udbhaṭa (9th century AD) introduced *Chekānuprāsa* in his work *Kāvyālaṅkāra-sāra-saṅgraha. Chekānuprāsa* is one repetition of two groups of consonants.[18] Here, the sequence may not be the same. The example given is as follows,

sa devo divasān ninye tasmin śailendrakandare |
gariṣṭhagoṣṭhī-prathamaḥ pramathaiḥ paryupāsitaḥ ||
3.3, *Kāvyālaṅkāra-sāra-saṅgraha*

[15] sarūpavarṇavinyāsamanuprāsaṃ pracakṣate| 2.5, *Kāvyālaṅkāra*
[16] grāmyānuprāsamanyttu manyante sudhiyo'pare| 2.6, *Kāvyālaṅkāra*
[17] pūrvānubhavasaṃskāra-bodhinī yadyadūrata| 1.55, *Kāvyādarśa*
[18] chekānuprāsastu dvayordvayoḥ susadṛśoktikṛtau| 3.2, *Kāvyālaṅkāra-sāra-saṅgraha*

In addition to this, *Vṛttyanuprāsa* was also defined. Here, the combination of the repeated consonants is considered. According to it, three *vṛttis* are defined viz. *Paruṣa* (harsh), *Upanāgarikā* (soft) and *Grāmyā* (other than the prior two). Again, the count and the definitions of *vṛttis* vary from scholar to scholar.

Bhoja from 12th century conducted a vast review on this *alaṅkāra.* In addition to *Vṛttyanuprāsa* and *Lāṭānuprāsa* he added 4 more types viz. *Śrutyanuprāsa*, *Varṇānuprāsa*, *Padānuprāsa* and *Nāma-dvirukti. Varṇānuprāsa* is similar to *Chekānuprāsa.* Also, *Padānuprāsa* and *Nāma-dvirukti* can be included into *Lāṭānuprāsa.*

Jayadeva (12th century AD) in *Candrāloka* introduced two types of *Anuprāsa* viz. *Sphuṭānuprāsa*, which is the repetition of consonants within a *pāda* or a half of a *pāda* and *Arthānuprāsa*, where the repetition of consonants occur in the two words which are connected semantically. For example,

candanaṃ khalu govinda-caraṇa-dvandva-vandanam| 5.6, *Candrāloka*

Here, the two words '*candanam*' and '*govind-caraṇa-dvandva-vandanam*' are connected with the 'upamāna-upameya' relation and possess the repetition of consonants '*nd*'.

The criteria for classifying *Anuprāsa* into different types is the number of repetitions, considering the order of the repeated letters and what letters are being repeated. Focusing on these points, different classifications are framed. Due to such finite classification dimensions, we observe a limited number of types. Also, similar kinds of types are explained in various other classification schemes.

For the tool development, we follow one comprehensive classification. *Viśvanātha's Anuprāsa* classification provided in his treatise *Sāhityadarpaṇa* has five classes that cover the extract of all the other interpretations available. Categories proposed by the rhetoricians like *Daṇḍin, Udbhaṭa, Vāmana, Rudraṭa, Mammaṭa, Jayadeva, Bhoja,* etc. are covered under the umbrella of the classification of *Viśvanātha.*

2.3.2 *Viśvanātha's* Classification

The tenure of *Viśvanātha* (14th century AD) comes in the later part of the tradition of *alaṅkāraśāstra*. He provides a well-defined and comprehensive theory for the classification of *Anuprāsa*, which facilitates the clarity for implementation. *Viśvanātha's* classification includes other prominent classifications. Moreover, it uses widely accepted nomenclature. According to him, the *Anuprāsa* is classified into 5 sub-classes. The examples for these 5 types are taken from 10th *pariccheda* of *Sāhityadarpaṇa*.

1. ***Chekānuprāsa***

   Chekānuprāsa is the double occurrence of consonants with the same sequence. In each repetition, vowel endings may vary. In the example given below, one repetition of '*n-d-h*', '*v-r*' and '*p-v-n*' is in the same order. That means the order of the repeated consonants is not changed. The repetition of '*v-r*' is not changed to '*r-v*' irrespective of the changing vowels in between.

   ādāya bakulagandhānandhīkurvan pade pade bhramarān |
   ayameti mandamandaṃ kāverīvāri-pāvanaḥ pavanaḥ||

2. ***Vṛttyanuprāsa***

   Vṛtti is the mood or emotion. It is defined as the arousal of a specific mood resulting from a certain combination of letters. Repetition of one or many consonants in any order to produce a specific mood (*vṛtti*) is called *Vṛttyanuprāsa*. The emotional effect differs according to the repetitive sound pattern and the combination of the letters used. This effect should complement the actual *rasa* of the poetry.

   unmīlanmadhugandhalubdhamadhupavyādhūtacūtāṅkura-
   krīḍatkokilakākalīkalakalairudgīrṇakarṇajvarāḥ |
   nīyante pathikaiḥ kathaṃ kathamapi dhyānāvadhānakṣaṇa
   prāptaprāṇasamāsamāgamaraṣollāsairamī vāsarāḥ ||

   In this example, the first foot has multiple repetitions of the consonant '*dh*'. In the second foot, there is repetition of the consonants '*k*' and '*l*' in any order. The third foot has the repetition of '*dh*' only once. The last foot has the repetition of '*p*', '*r*', '*s*' and '*m*' in different orders.

3. ***Śrutyanuprāsa***

   Śrutyanuprāsa is the repetition of a group of consonants with a similar manner of articulation. According to *Daṇḍin* it is also beneficial to the *rasa.*[19] These are further sub-classified into five classes according to the place of articulation.

   (a) *Kaṇṭhya* (Velar) - *{k, kh, g, gh, ṅ, h}*

   (b) *Tālavya* (Palatal) - *{c, ch, j, jh, ñ, y}*

   (c) *Mūrdhanya* (Retroflex) - *{ṭ, ṭh, ḍ, ḍh, ṇ, r, ṣ}*

   (d) *Dantya* (Dental) - *{t, th, d, dh, n, l, s, v}*

   (e) *Oṣṭhya* (Labiel) - *{p, ph, b, bh, m, v}*

   For example, the following verse

   > *dṛśā dagdhaṃ manasijaṃ jīvayanti dṛśaiva yāḥ |*
   > *virūpākṣasya jayinīstāḥ stumo vāmalocanāḥ ||*

   has a repetition of Palatal varṇas '*j*' and '*y*'.

4. ***Antyānuprāsa***

   Repetition of syllables at the end of the *padas* (words) or at the end of the foot. Specifically, after the penultimate vowel that is the last but one vowel of the *pāda* or *pada.* For example,

   > *keśaḥ kāśastabakavikāsaḥ kāyaḥ prakaṭitakarabhavilāsaḥ |*
   > *cakṣurdagdhavarāṭakakalpaṃ tyajati na cetaḥ kāmamanalpam ||*

   In this example, the ends of the first and second feet match, similarly the ends of the third and fourth feet.

5. ***Lāṭānuprāsa***

   Lāṭānuprāsa is not just the repetition of the consonants but the repetition of a word (*pada*) with similar meaning but different implications. This kind of *Anuprāsa* is similar to *Yamaka* as repetitions are considered for longer syllable sequences. Following is an example of *Lāṭānuprāsa.*

[19] *yayākayācicchrutyāyatsamānamanubhūyate|*
tadrūpāṃhipadāsattiḥsānuprāsārasāvahā|| 1.52, Kāvyādarśa

smerarājīvanayane nayane kiṃ nimīlite |
paśya nirjitakandarpaṃ kandarpavaśagaṃ priyam ||

In the above given example, the words '*nayane*' (meaning eyes) and '*kandarpam*' (meaning desire for love) are repeated with the same sense. The words are sometimes an independent word or a part of a compound word. According to the role of the word in the sentence, the implication changes. The first appearance of *nayane* in *smerarājīvanayane* (meaning - a woman with lotus like eyes) is in the form of an element of the compound, and contributes its meanings to form a meaningful compound. The other occurrence of '*nayane*' is an independent word to give the meaning as 'two eyes'. Similarly, in '*kandarpam*', both repetitions possess the same meaning, but the implication differs in each occurrence.

2.3.3 Implementation: Anuprāsa Identifier and Classifier

From the definitions of various *Anuprāsa*s, we note that some types have more stringent conditions than others. Hence, the examples that satisfy more stringent conditions may also satisfy less stringent conditions and thus can be categorised under two different types of *Anuprāsa*. For example, *Lāṭānuprāsa* demands that the repetition is of words and not syllables. *Chekānuprāsa* demands the repetition of syllables in the same order. Thus, any example of *Lāṭānuprāsa* is also potentially an example of *Chekānuprāsa* as well. However, due to the stringent conditions of *Lāṭānuprāsa*, it is appropriate to classify such an instance only under *Lāṭānuprāsa*. A similar situation exists with other pairs as well. In order to decide the proper exclusive sequence in which these *Anuprāsa*s should be identified, we look at the necessary conditions for each of them.

From the table 2.2, we understand that the natural order for identifying the *Anuprāsa* type is *Lāṭa, Cheka* and *Vṛtti*. The conditions of *śruti* and *antya* type of *Anuprāsa* do not clash with any other classes and hence can be identified either in the beginning or at the end.

We use the frequencies of n-grams[20] for identifying *Lāṭānuprasa* and frequencies of n-grams of the sequence of letters ignoring the vowels and their positions identifying the *Chekānuprāsa* and the frequencies of n-grams of consonants for identifying *Vṛttyanuprāsa* and only

[20] The sequences of letters of length 'n' are called n-grams.

Type	Count	Order	Unit	Position in the input
lāṭa	≥ 2	same	word	next to each other or with a few interventions.
cheka	2	same	sequence of syllables without vowels	anywhere within a proximity of $8 + 2*$ length of syllable sequence.
vṛtti	a)> 2 b)≥ 2	any	a) sequence of syllables without vowels b) a consonant	anywhere within a proximity of $8 + 2*$ length of syllable sequence.
śruti	≥ 2	any	syllables from the same class	within a proximity of 8 syllables.
antya	≥ 2	same	syllables after the second last vowel.	end of feet and words.

Table 2.2: Differentiation of the various types of *Anuprāsa* from the implementation point of view.

consonants having the same place of articulation and their positions in the input for identifying *Śrutyanuprāsa*. The *Pādāntyānuprāsa* is identified by looking at rhymes at the end of the *pādas*. For *Padāntyānuprāsa*, we consider the rhyming in the space-separated word endings.

Normalization of the input is an important step in the processing. To analyse *Antyānuprāsa*, the *daṇḍa*s ('|') and spaces to mark the word and the foot boundary are preserved. For other types, the normalization of various elements like removing the white spaces, *daṇḍas* and *avagraha*s, changing *anunāsikas* to *anusvāra* are done.

Algorithm for *Anuprāsa* identification and classification

Input: A verse in any of the supported scripts or transliteration schemes (Devanagari, IAST, SLP, WX, HK, etc.)
Output: Identified *Anuprāsa* patterns and their classification based on *Viśvanātha*'s theory.

1. Read the input.
2. Convert the input to WX notation.
3. Normalization:
    - Remove spaces (except for Pādāntyānuprāsa).
    - Convert all nasal stops to Anusvāra.
    - Remove *Avagraha* and *Daṇḍa* (except for Pādāntyānuprāsa).
4. Check for *Pādāntyānuprāsa* by dividing the input into 4 equal parts and comparing the sequence of letters after the second last vowel at the end of each part.

5. Check for *Padāntyanuprāsa* by comparing the word endings from the penultimate vowel to the end of the consecutive words. If the sequence of letters matches at least in two words mark the repeated sequence as *Padāntyānuprāsa*.

6. Get the n-grams ($n \geq 2$) along with their positions with and without vowels.

7. Remove all the small n-grams that can be subsumed by the large n-grams with matching positions (indices).

8. **If** the frequency of n-gram with vowels is more than 1, mark it as *Lāṭānuprāsa*.

9. **Else if** the frequency of n-grams without vowels is 2, mark such sequences as *Chekānuprāsa*.

10. **Else if** the frequency of n-grams ($n \geq 2$) without vowels is greater than 2 and if the frequency of single consonants is greater than or equal to 2, then the n-gram or the consonants are marked as *Vṛttyanuprāsa*.

11. **If** the frequency of consonants belonging to the same place of articulation is greater than 2, mark them as *Śrutyanuprāsa* of the type to which these consonants belong.

The use of else if ensures that the classification prefers a type with a more stringent definition than the others.

As a general rule in *Anuprāsa*, the repetitions should not be far away to make the reader forget the previous occurrence. If the distance is large, the instance will not be able to produce amusement for the reader. To strike out such cases, we have added a function in which the distance is calculated through the indices of the repeated consonants. For a single letter repetition, the maximum distance is considered to be 4 to 5 *akṣara*s, that is 8 to 12 letters approximately, considering the frequent conjuncts in Sanskrit.

2.3.4 Performance Evaluation

We tested our tool on a data set of 70 *ślokas* and 10 sample prose. The selected *ślokas* are primarily given as examples of *Anuprāsa* in the *alaṅkāraśāstra* texts and some from the *Raghuvaṃśa* of *Kālidāsa*. The prose examples were passages consisting of 2 to 3 sentences from *Bāṇabhaṭṭa's Kādambarī*. Most examples contained more than one type of *Anuprāsa*. This

tool could successfully handle these *Anuprāsa* instances. Since the three classes of *Anuprāsa* viz. *Lāṭānuprāsa*, *Chekānuprāsa*, and *Vṛttyanuprāsa* relax the conditions as we go from the first one to the third; the latter is a strict superset of the previous one. Hence, a pattern satisfying the conditions of *Lāṭānuprāsa*, though it is an example of *Chekānuprāsa* and *Vṛttyanuprāsa*, is shown only under the *Lāṭānuprāsa*. Similarly, the patterns satisfying the conditions for *Chekānuprāsa* are not again displayed under *Vṛttyanuprāsa*. Similarly, only those patterns that are not covered under *Lāṭānuprāsa* and *Chekānuprāsa*, are considered for *Vṛttyanuprāsa*.

Since this tool is not supported with word segmenter or meter identifier, for analysis of *Antyānuprāsa* it completely relies on the spaces and the *daṇḍas* to mark the *pada* and *pāda* boundary. If the user has not provided the *daṇḍa* or spaces in the appropriate place, the tool is not able to identify *Antyānuprāsa*.

2.4 Interface: Śabdālaṅkār Identifier and Classifier

We have designed a user-friendly interface to analyze *śabdālaṅkāra*s. This is an integrated tool for both *Yamaka* and *Anuprāsa*. User can provide their input text in various available encodings.

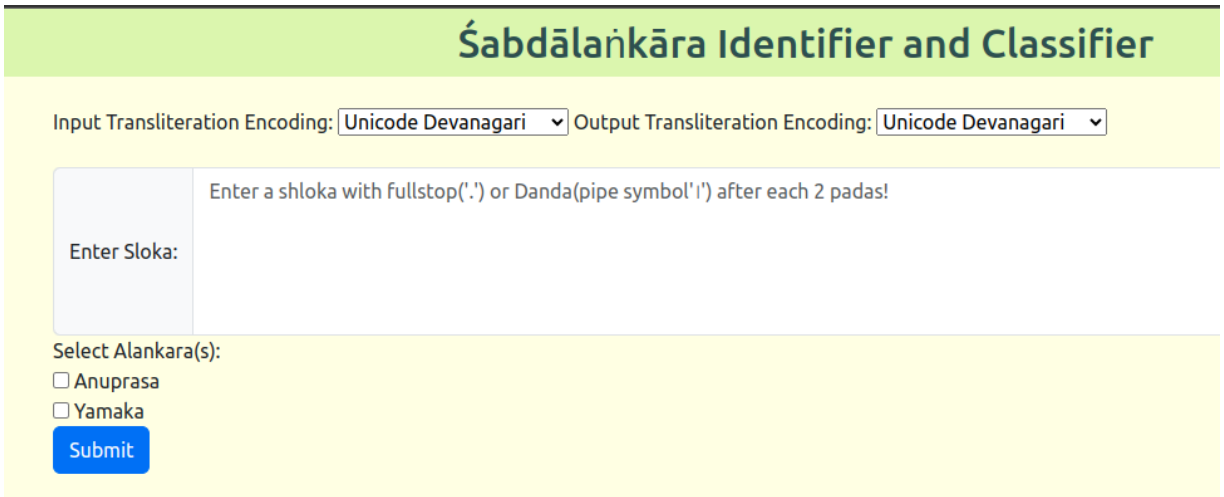


Figure 2.1: Śabdālaṅkāra Identifier and Classifier: Interface

To analyze *Yamaka*, we provide verse as an input without any special symbols except for the full stops or pipe symbols (Daṇḍa) to mark the two different lines of the verse. Sample input and output are shown in Figure 2.2 and Figure 2.3 respectively. In the given input, the

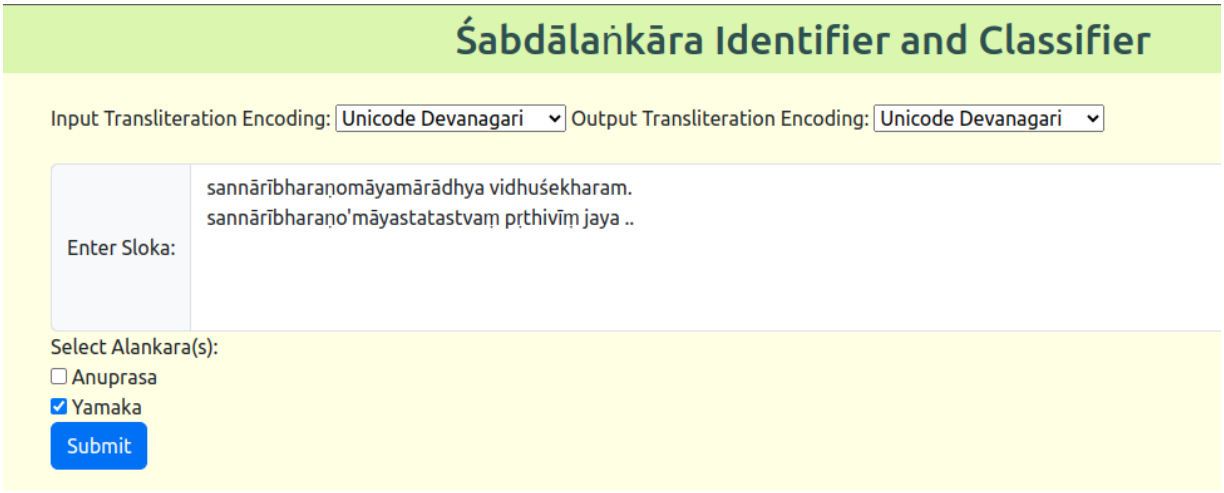


Figure 2.2: Śabdālaṅkāra Identifier and Classifier: Yamaka input

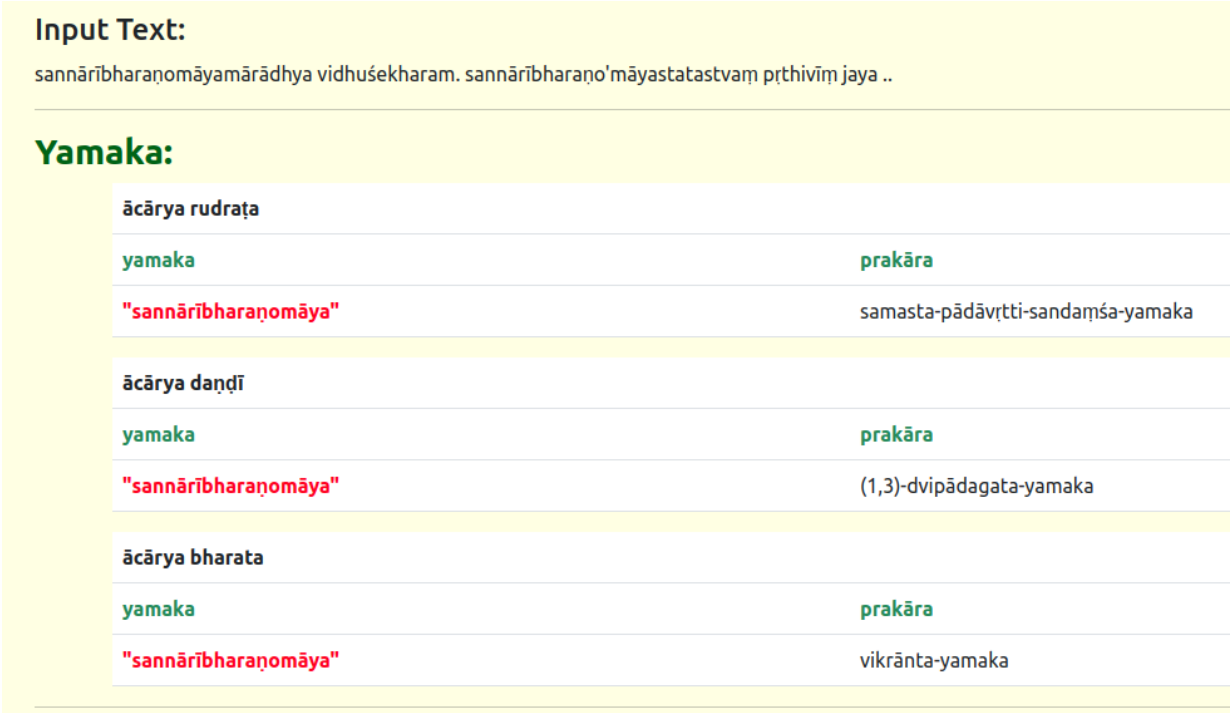


Figure 2.3: Śabdālaṅkāra Identifier and Classifier: Yamaka output

first and the third feet are identical. Hence, the output for the verse having this condition of *Yamaka* is called '*samasta-pādāvṛtti-sandaṃśa-yamaka*' in *Rudraṭa*s classification. According to *Daṇḍin* it is a case of '*(1,3)-dvipādagata-yamaka*'. Here, the numbers in the bracket are the *pāda* number of identical feet. In *Bharata*'s scheme, the same condition is named as '*vikrānta-yamaka*'.

Similarly for *Anuprāsa*, if the input is a verse it should be having fullstops or *Daṇḍa*s marking two separate lines of the verse. However, *Anuprāsa* can also be found in prose, it

also processes the input without the fullstops or *Daṇḍa*. If the input does not have line demarcation, it is considered that the input is in prose format. In this case, *Pādāntyānuprāsa* is not checked for the input. Figure 2.4 shows all five types of *Anuprāsa* highlighted in red colour corresponding to the input given by the user. The highlighted sequence facilitates the user with a better comprehension of the *alaṅkāra* and helps the user understand the difference between each type of *Anuprāsa* effectively and easily.

The interface is available in the 'tools' section at `https://sanskrit.uohyd.ac.in/scl/`

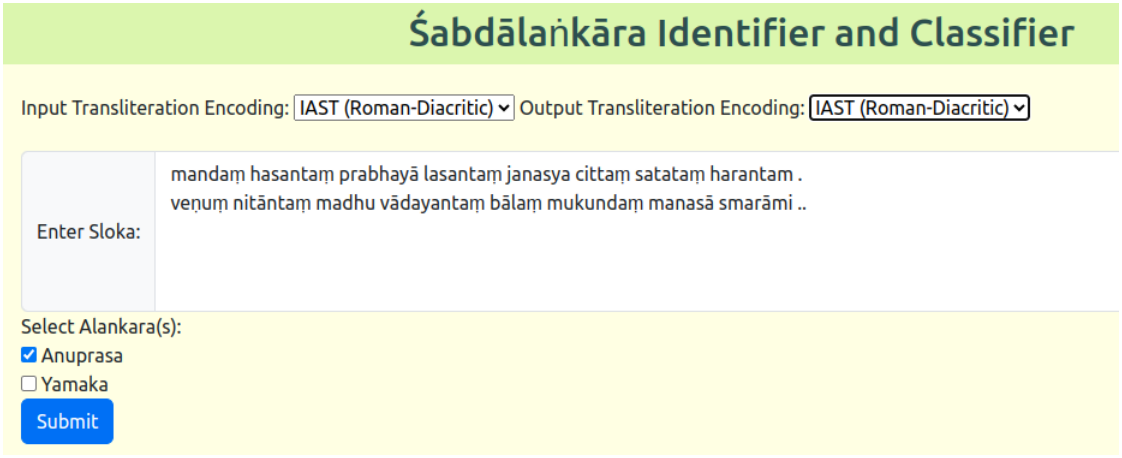


Figure 2.4: Śabdālaṅkāra Identifier and Classifier: Anuprāsa input

2.5 Conclusion

We have presented tools developed for the identification and classification of Sanskrit poetry focusing on two distinct *śbdālaṅkāra*s viz. *Yamaka* and *Anuprāsa*. The concept of *Yamaka* and *Anuprāsa* along with its classification is adopted by other Indian languages such as Hindi, Marathi, Telugu, Kannada, etc. The same model with the necessary amendments can be deployed for the identification and classification of *Anuprāsa* in modern Indian languages as well.

This module is extendable for other classifications presented in the tradition. The *Vṛttyanuprāsa*, a type of *Anuprāsa*, can be researched extensively to identify *rasa* depending upon the repetition of clusters of consonants.



Figure 2.5: Śabdālaṅkāra Identifier and Classifier: Anuprāsa output

These tools are useful for teaching *Yamaka* and *Anuprāsa*, the *śabdālaṅkāra*s along with various classification schemes by presenting a demonstration of different examples. While creating a masterpiece of poetry, a good poet does not deliberately enforce the figures in the poetry. Such upcoming masterpieces in Sanskrit can be tested with this tool.

We are providing the list of all the types of Yamaka dealt with in the this chapter along with the classical examples given by the rhetoricians in their treatises. This list has been published under the Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0) license. The list is available at `https://drive.google.com/drive/folders/1LaTyBd_YtrKJDaLVgVmHUAfFDPbXQns0?usp=sharing`.

Chapter 3

Mathematical Analysis of the Symmetries in Bharata's Yamaka

3.1 Introduction

The word symmetry (originated from the Greek word 'Symmetria') refers to the arrangement in dimensions or the due proportion. It refers to certain compositions of structural objects that preserve the structure. It acts as a measure of the beauty of the form or structure. Therefore it has always been considered as an important factor in visual aesthetics (Molnar and Molnar, 1986). Symmetry is abundant in nature. Several studies discuss various kinds of symmetry in living and non-living objects such as plants, living organisms, the structure of chemical molecules and organic compounds, etc. (Lederman and Hill, 2005), (Bunker and Jensen, 2004). The symmetry in arts, architecture, music and literature is also studied in great detail (Pavlovic and Triajstic, 1986).

Indian civilisation is no exception. According to Sihna (1989), 'Indian ethos has shaped through the centuries in evolving the facets of symmetry. Various shapes of Vedic altars constructed using the *Śulva sūtra*s are all symmetric in nature.' Some of the finest examples of symmetry are the fractal symmetry found in the temples and wells across India. Indian literature also exhibits symmetry in abundance. The *śabdālaṅkāra*s such as *Yamaka*, and various *Citrabandha*s[1] present examples of symmetry in literature.

Yamaka is used to enrich the beauty of the poetry. *Yamaka* is a figure of sound where a group of syllabic patterns is repeated, typically with a different meaning[2] at different locations in the same verse. The first classification of *Yamaka* is given by *Bharata* in his treatise *Nāṭya-*

[1] *Citrabandha* poetry is a kind of literary construction which creates visual patterns like Lotus (*Padma-bandha*), Sword (*Khaḍga-bandha*), Wheel (*Cakra-bandha*), etc. from the specific arrangement of syllables or word.

[2] *Bharata* does not insist on different meanings

śāstra. Later *Daṇḍin, Rudraṭa* and many other scholars have given various classifications covering different patterns of repetitions. There is an elegance and simplicity in *Bharata*'s classification. It is interesting to note that all the patterns of *Yamaka* mentioned by *Bharata* are not only simple but also form a symmetric group under the operations of reflection and rotation.

In mathematics, symmetry is rigorously captured through the framework of group theory. A symmetry group comprises all the transformations, such as rotations and reflections, that can be applied to an object without altering its overall appearance. For instance, a square possesses eight distinct symmetries, including rotations by 90°, 180°, and 270°, as well as reflections across its axes and diagonals. In simpler words, in spite of rotating a square in 180° (up side down rotation), etc., it retains its identity of being a square. The upper side may come to the bottom but the main shape of the square is preserved. Generally, the greater the number of symmetric operations such as rotation, reflection, etc. an object exhibits, the higher its degree of symmetry, often contributing to a stronger aesthetic appeal.

In this chapter, we apply the group theory to study the symmetry in *Bharata*'s *Yamaka* and show the simplicity of his classification by proving that each pattern forms a symmetric group.

3.2 Group

The concept of group theory is used to measure the degree of symmetry for a shape or any structure. We start with the concept of binary operation. We know that, when we mix blue color to yellow color we get green color.

Blue + Yellow = Green

A binary operation is simply a rule that operates two elements to give one output. Here, "+" is the binary operation, and blue and yellow are the two inputs. Green is the output. All three colors, Blue, Yellow, and Green, belong to the same set, which in this case is the set of colors. Also, these can be more than one operations.

Now, a group $(G, *)$ is a set G consisting of objects like colors, closed under a binary operation $*$ like mixing or adding, such that the following axioms are satisfied (Dummit and Foote, 2004).

- **Associativity:** When combining three or more elements (a, b, c) from the group G, the result is the same no matter how you group them — as long as the order of combining the elements stays the same.
  G is associative under the binary operation'$*$'.
  That is, for all $a, b, c \in$ [3]G, we have $(a * b) * c = a * (b * c)$.

  When we are doing $a * b * c$ if we group $(a * b)$ in $(a * b) * c$ or we group $(b * c)$ in $a * (b * c)$, the output remains same.

- **Identity:** There is an element I in G such that for all $a \in G$, $I * a = a * I = a$.

  When we combine a with I or I with a it gives a. After operating a with I element, a remains unchanged.

- **Inverse:** G has an inverse with respect to '$*$'.
  That is, corresponding to each element (eg. a) in G there is an inverse element (eg. a'[4]) in G such that $a * a' = a' * a = I$.

  When we combine an element with its inverse we should get the original identity element.

- **Closure**, a property of a group, also considered to be fulfilled. For all the elements $a, b, c \in G$, the result of the operation also belongs to the set G i.e. $a * b = c, c \in G$. It is implied in the binary operation.

'**Abelian group**' is a kind of group , if a group G is commutative under the binary operation $*$, that is, $a * b = b * a$, for all elements: $a, b \in G$
When the elements are combined in any order the result remains the same then, G is called an Abelian group.

In a nutshell, a group is a set of objects, like colors, shapes or structure that you can combine using a certain rule like mixing (i.e. binary operations), and no matter how you combine them, you always get another valid object from the same set. This combination

[3] = in
[4] =inverse of a

follows four special rules (closure, associativity, identity, and inverse), making the system structured and predictable.

3.3 Yamaka

According to *Bharata*, *Yamaka* is the repetition of sound.[5] Moreover, he described *Yamaka* with the definition of 10 sub-classes along with the examples, which we have elaborately discussed in section 2.2.1 in chapter 2. *Yamaka* is essentially employed in the metrical verses. *Yamaka* can occur at different positions in the verse. These 10 types of *Yamaka* are determined by the position and the number of repetitions.

Out of these ten classes, the last one, *Mālā-yamaka*being more of an *Anuprāsa* (see *Bharata* 10, Chapter 1), we ignore it. Hence, we study only the first nine. We describe each of the nine types by providing the abstract pattern associated with it as an symmetric structure (See Table 3.1) to perform symmetry operations.

The answers for the following questions we try to find out through this study, are:
While there can be other patterns of repetition possible, why does *Bharata* consider only these nine? What is special about them? Among these nine patterns, are some of them more symmetric compared to others?

To seek answers to these questions, we look at the symmetries each of these classes exhibits and the properties associated with such a set of symmetries. Further, to find out whether the symmetries involved in these nine different patterns are of the same order, or are some of them 'more' symmetric than the others, we use group theory, which provides rigorous tools to describe symmetries of geometric shapes. We also study other patterns with repetition of words that are not considered by *Bharata* to find out why *Bharata* did not find those patterns interesting.

In the next section, 3.4, we look at the symmetries associated with various *Yamaka*s. In section 3.5 we discuss the symmetries that are not considered by *Bharata*, and finally, we

[5] *śabdābhyāsastu yamakam* |16.60, *Nāṭyaśāstra*

Name & Repetition in pādas	**Pattern**	**Example**
Pādādi (Across)	$\begin{pmatrix} ya \\ yb \\ yc \\ yd \end{pmatrix}$	(y)**viṣṇuḥ**(a)sṛjati bhūtāni (y)**viṣṇuḥ**(b)saṃharate prajāḥ | (y)**viṣṇuḥ**(c)prasūte trailokyam (y)**viṣṇur**(d)lokādhi-daivatam ||
Pādānta (Across)	$\begin{pmatrix} ay \\ by \\ cy \\ dy \end{pmatrix}$	(a)dinakṣayāt saṃhṛtaraśmi(y)**maṇḍalam** (b)divīva lagnam tapanīya(y)**maṇḍalam** | (c)vibhāti tāmraṃ divi sūrya(y)**maṇḍalam** (d)yathā taruṇyāḥ stanabhāra(y)**maṇḍalam** ||
Sandaṣṭaka (Within)	$\begin{pmatrix} y^1y^1\ a \\ y^2y^2\ b \\ y^3y^3\ c \\ y^4y^4\ d \end{pmatrix}$	(y^1y^1)**paśya paśya**(a)ramaṇasya me guṇān (y^2y^2)**yena yena**(b)vāśagāṃ karoti mām || (y^3y^3)**yena yena**(c)hi sameti darśanam (y^4y^4)**tena tena**(d)vāśagāṃ karoti mām ||
Pādāntā -ṃreḍita (Within)	$\begin{pmatrix} a\ y^1y^1 \\ b\ y^2y^2 \\ c\ y^3y^3 \\ d\ y^4y^4 \end{pmatrix}$	(a) vijṛmbhitaṃ niḥśvasitaṃ | (y^1y^1)**muhurmuḥ** (b) kathaṃ vidheyaṃ smaraṇaṃ(y^2y^2)**pade pade** | (c) yathā ca te dhyānamidaṃ(y^3y^3)**punaḥ punaḥ** (d) dhuvaṃgatā te rajanī(y^4y^4)**vinā vinā** ||
Kāñcī (Within)	$\begin{pmatrix} y^1y^1\ a\ y^2y^2 \\ y^3y^3\ b\ y^4y^4 \\ y^5y^5\ c\ y^6y^6 \\ y^7y^7\ d\ y^8y^8 \end{pmatrix}$	(y^1y^1)**yāmaṃ yāmaṃ**(a)can(y^2y^2)**dravatīnām dravatīnām** (y^3y^3)**vyaktāvyaktā**(b)sā(y^4y^4)**rajanīnām rajanīnām** | (y^5y^5)**phulle phulle**(c)sam(y^6y^6)**bhramare vābhramare vā** (y^7y^7)**rāmā rāmā**(d)vi(y^8y^8)**smayate ca smayate ca** ||
Vikrānta (i) (Across)	$\begin{pmatrix} a \\ y \\ b \\ y \end{pmatrix}$	(a)sa pūrvaṃ vāraṇo bhūtvā (y)**dviśṛṅga iva parvataḥ** | (y)abhavaddantavaikalyāt (b)**dviśṛṅga iva parvataḥ** ||
Vikrānta (ii) (Across)	$\begin{pmatrix} y \\ a \\ y \\ b \end{pmatrix}$	(y)**sannārībharaṇo māyā** (a)mārādhya vidhuśekharam | (y)**sannārībharaṇo'māya** (b)stataḥ tvaṃ pṛthivīṃ jaya ||
Samudgaka (Across)	$\begin{pmatrix} y^1 \\ y^2 \\ y^1 \\ y^2 \end{pmatrix}$	(y^1)**ketakīkusumapāṇḍuradantaḥ** $\underline{(y^2)\text{śobhate pravarakānanahastī}}$ | $\underline{(y^1)}$**ketakīkusumapāṇḍuradantaḥ** $\underline{(y^2)\text{śobhate pravarakānanahastī}}$ ||
Cakrāvala (Across)	$\begin{pmatrix} a\ y^1 \\ y^1\ y^2 \\ y^2\ y^3 \\ y^3\ b \end{pmatrix}$	(a)śailā yathā śatrubhirāhatā(y^1)**hatā** (y^1)**hatā**śca bhūyastvanupuṅkhapuṅ(y^2)**khagaiḥ** | (y^2)**khagai**śca sarvairyudhi sañcitā(y^3)**ścitā** (y^3)**ścitā**(b)dhirūḍhā nihitāstalaistalaiḥ ||
Catur -vyavasita (Across)	$\begin{pmatrix} y \\ y \\ y \\ y \end{pmatrix}$	(y)**vāraṇānāmayameva kālo** (y)**vāraṇānāmayameva kālaḥ** | (y)**vāraṇānāmayameva kālo** (y)**vāraṇānāmayameva kālaḥ** ||

Table 3.1: Symmetry patterns in *Bharata*'s classification with Examples

conclude with what *Bharata* has regarded as the symmetries found in the literature, and how they correspond to symmetric groups of type $\mathbb{Z}_1$ or $\mathbb{Z}_2$ or $\mathbb{Z}_2 * \mathbb{Z}_2$.

3.4 Symmetry analysis of various types of *Yamaka*

For each type of *Yamaka*, now we consider the transformations that result in a structure that is similar in form to the original structure. We take the examples given by *Bharata* for each type and apply the symmetry operations to it.

1. ***Pādādi - Pādānta***

   We consider both *Pādādi* and *Pādānta* together since they are the reflections of each other if we divide each line in two parts. i.e. the repeated part and non-repeated part see table 3.2. In these types of *Yamaka*s, the pattern repeats either at the beginning or at the end of each *pāda*.

(y) viṣṇuḥ	**(a)** sṛjati bhūtāni	**(a)** dinakṣayāt saṃhṛtaraśmi	**(y)** maṇḍalam
(y) viṣṇuḥ	**(b)** saṃharate prajāḥ	**(b)** divīva lagnam tapanīya	**(y)** maṇḍalam
(y) viṣṇuḥ	**(c)** prasūte trailokyam	**(c)** vibhāti tāmraṃ divi sūrya	**(y)** maṇḍalam
(y) viṣṇur	**(d)** lokādhi-daivatam	**(d)** yathā taruṇyāḥ stanabhāra	**(y)** maṇḍalam
Pādādi Yamaka		***Pādānta Yamaka***	

Table 3.2: Representation of *Yamaka* using variables: *Pādādi* and *Pādānta* types

To perform various symmetry operations, we represent the verse mathematically using a variable for better understanding. In the example of Pādādi, we can see that in *Pādādi* type of *Yamaka*, the same sequence of letters, i.e. '*viṣṇuḥ*', repeats at the beginning of all four lines of a verse. So, let y be that common sequence of syllables and a, b, c, and d be the uncommon sequence of syllables in the four lines in order. Similarly, the variable y denotes the repeated pattern '*maṇḍalam*' and a, b, c, d be the uncommon sequence of the syllables. The *Pādādi* and *Pādānta Yamaka*s may be represented as,

$$\begin{pmatrix} ya \\ yb \\ yc \\ yd \end{pmatrix} \; and \; \begin{pmatrix} ay \\ by \\ cy \\ dy \end{pmatrix} \; respectively.$$

Now, let us consider the following operations on this mathematical representation such that we get the isomorphic form of the *Yamaka* within verse.

(a) The **vertical reflection** V, maps a *Pādādi* into *Pādānta* and vice versa. It swipes the places of the repeated sequence with the non-repeated sequence, as shown in table 3.3 and 3.4:

(y) viṣṇuḥ	**(a)** sṛjati bhūtāni		**(a)** sṛjati bhūtāni	**(y)** viṣṇuḥ
(y) viṣṇuḥ	**(b)** saṃharate prajāḥ	→	**(b)** saṃharate prajāḥ	**(y)** viṣṇuḥ
(y) viṣṇuḥ	**(c)** prasūte trailokyam		**(c)** prasūte trailokyam	**(y)** viṣṇuḥ
(y) viṣṇur	**(d)** lokādhi-daivatam		**(d)** lokādhi-daivatam	**(y)** viṣṇuḥ
Pādādi Yamaka			**After applying V (Vertical flipping)**	

Table 3.3: Representation of vertical reflection (V) using variables and verse: *Pādādi* type

(a) dinakṣayāt saṃhṛtaraśmi	**(y)** maṇḍalam		**(y)** maṇḍalam	**(a)** dinakṣayāt saṃhṛtaraśmi
(b) divīva lagnam tapanīya	**(y)** maṇḍalam	→	**(y)** maṇḍalam	**(b)** divīva lagnam tapanīya
(c) vibhāti tāmraṃ divi sūrya	**(y)** maṇḍalam		**(y)** maṇḍalam	**(c)** vibhāti tāmraṃ divi sūrya
(d) yathā taruṇyāḥ stanabhāra	**(y)** maṇḍalam		**(y)** maṇḍalam	**(d)** yathā taruṇyāḥ stanabhāra
Pādānta Yamaka			**After applying V (Vertical flipping)**	

Table 3.4: Representation of Vertical reflection (V) using variables and verse: *Pādānta* type

(b) **Reflection along the horizontal axis** H, divides the verse into two halves. We divide the verse that has four lines into two lines in each half and flip them, resulting in a pattern that retains the class of the original *Pādādi / Pādānta Yamaka*. Tables 3.5 and 3.6 show the transformations on *Pādādi* and *Pādānta* respectively.

(y) viṣṇuḥ	**(a)** sṛjati bhūtāni		**(y)** viṣṇur	**(d)** lokādhi-daivatam
(y) viṣṇuḥ	**(b)** saṃharate prajāḥ	→	**(y)** viṣṇuḥ	**(c)** prasūte trailokyam
(y) viṣṇuḥ	**(c)** prasūte trailokyam		**(y)** viṣṇuḥ	**(b)** saṃharate prajāḥ
(y) viṣṇur	**(d)** lokādhi-daivatam		**(y)** viṣṇuḥ	**(a)** sṛjati bhūtāni
Pādādi Yamaka			**After applying H (Horizontal flipped)**	

Table 3.5: Representation of of Horizontal reflection (H) using variables and verse: *Pādādi* type

(c) Let R represent the **rotation by an angle of** $180°$. This transformation is shown in Tables 3.7 and 3.8 for *Pādādi* and *Pādānta* types respectively.

Note that, R also maps a *Pādādi* pattern to *Pādānta* and vice versa.

(a) dinakṣayāt saṃhṛtaraśmi	**(y)** maṇḍalam	
(b) divīva lagnam tapanīya	**(y)** maṇḍalam	
(c) vibhāti tāmraṃ divi sūrya	**(y)** maṇḍalam	→
(d) yathā taruṇyāḥ stanabhāra	**(y)** maṇḍalam	

Pādānta Yamaka

(d) yathā taruṇyāḥ stanabhāra	**(y)** maṇḍalam
(c) vibhāti tāmraṃ divi sūrya	**(y)** maṇḍalam
(b) divīva lagnam tapanīya	**(y)** maṇḍalam
(a) dinakṣayāt saṃhṛtaraśmi	**(y)** maṇḍalam

After applying H (Horizontal flipping)

Table 3.6: Representation of of Horizontal reflection (H) using variables and verse: *Pādānta* type

(y) viṣṇuḥ	**(a)** sṛjati bhūtāni		**(d)** lokādhi-daivatam	**(y)** viṣṇuḥ
(y) viṣṇuḥ	**(b)** saṃharate prajāḥ		**(c)** prasūte trailokyam	**(y)** viṣṇuḥ
(y) viṣṇuḥ	**(c)** prasūte trailokyam	→	**(b)** saṃharate prajāḥ	**(y)** viṣṇuḥ
(y) viṣṇur	**(d)** lokādhi-daivatam		**(a)** sṛjati bhūtāni	**(y)** viṣṇuḥ

Pādādi Yamaka **After applying R rotation by an angle of** $180°$

Table 3.7: Representation of Rotational transformation (R) using variables and verse: *Pādādi* type

(a) dinakṣayāt saṃhṛtaraśmi	**(y)** maṇḍalam		**(y)** maṇḍalam	**(d)** yathā taruṇyāḥ stanabhāra
(b) divīva lagnam tapanīya	**(y)** maṇḍalam		**(y)** maṇḍalam	**(c)** vibhāti tāmraṃ divi sūrya
(c) vibhāti tāmraṃ divi sūrya	**(y)** maṇḍalam	→	**(y)** maṇḍalam	**(b)** divīva lagnam tapanīya
(d) yathā taruṇyāḥ stanabhāra	**(y)** maṇḍalam		**(y)** maṇḍalam	**(a)** dinakṣayāt saṃhṛtaraśmi

Pādānta Yamaka **After applying R rotation by an angle of** $180°$

Table 3.8: Representation of Rotational transformation (R) using variables and verse: *Pādānta* type

(d) Finally, let 'I' be an **identity mapping** that maps an element into itself. The Identity operation is shown in Table 3.9 and Table 3.10 for *Pādādi Yamaka* and *Pādānta Yamaka* respectively.

(y) viṣṇuḥ	**(a)** sṛjati bhūtāni			**(y)** viṣṇuḥ	**(a)** sṛjati bhūtāni
(y) viṣṇuḥ	**(b)** saṃharate prajāḥ			**(y)** viṣṇuḥ	**(b)** saṃharate prajāḥ
(y) viṣṇuḥ	**(c)** prasūte trailokyam	→		**(y)** viṣṇuḥ	**(c)** prasūte trailokyam
(y) viṣṇur	**(d)** lokādhi-daivatam			**(y)** viṣṇur	**(d)** lokādhi-daivatam
Pādādi Yamaka				**After applying *I* (Identity operation)**	

Table 3.9: Representation of Identity operation (I) using variables and verse: *Pādādi* type

(a) dinakṣayāt saṃhṛtaraśmi	**(y)** maṇḍalam			**(a)** dinakṣayāt saṃhṛtaraśmi	**(y)** maṇḍalam
(b) divīva lagnam tapanīya	**(y)** maṇḍalam			**(b)** divīva lagnam tapanīya	**(y)** maṇḍalam
(c) vibhāti tāmraṃ divi sūrya	**(y)** maṇḍalam	→		**(c)** vibhāti tāmraṃ divi sūrya	**(y)** maṇḍalam
(d) yathā taruṇyāḥ stanabhāra	**(y)** maṇḍalam			**(d)** yathā taruṇyāḥ stanabhāra	**(y)** maṇḍalam
Pādānta Yamaka				**After applying *I* (Identity operation)**	

Table 3.10: Representation of Identity operation (I) using variables and verse: *Pādānta* type

Having illustrated how the four transformations map a verse into different patterns, now we see the special characteristics of these patterns. Let $S = \{1, 2, 3, 4\}$ where $1, 2, 3, 4$ show all the symmetric patterns that we get after the symmetric transformations. Here, we are elaborating for *Pādānta Yamaka*, although if we start from second operation, it will be applicable for *Pādādi Yamaka*.

$$1 = \begin{pmatrix} ay \\ by \\ cy \\ dy \end{pmatrix}, 2 = \begin{pmatrix} ya \\ yb \\ yc \\ yd \end{pmatrix}, 3 = \begin{pmatrix} dy \\ cy \\ by \\ ay \end{pmatrix}, and \;\; 4 = \begin{pmatrix} yd \\ yc \\ yb \\ ya \end{pmatrix}.$$

Let $\sigma_v, \sigma_h, \sigma_r$ represent the vertical, horizontal and rotational transformations. Then we note that,

$$\sigma_v(1) = 2, \sigma_v(2) = 1, \sigma_v(3) = 4, \text{and} \sigma_v(4) = 3. \tag{3.1}$$

Here, the decomposition can be represented as $(13)(24)$.

$$\sigma_r(1) = 4, \sigma_r(4) = 1, \sigma_r(2) = 3, \text{and} \sigma_r(3) = 2. \tag{3.2}$$

That is, it is a decomposition of $1 \longleftrightarrow 2$ and $3 \longleftrightarrow 4$ or compactly represented as $(12)(34)$

$$\sigma_h(1) = 3, \sigma_h(3) = 1, \sigma_h(2) = 4, \text{and} \sigma_h(4) = 2. \tag{3.3}$$

The compact representation is $(14)(23)$

Let S_4 denote a symmetric group with all permutations of S, i.e., it consists of all patterns having after applying all operations, H, V, R and I. We notice that S with the above operations, viz. Horizontal reflection H, Vertical reflection V, and Rotation through $180°R$ form a subgroup of S_4, termed the permutation group represented compactly as

$G = \{I, (12)(34), (13)(24), (14)(23)\}$

where $(12)(34)$, $(13)(24)$, $(14)(23)$ are the cycle decompositions of S_4 representing vertical symmetry(σ_v), horizontal symmetry(σ_h) and rotational symmetry(σ_r) respectively, as above.

Let $*$ be the binary composition operation defined as
$(a * b)(X) = a[b(X)]$, where X is any verse of the type *Pādādi* or *Pādānta Yamaka* and $a, b \in G$.

The composition table for the set G is shown in Table 3.11.

$*$	I	σ_v	σ_h	σ_r
I	I	σ_v	σ_h	σ_r
σ_v	σ_v	I	σ_r	σ_h
σ_h	σ_h	σ_r	I	σ_v
σ_r	σ_r	σ_h	σ_v	I

Table 3.11: Composition matrix : *Pādādi / Pādānta Yamaka*

From the composition matrix (table 3.11), we notice that G has the following properties.

(a) **Closure**: $\forall\ a, b \in G, a * b \in G$, and hence, G satisfies the closure property.

(b) **Identity**: The identity mapping 'I' is the identity element, which is evident from the first row and the first column of the matrix.

(c) **Inverse**: The diagonal elements of the composition matrix are equal to the identity element, implying that each element is self-inverse.

(d) **Associativity**: To confirm, we need to check whether it satisfies the associative law, i.e.$\forall\ x, y, z \in G, x * (y * z) = (x * y) * z$.

i. If any of the elements is 'I' then the equality holds.

ii. The composition of any two of the three non-identity elements produces the third non-identity element.

iii. If all x, y, z are different and are different from the identity element,
then since the composition of two non-identity elements is the third non-identity element,
$(x * y) * z = z * z = I$,
and $x * (y * z) = x * x = I$.
Thus, equality holds.

iv. If two of the elements are equal,
let us assume $x = y \neq z$.
Then
$(x * y) * z = I * z = z$.
$x * (y * z) = x * w$, where w is different from y and z, and not I.
$= z$, since $x = y$.

Thus, we see that in all 64 possible combinations of *x,y,z* the associative law holds good.

(e) **Commutativity**: From the symmetry along the main diagonal of the table 3.11, it follows that the operation is commutative.

This confirms that $G \simeq \mathbb{Z}_2 \times \mathbb{Z}_2$ and specifically G is a Klein four group K_4.

2. ***Sandaṣṭaka* and *Pādāntāmreḍita***

Now, we consider the two types, viz. *Sandaṣṭaka* and *Pādāntāmreḍita*, where the same sequence of letters repeats twice at the beginning or end of each line of a verse. As in the above case, these patterns are vertical reflections of each other, so we can consider

these two types together. So let (y^1y^1), (y^2y^2), (y^3y^3), (y^4y^4) be that common sequence of syllables in each line in order. Let a, b, c, and d be the uncommon sequence of syllables in the four lines in order. Table 3.12 shows the parts in the verse and the corresponding variables for both the types.

(y^1y^1)paśya paśya	(a)ramaṇasya me guṇān	(a)vijṛmbhitaṃ niḥśvasitaṃ	(y^1y^1)muhurmuḥ
(y^2y^2)yena yena	(b)vāśagāṃ karoti mām	(b)kathaṃ vidheyaṃ smaraṇaṃ	(y^2y^2)pade pade
(y^3y^3)yena yena	(c)hi sameti darśanam	(c)yathā ca te dhyānamidaṃ	(y^3y^3)punaḥ punaḥ
(y^4y^4)tena tena	(d)vāśagāṃ karoti mām	(d)dhuvaṃgatā te rajanī	(y^4y^4)vinā vinā
Sandaṣṭaka Yamaka		***Pādāntāmreḍita Yamaka***	

Table 3.12: Representation of *Yamaka* using variables: *Sandaṣṭaka* and *Pādāntāmreḍita* types

Then the patterns for *Sandaṣṭaka* and *Pādāntāmreḍita* would correspond to

$$\begin{pmatrix} (y^1y^1)\ a \\ (y^2y^2)\ b \\ (y^3y^3)\ c \\ (y^4y^4)\ d \end{pmatrix} \ and \ \begin{pmatrix} a\ (y^1y^1) \\ b\ (y^2y^2) \\ c\ (y^3y^3) \\ d\ (y^4y^4) \end{pmatrix} \ respectively.$$

We show the symmetric transformations using the above-given representation.

Now, as in the previous case, here we also consider four transformations, viz. Vertical reflection (V), Horizontal reflection (H), rotation through $180°$ (R) and the identity mapping. These are defined below.

(a) **Vertical Reflection** (V)

$$V \ \text{(Sandaṣṭaka)} : \begin{pmatrix} (y^1y^1)\ a \\ (y^2y^2)\ b \\ (y^3y^3)\ c \\ (y^4y^4)\ d \end{pmatrix} \rightarrow \begin{pmatrix} a\ (y^1y^1) \\ b\ (y^2y^2) \\ c\ (y^3y^3) \\ d\ (y^4y^4) \end{pmatrix}.$$

$$V \text{ (Pādāntāmreḍita)} : \begin{pmatrix} a\ (y^1y^1) \\ b\ (y^2y^2) \\ c\ (y^3y^3) \\ d\ (y^4y^4) \end{pmatrix} \rightarrow \begin{pmatrix} (y^1y^1)\ a \\ (y^2y^2)\ b \\ (y^3y^3)\ c \\ (y^4y^4)\ d \end{pmatrix}.$$

Here the *Sandaṣṭaka Yamaka* gets transformed to *Pādāntāmreḍita* and vice versa.

(b) Let H represent the **horizontal reflection** defined as follows:

$$H \text{ (Sandaṣṭaka)} : \begin{pmatrix} (y^1y^1)\ a \\ (y^2y^2)\ b \\ (y^3y^3)\ c \\ (y^4y^4)\ d \end{pmatrix} \rightarrow \begin{pmatrix} (y^4y^4)\ d \\ (y^3y^3)\ c \\ (y^2y^2)\ b \\ (y^1y^1)\ a \end{pmatrix}.$$

$$H \text{ (Pādāntāmreḍita)} : \begin{pmatrix} a\ (y^1y^1) \\ b\ (y^2y^2) \\ c\ (y^3y^3) \\ d\ (y^4y^4) \end{pmatrix} \rightarrow \begin{pmatrix} d\ (y^4y^4) \\ c\ (y^3y^3) \\ b\ (y^2y^2) \\ a\ (y^1y^1) \end{pmatrix}.$$

This transformation retains the type of *Yamaka*.

(c) Let R represent the **rotation by an angle of** $180°$.

$$R \text{ (Sandaṣṭaka)} : \begin{pmatrix} (y^1y^1)\ a \\ (y^2y^2)\ b \\ (y^3y^3)\ c \\ (y^4y^4)\ d \end{pmatrix} \rightarrow \begin{pmatrix} d\ (y^4y^4) \\ c\ (y^3y^3) \\ b\ (y^2y^2) \\ a\ (y^1y^1) \end{pmatrix}.$$

$$R \text{ (Pādāntāmreḍita)} : \begin{pmatrix} a\ (y^1y^1) \\ b\ (y^2y^2) \\ c\ (y^3y^3) \\ d\ (y^4y^4) \end{pmatrix} \rightarrow \begin{pmatrix} (y^4y^4)\ d \\ (y^3y^3)\ c \\ (y^2y^2)\ b \\ (y^1y^1)\ a \end{pmatrix}.$$

Here, also we see that R transforms a *Sandaṣṭaka* to *Pādāntāmreḍita* and vice versa.

(d) Let 'I' be an **identity element** that maps an element to itself.

$$I \text{ (Sandaṣṭaka)} : \begin{pmatrix} (y^1y^1)\ a \\ (y^2y^2)\ b \\ (y^3y^3)\ c \\ (y^4y^4)\ d \end{pmatrix} \rightarrow \begin{pmatrix} (y^1y^1)\ a \\ (y^2y^2)\ b \\ (y^3y^3)\ c \\ (y^4y^4)\ d \end{pmatrix}.$$

$$I \text{ (Pādāntāmreḍita)} : \begin{pmatrix} a\ (y^1y^1) \\ b\ (y^2y^2) \\ c\ (y^3y^3) \\ d\ (y^4y^4) \end{pmatrix} \rightarrow \begin{pmatrix} a\ (y^1y^1) \\ b\ (y^2y^2) \\ c\ (y^3y^3) \\ d\ (y^4y^4) \end{pmatrix}.$$

Now, let $S = \{1, 2, 3, 4\}$ where

$$1 = \begin{pmatrix} (y^1y^1)\ a \\ (y^2y^2)\ b \\ (y^3y^3)\ c \\ (y^4y^4)\ d \end{pmatrix}, 2 = \begin{pmatrix} a\ (y^1y^1) \\ b\ (y^2y^2) \\ c\ (y^3y^3) \\ d\ (y^4y^4) \end{pmatrix}, 3 = \begin{pmatrix} (y^4y^4)\ d \\ (y^3y^3)\ c \\ (y^2y^2)\ b \\ (y^1y^1)\ a \end{pmatrix}, \textit{and}\ \ 4 = \begin{pmatrix} d\ (y^4y^4) \\ c\ (y^3y^3) \\ b\ (y^2y^2) \\ a\ (y^1y^1) \end{pmatrix}.$$

Then, as in the previous case, here also the permutation group G of S_4 is represented compactly as

$G = \{I, (12)(34), (13)(24), (14)(23)\}$

where $(12)(34)$, $(13)(24)$, $(14)(23)$ are the cycle decompositions of S_4 representing vertical symmetry(σ_v), horizontal symmetry(σ_h) and rotational symmetry (σ_r) respectively, which are defined as in equations 3.1, 3.2 and 3.3 with the same composition matrix defined in Table 3.11.

Thus, these two types of *Yamaka*s also confirm that $S \simeq \mathbb{Z}_2 \times \mathbb{Z}_2$ and specifically S is a Klein four group K_4.

3. ***Kāñcī***

In *Kāñcī* type of *Yamaka*, the same sequence of letters repeats twice at the beginning as well as at the end of each line of a verse. So let y^1y^1, y^3y^3, y^5y^5, and y^7y^7 be that common sequence of syllables at the beginning of each line in order. Let y^2y^2, y^4y^4, y^6y^6, and y^8y^8 be the common sequence of syllables at the end of each line in order. Let

a, b, c, and d be the uncommon sequence of syllables in the middle of four lines in order.

So the verse can be represented as

$$\begin{pmatrix} (y^1y^1)\ a\ (y^2y^2) \\ (y^3y^3)\ b\ (y^4y^4) \\ (y^5y^5)\ c\ (y^6y^6) \\ (y^7y^7)\ d\ (y^8y^8) \end{pmatrix}$$

This being a combination of both *Sandaṣṭaka Yamaka* as well as *Pādāntāmreḍita*, the same transformations I, H, V and R as previously described produce a K_4. For the sake of completeness, we describe the transformations below.

(a) The vertical reflection V is defined as

$$V\ (\text{Kāñcī}) : \begin{pmatrix} (y^1y^1)\ a\ (y^2y^2) \\ (y^3y^3)\ b\ (y^4y^4) \\ (y^5y^5)\ c\ (y^6y^6) \\ (y^7y^7)\ d\ (y^8y^8) \end{pmatrix} \rightarrow \begin{pmatrix} (y^2y^2)\ a\ (y^1y^1) \\ (y^4y^4)\ b\ (y^3y^3) \\ (y^6y^6)\ c\ (y^5y^5) \\ (y^8y^8)\ d\ (y^7y^7) \end{pmatrix}.$$

(b) H be the horizontal reflection, which is defined as

$$H\ (\text{Kāñcī}) : \begin{pmatrix} (y^1y^1)\ a\ (y^2y^2) \\ (y^3y^3)\ b\ (y^4y^4) \\ (y^5y^5)\ c\ (y^6y^6) \\ (y^7y^7)\ d\ (y^8y^8) \end{pmatrix} \rightarrow \begin{pmatrix} (y^7y^7)\ d\ (y^8y^8) \\ (y^5y^5)\ c\ (y^6y^6) \\ (y^3y^3)\ b\ (y^4y^4) \\ (y^1y^1)\ a\ (y^2y^2) \end{pmatrix}.$$

(c) Let R represent the rotation by an angle of $180°$. It is defined as

$$R\ (\text{Kāñcī}) : \begin{pmatrix} (y^1y^1)\ a\ (y^2y^2) \\ (y^3y^3)\ b\ (y^4y^4) \\ (y^5y^5)\ c\ (y^6y^6) \\ (y^7y^7)\ d\ (y^8y^8) \end{pmatrix} \rightarrow \begin{pmatrix} (y^8y^8)\ d\ (y^7y^7) \\ (y^6y^6)\ c\ (y^5y^5) \\ (y^4y^4)\ b\ (y^3y^3) \\ (y^2y^2)\ a\ (y^1y^1) \end{pmatrix}.$$

(d) Let 'I' be the identity mapping, viz.

$$I \text{ (Kāñcī)} : \begin{pmatrix} (y^1y^1)\ a\ (y^2y^2) \\ (y^3y^3)\ b\ (y^4y^4) \\ (y^5y^5)\ c\ (y^6y^6) \\ (y^7y^7)\ d\ (y^8y^8) \end{pmatrix} \rightarrow \begin{pmatrix} (y^1y^1)\ a\ (y^2y^2) \\ (y^3y^3)\ b\ (y^4y^4) \\ (y^5y^5)\ c\ (y^6y^6) \\ (y^7y^7)\ d\ (y^8y^8) \end{pmatrix}$$

Now, let $S = \{1, 2, 3, 4\}$ where

$$1 = \begin{pmatrix} (y^1y^1)\ a\ (y^2y^2) \\ (y^3y^3)\ b\ (y^4y^4) \\ (y^5y^5)\ c\ (y^6y^6) \\ (y^7y^7)\ d\ (y^8y^8) \end{pmatrix}, 2 = \begin{pmatrix} (y^2y^2)\ a\ (y^1y^1) \\ (y^4y^4)\ b\ (y^3y^3) \\ (y^6y^6)\ c\ (y^5y^5) \\ (y^8y^8)\ d\ (y^7y^7) \end{pmatrix},$$

$$3 = \begin{pmatrix} (y^7y^7)\ d\ (y^8y^8) \\ (y^5y^5)\ c\ (y^6y^6) \\ (y^3y^3)\ b\ (y^4y^4) \\ (y^1y^1)\ a\ (y^2y^2) \end{pmatrix}, \textit{and}\ \ 4 = \begin{pmatrix} (y^8y^8)\ d\ (y^7y^7) \\ (y^6y^6)\ c\ (y^5y^5) \\ (y^4y^4)\ b\ (y^3y^3) \\ (y^2y^2)\ a\ (y^1y^1) \end{pmatrix}.$$

Then, as in the previous case, here also the permutation group G of S_4 is represented compactly as

$G = \{I, (12)(34), (13)(24), (14)(23)\}$

where $(12)(34)$, $(13)(24)$, $(14)(23)$ are the cycle decompositions of S_4 representing vertical symmetry(σ_v), horizontal symmetry(σ_h) and rotational symmetry(σ_r) respectively, which are defined as in the previous case with the same composition matrix as defined in Table 3.11.

Thus $G \simeq \mathbb{Z}_2 X \mathbb{Z}_2$ or Klein four group K_4.

4. ***Vikrānta***

In *Vikrānta*, any one pair of alternative lines of a verse is the same. That is, either the first and third lines are identical (type 1) or the second and fourth lines are identical

(type 2). We represent the type 1 and type 2 of *Vikrānta* respectively as

$$\begin{pmatrix} a \\ y \\ b \\ y \end{pmatrix}, \quad and \quad \begin{pmatrix} y \\ a \\ y \\ b \end{pmatrix}$$

Now, let us consider the following transformations.

(a) Horizontal reflection: Let H represent the horizontal reflection defined as follows.

$$H \text{ (Vikrānta 1)} : \begin{pmatrix} a \\ y \\ b \\ y \end{pmatrix} \rightarrow \begin{pmatrix} y \\ b \\ y \\ a \end{pmatrix}$$

and

$$H \text{ (Vikrānta 2)} : \begin{pmatrix} y \\ b \\ y \\ a \end{pmatrix} \rightarrow \begin{pmatrix} a \\ y \\ b \\ y \end{pmatrix}$$

(b) Identity Operation : In identity operation, the element goes to itself. That is

$$I \text{ (Vikrānta 1)} : \begin{pmatrix} a \\ y \\ b \\ y \end{pmatrix} \rightarrow \begin{pmatrix} a \\ y \\ b \\ y \end{pmatrix}, \quad and \quad I \text{ (Vikrānta 2)} : \begin{pmatrix} y \\ a \\ y \\ b \end{pmatrix} \rightarrow \begin{pmatrix} y \\ a \\ y \\ b \end{pmatrix}$$

Thus, we notice that the horizontal reflection changes the type of the *Yamaka* from 1 to 2 and 2 to 1, while retaining the main class - *Vikrānta*.

Now, let $S = \{1, 2\}$ where

$$1 = \begin{pmatrix} a \\ y \\ b \\ y \end{pmatrix}, \quad and \quad 2 = \begin{pmatrix} y \\ a \\ y \\ b \end{pmatrix}.$$

Let S_2 denote a symmetric group with all permutations of S. We notice that S with the above operations, i.e. Horizontal reflection (H) and Identity (I), forms a subgroup of S_2, termed the permutation group represented compactly as

$G = \{I, (12)\}$ where (12) is the cycle decomposition of S_2 representing horizontal symmetry(σ_h) defined as below.

$\sigma_h(1) = 2, \sigma_h(2) = 1$.

Let $*$ be the binary composition operation.

The composition table for the set G is shown in Table 3.13.

$*$	I	σ_h
I	I	σ_h
σ_h	σ_h	I

Table 3.13: Composition matrix : *Vikrānta Yamaka*

From the composition matrix (table 3.13), we notice that G has the following properties.

(a) **Closure**: $\forall\ a, b \in G, a * b \in G$, and hence G satisfies the closure property.

(b) **Identity**: The identity mapping 'I' is the identity element, which is evident from the first row and the first column of the matrix.

(c) **Inverse**: The diagonal elements of the composition matrix are equal to the identity element, implying that each element is self-inverse.

(d) **Associativity**: For this table, we need to check whether it satisfies the associative law, viz. $\forall\ x, y, z \in G, x * (y * z) = (x * y) * z$.

   i. If any of the elements is 'I' then the equality holds.

   ii. If none of the elements is 'I', all of them will be σ_h, and thus, the equality holds.

   Thus, we see that in all 8 possible combinations of x, y, and z, the associative law holds.

(e) **Commutativity**: From the symmetry along the main diagonal of the table 3.13, it follows that the operation is commutative.

This confirms that $G \simeq \mathbb{Z}_2$.

5. ***Samudgaka***

Here, the alternative lines of a verse are the same. That is first and third lines are the same, and the second and fourth lines are the same. Thus, this is a combination of *Vikrānta* type 1 and type 2. So the verse may be represented as,

$$\begin{pmatrix} y^1 \\ y^2 \\ y^1 \\ y^2 \end{pmatrix}.$$

Now, let us consider the following operations on this.

(a) Let H represent the horizontal reflection, which is defined as follows :

$$H \text{ (Samudgaka)} : \begin{pmatrix} y^1 \\ y^2 \\ y^1 \\ y^2 \end{pmatrix} \rightarrow \begin{pmatrix} y^2 \\ y^1 \\ y^2 \\ y^1 \end{pmatrix}$$

(b) Let 'I' be the identity mapping.

$$I \text{ (Samudgaka)} : \begin{pmatrix} y^1 \\ y^2 \\ y^1 \\ y^2 \end{pmatrix} \rightarrow \begin{pmatrix} y^1 \\ y^2 \\ y^1 \\ y^2 \end{pmatrix}.$$

Now, let $S = \{1, 2\}$ where

$$1 = \begin{pmatrix} y^1 \\ y^2 \\ y^1 \\ y^2 \end{pmatrix}, \quad \textit{and} \quad 2 = \begin{pmatrix} y^2 \\ y^1 \\ y^2 \\ y^1 \end{pmatrix}.$$

Let S_2 denote a symmetric group with all permutations of S. We notice that S with the above operations, i.e. Horizontal reflection and identity, forms a subgroup of S_2, termed as the permutation group represented compactly as

$G = \{I, (12)\}$ where (12) is the cycle decomposition of S_2 representing horizontal symmetry(σ_h) defined as below.

$\sigma_h(1) = 2, \sigma_h(2) = 1.$

Let $*$ be the binary composition operation with the composition table shown in Table 3.13.

This being the same group as in the previous case, here also $G \simeq \mathbb{Z}_2$.

6. ***Cakravāla***:

In the *Cakravāla* type of *Yamaka*, the end sequence of letters of the first line (y^1) is the beginning sequence of the second line (y^1) similarly the end sequence of letters of the second line (y^2) is the beginning sequence of the third line (y^2) and the end sequence of letters of the third line (y^3) is the beginning sequence of the fourth line (y^3). So the verse may be represented as

$$\begin{pmatrix} a\ y^1 \\ y^1\ y^2 \\ y^2\ y^3 \\ y^3\ b \end{pmatrix}$$

Now, let us consider the following transformations.

(a) Let R represent the rotational reflection, which rotates by an angle of $180°$. That is,

$$R : \begin{pmatrix} a\ y^1 \\ y^1\ y^2 \\ y^2\ y^3 \\ y^3\ b \end{pmatrix} \rightarrow \begin{pmatrix} b\ y^3 \\ y^3\ y^2 \\ y^2\ y^1 \\ y^1\ a \end{pmatrix}.$$

(b) Let 'I' be the identity mapping.

Let S_2 denote a symmetric group with all permutations of S. We notice that S with the above operations, i.e. Rotational reflection (R) and Identity (I), forms a subgroup of S_2, termed the permutation group represented compactly as

$G = \{I, (12)\}$ where (12) is the cycle decomposition of S_2 representing horizontal symmetry(σ_h) defined as below.

$\sigma_h(1) = 2, \sigma_h(2) = 1$.

Let $*$ be the binary composition operation with the composition table shown in Table 3.13.

Hence, *Cakravāla* type of *Yamaka* also is a symmetry group isomorphic to $\mathbb{Z}_2$

7. ***Caturvyavasita***

   In the *Caturvyavasita* type of *Yamaka*, all the lines are identical, represented as below.

$$\begin{pmatrix} y \\ y \\ y \\ y \end{pmatrix}$$

   Although difficult to construct, it is not a very interesting pattern, having less number of permutations. The only distinct operation is the identity operation, as any other operation yields the same element, all the lines being the same. So, in this type $G = \{I\}$ and hence, $G \simeq \mathbb{Z}_1$.

We observe that all the 9 types of *Yamaka* with one or more symmetry operations like reflection about the vertical axis, reflection in the horizontal axis and rotation through $180°$ form abelian groups that are either isomorphic to $\mathbb{Z}_2 * \mathbb{Z}_2$ or $\mathbb{Z}_2$ or $\mathbb{Z}_1$.

3.5 Discussion on missing symmetries

To answer this question, let us look at the possible repetitions in 4 feet. The repetitions can be

(a) within the same foot resulting in a symmetry due to reflection across the vertical axis, or

(b) across the feet resulting in a symmetry due to reflection across the horizontal axis.

3.5.1 Repetition within the same *pāda*

In the case of repetition in the same *pāda*, there are possible patterns as shown in the table 3.14.

$\begin{pmatrix} y^1 y^1\ a \\ y^2 y^2\ b \\ y^3 y^3\ c \\ y^3 y^4\ d \end{pmatrix}$	$\begin{pmatrix} y^1\ a\ y^1 \\ y^2\ b\ y^2 \\ y^3\ c\ y^3 \\ y^4\ d\ y^4 \end{pmatrix}$	$\begin{pmatrix} a\ y^1 y^1 \\ b\ y^2 y^2 \\ c\ y^3 y^3 \\ d\ y^4 y^4 \end{pmatrix}$	$\begin{pmatrix} y^1 y^1 \\ y^2\ y^2 \\ y^3 y^3 \\ y^4 y^4 \end{pmatrix}$
(1)	(2)	(3)	(4)

Table 3.14: Repetition in the same foot

Among these, the first and the third correspond to the *Sandaṣṭaka* and *Pādāntāmreḍita* of *Bharata*. The fourth is a special case since it results from the previous three patterns when a, b, c, and d are empty. The second one forms an abelian group $\mathbb{Z}_2$ with G defined as $G = \{I, (12)\})$ where

$$1 = \begin{pmatrix} y^1\ a\ y^1 \\ y^2\ b\ y^2 \\ y^3\ c\ y^3 \\ y^4\ d\ y^4 \end{pmatrix}, 2 = \begin{pmatrix} y^4\ d\ y^4 \\ y^3\ c\ y^3 \\ y^2\ b\ y^2 \\ y^1\ a\ y^1 \end{pmatrix}.$$

The horizontal symmetry σ_h is defined as $\sigma_h(1) = 2$, $\sigma_h(2) = 1$ and I is the identity mapping. As discussed under *Vikrānta*, the set G, thus, is a group isomorphic to $\mathbb{Z}_2$

3.5.2 Repetition across the *pāda*s

Now, let us consider the repetition across the feet. This repetition may involve 2 or 3 or all the 4 feet. Further, a complete *pāda* may be repeated as another *pāda*, or only a part of a foot is repeated in some other foot in a verse.

- Repetition of complete foot:

  (a) With a complete *pāda* repeated twice, there are C_2^4 (= 6) possible combinations are shown in the table 3.15.

$\begin{pmatrix} y \\ y \\ a \\ b \end{pmatrix}$	$\begin{pmatrix} y \\ a \\ y \\ b \end{pmatrix}$	$\begin{pmatrix} y \\ a \\ b \\ y \end{pmatrix}$	$\begin{pmatrix} a \\ y \\ y \\ b \end{pmatrix}$	$\begin{pmatrix} a \\ y \\ b \\ y \end{pmatrix}$	$\begin{pmatrix} a \\ b \\ y \\ y \end{pmatrix}$
(1)	(2)	(3)	(4)	(5)	(6)

Table 3.15: One foot repeated

Out of these, *Bharata* has considered (2) and (5), which correspond to *Vikrānta*. Patterns (1) and (6) are horizontal reflections of each other, and they together result in a symmetry group isomorphic to $\mathbb{Z}_2$. Similarly, the patterns (3) and (4), where one repeated pattern is at the beginning of one hemistich and the other one at the end of the other, are horizontal reflections of each other. These two together also result in a symmetry group isomorphic to $\mathbb{Z}_2$. The patterns (1), (3), (4) and (6) do not seem to be appealing cases of symmetry from a literature point of view.

$\begin{pmatrix} y \\ y \\ y \\ a \end{pmatrix}$	$\begin{pmatrix} y \\ y \\ a \\ y \end{pmatrix}$	$\begin{pmatrix} y \\ a \\ y \\ y \end{pmatrix}$	$\begin{pmatrix} a \\ y \\ y \\ y \end{pmatrix}$
(1)	(2)	(3)	(4)

Table 3.16: Three feet repeated

There are 4 ways in which a pattern can repeat in 3 feet as shown in the Table 3.16 . All are isomorphic to $\mathbb{Z}_2$.

Finally, there is only one combination in which the pattern gets repeated in all four *pāda*s, which is classified by *Bharata* under *Caturvyayasita*.

(b) If there are two different repetitions, we can have 3 patterns as shown in Table 3.17. Of these, the (2) corresponds to *Samudgaka*. Patterns (1) and (3) would result in a symmetry group isomorphic to $\mathbb{Z}_2$ and $\mathbb{Z}_1$, respectively. But they are not of much interest from the literature point of view.

- Repetition of partial foot

  The texts on *Alaṅkāraśāstra* provide examples of partial repetition across the feet, where

(1)	(2)	(3)
$\begin{pmatrix} y^1 \\ y^1 \\ y^2 \\ y^2 \end{pmatrix}$	$\begin{pmatrix} y^1 \\ y^2 \\ y^1 \\ y^2 \end{pmatrix}$	$\begin{pmatrix} y^1 \\ y^2 \\ y^2 \\ y^1 \end{pmatrix}$

Table 3.17: Two feet repeated

the repetition is found in all the feet. Hence, we discussed only them. However, the definition of the *Yamaka* does not imply that the repetition should be across all the feet. Thus, one may have partial repetition in 2 or 3 feet as well. However, since examples of such patterns are not found, we do not study them here.

3.6 Conclusion:

We observe that *Bharata* has considered only the patterns that are symmetric under the horizontal and vertical reflection and rotation through $180°$. These symmetry operations of reflection and rotation form either a Klein four group K_4 or $\mathbb{Z}_2$ group. *Caturvyavasita* is an exception, where the only possible operation is that of identity. Some patterns are not considered by *Bharata*, and these might have been ignored by him, perhaps because of the absence of any evidence of such patterns in the earlier literature. We also notice that from the literature point of view, such patterns are not very appealing.

Chapter 4

Poetry Classifier Based on *Kāvyaguṇas*

4.1 Introduction

NLP is developing vastly for different languages with data-driven approaches. However, low-resource languages (LRL) face numerous challenges in NLP, including data scarcity, complex morphology, orthographic and historical variations, and a lack of digital resources. These issues, combined with the inadequacy of models designed for other languages, hinder the development of robust NLP solutions for Sanskrit.

In this chapter, we describe a novel approach used to perform the poetry classification task. Poetry classification is an important task in FLP. We consider three distinct *Kāvyaguṇa*s viz. *Mādhurya, Oja and Prasāda* as three different classes. We build a model that classifies the input into these three classes according to its features. To achieve this task, we exercise the feature engineering-based vectorization method, which is created with the consideration of the insights from the *Alaṅkāraśāstra* tradition. It ensures that the model captures the distinct features for each class despite having a limited dataset. We also compare the accuracy scores with the `Word2vec` vectorization-based models.

4.2 Background

Sanskrit poetics is primarily divided into six different schools, viz. *rasa* (aesthetic flavor), *alaṅkāra* (figures of speech), *rīti* (style), *dhvani* (suggestion), *aucitya* (propriety), and *vakrokti* (obliqueness). Each of these is based on what they consider the most essential factor in poetry. Various elements contribute to the beauty of poetry, and these six schools emphasize the aspects deemed the most important in the enjoyment of poetry. Although *Kāvyaguṇa* or *guṇa* (quality) is not considered a separate school, its role is vital in the enjoyment of poetry, as it catalyzes the experience of *rasa*, the aesthetic flavor. *Guṇa* means “quality” and

refers to syntactic and phonemic features in the literary text that enhance its overall aesthetic experience.

There are several opinions of theoreticians on the number and nature of *Kāvyaguṇa*s. Mainly, there are three kinds of perspectives over the classification of the *guṇa*s found in the tradition.

1. **Classification with Ten *guṇa*s**

   Bharata introduced the concept of *Kāvyaguṇa* in *Naṭyaśāstra*.

   These ten *Kāvyaguṇa*s are ***Śleṣa*** (Union of word and intended meaning), ***Prasāda*** (Perspicuity), ***Samatā*** (Uniformity), ***Samādhi*** (Concentration), ***Mādhurya*** (Sweetness), ***Oja*** (Grandeur), ***Padasaukumārya*** (Agreeableness), ***Arthavyakti*** (Directness of expression), ***Udāratā*** (Exaltedness), and ***Kānti*** (Polish).

   Vāmana (8th century AD) standardised *rīti* concept. He bestowed the spotlight to *ritī* in his *Kāvyālaṅkāra-sūtravṛtti (KS)*. According to him *rīti* (style) is the vital element of the *Kāvya*. The *rītis* are the style of poets, shown through the choice of the syllables, which are the smallest part of a word and the length of compound words. According to *Vāmana* there are ten *Śabdaguṇa*s and ten *Arthaguṇa*s. These ten *guṇa*s of each type have the same name but different attributes according to the category. These are instrumental in identifying the *rīti*. The features of these *guṇa*s are similar to those of *Bharata*. Poets like *Daṇḍin*, *Bāṇbhaṭṭa* use complex combinations of syllables and long compounds. Three dominant *rītis* are *Gauḍī*, *Pāñcālī* and *Vaidarbhī*. *Gauḍī rīti* abounds in the qualities of *Oja* (floridity) and *Kānti* (polish).[1] *Pāñcālī* is endowed with the qualities of *Mādhurya* (sweetness) and *Saukumārya* (softness).[2] *Vaidarbhī* diction consists of all the ten *guṇa*s in proportion.[3]

2. **Classification with Three *guṇa*s**

   The classification consists of three *guṇa*s, viz. ***Mādhurya*** (Sweetness), ***Oja*** (Floridity), and ***Prasāda*** (Lucidity)—was introduced by *Bhāmaha*. In his *Kāvyālaṃkāra* (6th century AD), *Bhāmaha* explains these three qualities in terms of their syntactic characteristics, although he does not explicitly label them as *guṇas*. Instead, these qualities

[1] *Ojaḥ kāntimati gaudiyā* | 2.12, *KS*
[2] *Mādhurya Saukumāryopapannā pāñcālī* | 13, *KS*
[3] *samagraguṇopetā vaidarbhī* | 2.11, *KS*

represent the essential attributes that a poet must employ to express sweetness, floridity, and lucidity in their work.

Bhāmaha and others linked the concept of *guṇas* with *rasa* (aesthetic experience), Unlike theorists who regarded *rīti* (style) as the soul of poetry. According to this view, *guṇas* help enhance and prepare the reader's mind for the experience of *rasa*. This approach was later followed by eminent scholars like *Ānandavardhana* (9th century AD) and *Mammaṭa*.

Mammaṭa (12th century AD), while defining kāvya (poetry), emphasized the inclusion of *guṇas* as essential elements.[4].He not only advocated the threefold classification of *guṇas* but also provided a rationale for not considering the ten types of *guṇas* outlined by earlier scholars. According to *Mammaṭa*, *Śleṣa*, *Samādhi*, *Udāratā*, and *Prasāda*, as defined by *Vāmana*, are subsumed under *Oja*. *Mādhurya* remains as it is, while *Vāmana*'s *Arthavyakti* corresponds to *Mammaṭa*'s *Prasāda*. *Vāmana*'s *Samatā* refers to uniformity in writing, which can sometimes be a flaw, as a poet needs to adapt their style to suit the mood of the events described in the kāvya. *Saukumārya* (Softness) is understood as the absence of harshness, and *Kānti* (Polish) is regarded as the absence of vulgarity.

3. **Other Classifications**

Kuntaka and *Bhoja* contributed with other prominent classification schemes. *Kuntaka* (10th century AD) identified *guṇas* as indicators of the three *mārgas* (paths of style): ***Saukumārya*** (softness), ***Prasāda*** (lucidity), ***Lāvaṇya*** (charm), and ***Abhijātya*** (nobility or delight). While the names of these *guṇas* remain the same, their characteristics vary across each *mārga*. Additionally, *Kuntaka* emphasized the importance of *Aucitya* (propriety) and *Saubhāgya* (grace or delight), which are essential *guṇas* without which poetry would lose its appeal.

Bhoja, who was a renowned critic in the 12th century AD, expanded on the concept of *guṇas* in his treatise Sarasvatīkaṇṭhābharaṇa. In this work, he detailed 24 śabdaguṇas (qualities of sound) and 24 arthaguṇas (qualities of meaning), offering an extensive critique of each aspect of poetic expression.

[4] *tadadoṣau śabdārthau*
saguṇau *analaṅkṛtī punaḥ kvāpi* | 1.1, *Kāvyaprakāśa*

	Mādhurya	**Oja**
Consonants	all except *t, ṭ, d, ḍ, ś, ṣ*	*t, ṭ, d, ḍ, ś, ṣ*
Conjuncts	nasal C + C *r*/ṇ + short V	unaspirated C + aspirated C *r*/ṇ + long V *r* + C / C + *r* C + C
Compounds	No or of smaller length	lengthy
Construction	moderate complex	extreme complex

Table 4.1: Phonetic and Syntactic Features of *guṇa*s
(Note: C = consonant, V = vowel)

Given the clarity of the three *guṇa* classification, we adopted this scheme for our poetry classification model. These three *guṇas* viz. *Mādhurya* (sweetness), *Oja* (floridity), and *Prasāda* (lucidity) effectively encapsulate the essence of other classification systems, distilling multiple qualities into a simpler framework. *Mādhurya* is characterized by soft and sweet constructions with minimal compounds, while *Oja* involves harsher, more complex constructions with longer compounds. *Prasāda*, the neutral *guṇa*, is present in all types of constructions and across all *rasas*.

In the ten *guṇa* classification, many qualities are defined by the absence of faults, making it difficult to assign precise vector values for supervised machine learning tasks. In contrast, the three *guṇa* scheme differentiates the types, reducing the likelihood of class confusion. The distinct features of *Mādhurya*, *Oja*, and *Prasāda* provide a solid foundation for classification, ensuring better clarity and separation in the output of the model. The detailed characteristics of these three *guṇas* are as follows.

1. **Mādhurya (Sweetness):** It invokes a sense of mollification and melting of the reader's mind. Commonly associated with *Śṛṅgāra* (erotic), *Karuṇa* (compassionate), and *Śānta* (peaceful) *rasas*. It is characterized by a pleasing construction using soft consonants and short or no compound words.

2. **Oja (Floridity):** It provides a sense of lustrous expansion and is effective in depicting *Vīra* (heroic), *Bībhatsa* (disgusting), and *Raudra* (furious) *rasas*. It is identified by the use of harsh consonant combinations and lengthy compounds.

3. **Prasāda (Lucidity):** This quality ensures that the meaning of the text is easily comprehensible upon hearing. It is present across all kinds of *rasas* and constructions.[5] There is no consonant combination and length of compound words specified to identify *Prasāda guṇa*. However, the literary text where there is the absence of *Mādhurya* and *Oja* can be considered under this category.
   (The syntactic and phonetic features for *Mādhurya* and *Oja* are shown in the table 4.1)

4.3 Implementation and Algorithm

In the previous section, we discussed the characteristics of *guṇas* as defined in the treatises of Sanskrit poetics. For individuals, it can be challenging to focus on and recall these nuanced features, as a human mind often processes these poetic effects collectively, intuitively, to classify poetry into the respective *guṇa*s. Despite clearly defined rules and boundaries, rule-based systems struggle to handle the inherent uncertainty and complexity of Sanskrit poetry. Moreover, there is currently no model capable of effectively processing such poetry without supervision. Therefore, we chose to pursue a balanced approach using supervised machine learning with feature engineering.

4.3.1 Dataset

The learning dataset consists of 672 Sanskrit verses, equally distributed among the three *guṇas*. These verses are primarily sourced from classical Sanskrit literature, including works such as *Kumārasambhava*, *Buddhacarita*, *Ṛtusaṃhāra*, *Amaruśataka*, *Manusmṛti, Śiśupālavadha, Daśakumāracarita, Rāmāyaṇa* and *Mahābhārata*. To introduce more variety, selected hymns (*stotra*s) like *Śivatāṇḍava-stotra, Mahiṣāsura-mardinī-stotra*, etc. have also been added to the dataset.

For classification purposes, we consider each verse as a unit of input. While the diction of a poet often remains consistent throughout a poem, the choice of a specific *guṇa* is influenced by the *rasa* (emotional essence) of the event being described. Consequently, 224 verses have been assigned to each of the three *guṇas* in the dataset, providing a balanced distribution for training the model.

[5] *śrutimātreṇa śabdāttu yenārthapratyayo bhavet |*
sādhāraṇaḥ samagrāṇāṃ sa prasādo guṇo mataḥ || 8.76, *Kāvyaprakāśa*

These verses undergo simple steps of data normalization. We remove punctuation marks and *avagraha* ('ऽ'), a special symbol used to indicate the elided *akāra* or *ākāra*. Once the data is cleaned, it is ready for the feature extraction process. This ensures that the dataset is prepared for machine learning algorithms to process it efficiently and produce accurate classifications.

4.3.2 Vectorization

Machines do not inherently understand words or alphabets, they only process numbers. Therefore, it is essential to convert text into numerical values based on specific criteria. This numerical representation of text features is called a vector. Several automatic models are available for converting text into vectors, such as `Word2vec`. `Word2vec` is a neural network-based model that generates vector representations of words, capturing their semantic meanings and relationships by training on large datasets. It produces word embeddings, where similar words are mapped to nearby points in the vector space, enabling efficient similarity calculations and other linguistic tasks.

However, in the case of Sanskrit, tokenization for vectorization becomes particularly challenging due to the phenomena of *sandhi* (phonetic joining of words) and *samāsa* (compound words). Additionally, the presence of polysemous words (words with multiple meanings) complicates the vectorization process while using models trained on for other languages. To address these challenges, manual feature extraction based on the clues provided by Sanskrit poetics is a more suitable approach.

Here, for comparison purposes, we apply both the techniques of vectorization, viz. `Word2vec` and feature engineering.

Custom Feature Engineering

The clues to identify *guṇa*s are already provided in the poetic tradition. We define eight different features. These features are as follows.

1. **Number of syllables:** This feature is a count of a total number of syllables in a verse.

2. **Number of words:** This feature represents the proportion of the number of words in a verse to the total number of letters. The number of words and letters may vary depending on the meter. For instance, the *Anuṣṭup* meter has 32 *akṣaras*, while the

Śārdūlavikrīḍita meter has 48 *akṣaras*. To maintain uniformity across meters, we calculate the proportion of words to letters.

3. **Number of lengthy compound words:** This feature measures the number of long compound words. A compound is deemed to be long if it has more than 9 syllables. The number is arrived from the heuristics of the data used for training. The count of such words is divided by the total number of words in the verse. Generally, verses with more compounds or *sandhi* have fewer independent words. *Oja guṇa* verses tend to have fewer words than *Mādhurya* verses, with *Prasāda* constructions having even more independent words.

4. **Number of *Oja* syllables and conjuncts:** In *Kāvyaprakāśa*, *Mammaṭa* defines typical syllables and conjunct combinations of *Oja*[6] and *Mādhurya*. For *Oja*, we count:

   - All retroflex consonants except nasals (*ṭ, ṭh, ḍ, ḍh, ś, ṣ*).
   - *r* and *ṇ* followed by long vowels.
   - Unaspirated consonants paired with aspirated consonants from the same articulation group (e.g., *k+kh*, *g+gh*, *c+ch*, *j+jh*).
   - Any consonant combined with itself (e.g., *t+t*, *th+th*, *v+v*).

   These combinations are counted across the verse and divided by the total number of letters. The percentage is taken as the feature value.

5. **Number of conjunct consonants for *Mādhurya*:** For *Mādhurya*, *Mammaṭa* provides the following guidelines[7]:

   - The combinations used for *Oja* should be avoided.
   - Count conjunct consonants where any nasal is combined with a consonant of the same articulation point (e.g., *ṇ+k, ñ+c, ṇ+ṭ, n+t, m+p*). This count is divided by the total number of letters, and the percentage is considered the feature value.

6. **Number of long vowels:** This feature represents the percentage of long vowels (*ā, ī, ū, e, ai, o, au*) out of the total number of vowels in the verse.

[6] *yoga ādyatṛtīyāmyāmantyayo reṇa tulyayoḥ |*
ṭādiḥ śaṣau vṛttidairghyaṃ gumpha uddhṛta ojasi || 8.75, *Kāvyaprakāśa*

[7] *murdhni vargāntyagāḥ sparśā aṭavargā raṇau laghū |*
āvṛttirmadhyavṛttirvā mādhurye ghaṭanā tathā || 8.74, *Kāvyaprakāśa*

7. **Number of *r* and *ṇ* followed by short vowels:** This feature is the frequency of *r* and *ṇ* followed by short vowels, divided by the total number of letters in a verse.

8. **Number of unaspirated consonants:** This feature captures the percentage of unaspirated consonants (*k, g, c, j, ṭ, ḍ, p, b*) that occur in *Mādhurya* constructions, divided by the total number of letters in the verse.

These features are calculated for each verse of the dataset as an eight-dimensional vector.

Word2vec

To compare the performance of the mentioned feature engineering assisted model with a neural network-based word embedding model, we employed the `Word2vec` technique, which generates vectors for each word in the text based on its semantic meaning and relationships with neighboring words.

The Word2Vec was trained from scratch on ~700 verses. We use syllables instead of words, because the input text is not segmented, so the word boundaries are not clear. Further, for the Guṇa classification, what matters is the syllable patterns, not necessarily the words. We set the vector size to 500 and, since our classification is at the verse level, we calculated a sentence vector that is the average of the vectors for all *akṣaras* within each verse to represent the entire verse. The total number of syllables in the 700 verses is 34,347, and the unique number of syllables is 1,939. The size of the syllables varies from 1 letter to 6 letters with the following distribution:

Number of Letters	Frequency
1	718
2	23,372
3	8,415
4	1,121
5	52
6	2

Table 4.2: Distribution of syllable lengths.

The distribution is very skewed, indicating the impossibility of certain combinations of the letters. This data provides a lot of insights regarding the 'semantics' associated with the letters such as their properties - soft/hard, place of articulation, etc.

The Word2Vec captures semantic similarities, but the task does not rely on semantic similarities but syntactic patterns, let us see what kind of semantics is captured by Word2Vec with syllables, and how it is useful for the classification task at hand. The syllables tell you which letters (varṇas) can go with which other letters. For example the combinations such as 'kc' or 'kp' are not found in the data, while 'kv', 'ky', 'kl' are found in the data. Thus, the machine indirectly learns the place and effort (sthāna and prayatna) corresponding to each letter. Also from the sequence of these syllables, it learns which syllables can follow which others. This way, it indirectly learns the soft (mṛdu) and hard (kaṭhora) consonants, and so on. The semantics needed for the identification of the *guṇa* is the semantics associated with the consonants. And thus, it indirecly learns the features used in feature engineering. The space separates the words, and thus 'space' acts as a pointer to know whether the words are long or short. This word2vec captures the number of syllables as well, indirectly.

4.3.3 Learning Models

The extracted vectors are stored for each verse and are used to train the machine learning models. For training purposes, we split the dataset into training and test sets. Here, we have split the dataset into 80% training and 20% testing to ensure that the model generalizes well.

We consider multiple models that are developed to train the machine, namely,

- **Random Forest (RF)**
- **Gradient Boosting (GB)**
- **Support Vector Machine (SVM)**
- **Multi-layer Perceptron (MLP)**
- **Extra Trees (ET)**
- **Extreme Gradient Boosting (XGB)**

We optimise the performance of MLP by setting `max-iter = 2000` and in SVM we set the `gamma = `auto'`.

4.4 Performance Evaluation

4.4.1 Validation of Word2vec

The six models were then trained using these automatically generated vectors. The precision and standard deviation of these models evaluated using K-fold cross-validation are detailed in the table 4.3. We observed was obtained in the Extreme Gradient Boosting (XGB) at 59%, but has the standard deviation of 0.11. The models XGB and GB having the accuracy of 0.58 performs better with a lower standard deviation of 0.06 and 0.07 respectively.

Model	Accuracy	Standard Deviation
RF	0.56	0.06
GB	0.58	0.07
SVM	0.33	0.02
MLP	**0.59**	**0.11**
ET	0.56	0.06
XGB	0.58	0.06

Table 4.3: Performance Metrics in K-fold Validation of Different Models using `Word2vec`

4.4.2 Validation of Feature Engineering-Based Models

To thoroughly test models trained on the custom features given by the poetic theorists, we considered two types of validation techniques.

10-fold Cross-validation

Model	Accuracy	Standard Deviation
RF	0.885106	0.031844
GB	0.870213	0.032197
SVM	0.906383	0.043811
MLP	**0.893617**	**0.026913**
ET	0.893617	0.028546
XGB	0.893617	0.026913

Table 4.4: Accuracy and Standard deviation scores of different models according to 10-fold cross-validation

The table 4.4 shows the accuracy and standard deviation of various machine learning models. The Support Vector Machine model achieved the highest accuracy at 90.6%, though with

a slightly higher standard deviation (0.0438). Models like Gradient Boosting, Multi-Layer Perceptron, Extra Trees, Random Forest and XGBoosting all performed well, with accuracies around 87-89%, and comparable stability. The standard deviations for all models are relatively low, suggesting consistent performance across different runs, although the SVM model displayed slightly greater variability than the others. Overall, SVM stands out in accuracy, while other models offer a balanced mix of performance and consistency.

Holdout Validation

The other technique we employ to evaluate the model is the holdout validation technique, in which the dataset is split into two subsets. One for training the model and the other for testing its performance. This ensures that the model is evaluated on unseen data, providing a more reliable and generalized model when run on unseen data. There are various scores to compare the performances (see table 4.5). These scores cover various conditions for the reliability of the models. In addition, a confusion matrix is also generated to analyze the learning of the model for each of the classes.

Model	Accuracy
RF	0.88
GB	0.87
SVM	0.90
MLP	0.87
ET	**0.92**
XGB	0.91

Table 4.5: Performance Comparison of Machine Learning Models According to Holdout Validation

In the context of multi-class classification, a confusion matrix provides a detailed breakdown of the model's performance by showing how predictions for each class correspond to the true labels. It gives insight into how often the model confuses one class with another, highlighting strengths and weaknesses in classification accuracy across different classes. We reserved 20% of the data in the test set. 135 verses in the test set were randomly distributed among these classes as, 45 for *Oja*, 47 for *Mādhurya* and 43 for *Prasāda*.

In the matrices given for each model in tables 4.6, 4.7, 4.8, 4.9, 4.10 and 4.11 the rows indicate gold values for *Mādhurya* (M), *Prasāda* (P) and *Oja* (O). The columns M, P and O show the predicted classes.

	M	P	O
M	37	4	4
P	3	44	0
O	5	0	38

Table 4.6: Random Forest

	M	P	O
M	36	5	4
P	4	43	0
O	4	0	39

Table 4.7: Gradient Boosting

	M	P	O
M	38	4	3
P	3	44	0
O	4	0	39

Table 4.8: Support Vector Machine

	M	P	O
M	37	6	2
P	4	43	0
O	6	0	37

Table 4.9: Multi-layer Perceptron

	M	P	O
M	39	4	2
P	2	45	0
O	3	0	40

Table 4.10: Extra Trees

	M	P	O
M	38	4	3
P	2	45	0
O	3	0	40

Table 4.11: XGBoost

The confusion matrices indicate that all the models were able to clearly distinguish between the classes *Oja* (O) and *Prasāda* (P). Notably, there are no instances of O being falsely predicted as P or P being falsely predicted as O, demonstrating the models' strong ability to differentiate between these two *guṇas*. However, the models encountered difficulty in distinguishing *Mādhurya* (M) from both O and P. Specifically, the models misclassified 5–9 examples between M and P, while a similar range of misclassifications (5–9 instances) was observed be-

tween M and O. This fact highlights the greater challenge in accurately distinguishing between M and other two classes.

4.5 Error Analysis

As we have seen in table 4.1, the characteristics of *Mādhurya* and *Oja* are contradictory to each other. Focusing on it we described various features. With this information, one would expect the models to perform better for *Oja* and *Mādhurya* in the first place. Contrary to our expectation, we find that some *Mādhurya* examples are incorrectly classified as *Oja* and *Prasāda* and vice versa. In order to analyze the failure cases, we take the average of the values for each feature in different classes, and compare the values of the incorrectly analysed cases. The verse with the feature values (Table 4.13) for the following four cases are as follows.

Feature	**M**	**P**	**O**
Syllables (**v**)	126.76	80.98	137.66
Words (**w**)	10.90	13.13	6.27
Lengthy Compounds (**lc**)	4.72	0.98	28.58
Oja Syllables (**os**)	18.05	14.64	23.71
Long Vowels (**lv**)	17.46	20.90	17.19
Unaspirated Consonants (**uc**)	34.51	33.90	33.80
Mādhurya Syllables (**ms**)	8.73	7.79	6.66
r and *n* with short vowels (**rn**)	5.62	4.67	6.99

Table 4.12: Average value for each variable for categories M, P and O

1. ***Mādhurya* predicted as *Oja*:** In Table 4.13, the first example shows a verse, presented below, from the *Mādhurya* class incorrectly classified as *Oja*. The values of key variables such as `v`, `w`, `lc`, `os`, and `rn` closely align with the average values for the *Oja* class, leading to the misclassification.

Sr. No.	v	w	lc	os	lv	uc	ms	rn	True class	Predicted
1	183.0	6.5574	16.6667	34.0000	14.8649	34.4262	12.0219	6.0109	M	O
2	121.0	9.9174	0.0000	18.0000	17.6471	38.8430	4.9587	4.1322	O	M
3	99.0	13.1313	0.0000	13.0303	22.7273	37.3737	5.0505	1.0101	M	P
4	112.0	10.7143	0.0000	15.0000	14.8936	26.7857	10.7143	9.8214	P	M

Table 4.13: Misclassified examples with their feature values and predicted labels.

dharasyoddhartāsi tvamiti nanu sarvatra jagati
pratītastatkiṃ māmatibharamadhaḥ prāpipāyiṣuḥ |
upālabdhevoccairgiripatiriti śrīpatimasau
balākrāntaḥ kriḍaddviradamathitorvīruharavaiḥ||

2. ***Oja* predicted as *Mādhurya*:**
   In Table 4.13, the second example demonstrates a verse, shown below, from the *Oja* class being incorrectly classified as *Mādhurya*. The variables `v`, `w`, `lc`, `os`, and `rn` exhibit values that match the average characteristics of the *Mādhurya* class, which likely contributed to the error.

   śrīmadbhirjitapulināni mādhavīnāmārohairnibiḍabṛhannitambabimbaiḥ|
   pāṣāṇaskhalanavilolamāśu nūnaṃ vailakṣyādyayuravarodhanāni sindhoḥ ||

3. ***Mādhurya* predicted as *Prasāda*:**
   The third instance in Table 4.13 represents a verse, given below, belonging to the *Mādhurya* class that was misclassified as *Prasāda*. Here, the values of the variables `v`, `w`, and `lv` align more closely with the average values for *Prasāda*, `ms` value does not fit with the average value for *Mādhurya* resulting in the misclassification.

   tadīyamālokya sugāḍha bhakte mahāntamāveśamuvāca lokaḥ |
   baddhādarosau yadi pāṇḍuraṅge tadantike tiṣṭhati kiṃ na nityam ||

4. ***Prasāda* predicted as *Mādhurya*:**
   The fourth instance in Table 4.13 shows a verse, presented below, from the *Prasāda* class being classified as *Mādhurya*. This misclassification can be attributed to the variables `v`, `w`, and `lv`, whose values align closely with the average values of the *Mādhurya* class.

   acyutaṃ keśavaṃ rāmanārāyaṇaṃ kṛṣṇadāmodaraṃ vāsudevaṃ harim |
   śrīdharaṃ mādhavaṃ gopikāvallabham jānakīnāyakaṃ rāmacandraṃ bhaje||

Its complexity characterizes Oja, while *Prasāda* is associated with simplicity. In contrast, *Mādhurya*, with its soft combinations of consonants and a moderate level of complexity, occupies a midpoint between these two *guṇas*. Even for a human annotator, it is tricky to classify the ślokas, especially those identified as failed cases, into a definitive *guṇa*. This observation provides a plausible explanation for the misclassifications, as texts exhibiting *Mādhurya* may sometimes share features with *Oja* and *Prasāda*.

This discrepancy opens avenues for further exploration. Future research could address this issue by employing multilabel classification techniques, allowing texts to be assigned to one or more *guṇas* based on their dominant features.

In Table 4.12, features such as `v`, `w`, `lc`, and `os` have shown a significant impact on the classification task. However, features like `lv`, `uc`, `ms`, and `rn` exhibit less pronounced differences in their average values across *guṇas*. Assigning weights to these features based on their relevance to specific classes could improve classification results. For instance, greater weight could be assigned to `lv`, `uc`, `ms`, and `rn` when they appear in *Mādhurya*.

Such approaches could offer deeper insights into the overlapping characteristics of these poetic styles and enhance the accuracy of classification tasks.

4.6 Comparison of Feature Engineering with Word2vec

`Word2vec` is a widely used unsupervised learning technique that captures semantic relationships between words by mapping them into continuous vector spaces. However, in our task of classifying Sanskrit poetry into *guṇas*, we observed certain limitations when relying solely on `Word2vec` embeddings.

The low score with the `Word2vec` model can be attributed to the limited size of the training data. Models like `Word2vec` typically require large datasets, often in the thousands or even millions of examples, to perform effectively. Additionally, the vector sizes used in such models usually range from hundreds to thousands, demanding powerful computational resources. Training on smaller datasets with limited resources poses significant challenges. Despite these constraints, our approach of training the model on minimal data with available resources has produced commendable results, highlighting the effectiveness of the feature engineering technique.

The custom feature engineering method is specifically tailored to the nuances of Sanskrit poetics. By incorporating features which indirectly reflect morphological structure, consideration of metrical patterns, and stylistic markers, our model was able to better capture the inherent characteristics that define different *guṇas*. This domain-specific approach resulted in higher accuracy compared to the more general-purpose `Word2vec` embeddings.

4.7 Conclusion

The computational processing of poetic language presents significant challenges. To address these complexities, we introduced the module 'Kāvyaguṇa Classifier', designed to classify literary texts into three distinct *guṇas*: *Mādhurya*, *Oja*, and *Prasāda*. For this task, we explain two different approaches used for vectorization. `Word2vec` is useful for capturing semantic information; our custom feature engineering leveraged expert knowledge of the literary domain, allowing a more precise classification in this context. This demonstrates that, in specialized tasks like the classification of Sanskrit poetry, combining traditional machine learning techniques with domain-specific insights can offer significant advantages over purely data-driven methods.

This module not only serves its primary purpose, but can also be adapted for related tasks. As discussed previously, these *guṇa*s are essential components of both *rītis* and *rasas*. The *rītis* reflect the stylistic inheritance of poets, while the rasas encapsulate the emotions arising from aesthetic experiences. The interplay of *guṇa*s in these contexts is crucial, making the *Kāvyaguṇa* classifier a valuable tool for identifying both *rītis* and *rasas*.

For the further research, we are publishing the training dataset of 672 verses under the Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0) license. The data can be found at `https://drive.google.com/drive/folders/1LaTyBd_YtrKJDaLVgVmHUA\protect\discretionary{\char\hyphenchar\font}{}{}fFDPbXQns0?usp=sharing`.

Moreover, given the linguistic similarities between Sanskrit and other Indian languages, their grammatical structures and poetic traditions are deeply influenced by Sanskrit. Concepts analogous to *Kāvyaguṇa* can be found in these languages as well. By employing language-specific training data and adapting feature extraction methodologies accordingly, similar classification tasks can be effectively achieved across various Indian languages.

Chapter 5

Sentential Parsing of Upamā

5.1 Introduction

In chapter 2 we have described the *Yamaka* and *Anuprāsa* which are *śabdālaṅkāra*s. Another important category of *alaṅkāra* is *arthālaṅkāra*, or the ornamentation of meaning. *Arthālaṅkāras* represent various ways in which meaning can be expressed with artistic elegance. Their use elevates a simple sentence into something more impactful and refined. These include comparisons as in *upamā, utprekṣā, rūpaka, ananvaya*, cause-and-effect relations as in *vibhāvanā, viśeśokti*, inversions as in *nidarśanā*, and other conceptual structures. Literary language, or *kāvya*, is not just used to convey information, it expresses ideas in a rich and multi-layered way, full of depth and subtlety. The careful selection of words, their arrangement, and the implied meanings distinguish poetic language from ordinary speech.

Contemporary natural language processing tools, however, are primarily trained to focus on grammatical aspects such as syntax and morphology. When applied to poetic texts, especially those rich in *alaṅkāras*, often fail to capture the semantic depth and aesthetic dimensions inherent in such language. They can offer structural analysis but fall short of interpreting the poetic significance.

Among the various *arthālaṅkāras*, *Upamā*, holds a central place. It expresses similarity between two entities and typically involves four components:

1. ***Upameya***: The subject of comparison.
2. ***Upamāna***: The object of comparison.
3. ***Sādhāraṇa-dharma***: The shared quality in the form of an Event or a property or a State.

4. ***Upamā-dyotaka***: Comparative marker or comparator, such as '*iva*', 'yathā', 'tulya', etc.

A sentence containing *Upamā* conveys additional information by highlighting the similarity between the *upameya* and the *upamāna*, marked by a comparative expression and grounded in a shared property. Such sentences carry greater informational and aesthetic depth than ordinary statements. When a sentential parser lacks the capacity to process these figurative elements and their interrelations, essential semantic nuances may be lost. Owing to its structural clarity and frequent occurrence in Sanskrit poetry and poetics, *Upamā* stands out as both theoretically significant and practically approachable.

Some attempts have been made towards processing of *Upamā*. Jadhav et al. (2023) provided graphs for human cognition of *upamā* sentences and proposed a representation schema for machines to process the limited range of 6 subtypes of *pūrṇopamā* category. Jadhav et al. (2025) explored the potential of Large Language Model in identification and classification of *Upamā*. Kulkarni and Kulkarni (2025) has proceeded towards the refined manual modelling with a limited instances of *Upamā* occurring in the second canto of *Raghuvaṃśa*, primarily consisting of *Samāsagā-upamā* denoted by *iva*. However, no comprehensive guidelines covering all 25 subtypes of *Upamā* have been identified. Their systematic implementation for parsing sentences containing *Upamā Alaṅkāra* also remains unexplored.

While Anusārakam, a sentential analyser based on Pāṇinian *kāraka* and *śabdabodha* theories, primarily focuses on syntactic and semantic analysis based on grammatical roles, it does not account for the semantic ornamentation introduced by figures of speech (*alaṅkāras*). To address this, our aim is to extend the parser with a specialised module designed to identify and analyse *Upamā alaṅkāra* in Sanskrit sentences. In order to extend it, we first propose the tagging guidelines for the the 25 types of *Upamā* constructions.

In this chapter, first we describe the subtypes of *Upamā* given by *Mammaṭa* in his treatise *Kavyaprakāśa*. This is followed by the grammatical classification and tagging scheme for annotating *Upamā*. We also, deal with the examples and special cases found in literature along with other *alaṅkāras* having similar constructions.

5.2 Upamā Alaṅkāra

Arthālaṅkāras are commonly classified into various types based on similarity, dissimilarity, cause and effect, association, and so on. Among these, the category grounded in similarity stands out both in terms of conceptual prominence and the number of individual *alaṅkāras* it contains. *Upamā* is the foremost figure in this group, occupying a foundational place in Sanskrit poetics. Instances of *Upamā* appear even in the Vedic hymns, indicating its deep roots in the earliest Sanskrit literature. Several literary theorists have regarded *Upamā* as the source or model for other *arthālaṅkāras.*[1]

Bharata initiated the theoretical treatment of *alaṅkāras* in his *Nāṭyaśāstra*, identifying *Upamā* as one of just four figures of speech. The essence of *Upamā* proposed by *Bharata* was adopted by later scholars with only minor changes. However, a variety of classifications emerged among theoreticians, shaped by their differing interpretive frameworks.

Bharata and *Daṇḍin* focused on the nature of the subject matter when classifying *Upamā*. ***Bharata*** mentions five types of *Upamā*[2]: *Praśaṃsā* (praise), *Nindā* (censure), *Kalpitā* (conceit), *Sadṛśī* (likeness), and *Kiñcitsadṛśī* (partial resemblance). *Daṇḍin*, in contrast, expands the scope significantly by identifying 32 distinct types of comparisons classified under *Upamā*[3].

Vāmana, recognises two types—*Kalpitā* and *Laukikī*—depending on whether the comparison is imaginative or grounded in common worldly perception. *Vāmana*, approaching from a different angle, contributes to a more detailed classification based on syntactic structure. As discussed earlier, an *Upamā* typically consists of four elements: the subject of comparison (*Upameya*), the object of comparison (*Upamāna*), a shared quality (*Sādhāraṇa-dharma*), and a comparative marker (*Upamā-dyotaka*), often a word like 'iva', 'yathā', or 'tulya', etc. When

[1] upamāprapañcaśca sarvo'laṅkāra iti vidvadbhiḥ pratipannameva | 17th Chapter, *Abhinavabhāratī*

[2] praśaṃsā caiva nindā ca kalpitā sadṛśī tathā |
yā kiñcitsadṛśī jñeyā sopamā pañcadhā budhaiḥ|| 17.47, *Nāṭyaśāstra*

[3] *dharmopamā, vastūpamā, viparyāsopamā, anyonyopamā, niyamopamā, aniyamopamā, samuccayopamā, atiśayopamā, utprekṣitopamā, adbhutopamā, mohopamā, saṃśayopamā, nirṇayopamā, śleṣopamā, samānopamā, nindopamā, praśaṃsopamā, ācikhyāsopamā, virodhopamā, pratiṣedhopamā, caṭūpamā, tattvākhyānopamā, asādhāraṇopamā, abhūtopamā, asambhāvitopamā, bahūpamā, vikriyopamā, mālopamā, vākyārthopamā, prativastūpamā, tulyayogopamā and hetūpamā | Kāvyaādarśa*, 2.15 - 2.50

all the four components are explicitly present, *Vāmana* terms it a *Pūrṇopamā* (complete simile).[4] If one or more of these elements are omitted, the simile is called *Luptopamā* (elliptical).[5] Additionally, *Vāmana* draws a distinction between similes expressed through a single word and those constructed with more than one word or a paraphrase.

Udbhaṭa, in his seminal contributions, builds upon *Vāmana*'s syntactic model by introducing grammatical categories into the classification. He includes forms such as compound words (*Samāsa*) and derivatives with *Taddhita* suffixes that inherently convey comparison. This attention to grammatical structure sets a precedent followed by later scholars such as *Mammaṭa* and *Viśvanātha*. A classification based on grammatical cues offers more precision and can be effectively translated into computational analysis. In contrast, the earlier subject-based classifications, while insightful, pose challenges for machine interpretation and pattern recognition.

We follow the classification provided by *Mammaṭa* in *Kāvyaprakāśa*, which is based on *Udbhaṭa* with precise definitions and examples. In the following section, we explain different types of *Upamā* with an example and suggest a tagging scheme representing the syntax faithfully. *Upamā* being a figure of spēch, deals more with semantics rather than syntax. The semantic clues guide us to its semantics. Therefore, we also suggest a semantic representation of the same sentence.

5.3 Types of *Upamā*

In *Kāvyaprakāśa*, the classification of *Upamā* rests primarily on syntactic considerations. The distinctions are drawn according to how the four essential components of *Upamā*—the subject of comparison (*upameya*), the object of comparison (*upamāna*), the shared property (*sādhāraṇa-dharma*), and the comparative marker (*upamā-vācaka*)—are expressed in a construction. Broadly, two principal types are recognised: *Pūrṇopamā* (complete) and *Luptopamā* (elliptical).

[4] guṇadyotakopamānopameyaśabdānāṃ sāmagnye pūrṇā| 2.5, *Kāvyālaṅkārasūtravṛtti*
[5] lope luptā| 4.6, *Kāvyālaṅkārasūtravṛtti*

In *Pūrṇopamā*, all four components are explicitly present, while in *Luptopamā* one or more of these are absent. Another basis of classification is the nature of similarity indicated by the comparative marker: *Śrautī*, where the resemblance is grasped immediately upon hearing the word, and *Ārthī*, where the listener needs to process the meaning to discern the implied similarity. A further division concerns the syntactic setting of the marker: *Vākyagā* (appearing as stand-alone words), *Samāsagā* (occurring components of compound), and *Pratyayagā* (expressed by affixes denoting similarity).

Thus, *Pūrṇopamā* comprises of six subtypes arising from the intersection of the two semantic kinds (*Śrautī* and *Ārthī*) with the three syntactic settings (*Vākyagā*, *Samāsagā*, *Pratyayagā*).

Table 5.1: Classification of *Pūrṇopamā*

Meaning indicated	**Syntactic environment**
Śrautī (implicit)	Vākyagā Samāsagā Pratyayagā
Ārthī (explicit)	Vākyagā Samāsagā Pratyayagā

In contrast, *Luptopamā* is divided by the number of omitted elements: *Ekāluptā* (one element omitted), *Dviluptā* (two omitted), and *Triluptā* (three omitted), producing a total of nineteen recognised varieties. *Ekāluptā* comprises omissions of either the shared property (*Dharmaluptā*), the object of comparison (*Upamānaluptā*), or the comparative marker (*Vācakaluptā*). *Dviluptā* involves the loss of any two components, yielding *Dharma-vācaka-luptā*, *Dharma-upamāna-luptā*, and *Vācaka-upameya-luptā*. Finally, *Triluptā* denotes the omission of three—*sādhāraṇa-dharma*, *upamāna*, and *upamā-vācaka*, simultaneously. Refer to Table 5.2 for details.

In the 10th *ullāsa* of *Kāvyaprakāśa*, examples are provided for each of the *Upamā* types. This classification preserves the subtle distinctions of classical Sanskrit poetics while offering a clear and systematic framework, particularly useful for the computational analysis of *Upamā*.

Table 5.2: Classification of *Luptopamā*

Category	**Subtypes and further divisions**
Ekāluptā	• *Dharmaluptā* (5): *Śrautī Vākyagā*, *Śrautī Samāsagā*, *Ārthī Vākyagā*, *Ārthī Samāsagā*, *Ārthī Pratyayagā* • *Upamānaluptā* (2): *Ārthī Vākyagā*, *Ārthī Samāsagā* • *Vācakaluptā* (6): *Samāsagā*; *Pratyayagā—kyac* (karma), *kyac* (ādhāra), *kyaṅ* (kartā), ṇamul (karma), ṇamul (kartā)
Dviluptā	• *Dharma-vācaka-luptā* (2): *Samāsagā*, *Pratyayagā—kvip affix* • *Dharma-upamāna-luptā* (2): *Vākyagā*, *Samāsagā* • *Vācaka-upameya-luptā* (1): *Pratyayagā—kyac affix*
Triluptā	*Dharma-upamāna-vācaka-luptā* (1): *Samāsagā*

5.4 Tagging guidelines for *Upamā*

Prior to the development of a parser - whether it is developed as an expert system following the grammar rules or as ML (Machine Learning) system using annotated data, we need guidelines for the annotation.

In this section, we present the tag scheme and the guidelines devised for the annotation of the four essential elements of *Upamā* and describe the annotation of different constructions enumerated in Section 5.3, including subtle variations found in literary texts.

5.4.1 About the Notation and Convention

Here, we explain the tagging scheme with the following example (see figure 5.1):

Skt: *sundaram mukham paśya,*

yathā kamalam.

Gloss: beautiful{sg,nom,neut} face{sg,nom,neut} see{imper,2p,sg}
like lotus{sg,nom,neut}
Eng: See the beautiful face,
like a lotus.

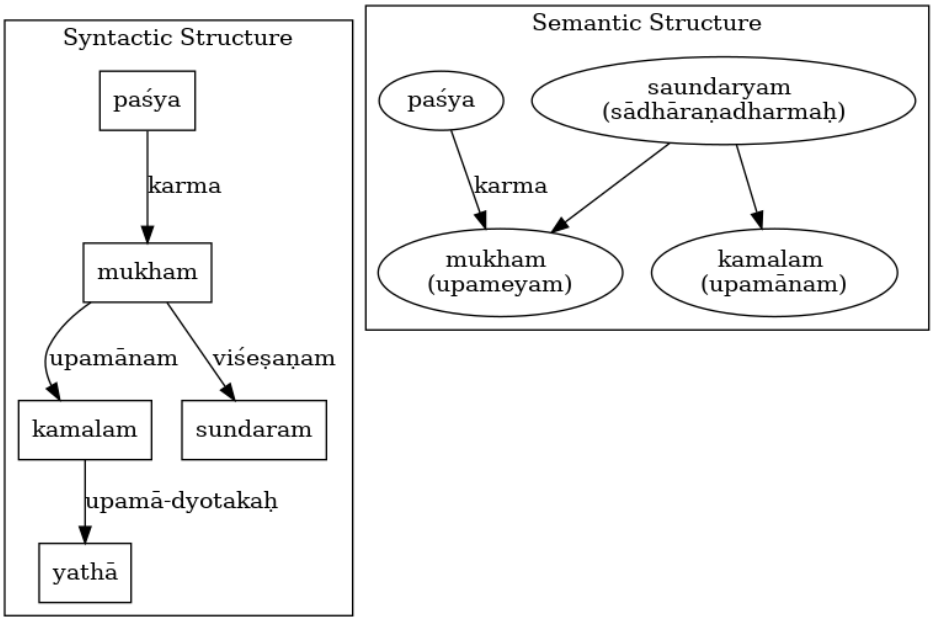


Figure 5.1: Example parse tree with *Upamā* specific tags

Here, the main sentence is *sundaraṃ mukhaṃ paśya* where *sundara* (beautiful) is an adjective (*viśeṣṇa*) of *mukham* (face) which is a *karma* of the activity of seeing *(paśya).* Now, there is a comparison of *mukha* with *kamala* (lotus) as an object of comparison, *mukha* being a subject of comparison and the shared property being beauty *(saundarya).* Now, in dependency parsing we just show the relation between *kamalam* and *mukham*, *kamalam* being the *upamāna* and *yathā* the marker (*Upamā-dyotaka*).

Since, in the dependency parser, we mark only the relations that are expressed through position or through markers, the semantic property of being a *sādhāraṇa-dharma* is not shown in the dependency diagram. Thus, *sundara* is marked only as a *viśeṣaṇa* of mukham and not shown explicitly as the *sādhāraṇa-dharma.* This decision is a very conscious decision. Only the syntactic information is marked. In addition, we decided to name the relation as *'upamāna'* following the convention we followed in dependency tagging, rather than *'upamāna-upameya-bhavaḥ'*, to keep it compact. The system followed in the marking the relations is as follows.

If the word 'Y' has a relation of 'R' with respect to 'X' then we mark it as: X->Y [R] (see diagram 5.2)

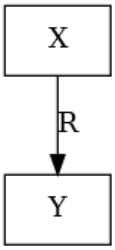


Figure 5.2: Representation of X->Y

For example, *rāmaḥ gacchati| (Rāma goes.) rāmaḥ -> gacchati [kartā]*, refer to figure 5.3,

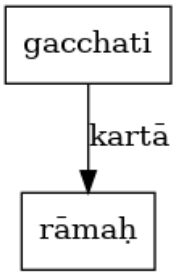


Figure 5.3: Parse Tree for *'rāmaḥ gacchati|'*

On similar lines, we decided to mark the relation between *mukham* and *kamalam* as *'upamāna'*. We have also shown the semantic structure of *Upamā* in the sentence in figure 5.1. This semantic structure is arrived at after discarding the *'dyotaka'* elements. We also take the same word form as in the sentence for easy understanding of the semantic graph.

5.4.2 Annotating *Upamā*: Guidelines

In this part, we present the guidelines for annotations in the form of parse trees for each of the types of *Upamā* described earlier. We present examples according to their grammatical category discussed in the previous section, viz. *Vākyagā, Samāsagā* and *Pratyayagā*.

- ***Vākyagā Upamā***: In this category, the *Upamā alaṅkāra* is expressed using explicit words. We get all four elements of *Upamā* as separate words. In *Vākyagā Śrautī Upamā* the *upamā_dyotaka*s are indeclinables *(avyayas)*.

  1. ***Pūrṇopamā Śrautī Vākyagā***:

     Skt: *mukham kamalam yathā sundaram asti.*

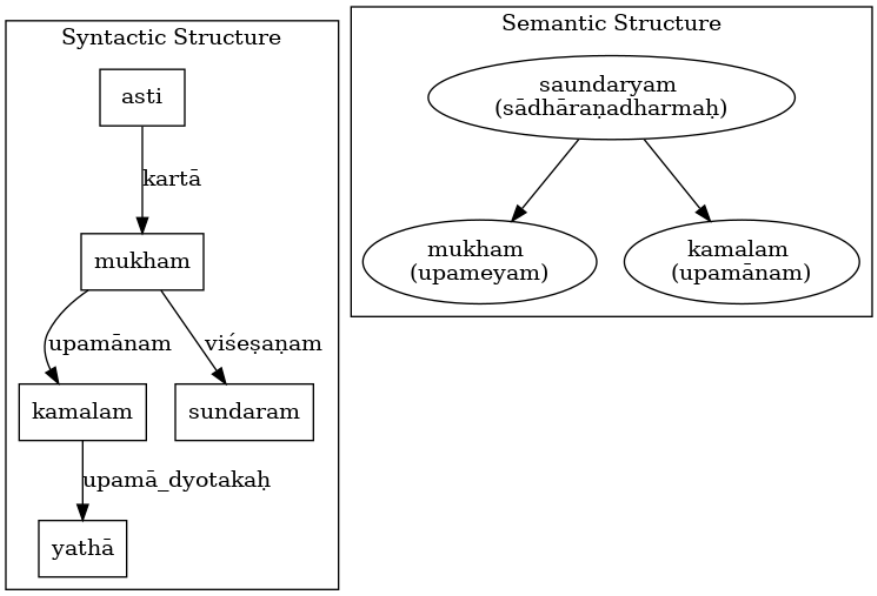


Figure 5.4: *Vākyagā Śrautī Pūrṇopamā*

Gloss: face{sg,nom,neut} lotus{sg,nom,neut} like{indecl} beautiful{sg,nom,neut} is{pres,3p,sg}

Eng: The face is beautiful like a lotus.

Upameyam: *mukham*, Upamānam: *kamalam*,

Upamā-dyotakaḥ: *yathā*, Samāna-dharma: *saundarya*.

Above are the four components mentioned in this sentence. Figure 5.4 shows the relations marked for the sentence, in syntactic as well as semantic level. *Mukham* being the *upameya* has a primary role of *kartā* in the sentence on primary layer. At semantic level, due to the *upamāna* relation with *kamalam*, we understand that the *mukham* is the *upameya*. Here, we have to note that both *mukham* and *kamalam* having same number, gender and case, making them compatible to be marked as *kartā*. But the comparator is always positioned after the *upamāna* in *śrautī upamā*. Hence, we can deduce that *kamalam* is the *upamāna* and *mukham* is the *upameya*.

For humans, differentiating between these factors is accomplished due to their access to world knowledge and the contextual cues provided in discourse. For machines, which lack such inherent world knowledge and contextual understanding, this task becomes significantly more challenging. Therefore, lexical clues provided by classical literary analysis prove to be of great help in the automatic processing

of such figurative expressions.

Similarly, in *Śrautī Upamā*, the comparative marker is always associated with and positioned next to the *Upamāna*.[6] For instance, we observe that *yathā* is marked as the *Upamā-dyotaka* along with *kamalam*.

The adjective *sundaram* indicates the shared property of beauty (*saundarya*). In this sentence, it is the only adjective present. Sometimes, multiple adjectives may appear to represent the shared quality. At times, adjectives may belong either to the *Upamāna* or to the *Upameya*, or in some cases, the action itself may be the common element.

2. ***Vākyagā Śrautī Dharmaluptā*** :

   Skt: *mukham kamalam yathā asti.*

   Gloss: face{sg,nom,neut} lotus{sg,nom,neut} like{indecl} is{pres,3p,sg}

   Eng: The face is like a lotus.

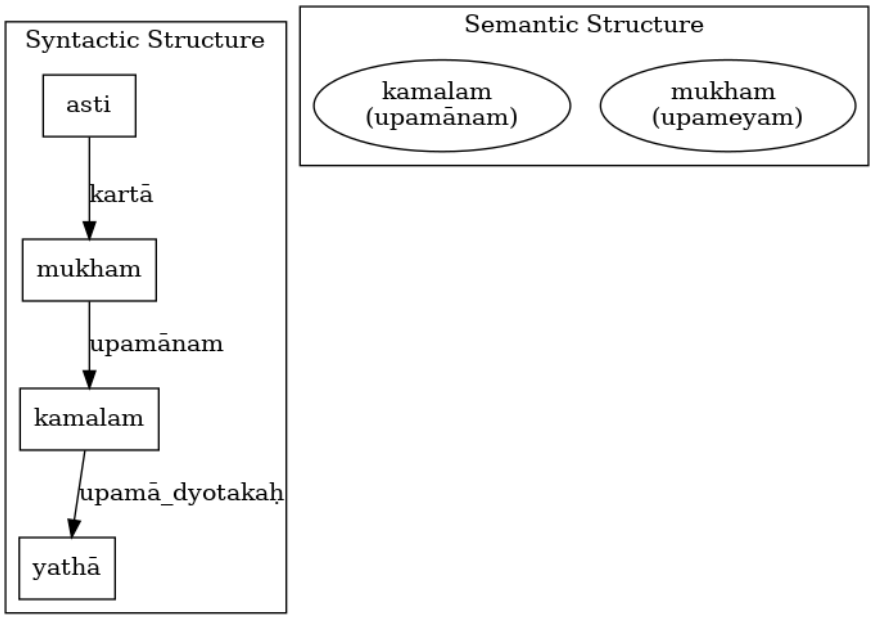


Figure 5.5: *Vākyagā Śrautī Dharmaluptā*

[6] yathevavādiśabdā yatparāstasyaivopamānatāpratītiriti yadyapyumānaviśeṣaṇānyete tathāpi śabdaśaktimahimnā śrutyaiva ṣaṣṭhīvat saṃbandhaṃ pratipādayantīti tatsadbhāve śrautī upamā| *Kāvyaprakāśa*,87, 10th Ullasa||

In this sentence, the shared property is not mentioned. However, the parse tree will be similar to *Pūrṇopamā Śrautī Vākyagā* without the word '*sundaram*'. See figure 5.5 for the syntactic parse structure and semantic representation.

3. ***Vākyagā* Ārthī Pūrṇopamā** :

   Skt: *mukham kamalena tulyam sundaram asti.*

   Gloss: face{sg,nom,neut} lotus{sg,instr,neut} comparable{sg,nom,neut} beautiful{sg,nom,neut} is{pres,3p,sg}

   Eng: The face is beautiful, comparable to a lotus.

   Here, *tulyam* is the comparative marker which is an independent word. In *Ārthī Upamā*, the structure of the sentence will be a bit different than *śrautī upamā*. *tulyam* has agreement with *mukham* which is the *upameya* unike *śrautī upamā*. But *tulya* has the third case expectancy for *kamalena* (*upamāna*). So, the parser can mark the *upamāna*. Figure 5.6 shows the parse tree for the given sentence with the semantic representation of *Upamā*.

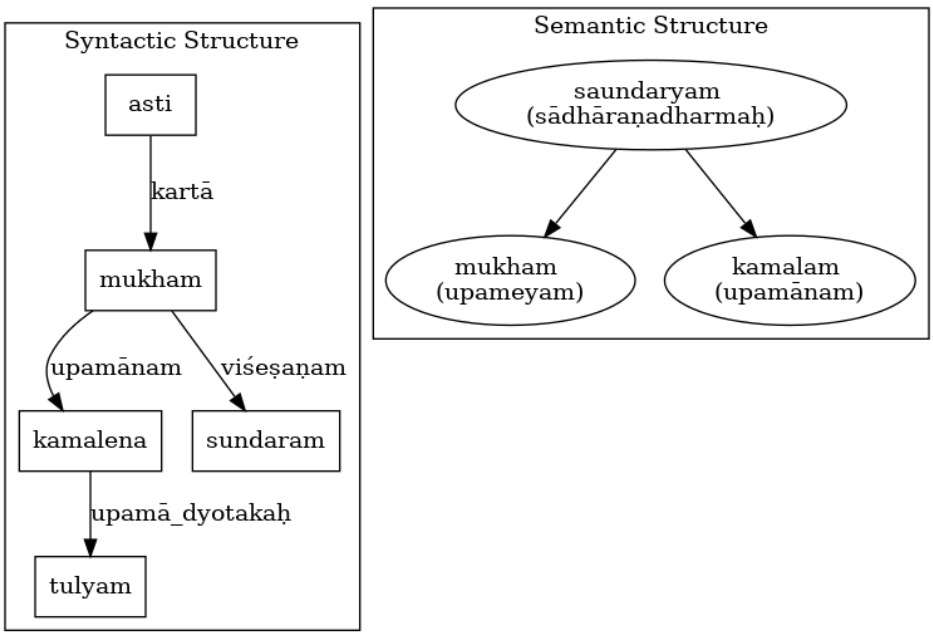


Figure 5.6: *Vākyagā Ārthī Pūrṇopamā: type I*

There are three ways the comparative marker can have the agreement with the other elements.

(a) **Type I:** The marker can be related to *upmeya* as in the given example.

(b) **Type II:** It can be related to *upamāna.*

For example, '*mukhasya tulyam sundaram kamalam asti*| (See figure 5.7)

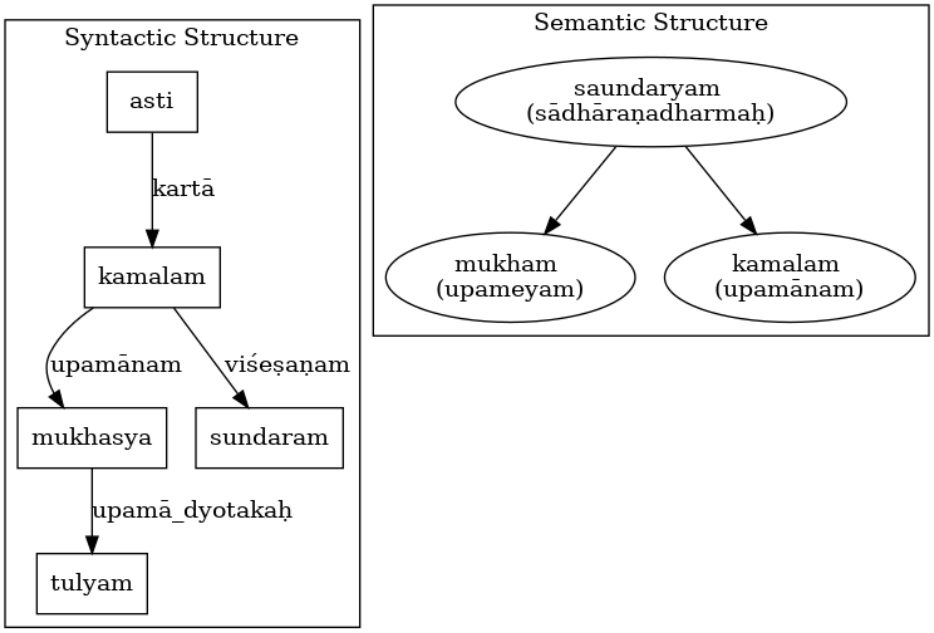


Figure 5.7: *Vākyagā Ārthī Pūrṇopamā: type II*

(c) **Type III:** it also can be related to both *upameya* and *upamāna.*

For example, '*mukham ca kamalam ca tulyam sundaram asti.* (See figure 5.8)

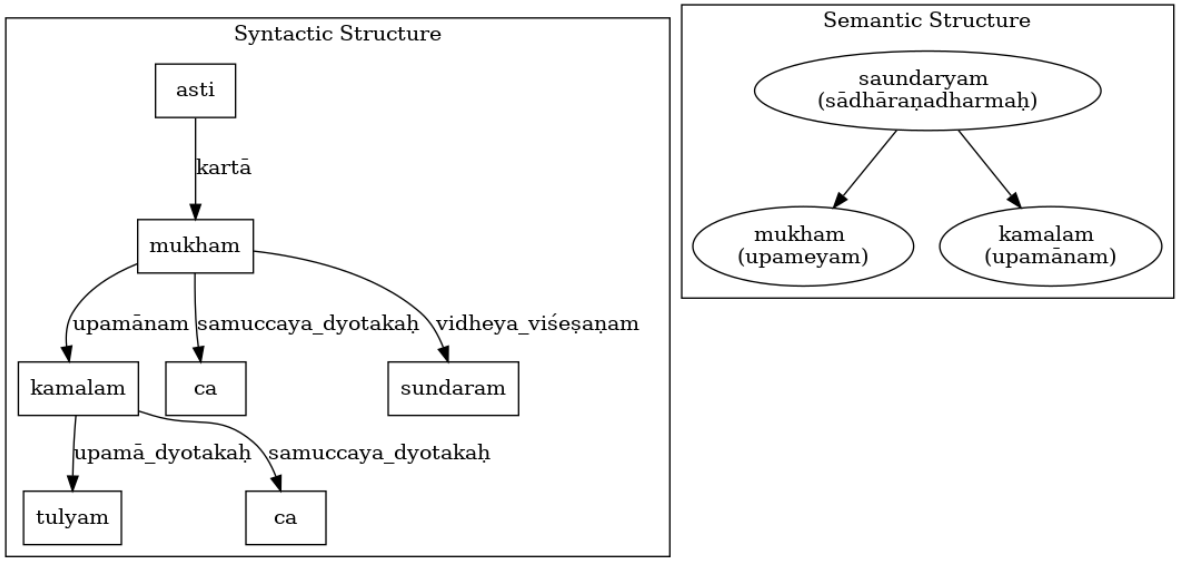


Figure 5.8: *Vākyagā Ārthī Pūrṇopamā:type III*

It makes it challenging for the parser to identify the *upamāna* and *upameya* when only the case, number and gender agreement provides the hint for identification. These words can also bear the *kāraka* roles as per the meaning of the sentence.

4. ***Dharmaluptā Ārthī Vākyagā***

   Skt: *mukham kamalena tulyam asti.*

   Gloss: face{sg,nom,neut} lotus{sg,instr,neut} comparable{sg,nom,neut} is{pres,3p,sg}

   Eng: The face is comparable to a lotus.

   The analysis of this example will be as *Pūrṇopamā Ārthī Vākyagā* without the mention of common property, that is *sundaram*. Figure 5.9 provides the parse tree for this sentence along with the semantic representation of the components of *Upamā*.

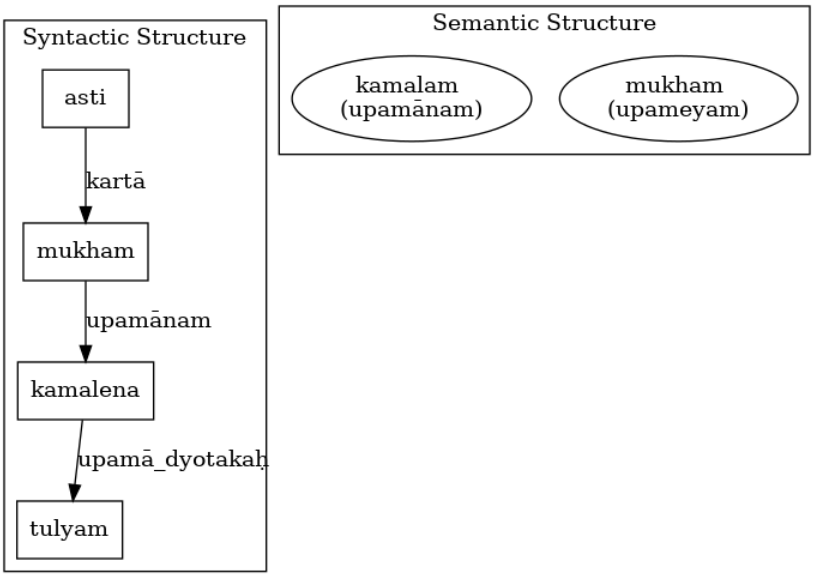


Figure 5.9: *Vākyagā Ārthī Dharmaluptā*

5. ***Vākyagā Ārthī Upamānaluptā***

   Skt: *mukhasya sadṛśaṃ sundaraṃ kim api na asti.*

   Gloss: face{sg,gen,neut} similar{sg,nom,neut} beautiful{sg,nom,neut} what{sg,nom,neut} any not is{pres,3p,sg}

   Eng: There is no beauty comparable to the face.

   In this sentence, 'sadṛśam' is the *dyotaka* which is an independent word. *Upameya* and the *samāna-dharma* are as before. There is no entity found to be compared

with the face, therefore this expression does not have a *upamāna.* Here, the phrase '*kim api na*' triggers the *upamānalupta* type. The syntactic parse structure and semantic representation is shown in figure 5.10.

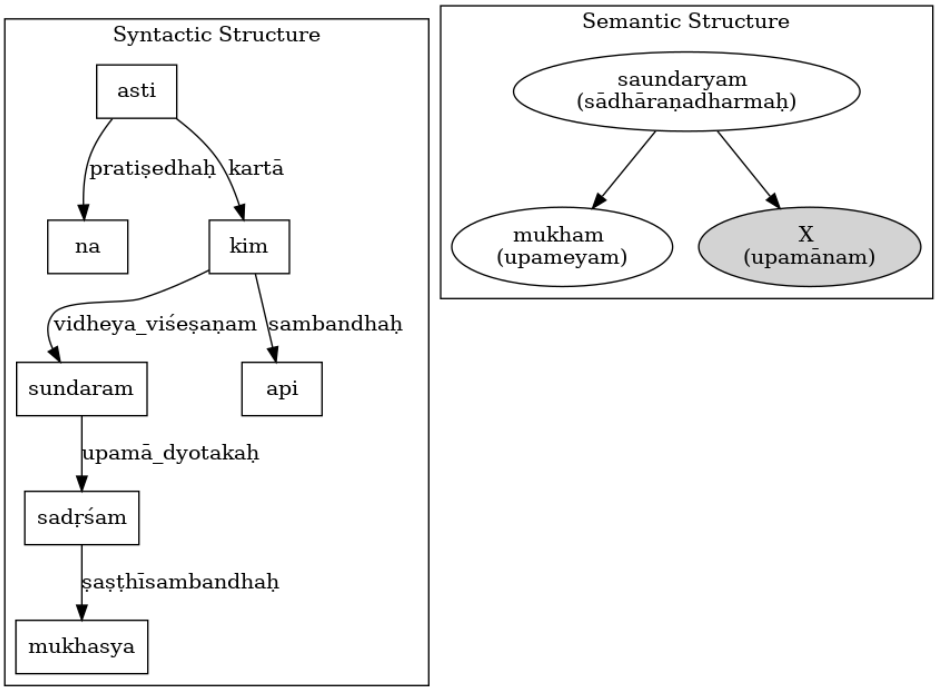


Figure 5.10: *Vākyagā Ārthī Upamānaluptā*

6. ***Vākyagā Ārthī Dharma-upamānaluptā***

   Skt: *mukhasya sadṛśam kim api na asti.*

   Gloss: face{sg,gen,neut} similar{sg,nom,neut} what{sg,nom,neut} any{indecl} not is{pres,3p,sg}

   Eng: Nothing similar to the face exists.

   The syntactic and semantic annotation for this example (see figure 5.11) will be similar to the previous type (*Upamānaluptā Ārthī Vākyagā*) without the mention of the common property '*sundaram*'.

- ***Samāsagā Upamā***

The *Upamā* expression using compounds is considered in *samāsagā Upamā.* There are a total of 9 types of *samāsagā Upamā.* In this category the *dyotaka*s are either elided or compounded with the other element. Thus, we mark the compound itself as *upamānam.* In some types, the compound which is having the elements of *Upamā* carries the *kāraka*

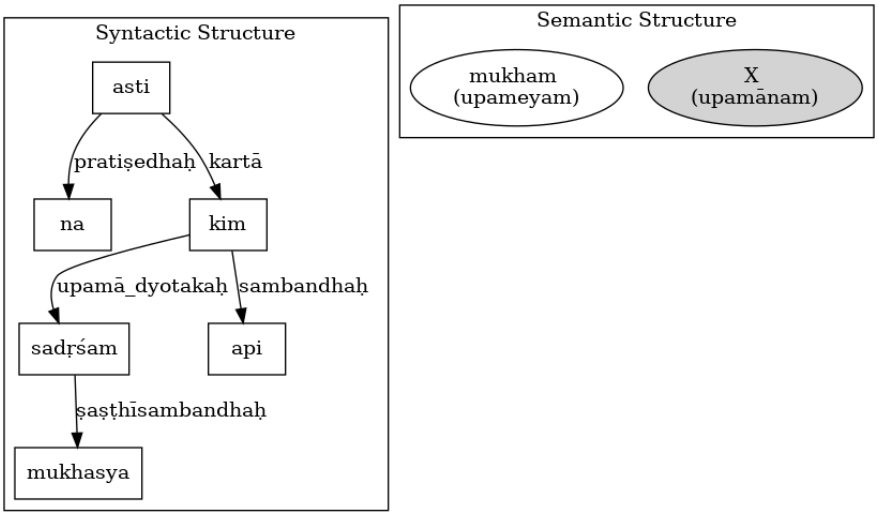


Figure 5.11: *Vākyagā Ārthī Dharma-upamānaluptā*

role in the sentence. In such cases, we have to understand the pesence of *Upamā* from the compound type.

1. ***Samāsagā Śrautī Pūrṇopamā***

   Skt: *mukham* *kamalam-iva* *sundaram* *asti.*

   Gloss: face{sg,nom,neut} lotus{sg,nom,neut}-like{indecl} beautiful{sg,nom,neut} is{pres,3p,sg}

   Eng: The face is beautiful like a lotus.

   In this sentence, except for the *Upamā-dyotakaḥ* that is *iva*, all other components are the same as *Pūrṇopamā Śrautī Vākyagā*. According to *'ivena samāso vibhaktyalopaśca'*|, a *vārtika* under *Aṣṭādyāyī* 2.1.4 and 2.4.71, *iva* is a particle (*avyaya*) which always forms a compound '*kevala-samāsaḥ*' with the *upamāna* without deletion of the case marker of the previous word. The syntactic parse tree and the semantic structure is as shown in figure 5.12.

2. ***Dharmaluptā Śrautī Samāsagā***

   Skt: *mukham* *kamalam-iva* *asti.*

   Gloss: face{sg,nom,neut} lotus{sg,nom,neut}-like{indecl} is{pres,3p,sg}

   Eng: The face is like a lotus.

   For this sentence, the analysis will be the same as *Pūrṇopamā Śrautī Samāsagā* without the entry of the common property *sundaram*. Figure 5.13 demonstrates the syntactic parse tree and the semantic representation of the given sentence.

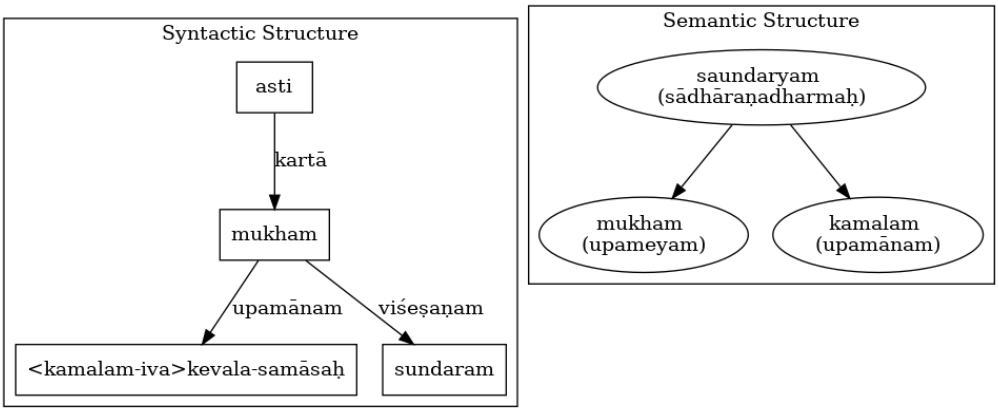


Figure 5.12: *Samāsagā Śrautī Pūrṇopamā*

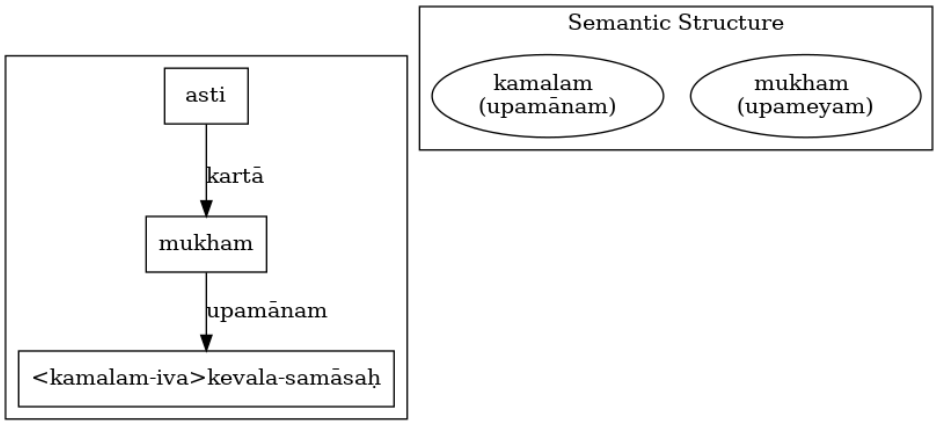


Figure 5.13: *Samāsagā Śrautī Dharmaluptā*

3. ***Samāsagā Śrautī Vācakaluptā***:

   Skt: *mukham kamala-sundaram asti.*

   Gloss: face{sg,nom,neut} lotus(like)-beautiful{sg,nom,neut} is{pres,3p,sg}

   Eng: The face is beautiful like a lotus.

   Here, the *upameya* and the *samāna-dharma* joined as a compound to give the meaning 'beautiful like a lotus' ('*kamala iva sundaram* |) by '*upamānāni sāmānyavacanaiḥ*|' (*Aṣṭādhyāyī*, 2.1.55). In this compound, the comparative marker gets elided. It is a kind of '*karmadhāraya-samāsa*'. The parser tree shows the tag 'karmadhāraya-samāsaḥ' for this compound, which provides the information of the compound type as *karmadhāraya*. Through this information, we get that the first component of the compound '*kamala*' is a *upamāna* and the second component, i.e. '*sundaram*', is a quality which is shared by the *upamāna* and the *upameya*. Figure 5.14 shows the syntactic structure and the semantic representation of this example.

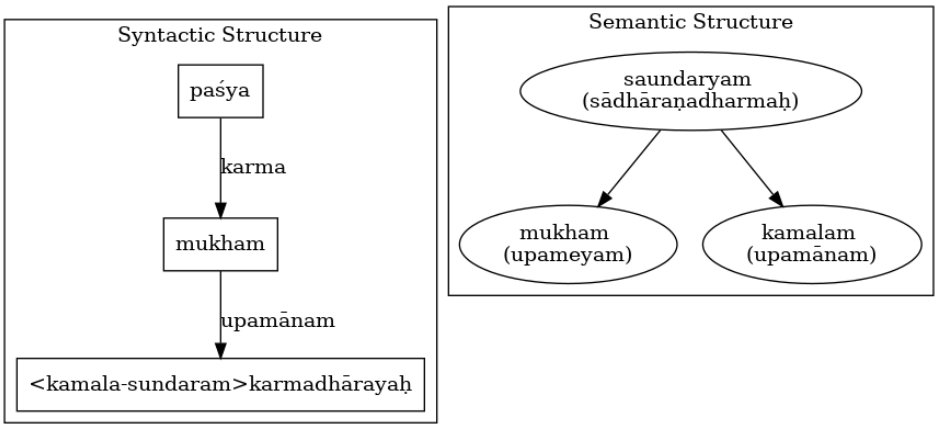


Figure 5.14: *Samāsagā Śrautī Vācakaluptā*

4. ***Samāsagā Śrautī Dharma-vācakaluptā***:

   Skt: *asau rājakuñjaraḥ rājate.*

   Gloss: that{sg,nom,masc} king-elephant(like){sg,nom,masc} shines{pres,3p,sg}

   Eng: That, elephant like king, shines.

   In this example, the word *rājakuñjaraḥ* is a compound, meaning '*rāja kuñjaraḥ iva*', expressing the *dyotaka* and the *dharma* with *samāsa*. This is an instance of *karmadhāraya* compound, which is formed by the *Aṣṭādhyāyī* rule -'*upamitaṃ vyāghrādibhiḥ sāmānyā'prayoge*| *(2.1.56)*, where the first component (here, *rājā*) is

the *upameya* and the second component (here, *kuñjara*) is a *upamāna*. It is tagged as 'karmadhāraya' in the parse tree in figure 5.15.

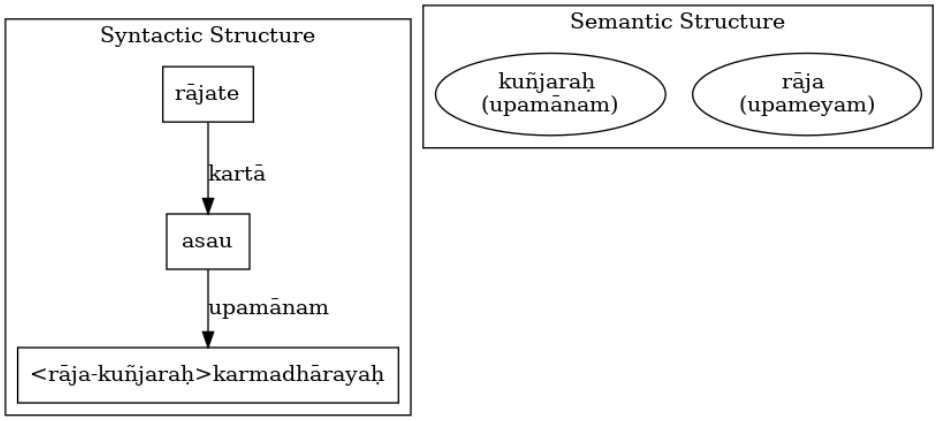


Figure 5.15: *Samāsagā Śrautī Dharma-vācakaluptā*

5. ***Samāsagā Ārthī Pūrṇopamā*** :

Skt: *mukham kamala-tulyam sundaram asti.*

Gloss: face{sg,nom,neut} lotus-similar{sg,nom,neut} beautiful{sg,nom,neut} is{pres,3p,sg}

Eng: The face is beautiful, comparable to a lotus.

Here, the marker, *tulyam* is compounded with the *upamāna - kamalam*. In the parser, it is treated as a single word. With the help of the *samāsa* module, we get the information of the type that is *Tṛtīyā-tatpuruṣa* of the compound word. From the paraphrase, we can decipher the *upamāna* and the *dyotaka*. The parse tree along with the semantic structure for the given instance is as shown in the figure 5.16.

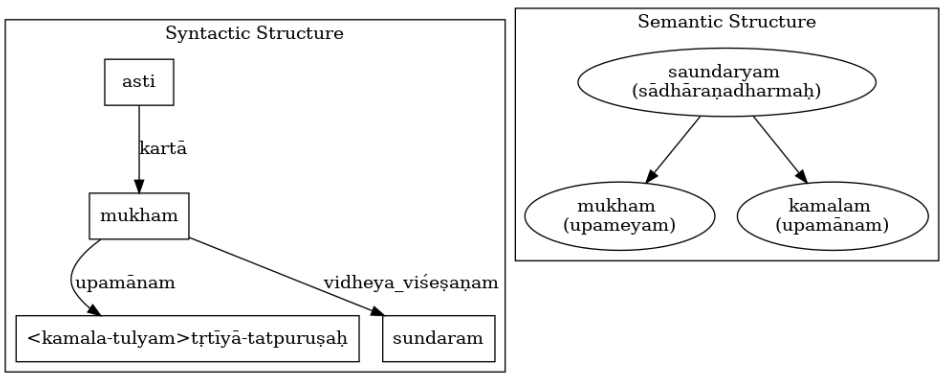


Figure 5.16: *Samāsagā Ārthī Pūrṇopamā*

6. ***Samāsagā Ārthī Dharmaluptā*** :

   Skt: *mukham kamala-tulyam asti.*

   Gloss: face{sg,nom,neut} lotus-comparable{sg,nom,neut} is{pres,3p,sg}

   Eng: The face is comparable to a lotus.

   The parse of the given example will be similar to the *Pūrṇopamā Ārthī Samāsagā* graph without the shared property. Figure 5.17 shows the syntactic parse structure and semantic representation of the given sentence.

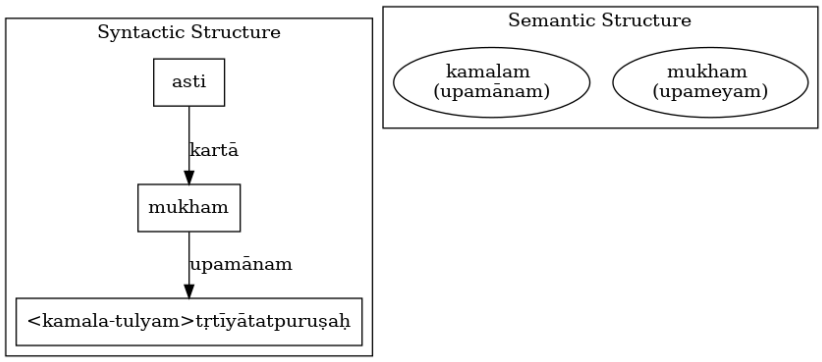


Figure 5.17: Samāsagā Ārthī Dharmaluptā

7. ***Samāsagā Ārthī Upamānaluptā***

   Skt: *mukha-sadṛśam sundaram kim api na asti.*

   Gloss: face-similar{sg,nom,neut} beautiful{sg,nom,neut} what{sg,nom,neut} any{indecl} not{indecl} is{pres,3p,sg}

   Eng: Nothing beautiful, similar to the face, exists.

   Here, *sadṛśam* is compounded with *mukham* with *Ṣaṣṭhītatpuruṣa samāsa*, therefore it is *samāsagā Upamā*. The figure 5.18 shows the parse tree and semantic structure.

8. ***Samāsagā Ārthī Dharma-upamānaluptā***

   Skt: *mukha-sadṛśam kim api na asti.*

   Gloss: face-similar{sg,nom,neut} what{sg,nom,neut} any{indecl} not{indecl} is{pres,3p,sg}

   Eng: Nothing similar to the face exists.

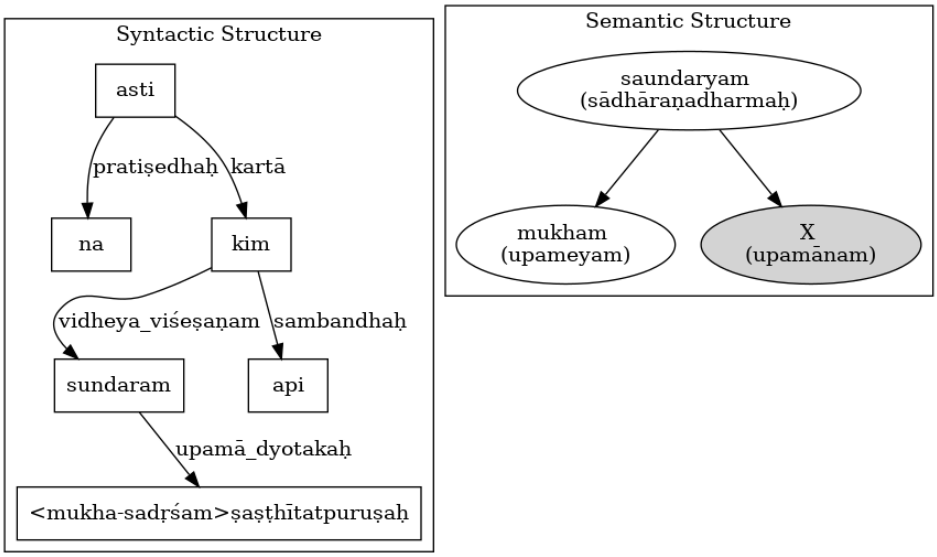


Figure 5.18: Samāsagā Ārthī Upamānaluptā

The analysis for this example will be similar to the previous *Upamānaluptā Ārthī Samāsagā* type without the common property. This example is demonstrated with the parse tree and semantic structure in figure 5.19.

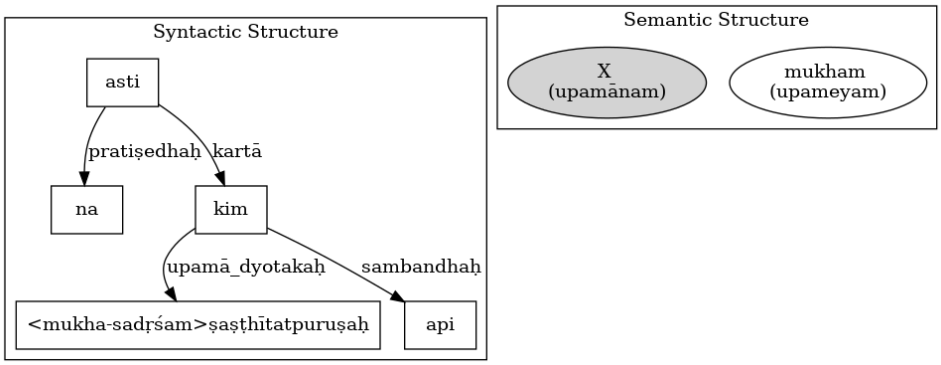


Figure 5.19: Samāsagā Ārthī Dharma-upamānaluptā

9. **Triluptā**: ***Samāsagā Ārthī Dharma-upamāna-vācaka-luptā***

   Skt: *mṛganayanā manaḥ harate.*

   Gloss: deer-eyed{sg,nom,fem} mind{sg,acc,neut} steals{pres,3p,sg}

   Eng: The doe-eyed (woman) steals the mind.

   mṛganayanā is a *bahuvrīhi compound (upamānapūrvapada-vyadhikaraṇa-bahuvrīhi)* under the *Aṣṭādhyāyī* rule *anekamanyapadārthe| (2.2.24)* and the *vārtika - 'saptamyupamānapūrvapadasyottarapadalopaśca|'. Upamāna-pūrvapada-vyadhikaraṇa-bahuvrīhi* is a kind of *bahuvrīhi* compound, where the preceding component is *up-*

amāna. This compound has the meaning '*mṛga-locane iva cañcale nayane yasyaḥ sā*'. Here, *(mṛga-)locane* is *upamāna* out of which only a part *mṛga* is explicit in the sentence, and the latar component- *nayane* is *upameya.* The parse tree, along with the semantic representation of the given instance, is shown in the figure 5.20.

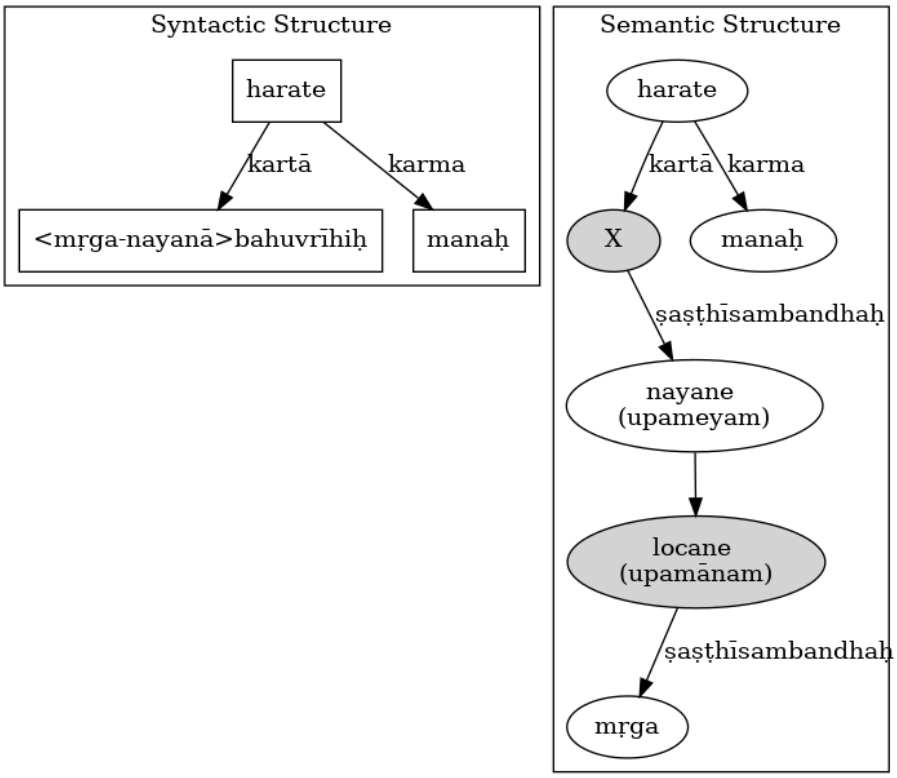


Figure 5.20: *Samāsagā Ārthī Dharma-upamāna-vācaka-luptā*

- ***Pratyayagā Upamā***:

  The comparison in *upamā* expressed through affixes is considered as *Pratyayagā Upamā. Taddhita* affixes are attached to a noun to form a new *prātipadika* (nominal stem). Similarly, some *sanādi* (*kyac, kyaṅ, kvip*) and *kṛt* (*ṇamul*) affixes are attached to a noun to form a new verb. There are 10 types of affixes which are used in *Upamā* to denote similarity.

  1. ***Pratyayagā Śrautī Pūrṇopamā***

     Skt: *mukhasya saundaryam kamalavat asti.*

     Gloss: face{sg,gen,neut} beauty{sg,nom,neut} lotus-like{indecl} is{pres,3p,sg}

     Eng: The beauty of the face is like a lotus.

Here, the *upamāna* along with the *dyotaka* are presented by the word *kamalavat*. By '*tatra tasyeva*|'(*Aṣṭādhyāyī, 5.1.116*) to depict the similarity the '*vati*' *tadhhita* affix is added to the *upamāna*. Figure 5.21 demonstrates the parse tree and the semantic structure for the sentence.

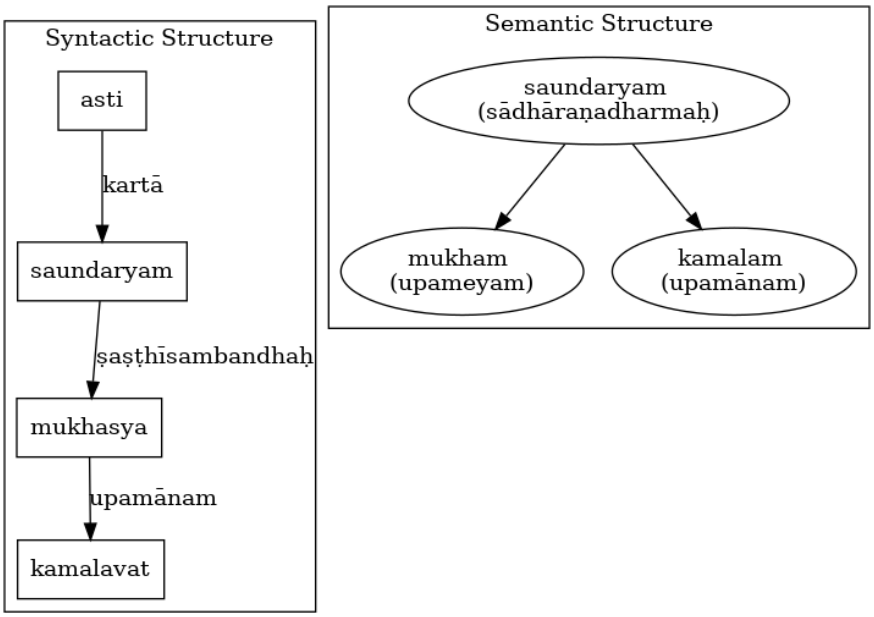


Figure 5.21: *Pratyayagā Śrautī Pūrṇopamā*

2. ***Pratyayagā Ārthī Pūrṇopamā***

   Skt: *mukham kamalavat sundaram asti.*

   Gloss: face{sg,nom,neut} lotus-like{indecl} beautiful{sg,nom,neut} is{pres,3p,sg}

   Eng: The face is beautiful like a lotus.

   In this sentence, *vati*, a *taddhita* affix is added in the meaning of similarity by *'tena tulyam kriyā cedvatiḥ*|'(*Aṣṭādhyāyī*, 5.1.116). in stead of *kamalena tulyam* the taddhita ending word *kamalavat* has come. Figure 5.22 demonstrates the syntactic and semantic representation of the example.

   The following verse has the examples of *kyac* and *kyaṅ* affixes.

   *pauraṃ **sutīyati** janaṃ samarāntare'sāv**antaḥpurīyati** vicitracaritracuñcuḥ|*
   __nārīyate__ samārasīmni kṛpāṇapāṇerālokya tasya caritāni sapatnasenā||

3. ***Pratyayagā Śrautī Vācakaluptā karma - kyac***:

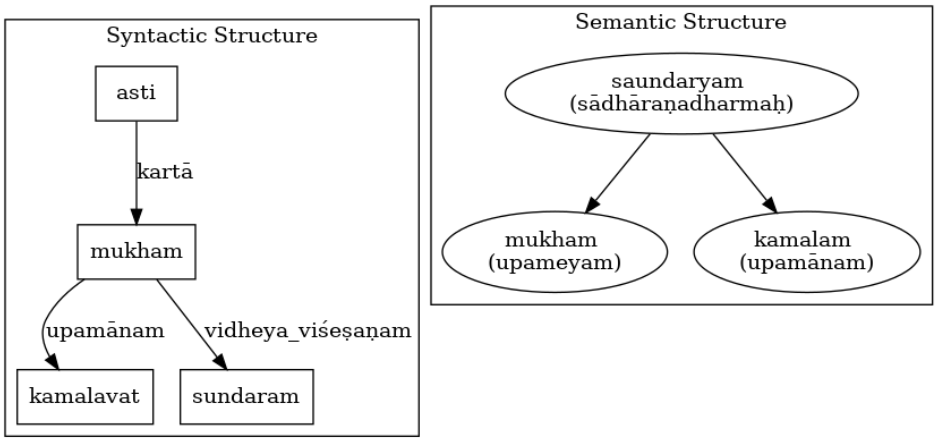


Figure 5.22: *Pratyayagā Ārthī Pūrṇopamā*

Skt: *asau pauraṃ sutīyati.*

Gloss: that{sg,nom,masc} citizen{sg,acc,neut} treats-like-children{pres,3p,sg}

Eng: That one treats citizens as children.

The word '*sutīyati* has the *kyac* affix in the meaning of *'sutam iva ācarati|'* by *Aṣṭādhyāyī rule - upamānādācāre| (3.1.10).* Refer to figure 5.23 for the semantic and syntactic structure of the sentence.

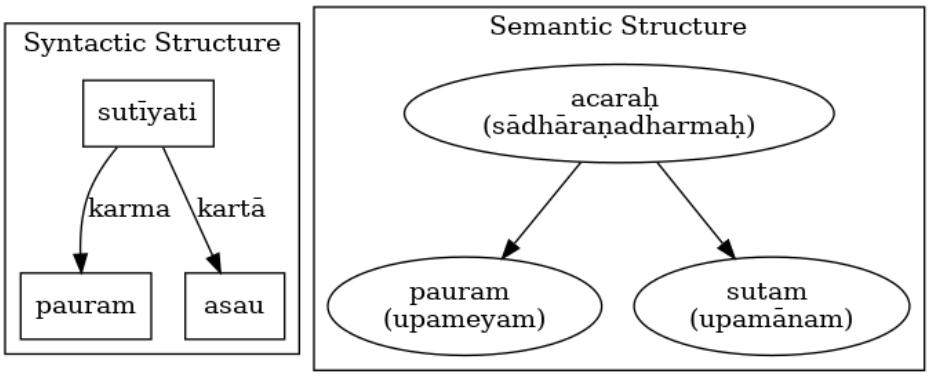


Figure 5.23: *Pratyayagā Śrautī Vācakaluptā karma - kyac*

4. ***Pratyayagā Śrautī Vācakaluptā ādhāra - kyac***:

   Skt: *asau samare antaḥpurīyati.*

   Gloss: that{sg,nom,masc} battle{sg,loc,neut} behaves-as-in-apartment{pres,3p,sg}

   Eng: That one behaves in battle, as he is in his home.

   The word '*antaḥpurīyati*' having *kyac* suffux gives the meaning of *'antaḥpure iva ācarati|'* by the *vārtika* '*adhikaraṇācceti vaktavyam*' on *Aṣṭādhyāyī* rule *up-*

amānādācāre| (3.1.10) . Figure 5.25 shows the annotation parse tree and syntactic representation of the given example.

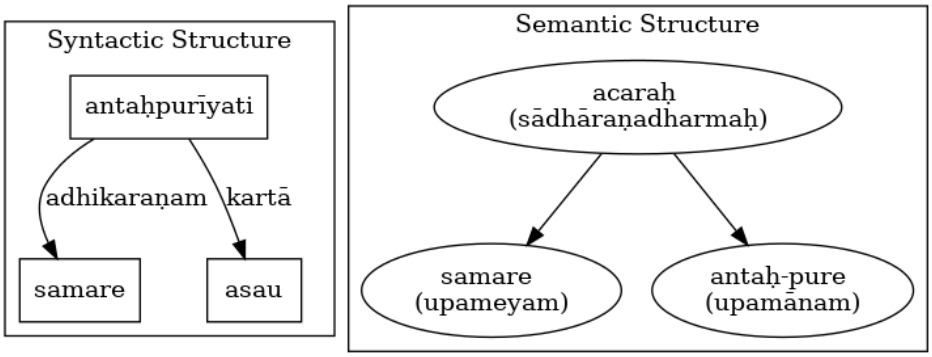


Figure 5.24: *Pratyayagā Śrautī Vācakaluptā ādhāra - kyac*

5. ***Pratyayagā Śrautī Vācakaluptā kartā - kyaṅ***:

   Skt: *senā nārīyate.*

   Gloss: army{sg,nom,fem} behaves-like-woman{pres,3p,sg}

   Eng: The army behaves like a woman.

   The word from the above given verse, *nārīyate* has *kyaṅ* affix in the meaning of *kartā* for *nārī iva ācarati|* by *Aṣṭādhyāyī* rule '*kartuḥ kyaṅ salopaśca|*' (3.1.11)|. Figure 5.28 shows the parse tree with the semantic structure for the same.

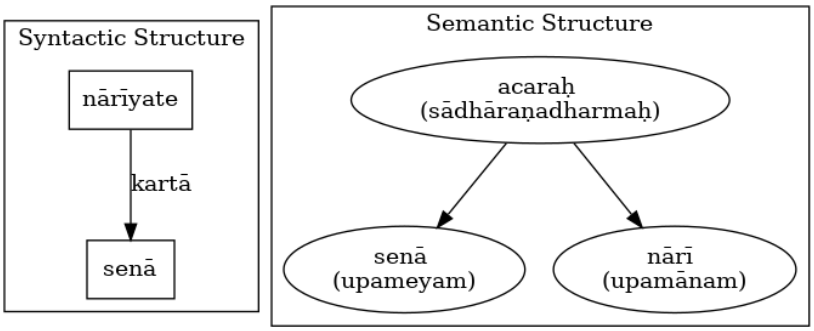


Figure 5.25: *Pratyayagā Śrautī Vācakaluptā kartā - kyaṅ*

6. **Pratyayagā Śrautī Vācakaluptā karma - ṇamul**:

   Following verse demonstrates the affix *ṇamul* in two different meanings. The first line has the affix in *karma* meaning and the second line has it in *kartā* meaning.

   mṛdhe ***nidāghagharmāṃśudarśaṃ*** *paśyanti taṃ pare |*

   sa punaḥ ***pārthasaṃcāraṃ*** *saṃcaratyavanīpatiḥ ||*

Skt: *taṃ gharmaṃśudarśaṃ paśyanti.*

Gloss: he{sg,acc,masc} Sun-see{sg,acc,masc} see{pres,3p,pl}

Eng: He was looked upon as the sun.

Here, '*nidāghagharmāṃśudarśaṃ*' (for *nidāghagharmāṃśum iva paśyati*') has the *ṇamul* affix in the meaning of *karma* by *Aṣṭādhyāyī* rule '*upamāne karmaṇi ca|' (3.4.45).* Refer to the figure 5.26 for the detailed parse tree and semantic structure of the sentence.

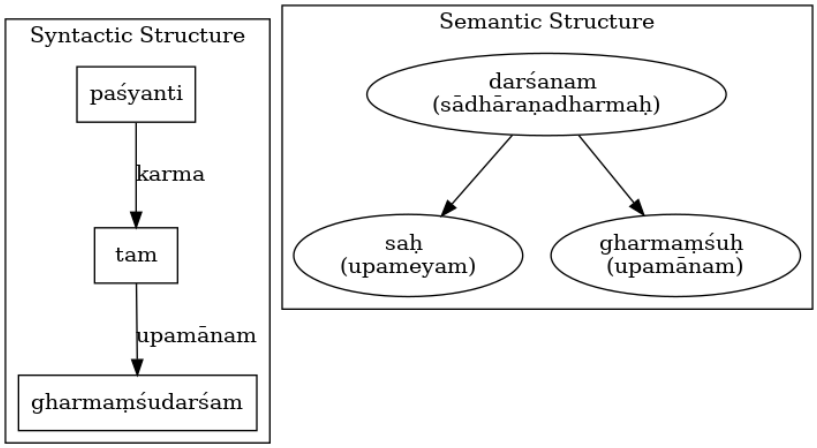


Figure 5.26: *Pratyayagā Śrautī Vācakaluptā karma - ṇamul*

7. ***Pratyayagā Śrautī Vācakaluptā kartā - ṇamul***:

   Skt: *saḥ pārthasañcāraṃ sañcarati.*

   Gloss: he{sg,nom,masc} movement-of-Arjuna{sg,acc,masc} move{pres,3p,sg}

   Eng: He moves like Arjuna.

   '*pārthasaṃcāraṃ*' (for pārthaḥ iva sañcaraṇam) has the *'ṇamul'* affix in the meaning of *kartā* by the same *Aṣṭādhyāyī* rule as previous. Figure 5.27 shows the syntactic and semantic structure for the given example.

8. ***Pratyayagā Śrautī Vācaka-upameyaluptā : kyac***

   Skt: *rājā* ***sahasrāyudhīyati|***

   Gloss: king{sg,nom,masc} behaves-like-having-thousand-weapons{pres,3p,sg}

   Eng: The king acts as if he is the 'kārtavīrya' having thousand arms.

   Here, '*sahasrāyudhīyati*' has *kyac* affix giving the meaning - '*sahasrāyudham iva ācarati ātmānam*'. Figure 5.28 provides the parse tree and the semantic representation of the stated instance.

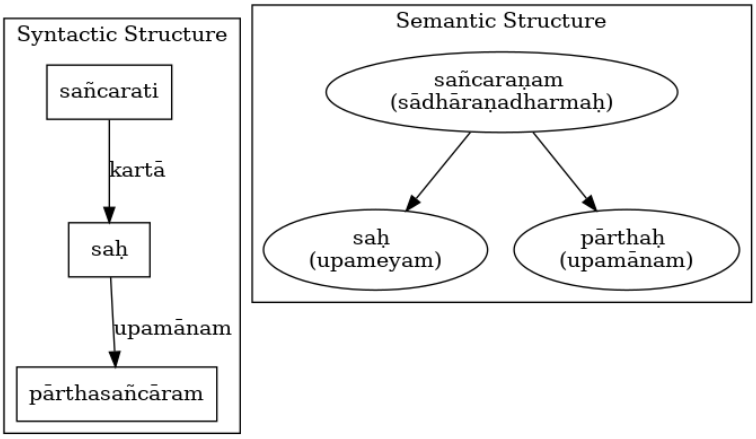


Figure 5.27: *Pratyayagā Śrautī Vācakaluptā kartā - ṇamul*

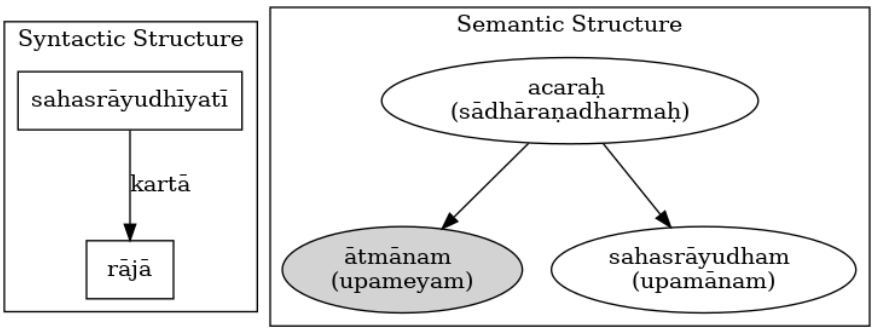


Figure 5.28: *Pratyayagā Śrautī Vācaka-upameyaluptā : kyac*

9. ***Pratyayagā Śrautī Dharma-vācakaluptā - kvip*** :

   Skt: *yāminyaḥ* ***dinanti|***

   Gloss: night{pl,nom,fem} acts-like-day{pres,3p,pl}

   Eng: The nights feels like days.

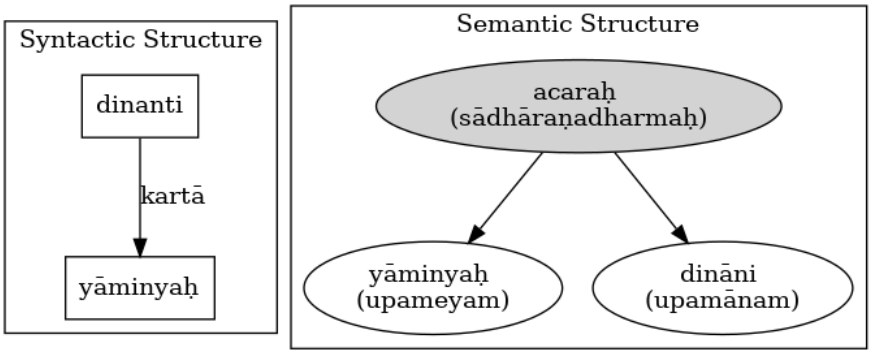


Figure 5.29: *Pratyayagā Śrautī Dharma-vācakaluptā - kvip*

Here, '*yāminyaḥ*' (nights) is the *upameya*. The *kvip* affixed word *'dinanti'* having the meaning *'dinam iva ācarati |'* expressing *upamānam*(*dinam*), while implicitly provides the *dyotaka* and the *dharma*. The affix '*kvip*' gets elided due to the rule *verapṛktasya| (Aṣṭādhyāyī, 6.1.67)*. Therefore, the common property 'ācāraḥ', which is expressed through the suffix, is also considered as omitted.[7] Figure 5.29 shows the parse tree and semantic representation.

10. ***Pratyayagā Ārthī Dharmaluptā***

    Skt: *duṣṭasya manaḥ* ***viṣakalpaṃ*** *asti|*

    Gloss: wicked{sg,gen,masc} mind{sg,nom,neut} poison-like{sg,nom/acc,neut} is{pres,3p,sg}

    Eng: The mind of a wicked person is like a poison.

    In this example, the word *viṣakalpaṃ* has the *taddhita* affix '*kalpap*' by the *Aṣṭādhyāyī* rule '*īṣadasamāptau kalpabdeśyadeśīyaraḥ| (5.3.67)*'. Figure 5.30 shows the parse tree along with the semantic structure for the given example.

We observe that out of 25 subtypes of *Upamā*, the parser can analyse *Vākyagā Upamās* of 6 types and *Samāsagā* of 9 types, making a total of 15. The 10 *Upamā*s involving *Taddhita*,

[7] *vivaraṇakārāstu kippratyaya eva tulyācārarūpasādhāraṇadharmavācakaḥ| tasya "verapṛktasya"-iti sūtreṇa nityaṃ lupta-tvāddharmalopa ityāhuḥ| 'Balabodhinī' commentary on 10.89, Kāvyaprakāśa*

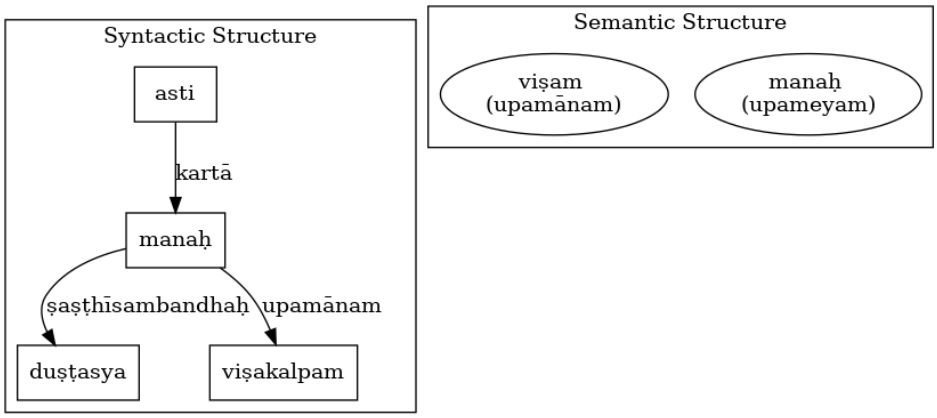


Figure 5.30: *Pratyayagā Ārthī Dharmaluptā*

sanādi and *kṛt* suffixes cannot be handled by the current system due to the non-availability of morphological analysers to analyse these type of words. In the analysis, the major cue to capture the *Upamā* is the comparative marker. In the types *Vākyagā*, *Samāsagā*, where the *dyotaka* is explicit, the parser is able to recognise it as *Upamā* and assign appropriate tags, without any difficulty. Due to the addition of the 'Compound-type Identification module' (Krishnan et al., 2025) the 9 *samāsagā upamā* sentences can also be processed by the parser.

5.4.3 Handling of special nuances from literature

In *Kāvyaprakāśa*, *Mammaṭa* systematically addresses the major grammatical categories and illustrates each of them with corresponding examples. These examples typically employ a single *upamāna–upameya* pair and conform to the standard construction of *Upamā*. However, in actual poetic practice, the use of *Upamā* often departs from this standard form. A poet, exercising creative freedom, may construct expressions involving more than one *upamāna–upameya* pair. In some instances, a single *upameya* may be compared with multiple *upamānas*. Thus, poetic constructions frequently display structural variations beyond the canonical type.

When designing guidelines for parse tree representation, it is therefore necessary to accommodate such creative possibilities alongside the standard forms. Furthermore, certain other *alaṅkāras* exhibit structures similar to *Upamā*. To avoid ambiguity, these must be represented in a manner that distinguishes them from *Upamā*. Accordingly, the guidelines are extended to incorporate these nuances, ensuring that both the creativity of poetic expression and the structural particularities of related *alaṅkāras* are adequately captured.

- **More than one *Upamāna*s and *upameya*s with one comparative marker**:

  Skt: *sītā api anugatā rāmaṃ śaśinaṃ rohiṇī yathā |*

  Gloss: Sītā{sg,nom,fem} also{indecl} followed{ptcp,past,nom,sg,fem} Rāma{sg,acc,masc} moon{sg,acc,masc} Rohiṇī{sg,nom,fem} like{indecl}

  Eng: Sītā also followed Rāma, as Rohiṇī follows the moon.

  This sentence is taken from *Rāmāyaṇam* 1.1.28. In this sentence, there are two *upamāna*s and their respective *upameya*s. But only one comparative marker is given. In this case, we can mark the relation of *yathā* with *rohiṇī* after which *upamā_dyotaka* is present in the sentence. We mark the relation of *upamānam* between *rāmam* and *śaśinam* ; also in between *sītā* and *rohiṇī*. Figure 5.31 shows the parse tree for the given example.

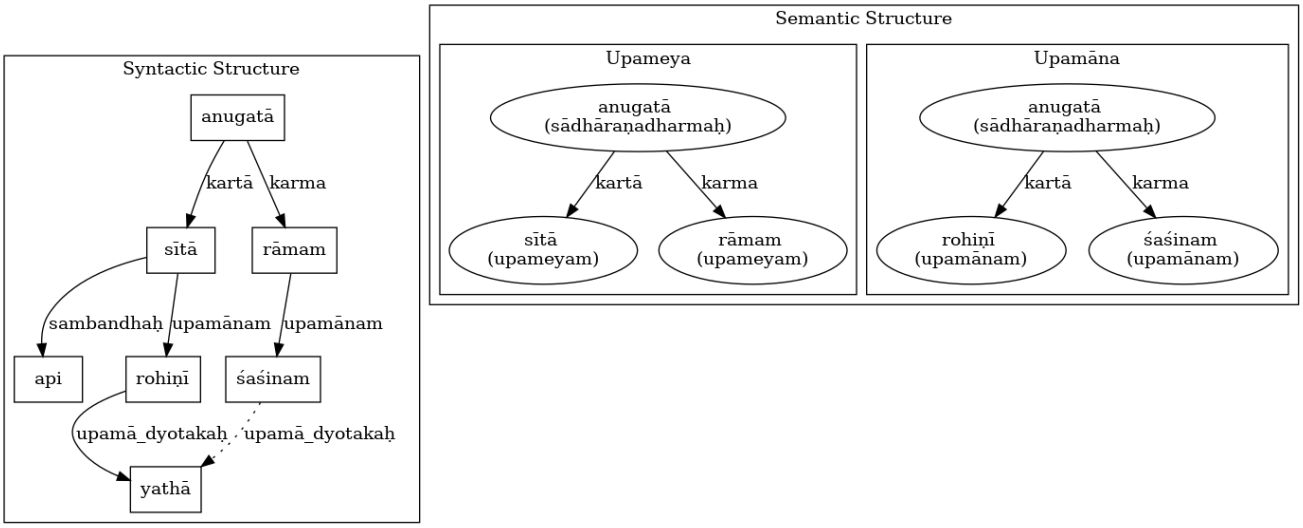


Figure 5.31: Representation of more than one *upamāna*s and *upameya*s

- ***Iva* not connected with the *upamāna* in *samāsa*.**

  Iva forms a compound with *upamāna* without the deletion of the case marker even when it is a *samāsa*.[8] The samāsa which is formed with *iva* should have the *upamāna* as the initial component. But, in some verses[9] we have observed that *iva* is attached to some other word than the *upamāna*. In such cases, we treat it as *Vākyagā Upamā*. We

[8] *ivena samāso vibhaktyalopaśca| vārtikam under Aṣṭādyāyī 2.1.4 and 2.4.71*

[9] *udbāhuriva vāmanaḥ| 1.3, Raghuvaṃśa*
jugopa gorūpadharāmivormim| 2.3, Raghuvaṃśa

mark *iva* as an independant word with the relation *upamā_dyotakaḥ* with the *upamāna* and not with the word with which *iva* is attached.

iti vismṛtānyakaraṇīyamātmanaḥ sacivāvalambidhuraṃ dharādhipam |
parivṛddharāgamanubandhasevayā mṛgayā jahāra catureva kāminī ||RV, 9.69||

Gloss: (*iti*) thus{indecl} (*vismṛtānyakaraṇīyamātmanaḥ*) whose.other.duties. forgotten.by.himself{sg,gen,masc} (*sacivāvalambidhuraṃ*) minister.reliance. burden{sg,acc,masc}
(*dharādhipam*) king{sg,acc,masc} (*parivṛddharāgamanubandhasevayā*) with.service.of.increasing.passion{sg,instr,fem} (*mṛgayā*) hunt{sg,nom,fem}
(*jahāra*) carried.off{aor,3p,sg,act} (*caturā*)cleverly (*iva*) like{indecl}
(*kāminī*) beloved.woman{sg,nom,fem}

Eng: Thus, the hunt, with service bound up with growing passion, carried off the king—who, having forgotten other duties, bore the burden of reliance on his ministers—just as a clever woman seizes her lover.

This verse is taken from *Raghuvaṃśam*(RV). In this verse the *upamā_dyotaka* is *iva* which is attached with an adjective of the *upamāna* that is *caturā* and not with the *upamāna - kāminī*. To show the annotation for such sentence, here, we form a sentence from the above mentioned verse having the components of *Upamā*. Figure 5.32 shows the parse tree for the given example. Example:
adhipam mṛgayā jahāra catureva kāminī |

- **Differentiating between *Upamā* and other *arthālaṅkāras* when *iva* is the marker.**

  In addition, several other *arthālaṅkāras* also employ a *dyotaka* (comparative marker) similar to that used in *Upamā*. For instance, *Utprekṣā* and *Ananvaya* exhibit surface structures that can be mistaken for *Upamā* by a computational parser. However, their semantics and poetic effect differ significantly. In *Utprekṣā*, the *Upamāna* is imaginary or hypothetical, whereas in *Ananvaya* the *Upamāna* and the *Upameya* refer to the same entity. On the same grounds, *Upameyopamā* is an *alaṅkāra* where the comparison is done reciprocally. The *upameya* becomes the *upamāna*. Here, we show the representative parse trees for these *alaṅkāras*.

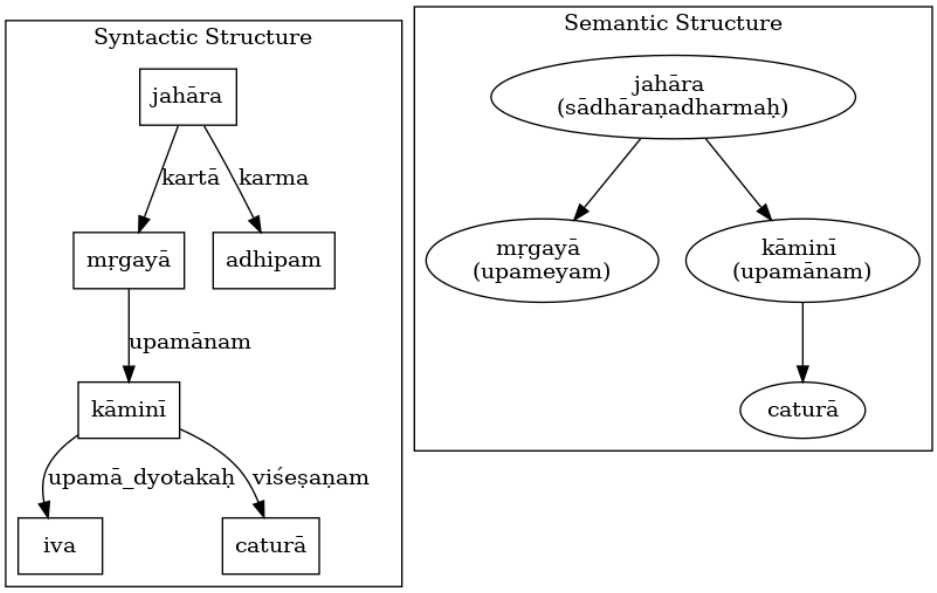


Figure 5.32: Example parse tree for *Iva* not connected with the *upamāna*

– ***Ananvaya***

The following example is taken from *Rāmāyaṇa* (6.107.52). In the source text the śloka is as 'rāmarāvaṇayoryuddham rāmarāvaṇayoriva|'. Here, we have to supply '*yuddham*' as the *upamāna*.

Skt: *rāma-rāvaṇayoḥ yuddham rāma-rāvaṇayoḥ yuddham iva asti |*

Gloss: Rāma-Rāvaṇa{du,gen,masc} battle{sg,nom/acc,neut} Rāma-Rāvaṇa{du,gen,masc} battle{sg,nom/acc,neut} as.if{indecl} be{pres,3p,sg}

Eng: The battle of Rāma and Rāvaṇa is only like the battle of Rāma and Rāvaṇa.

In *ananvayā*, the *upameya* is depicted as there is no other entity to compare with it. In other words, the *sādhāraṇadharma* which is being compared cannot be seen in any other object. Hence, the *Upameya* itself is presented as the *upamāna* as in the given example. The war between *Rāma* and *Rāvaṇa* is so great that it can be compared only with itself. The parse tree representation is shown is figure 5.33.

– ***Upameyopamā***

Skt: *dharaṇī iva tasya dhṛtiḥ asti| tasya dhṛtiḥ iva dharaṇī asti|*

Gloss: earth{sg,nom,fem} like{indecl} his{sg,gen,masc} firmness{sg,nom,fem} be{pres,3p,sg} his{sg,gen,masc} firmness{sg,nom,fem} like{indecl} earth{sg,nom,fem} be{pres,3p,sg}

Eng: His firmness is like the Earth; and the Earth is like his firmness.

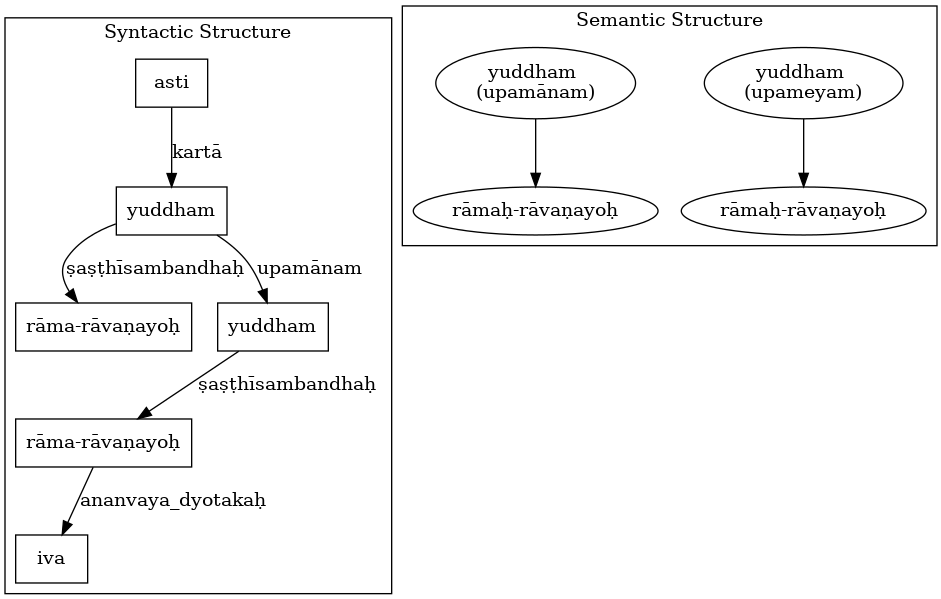


Figure 5.33: Representation of *Ananvayā*

This *alaṅkāra*, to show the degree of similarity between the *upamāna* and the *upameya*, both are used interchangeably. The example 'his fortitude' (tasya dhṛtiḥ) is compared with 'the earth' (dharaṇī) and vice versa as demonstrated in figure 5.34.

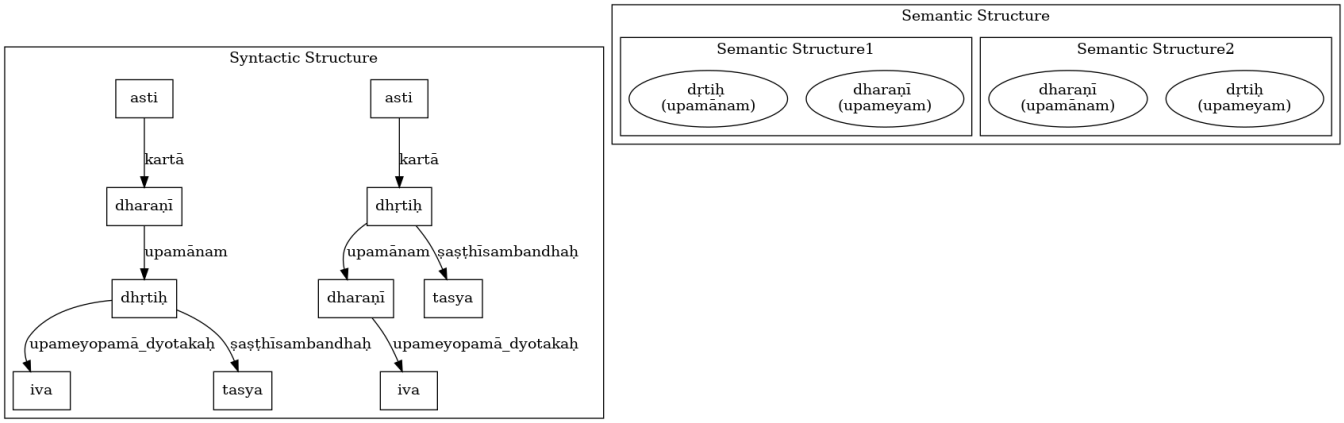


Figure 5.34: Representation of *Upameyopamā*

- ***Utprekṣā***:

The figures explained before have the common property as the basis of the comparison. In *Utprekṣā* the common property is the subject of suggestive meaning (*vyaṅgyasādharmya*). In this *alaṅkāra*, the object is imagined or forced with the said common property to be compared as *upamāna*. Hence, we do not show the suggested *sādhāraṇadharma* in the semantic representation as it may vary from person to person.

Skt: *limpati iva tamaḥ aṅgāni |*

Gloss: cling{pres,3p,sg} as.if{indecl} darkness{sg,nom,neut} limbs{pl,acc,neut}

Eng: It is as if darkness clings to the limbs.

In this example, the marker *iva* is attached to the action *limpati*, which is a clue for identifying *utprekṣā*. Unlike *Upamā*, there is no *upamāna-upameya-bhāva* present in the sentence. The action of smearing (*limpati*) is imposed by the poet. Figure 5.35 shows the parse tree for the same.

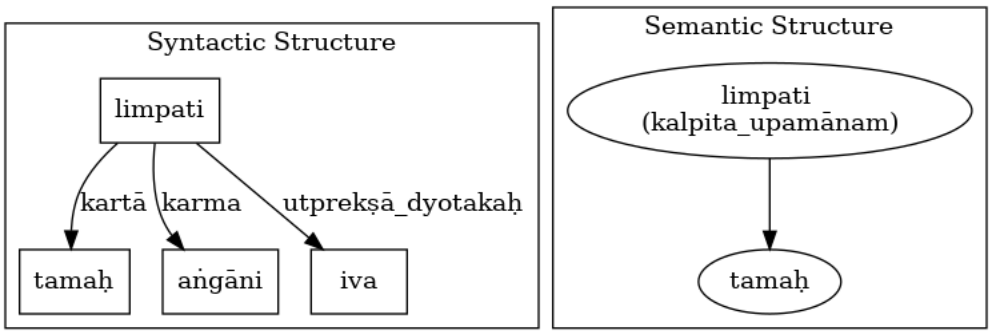


Figure 5.35: Representation of *Utprekṣā*

From a machine processing perspective, these constructions may appear structurally similar, since all contain a comparative marker, *upamāna* and *upameya*, but the underlying meaning and the interpretative function in each *alaṅkāra* are distinct.

5.5 Conclusion and future direction

Upamā is primarily found in all litarary texts to enhance the richness of the poetry. To analyse sentences carrying *Upamā*, the parser must be designed to recognise and appropriately tag these constructions.

In this chapter, we presented a tagging scheme to annotate the *Upamā* construction with different grammatical categories along with the special literary constructions and similar other *alaṅkāra*s. However, there are further challenges to make the parser capable of processing such complex constructions. Presented below is a list of challenges that are particularly difficult for a machine to tag at the initial stage of processing.

To improve the parser's coverage across a broader range of *Upamā* constructions, the following four areas of enhancement are proposed:

1. **Integration of a *taddhita, sanādi* and *kṛt* Module:** A dedicated module to analyze *taddhita*, *sanādi* and *kṛt* affixes and their semantic functions in conveying comparison.
2. **Enhancement of the *Samāsa* Module:** A robust *samāsa* analysis system to identify embedded relationships within compounds where explicit comparative markers are absent.
3. **Expansion of the Comparative Marker Lexicon:** An enriched list of comparative markers (*dyotaka*s) drawn from classical literature to cover a wider spectrum of linguistic realisations.
4. **Formulation of Special Case Rules:** Custom rules to address nuanced literary examples not fitting standard classifications, such as verses with only one comparative marker, or multiple *upamāna–upameya* pairs, so that the parser can handle such deviations gracefully.

By addressing the listed areas, the system can be better equipped to analyze the diverse and complex instantiations of *Upamā* found in Sanskrit literature.

Chapter 6

Conclusion

This study has attempted to approach Sanskrit poetry through the lens of computational analysis, with a focus on figurative language and poetic qualities. The primary aim was to explore the feasibility of identifying stylistic devices—Yamaka and Anuprāsa, along with the broader aesthetic markers defined by the concept of kāvyaguṇa. Rather than treating these features as isolated literary ornaments, the work has examined them as patterns that can be formalized, modeled, and analysed using linguistic and algorithmic methods.

The contributions of this research lie in bridging classical Sanskrit poetics with computational models, thereby introducing a novel direction in the field of Sanskrit figurative language processing. By combining traditional literary insights with contemporary techniques such as rule-based modeling, feature engineering, and machine learning classification, the study provides a framework for identifying complex poetic features at scale. The *Kāvyaguṇa* Classifier, developed as a part of this work, demonstrates how such models can be applied to classify verses having the properties of Mādhurya, Oja, and Prasāda, drawing on both statistical and stylistic cues. Alongside, the detection and classification models for *Yamaka* and *Anuprāsa* offer a structured approach to mapping sound-based ornamentation in verse. Parsing of *Upamā alaṅkāra* will help to improve the translation and understanding of sentences.

6.1 Limitations

The scope of this study is limited to selected Alaṅkāras—Yamaka, Anuprāsa, and Upamā—and three Kāvyaguṇas. The datasets, though diverse in textual sources, remain modest in size due to the availability of manually verifiable annotated material. The computational models, while effective for the chosen literary features, are tailored to Sanskrit and may require adaptation for other languages. Furthermore, the symmetry-based formalization is demonstrated for

specific Yamaka types and may not cover the entire spectrum of poetic variations found in the tradition.

6.2 Future work

To address these challenges, future research in Sanskrit FLP must focus on developing large-scale annotated corpora, creating transformer-based language models fine-tuned for Sanskrit, and integrating computational techniques for identifying *Rasa* and *Alaṅkāras*. Furthermore, advancements in prosody-aware NLP models could enhance the computational understanding of Sanskrit poetic structures. The integration of these approaches will be crucial in enabling more sophisticated analysis and generation of figurative language in Sanskrit texts.

In conclusion, this thesis presents an early but meaningful step towards the formalization of Sanskrit poetic analysis in computational terms. While much remains to be explored, the convergence of literary tradition and modern technology holds promise for new insights into classical texts, and for building tools that support digital scholarship in the Indian knowledge systems.

Bibliography

Ācārya Viśveśvara (2017). Kāvyaprakāśa. jñānamaṇḍala Limited, Varanasi.

Agrawal, P. and Madaan, V. (2020). A Sanskrit to Hindi language machine translator using rule based approach. In Proceedings of the 17th International Conference on Natural Language Processing (ICON): System Demonstrations, pages 13–15, Patna, India. NLP Association of India (NLPAI).

Ahmad, S., Zubair, M., Alotaibi, F., and Khan, S. (2020). Classification of poetry text into the emotional states using deep learning technique. IEEE Access, PP:1–1.

Audichya, M. K. and Saini, J. R. (2021). Towards Natural Language Processing with figures of speech in Hindi poetry. In International Journal of Advanced Computer Science and Applications.

Balasubramanyam, V. (2017). Citram - Poetry of Sound, volume 1. Rashtriya Sanskrit Sansktan, New Delhi.

Barbadikar, A. and Kulkarni, A. (2024a). Anuprāsa identifier and classifier: A computational tool to analyze Sanskrit figure of sound. In Bhattacharya, A., editor, Proceedings of the 7th International Sanskrit Computational Linguistics Symposium, pages 102–112, Auroville, Puducherry, India. Association for Computational Linguistics.

Barbadikar, A. and Kulkarni, A. (2024b). Yamaka identifier and classifier: A computational tool for the analysis of Sanskrit figure of sound (upcoming). Annals of Bhandarkar Oriental Research Institute, 102:29–45.

Bizzoni, Y. and Haider, T. (2023). Understanding poetry using natural language processing tools. Digital Scholarship in the Humanities, 39(2):500–517.

Bunker, P. and Jensen, P. (2004). Fundamentals of Molecular Symmetry.

Chakrabarty, T., Muresan, S., and Peng, N. (2021). It's not rocket science: Interpreting figurative language in narratives. arXiv preprint arXiv:2109.00087.

Chakrabarty, T., Muresan, S., and Peng, N. (2022). Flute: Figurative language understanding through textual explanations. arXiv preprint arXiv:2205.12404.

Chowdhary, S. (1965). Kāvyālaṅkāra. Vasudev Prakashan, Delhi.

Deshmukh, R., Kore, S., Chavan, N., Gole, S., and Adarsh, K. (2021). Marathi poem classification using machine learning. International Journal of Recent Technology and Engineering, 8.

Dummit, D. S. and Foote, R. M. (2004). Abstract Algebra. John Wiley and Sons, Inc, Hoboken, NJ, USA, third edition edition.

Durgaprasad and Parab, K. (1886). Kāvyālaṅkāraḥ. Nirnayasagar Press, Mumbai.

Englard, B. (2013). A rhetorical analysis approach to natural language processing. In ArXiv.

Gerow, E. (1971). A Glossary of Indian Figures of Speech. Mouton and Co. N. V. Publishers, Hague.

Ghosh, M. (1951). The Nāṭyaśāstra, volume 1. Asiatic Society of Bengal, Calcutta.

Goyal, P., Arora, V., and Behera, L. (2009). Analysis of sanskrit text: Parsing and semantic relations. In Huet, G., Kulkarni, A., and Scharf, P., editors, Sanskrit Computational Linguistics, pages 200–218, Berlin, Heidelberg. Springer Berlin Heidelberg.

Goyal, P. and Huet, G. (2013). Completeness analysis of a sanskrit reader. In Proceedings, 5th International Symposium on Sanskrit Computational Linguistics. DK Printworld (P) Ltd.

Hämäläinen, M. (2018). Poem machine - a co-creative NLG web application for poem writing. In Krahmer, E., Gatt, A., and Goudbeek, M., editors, Proceedings of the 11th International Conference on Natural Language Generation, pages 195–196, Tilburg University, The Netherlands. Association for Computational Linguistics.

Hattori, M. (1997). On the rhyme (yamaka) in Sanskrit poetics. Annals of the Bhandarkar Oriental Research Institute, 78(1/4):263–274.

Jadhav, B., Barbadikar, A., Kulkarni, A., and Kulkarni, M. (2023). Issues in the computational processing of upamāalaṅkāra. In D. Pawar, J. and Lalitha Devi, S., editors, Proceedings of the 20th International Conference on Natural Language Processing (ICON), pages 180–190, Goa University, Goa, India. NLP Association of India (NLPAI).

Jadhav, B., Dutta, H., Kanitkar, S., Kulkarni, M., and Bhattacharyya, P. (2025). An introduction to computational identification and classification of upamā alaṇkāra. In Kulkarni, A. and Hellwig, O., editors, Computational Sanskrit and Digital Humanities - World Sanskrit Conference 2025, pages 1–14, Kathmandu, Nepal. Association for Computational Lingustics.

Jurafsky, D. and Martin, J. H. (2009). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Pearson Prentice Hall, 2nd edition.

Kaur, J. and Saini, J. (1970). Designing punjabi poetry classifiers using machine learning and different textual features. The International Arab Journal of Information Technology (IAJIT), 17(01):38 – 44.

Krishnan, S., Satuluri, P., Barbadikar, A., Prasanna Venkatesh, T. S., and Kulkarni, A. (2025). Compound type identification in Sanskrit. In Kulkarni, A. and Hellwig, O., editors, Computational Sanskrit and Digital Humanities - World Sanskrit Conference 2025, pages 90–108, Kathmandu, Nepal. Association for Computational Lingustics.

Kulkarni, A. (2021). Sanskrit parsing following indian theories of verbal cognition. ACM Trans. Asian Low-Resour. Lang. Inf. Process., 20(2).

Kulkarni, A. and Das, M. (2012). Discourse analysis of Sanskrit texts. In Proceedings of the Workshop on Advances in Discourse Analysis and its Computational Aspects, pages 1–16, Mumbai, India. The COLING 2012 Organizing Committee.

Kulkarni, A. and Kumar, A. (2011). Statistical constituency parser for Sanskrit compounds. In ICON 2011.

Kulkarni, A. and Kumar, A. (2013). Clues from Aṣṭādhyāyī for compound type identification. In 5th international SCLS 2013.

Kulkarni, A. and Pai, M. (2019). Sanskrit sentence generator. In Proceedings of the 6th International Sanskrit Computational Linguistics Symposium, pages 1–13, IIT Kharagpur, India. Association for Computational Linguistics.

Kulkarni, A., Satuluri, P., Panchal, S., Maity, M., and Malvade, A. (2020). Dependency relations for Sanskrit parsing and treebank. In Proceedings of the 19th International Workshop on

Treebanks and Linguistic Theories, pages 135–150, Düsseldorf, Germany. Association for Computational Linguistics.

Kulkarni, A. and Shukl, D. (2009). Sanskrit morphological analyser: Some issues. In the Festscrift volume of Bh. Krishnamoorty, Indian Linguistics.

Kulkarni, R. M. and Kulkarni, M. A. (2025). Morph-dependency model for upamā. Language in India, 25(9):146–156.

Kumar, A. et al. (2023). A novel approach for identification of figurative language types in devanagari-scripted languages. International Journal of Advanced Computer Science and Applications, 14(8).

Lahiri, P. C. (1937). Concepts of Rīti and Guṇa in Sanskrit Poetics. Phd thesis, University of Dacca.

Lederman, L. and Hill, C. T. (2005). Symmetry and the Beautiful Universe. Prometheus Books, Amherst, NY.

Liu, H., Chakrabarty, T., and Peng, N. (2022). Testing the ability of language models to interpret figurative language. arXiv preprint arXiv:2204.12632.

McManus, I. C. (2005). Symmetry and aymmetry in aesthetics and the arts. European Review, 13(2):157–180.

Melnad, K., Scharf, P., and Goyal, P. (2015). Meter identification of Sanskrit verse. In Sanskrit Syntax: Selected Papers Presented at the Seminar on Sanskrit Syntax and Discourse Structures.

Mishra, D. K. N. (2006). Sarasvatī-kaṇṭhābharaṇam by Bhoja. Chaukhambha Publishers.

Mishra, R., editor (2012). Vakrokti-jīvita. Chowkhamba Sanskrit Series, Benares.

Mishra, R. et al. (2023). Utilizing computational linguistics tools for enhanced poetic interpretation. ResearchGate.

Mishra, S. (1996). Kāvyādarśa. Chowkhamba Vidyabhavan, Varanasi.

Molnar, V. and Molnar, F. (1986). Symmetry making and breaking in visual art. Computers and Mathematics with Applications, 12B(1/2):291–301.

Naaz, K. and Singh, N. K. (2022). Design and development of computational tools for analyzing elements of hindi poetry. In IEEE Access.

Neil, T. (2023). Skrutable: Another step toward effective Sanskrit meter identification. In Proceedings of the Computational Sanskrit & Digital Humanities: Selected papers presented at the 18th World Sanskrit Conference.

Pavlovic, B. and Triajstic, N. (1986). On symmetry and asymmetry in literature. Computers and Mathematics with Applications, 12B(1/2):197–227.

Qadir, A., Riloff, E., and Walker, M. A. (2016). Automatically inferring implicit properties in similes. In Knight, K., Nenkova, A., and Rambow, O., editors, Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1223–1232, San Diego, California. Association for Computational Linguistics.

Rajagopalan, S. (2018). A user-friendly tool for metrical analysis of Sanskrit verse. In Computational Sanskrit & Digital Humanities, Selected papers presented at the 17th World Sanskrit Conference.

S, P. (2022). Spellchecker for Sanskrit:the road less taken. In Akhtar, M. S. and Chakraborty, T., editors, Proceedings of the 19th International Conference on Natural Language Processing (ICON), pages 290–299, New Delhi, India. Association for Computational Linguistics.

Sanchez-Behar, A. (2014). Symmetry in the music of john adams. Tempo, 68 (268):46–60.

Sastri, C. S. R. (1956). Kāvyālaṅkāra of Bhāmaha. The Sri Balamanorama Press, Mylapore, Madras.

Satuluri, P. and Kulkarni, A. (2013). Generation of sanskrit compounds. In ICON 2013.

Shastri, P. R. R. (1938). Kāvyādarśa. Bhandarkar Oriental Research Institute, Pune.

Shrikrishnamoori (1909). Kāvyālaṅkārasūtravṛttiḥ. Sri Vani Vilas Press, Srirangam.

Shutova, E. V. (2011). Computational approaches to figurative language. Phd thesis, University of Cambridge.

Sihna, D. K. (1989). Symmetry in the Indian Context: A critique. Hungary.

Singh, N. K., Naaz, K., and Chakraborty, S. (2023a). Komala and kathora: A novel approach towards classification of hindi poetry. ACM Trans. Asian Low-Resour. Lang. Inf. Process., 22(6).

Singh, N. K., Naaz, K., and Chakraborty, S. (2023b). Komala and kaundefinedhora: A novel approach towards classification of hindi poetry. ACM Trans. Asian Low-Resour. Lang. Inf. Process., 22(6).

Sriram, K., Kulkarni, A., and Huet, G. (2023). Validation and normalization of DCS corpus and development of the Sanskrit heritage engine's segmenter. In Proceedings of the Computational Sanskrit & Digital Humanities: Selected papers presented at the 18th World Sanskrit Conference, pages 38–58, Canberra, Australia (Online mode). Association for Computational Linguistics.

Söhnen, R. (1995). On the concept and presentation of "yamaka" in early indian poetic theory. Bulletin of the School of Oriental and African Studies, 58(3):495–520.

Terdalkar, H. and Bhattacharya, A. (2023). Chandojnanam: A Sanskrit meter identification and utilization system. In Proceedings of the Computational Sanskrit & Digital Humanities: Selected papers presented at the 18th World Sanskrit Conference.

Tripathi, R. (1972a). Sanskṛta Sāhitya me śabdālaṅkāra. Shri Lal Bahadur Shastri Kendriya Sanskrit Vidyapeeth, Delhi.

Tripathi, R. (1972b). Saṁskṛta-Sāhitya meṁ Śabdālaṅkāra. Number Navama Puṣpa in Saṁskṛta-Vidyāpīṭha-Granthamālā. Śrī Lāla Bahādura Śāstrī Kendrīya Saṁskṛta Vidyāpīṭha, Dilli. Foreword by Śrī Devanandana Prasāda Yādava.

Veale, T. and Hao, Y. (2018). Proceedings of the workshop on figurative language processing. In ACL Anthology.

Vrajaratnadas. Kāvyādarśa. Vrajaratnadas, Shrikamalamani Granthamaka Karyalaya, Kashi.

Wang, C. (2022). Analysis of poetry style based on text classification algorithm. Scientific Programming, 2022.

Wujastyk, D. (1978). Automatic scansion of sanskrit poetry for authorship criteria. Bulletin - Association of Literary and Linguistic Computing, 6(2):122–135.

Sl No. 40441

Notification No. 1

हैदराबाद विश्वविद्यालय

UNIVERSITY OF HYDERABAD

(A Central University Established by an Act of Parliament)

P.O. CENTRAL UNIVERSITY, GACHIBOWLI, HYDERABAD - 500 046 (INDIA)

REGULAR EXAMINATION

SEMESTER GRADE TRANSCRIPT

REG. NO. **20HSPH02**

NAME OF THE STUDENT **BARBADIKAR AMRUTA VILAS**

MONTH AND YEAR OF **FEB 2021** **SEMESTER 1**

COURSE **Ph.D. SANSKRIT STUDIES**

PARENT'S NAME **VILAS BARBADIKAR / SHRADDHA BARBADIKAR**

COURSE NO	TITLE OF THE COURSE	CREDITS	RESULTS
SK801	RESEARCH METHODOLOGY	4	PASS
SK812	TOPICS IN NATURAL LANGUAGE PROCESSING	4	PASS

SEMESTER GRADE POINT AVERAGE (SGPA) :**10.0**

(In words) :TEN POINT ZERO

DEPUTY REGISTRAR
(ACAD & EXAMS)

Date Of Result Notification : **Mar 10, 2021**

Date Of Download : **Oct 23, 2025**

This is system generated certificate issued by O/o The Controller of Examinations, University of Hyderabad

Sl No. 47830

Notification No. 1

हैदराबाद विश्वविद्यालय

UNIVERSITY OF HYDERABAD

(A Central University Established by an Act of Parliament)

P.O. CENTRAL UNIVERSITY, GACHIBOWLI, HYDERABAD - 500 046 (INDIA)

REGULAR EXAMINATION

SEMESTER GRADE TRANSCRIPT

REG. NO. **20HSPH02**

NAME OF THE STUDENT **BARBADIKAR AMRUTA VILAS**

MONTH AND YEAR OF **JUL 2021** **SEMESTER 2**

COURSE **Ph.D. SANSKRIT STUDIES**

PARENT'S NAME **VILAS BARBADIKAR / SHRADDHA BARBADIKAR**

COURSE NO	TITLE OF THE COURSE	CREDITS	RESULTS
EN875	RESEARCH AND PUBLICATION ETHICS	2	PASS
SK830	DISSERTATION RELATED READINGS	4	PASS

SEMESTER GRADE POINT AVERAGE (SGPA) :**10.0**

(In words) :TEN POINT ZERO

DEPUTY REGISTRAR
(ACAD & EXAMS)

Date Of Result Notification : **Aug 5, 2021**

Date Of Download : **Aug 15, 2025**

This is system generated certificate issued by O/o The Controller of Examinations, University of Hyderabad

Gmail **Amruta Barbadikar <amruta.barbadikar@gmail.com>**

Fwd: Your Submission IJHS-D-23-00135R1 - [EMID:13739683e55e797d]

Amba Kulkarni <ambapradeep@gmail.com> Tue, Oct 1, 2024 at 4:49 PM
To: Amruta Barbadikar <amruta.barbadikar@gmail.com>, Bharati Bhat <bharatimundgodi@gmail.com>

Finally, our manuscript is accepted for publication!

-- Amba

---------- Forwarded message ---------
From: **Indian Journal of History of Science Editorial Office** <em@editorialmanager.com>
Date: Tue, 1 Oct 2024 at 16:00
Subject: Your Submission IJHS-D-23-00135R1 - [EMID:13739683e55e797d]
To: Amba Kulkarni <ambapradeep@gmail.com>

Dear Dr Kulkarni,

We are pleased to inform you that your manuscript, "Symmetry in Bharata's Yamaka: A Mathematical Analysis", has been accepted for publication in
Indian Journal of History of Science.

You will receive an e-mail in due course regarding the production process.

Please remember to quote the manuscript number, IJHS-D-23-00135R1, whenever inquiring about your manuscript.

With kind regards,
Madhvendra Narayan
Associate Editor
Indian Journal of History of Science

Comments to the author (if any):
Reviewer #2: In this manuscript, the authors draw intriguing parallels between Bharata's yamaka classification and symmetry groups. The manuscript is well-written, making it accessible to both mathematicians and non-mathematicians. The authors have also adequately addressed the concerns raised by the other reviewer.

While Bharata recognized and formalized several symmetries in the yamaka class of prosody, he did not abstract these into a general theory, as seen in the Chandaḥśāstram. However, this analysis offers a novel perspective or linguistic tool for exploring literature.

This letter contains confidential information, is for your own use, and should not be forwarded to third parties.

Recipients of this email are registered users within the Editorial Manager database for this journal. We will keep your information on file to use in the process of submitting, evaluating and publishing a manuscript. For more information on how we use your personal details please see our privacy policy at https://www.springernature.com/production-privacy-policy. If you no longer wish to receive messages from this journal or you have questions regarding database management, please contact the Publication Office at the link below.

_____
In compliance with data protection regulations, you may request that we remove your personal registration details at any time. (Use the following URL: https://www.editorialmanager.com/ijhs/login.asp?a=r). Please contact the publication office if you have any questions.

Indian Journal of History of Science (2024) 59:341–350
https://doi.org/10.1007/s43539-024-00140-5

ARTICLE

Symmetry in Bharata's *Yamaka*: A mathematical analysis

Amba Kulkarni[1] · **Amruta Barbadikar**[1] · **Bharati Bhat**[1]

Received: 7 November 2023 / Accepted: 1 October 2024 / Published online: 20 January 2025
© Indian National Science Academy 2025

Abstract
Indian literature exhibits symmetry in abundance. The *śabdālaṅkāra*s (figure of sound) such as *Yamaka* present examples of symmetry in literature. The first classification of *Yamaka* is given by Bharata in his treatise *Nāṭyaśāstra*. It is interesting to note that all the patterns of *Yamaka* mentioned by Bharata are not only simple but also exhibit symmetry under the operations of reflection and rotation. In this paper, we show the simplicity of the classification of *Yamaka* by Bharata by proving that the patterns corresponding to each type of *Yamaka* form a group isomorphic to either the Klein four-group $\boldsymbol{K_4}$ or the groups $\mathbb{Z}_2$ or $\mathbb{Z}_1$.

Keywords Symmetry · Symmetry group · *Yamaka alaṅkāra* · Klein 4 group

1 Introduction

The word symmetry (originated from the Greek word symmetria) refers to the arrangement in dimensions or the due proportion. It refers to certain compositions of structural objects that preserve the structure. It acts as a measure of the beauty of the form or structure. Therefore it has always been considered as an important factor in visual aesthetics (Molnar & Molnar, 1986). Symmetry is abundant in nature. Several studies discuss various kinds of symmetry in living and non-living objects such as plants, living organisms, the structure of chemical molecules and organic compounds, etc. (Lederman & Hill, 2005; Bunker & Jensen, 2004). The symmetry in arts, architecture, music, and literature are also studied in great detail (Pavlovic & Triajstic, 1986).

Indian civilisation is no exception. According to Sihna (1989), 'Indian ethos has shaped through the centuries in evolving the facets of symmetry. Various shapes of Vedic altars constructed using the *Śulbasūtr* as are all symmetric.' Some of the finest examples of symmetry are the fractal symmetry found in the temples and wells across India. Indian literature also exhibits symmetry in abundance. The *śabdālaṅkāra*s such as *Yamaka*, and various *citrabandha kāvya*s (poetries composed in the form of drawings) present examples of symmetry in literature.

Yamaka is used to enrich the beauty of the poetry. *Yamaka* is a figure of sound where a group of syllabic patterns is repeated typically with a different meaning at different locations in the same verse. The first classification of *Yamaka* is given by Bharata in his treatise *Nāṭyaśāstra*. Later Daṇḍi, Rudraṭa, and many other scholars gave various classifications covering different patterns of repetitions. There is an elegance and simplicity in Bharata's classification.[1] It is interesting to note that all the types of *Yamaka* mentioned by Bharata are not only simple but also exhibit symmetry under the operations of reflection and rotation. In this paper, we show the simplicity of the classification of *Yamaka* by Bharata by proving that each type forms a group isomorphic to either the Klein four-group K_4 or the groups $\mathbb{Z}_2$, or $\mathbb{Z}_1$.

2 *Yamaka*

The embellishment in the poetry makes it different from the language which is used for regular communication. It enhances the beauty of the poetry. The systematic study of this ornamentation is known as the theory of *alaṅkāra*. It is one of the important disciplines in the *Kāvyaśāstra* tradition. Scholars of the *Alaṅkāra* school regard the poetic

[1] Bharata does not insist on different meanings.

✉ Amba Kulkarni
ambakulkarni@uohyd.ac.in

Amruta Barbadikar
amruta.barbadikar@gmail.com

Bharati Bhat
bharatimundgodi@gmail.com

[1] Department of Sanskrit Studies, University of Hyderabad, Hyderabad 500046, India

Subject: Annals, Vol. 102: Certificate of Publication
From: Bhandarkar Institute <annals.bori@gmail.com>
Date: 12/11/24, 3:32 pm
To: Amruta Barbadikar <amruta.barbadikar@gmail.com>, Amba Kulkarni <ambapradeep@gmail.com>

To
Smt. Amruta Barbadikar and Dr. Amba Kulkarni

Namaskaar.

Please find attached the certificate of publication of your research paper entitled '*Yamaka Identifier and Classifier: A Computational Tool for the Analysis of Sanskrit Figure of Sound*', that has been published in the 102nd volume of our research journal *Annals of the Bhandarkar Oriental Research Institute* (ISSN: 0378-1143) at pages 29-45.

The said 102nd volume of the *Annals* was released on July 06, 2024.

Shreenand L. Bapat
Registrar and Curator, BORI
and **Managing Editor, ABORI**

Certificate-of-Publication=Barbadikar-Kulkarni.jpg

Bhandarkar Oriental Research Institute

812, Shivajinagar, Chiplunkar Road(ILS Law College Road), Pune - 411004(India)
Email: info@boriindia.org Website: www.bori.ac.in

Outward No. 218/2024-25 Date: 12/11/2024

CERTIFICATE OF PUBLICATION

This is to certify that a research paper entitled '*Yamaka Identifier and Classifier: A Computational Tool for the Analysis of Sanskrit Figure of Sound*', by Smt. Amruta Barbadikar and Dr. Amba Kulkarni, has been published in the 102nd volume of our research journal *Annals of the Bhandarkar Oriental Research Institute* (ISSN: 0378-1143) at pages 29-45.

The said 102nd volume of the *Annals* was released on July 06, 2024.

BHANDARKAR O. R. INSTITUTE PUNE 411004

(Shreenand L. Bapat)
Registrar and Curator, BORI
and **Managing Editor, ABORI**

Contact: +91-20-25656932 & +91-7796699822 (Weekdays: 09:30 am to 05:30 pm, Sat. 09:30 am to 01:00 pm)

Attachments:

Certificate-of-Publication=Barbadikar-Kulkarni.jpg 417 KB

YAMAKA IDENTIFIER AND CLASSIFIER: A COMPUTATIONAL TOOL FOR THE ANALYSIS OF SANSKRIT FIGURE OF SOUND

By

Amruta Barbadikar and Amba Kulkarni

The science dealing with various aspects of the beauty of a literary text is termed as Alaṅkāraśāstra in Sanskrit. The literal meaning of 'Alaṅkāra' is an ornament or a decoration. In the context of literature, these ornaments are either related to the sounds of the phonemes in the words or to the meanings associated with the words or to both. The beautiful rhythmic patterns of sound are termed as Śabdālaṅkāras (ornaments of sound) and the patterns that give rise to pleasure due to the meanings of the words are called Arthālaṅkāras. There are constructions with rhythmic sound patterns that are also meaningful. These fall under the category of Ubhayālaṅkāras. Śabdālaṅkāras being governed by the rhythmic, repetitive sound patterns, are very easy to capture and can be identified with the pattern matching algorithms.

Repeated use of the same sequence of letters is one of the techniques that poets use for enhancing the beauty and charm of poetry. Anuprāsa (alliteration) and Yamaka are two popular śabdālaṅkāras. Both use repetitive patterns. Among these two, in alliteration typically a single letter or a syllable is repeated. For example, in the following phrase, the syllable 'ra' is repeated and so, it is an example of Anuprāsa.

'rāmo rājamaṇiḥ sadā vijayate
rāmaṁ rameśaṁ bhaje |'

In a variation of Anuprāsa called Lāṭānuprāsa, one or more nominal stems are repeated. The nominal stem may occur with different case markers.

Yamaka also shows the repetition of sound patterns. But these patterns are larger sequences of syllables, typically consisting of at least two syllables. For example, "samarasamaraso 'yam" Here, the sequence 'samara' is repeated. The first occurrence is the left part of a compound and the second occurrence is a part of the word 'samarasaḥ.'

Gmail **Amruta Barbadikar <amruta.barbadikar@gmail.com>**

Your ICON 2024 Submission (Number 55)

Committee, ICON 2024 <papers_icon2024@softconf.com> Tue, Dec 3, 2024 at 11:44 AM
Reply-To: papers_icon2024@softconf.com
To: amruta.barbadikar@gmail.com

Dear Amruta Barbadikar:

On behalf of the ICON 2024 Program Committee, I am delighted to inform you that the following submission has been accepted to appear at the conference:

Automatic Sanskrit Poetry Classification Based on KÄ vyaguá¹‡a

The Program Committee worked very hard to thoroughly review all the submitted papers. Please repay their efforts, by following their suggestions when you revise your paper.

When you are finished, you can upload your final manuscript at the following site:

https://softconf.com/icon2024/papers/

You will be prompted to login to your START account. If you do not see your submission, you can access it with the following passcode:

55X-A8C7J5H2A8

Alternatively, you can click on the following URL, which will take you directly to a form to submit your final paper (after logging into your account):

https://softconf.com/icon2024/papers/user/scmd.cgi?scmd=aLogin&passcode=55X-A8C7J5H2A8

The reviews and comments are attached below. Again, try to follow their advice when you revise your paper.

Congratulations on your fine work. If you have any additional questions, please feel free to get in touch.

Best Regards,
Program Committee, ICON 2024
ICON 2024

```
============================================================================
ICON 2024 Reviews for Submission #55
============================================================================

Title: Automatic Sanskrit Poetry Classification Based on KÄ vyaguá¹‡a
Authors: Amruta Barbadikar and Amba Kulkarni


============================================================================
                            REVIEWER #1
============================================================================

What is this paper about, and what contributions does it make?
----------------------------------------------------------------------------
This paper presents the Kavyaguna Classifier, a machine learning module, designed to classify
Sanskrit literary texts into three distinct gunas, by employing a diverse range of machine
learning algorithms, including Random Forest, Gradient Boosting, XGBoost, MultiLayer
Perceptron and Support Vector Machine.
It classifies poetry based on various features such as Lucidity, Floridity and sweetness
known as the Kavyagunas.
----------------------------------------------------------------------------


What strengths does this paper have?
----------------------------------------------------------------------------
Automatic classification and identification of three kavyagunas.
----------------------------------------------------------------------------
```

What weaknesses does this paper have?

Limited work

Questions for the Author(s)

Typo: * is endowed wtih the qalities of ... => is endowed with the qualities of

Question: In Table 6: Gradient Boosting: Confusion Matrix, the authors say that for Prasada, 39 instances were correctly classified with no misclassifications.
My question is: aren't there 04 misclassifications out of the total 43 verses for Prasada?

==========
REVIEWER #2
==========

What is this paper about, and what contributions does it make?

This paper proposes an automated way of identifying and classifying Kavyaguna in Sanskrit poetry.

What strengths does this paper have?

S1. The problem statement is interesting and difficult.

S2. The description about what Kavyaguna are and what its different variants are, is very good.

S3. The results are good.

What weaknesses does this paper have?

W1. What are the features for Prasada?

W2. Putting multiple confusion matrices for multiple methods is unnecessary. Even more unnecessary is repeating its values in the text. This space should be utilized for something more fruitful.

W3. For example, it needs to be analysed and understood why there is no confusion between Madhurya and Prasada? I would have expected the models to do very well on Oja versus Madhurya since they seem to be complementary. Please explain this.

W4. Similarly, please show some anecdotal examples of poetry that is correctly classified, but more importantly, poetry that is wrongly classified. Please then analyse its features and why they are wrongly classified. At least one such example would vastly improve the value of the paper.

W5. Typos

(a) Sec 3 title

(b) No bold in Padasaukumarya

(c) word2vect --> word2vec

Questions for the Author(s)

W1 - W4

Automatic Sanskrit Poetry Classification Based on Kāvyaguṇa

Amruta Barbadikar and **Amba Kulkarni**
amruta.barbadikar@gmail.com, ambakulkarni@uohyd.ac.in
Department of Sanskrit Studies,
University of Hyderabad.

Abstract

Kāvyaguṇa denotes the syntactic and phonetic attributes or qualities of Sanskrit poetry that enhance its artistic appeal, commonly classified into three categories: Mādhyurya (Sweetness), Oja (Floridity), and Prasāda (Lucidity). This paper presents the Kāvyaguṇa Classifier, a machine learning module, designed to classify Sanskrit literary texts into three distinct guṇas, by employing a diverse range of machine learning algorithms, including Random Forest, Gradient Boosting, XGBoost, Multi-Layer Perceptron and Support Vector Machine. For vectorization, we employed two methods: the neural network-based `Word2vec` and a custom feature engineering approach grounded in the theoretical understanding of Kāvyaguṇas as described in Sanskrit poetics. The feature engineering model significantly outperformed, achieving an accuracy of up to 90.6% in K-fold cross-validation and 92% in Holdout validation.

1 Introduction

Natural language processing (NLP) involves the intersection of computer science, artificial intelligence, and computational linguistics, focusing on how computers can understand, interpret, and respond to human languages (Jurafsky and Martin, 2009). NLP is developing vastly for different languages with data-driven approaches. However, Low Resource Languages face numerous challenges in NLP, including data scarcity, complex morphology, orthographic and historical variations, and a lack of digital resources. These issues, combined with the inadequacy of models designed for other languages, hinder the development of robust NLP solutions for Sanskrit. Although Sanskrit has traditionally been categorized as a Low-resource language, its NLP development has gained significant momentum in recent years. This progress can be attributed to the well-defined linguistic theories and the association of modern computational approaches. Several computer applications are available for the analysis of the syntactic components of the language (Goyal and Huet, 2013; Satuluri and Kulkarni, 2013; Kulkarni and Shukl, 2009; Kulkarni and Kumar, 2013, 2011; Goyal et al., 2009; Kulkarni et al., 2020; Kulkarni and Das, 2012; S, 2022).

The poetic literature of a language reflects the higher intellect of society. Sanskrit literature, dating back to the Vedic era, has fascinated humanity with its sophisticated aesthetic dimensions. Classical Sanskrit literature is rich in examples of such poetic expression. Understanding and processing this literature require a high level of intellect. ĀNANDAVARDHANA emphasizes the importance of deeper understanding, stating, "Simply knowing the words and their meanings does not enable one to fully appreciate poetry; true enjoyment comes to those experts who grasp its deeper essence."[1] For machines lacking world knowledge, this becomes a complex task. Without advanced methods for the automatic interpretation of implied meanings, machines struggle to process the ornate language accurately.

While some progress has been made in the processing of literary regard, machines can more easily analyze aspects where meaning is not deeply embedded. For example, tasks such as meter identification (Rajagopalan, 2018; Melnad et al., 2015; Neil, 2023; Terdalkar and Bhattacharya, 2023), and the identification and analysis of sound figures like Yamaka (Barbadikar and Kulkarni, 2023) and Anuprāsa (Barbadikar and Kulkarni, 2024) have been successfully achieved.

Poetry classification is performed using various standards such as sentiment, poetic style, etc. Wujastyk (1978) worked out the classification for

[1] śabdārthaśāsanajñānamātreṇaiva na vedyate |
vedyate sa tu kāvyārthatattvajñaireva kevalam ‖ 1.7, Dhvanyāloka

View Reviews

Paper ID
7

Paper Title
Anuprāsa Identifier and Classifier: A computational tool to analyze Sanskrit figure of sound

Track Name
ISCLS2024

Reviewer #3

Questions

1. Summary of the paper
The authors discuss a tool useful for the identification and classification of the Anuprāsa alaṅkāra.

2. Strengths of the paper
Well written. Well presented. Unexplored area in Sanskrit Computational linguistics.

3. Weaknesses of the paper
No.

4. Details
1. Anuprāsa is repletion of sounds not letters. Please check again use of the word "letter" in the paper.
2. Add dates of Bhāmaha, Daṇḍin, and Jayadeva. It is very useful to get a brief history of development of Anuprāsa.
3. Please use italics for the all the Sanskrit words rather than capitalizing the first letter of a Sanskrit word or putting inverted commas around the word.
4. What is "Ii" on page 4 in the point 2 Vṛttyanuprāsa?
5. Please follow proper style in the bibliography. Your titles are not italicized in the references.

5. Decision
Accept

Reviewer #4

Questions

1. Summary of the paper
The paper proposes an algorithm and a tool to detect Anuprasa alankara in Sanskrit text.

2. Strengths of the paper

S1. The problem is interesting.
S2. The algorithm is designed with rigour.
S3. An interface is provided.
S4. Alankara is explained in detail.

3. Weaknesses of the paper

W1. Examples of different alankaras should be given when they are mentioned or explained.
In particular, examples of yamaka, and different types of anuprasa such as gramyanuprasa, etc. are important.
W2. Why is only Visvanatha's classification of Anuprasa followed?
W3. What are the results of testing? Were all alankaras identified correctly?
W4. Is the algorithm a heuristic, or is it guaranteed to produce a correct answer always?

4. Details

Typo: Dandn -> Dandin

4-5 aksharas may not always be 8-10 letters since conjunct characters are quite common in Sanskrit

5. Decision

Neutral

Reviewer #5

Questions

1. Summary of the paper

This paper gives a classification of poetic figures of style based on sound repetitions, according the various historical definitions. It gives an analysis of their respective generality, and deduces an algorithm for the corresponding classification of poetry. The algorithm has been implemented, tested and made publicly available with a Web user interface.

2. Strengths of the paper

Very sound and innovative work, on an important emerging topic in poetry evaluation according to tradition

3. Weaknesses of the paper

None

4. Details

Typos and style recommendations :
- makes the use of semantics the least => makes the least use of semantics
- The tradition of Alaṅkāraśastra (poetics) is developed over the time period => use

« some « or « a long », but if you maintain « the », state which.
- footnote 6: G=> g
- in the text, anusvaram is indicated in transliteration as m with a dot diacritic, sometimes above, eg « aṁka » sometimes below eg « (ṃ). » A uniform notation should be used.
- Daṇḍn => Daṇḍin
- nutrifies => nourishes ?
- posses => possesses
- twice occurrence => double occurrence
- below given example => example given below
- at the end of the feet => at the end of the pada (?)
- « is > 2 » is wrong technical writing, use « is greater than 2 »
- Since, the => Since the
- « The nasal stops viz. ṅ, ñ, ṇ, n and m can be written as an anusvāra (ṃ) when it is followed by a homorganic stop. » is wrong for eg saṅga and other derivatives of sañj, and also aṅka, so it should be the other way around.
- Various examples are given without precise quotation of the work in which they appear. If you have such quotations, it may be interesting to give them (I know it is not easy, with recensions, etc.)

- Bibliography. A classical manual on poetics is Kāvyamīmāṃsā by Rājaśekhara (10th century). Although it is an introductory manual for beginners, it might be interesting to check whether its terminology and examples are consistent with the cited works.

5. Decision
Accept

Anuprāsa Identifier and Classifier: A computational tool to analyze Sanskrit figure of sound

Amruta Barbadikar
Department of Sanskrit Studies,
University of Hyderabad
amruta.barbadikar@gmail.com

Amba Kulkarni
Department of Sanskrit Studies,
University of Hyderabad
ambakulkarni@uohyd.ac.in

Abstract

Anuprāsa is a *śabdālaṅkāra* (figure of sound), in which the poetry is embellished by the repetitive occurrence of letters.[1] The task of processing the decorative language consisting of such figures is a path not explored in the field of Sanskrit computational linguistics. This paper discusses a tool that identifies and classifies *anuprāsa alaṅkāra*. *Anuprāsa*, being a figure of sound, makes the least use of semantics. This tool is essentially developed upon the insights taken from the school of *alaṅkāra*s, especially from the treatise of ĀCĀRYA VIŚVANĀTHA from 14th century AD.

1 Introduction

There is a varied scope for research in the field of Sanskrit computational linguistics. Segmentation(Goyal and Huet, 2013), Morph-analysis and Generation (Kulkarni and Shukl, 2009), Compound type analysis (Kulkarni and Kumar, 2013; Kulkarni and Kumar, 2011) and Generation (Satuluri and Kulkarni, 2013), Sentence analysis (Goyal et al., 2009; Kulkarni et al., 2020) and Generation (Kulkarni and Pai, 2019), Discourse analysis (Kulkarni and Das, 2012), Translation (Agrawal and Madaan, 2020), etc. tasks are being carried out with extensive efforts. Most of the tasks are grammar oriented. Other works which attempt to process the rhetoric of the poetic language are very limited. Except for the Meter identification (Melnad et al., 2015; Rajagopalan, 2018; Neil, 2023; Terdalkar and Bhattacharya, 2023) and a tool to identify and classify the *yamaka alaṅkāra* (Barbadikar and Kulkarni, 2023), the Natural Language Processing (NLP) tools are far away from processing poetic beauty in Sanskrit.

Although, some examples in languages other than Sanskrit for processing figurative language can be found. Shutova (2011) presented a computational approach to process metaphor using statistical methods. Englard (2013) used the rhetorical analysis of text to predict the author. For Hindi, Audichya and Saini (2021), worked out the *alaṅkāra*s in Hindi to present the hierarchical structure with a taxonomical listing of *alaṅkāra*s. However, the computational implementation was not exercised. Naaz and Singh (2022) were able to contribute by presenting three different tools for Hindi. 'Text2Mātrā' produces the *laghu* and *guru mātras* for the input, 'RPaGen' detects the rhyming quality of the poem and 'FoSCal' generates a score according to the quantity of *anuprāsa* used over the poem. For Sanskrit, the automatic Meter Identification task has been worked out, from various applications perspectives, by different scholars. Melnad et al. (2015), Rajagopalan (2018), Neil (2023), Terdalkar and Bhattacharya (2023) are some of the notable contributors to the available state-of-the-art Sanskrit Meter Identification systems.

Highly complex language structures, use of intended and implicit meaning, multiple meanings of a word, multiple words having a similar meaning, and unavailability of useful state-of-the-art tools are the factors that discourage the processing of decorative language used in poetry. The tradition of *alaṅkāraśastra* (poetics) is developed over a long period ranging from the 1st century AD. The study of figures of speech is an important stream of this tradition. Figures of speech

[1]Here, the repetition of sounds is desirable. In Sanskrit, there is one to one mapping of sound with the denoting letter and we compute letters, not sounds. Hence we use the terms letters and sounds interchangeably.

Gmail **Amruta Barbadikar <amruta.barbadikar@gmail.com>**

Fwd: Your ICON 2023 Submission (Number 88): Accepted for Oral Presentation (Long Paper)

Bhakti Jadhav <bhaktij96@gmail.com> Sat, Dec 2, 2023 at 10:33 AM
To: Amba Kulkarni <ambapradeep@gmail.com>, Malhar Kulkarni <malharku@gmail.com>
Cc: Amruta Barbadikar <amruta.barbadikar@gmail.com>

Dear sir and madam,

We are delighted to convey you that our paper has been selected for ICON 23. Thank you for all your guidance and support. It is only because of your guidance that we could do this. I request you to kindly find attached the reviews on our paper, so that we modify accordingly.

---------- Forwarded message ---------
From: **Program Chairs, ICON 2023** <papers_icon2023@softconf.com>
Date: Sat, Dec 2, 2023 at 10:17 AM
Subject: Your ICON 2023 Submission (Number 88): Accepted for Oral Presentation (Long Paper)
To: <bhaktij96@gmail.com>

Dear Bhakti Jadhav:

On behalf of the ICON 2023 Program Committee, I am delighted to inform you that the following submission has been accepted as a Paper to appear at the conference:

Issues in the computational processing of UpamÄ alaá¹…kÄ ra.

The Program Committee worked very hard to thoroughly review all the submitted papers. Please repay their efforts, by following their suggestions when you revise your paper.

Please submit the final version of your Paper by Thursday, 7th December 2023 (11:59 PM IST) in compliance with the stylesheet of the Conference.

Guidelines for Camera-ready Submission

---

1. Authors must follow the ACL Author Guidelines

2. Authors must download the ACL 2023 style sheets from ICON 2023 main page and add \aclfinalcopy in the .tex file.

3. The deadline for submission of the camera-ready paper is Thursday, 7th December 2023 (11:59 PM IST).

4. The final submission site is available at https://softconf.com/icon2023/papers/

When you are finished, you can upload your final manuscript at the following site:

https://softconf.com/icon2023/papers/

You will be prompted to log in to your START account and thus access to your User Console: from here choose "Your current Submission(s)" for accessing the list of your submission(s). Click on the submission id for submitting the final paper.

If you do not see your submission, you can access it with the following passcode:

88X-G8E6E5P9H3

Alternatively, you can click on the following URL, which will take you directly to a form to submit your final paper (after logging into your account):

https://softconf.com/icon2023/papers/user/scmd.cgi?scmd=aLogin&passcode=88X-G8E6E5P9H3

The reviews and comments are attached below. Again, try to follow their advice when you revise your paper.

IMPORTANT INFORMATION

1. The time duration (tentative) for accepted short paper presentations is as follows: 10 minutes presentation + 3 minutes QA

2. For each accepted paper, at least one author of the paper must register for the conference.
For more information about registration, please visit: ICON 2023 main page

3. The early registration period for this paper ends December 10, 2023 [11:59 PM IST (Anywhere on Earth) Time Zone (UTC+5:30)]

Congratulations on your fine work. If you have any additional questions, please feel free to get in touch.

Thanks and Best Regards,
Prof. Jyoti D. Pawar and Prof. Shobha L.
PC Chairs, ICON 2023

```
==========================================================================
ICON 2023 Reviews for Submission #88
==========================================================================

Title: Issues in the computational processing of UpamÄ  alaá¹…kÄ ra.
Authors: Bhakti Jadhav, Amruta Barbadikar, Amba Kulkarni and malhar kulkarni


==========================================================================
                            REVIEWER #1
==========================================================================

What is this paper about, and what contributions does it make?
--------------------------------------------------------------------------
The paper focuses on the computational processing and understanding of figurative speech,
specifically addressing the UpamÄ  alaá¹…kÄ ra, which is a type of simile in classical Indian
literary theory.

The paper presents a methodology for representing the verbal cognition of UpamÄ  alaá¹…kÄ ra
by humans as a dependency tree. This tree involves the identification and analysis of various
components, including the vehicle, topic, common property, and the word indicating
similitude.

The work done provides the ability to represent the dependency tree without any loss of
information and without the repetition of elliptical elements. This representation can be
beneficial for the computational processing of alaá¹…kÄ ras, thereby making way for efficient
approaches to process figurative speech.
--------------------------------------------------------------------------


What strengths does this paper have?
--------------------------------------------------------------------------
The paper tries to tackle a particularly hard problem that has not been solved before. The
area of problem is also interesting and can have practical relevance in understanding old
texts.

Dependency tree based representation is a proved and tested solution for difficult natural
language problems and the paper employs the same to provide an alternate representation
without loss of information.
--------------------------------------------------------------------------


What weaknesses does this paper have?
--------------------------------------------------------------------------
Even though the paper sets out to tackle a challenging problem, I felt it could have done
more in the following areas:

- Instead of limiting the scope to the Purna variety, the authors could have performed some
analysis on the Lupta variation as well.

- The construction of parse trees requires identification of the six broad patterns. A
thorough analysis and solution should have been provided to perform the initial pattern
recognition.

- The potential problems that section 5 talks about are actually the key to solving this
problem, and the paper mentioning them in passing (instead of tackling) doesn't add much
```

value to the work.

Reviewer's Scores

Overall Recommendation: 3

Presentation Improvements

The figures in the paper are not in close proximity to the related text. This can impact the reader's ability to easily correlate the content in the text with the corresponding figures, potentially leading to confusion or difficulty in understanding the presented information.

Using a variety of colors in figures can pose challenges for individuals with color vision deficiencies, commonly known as color blindness. To enhance accessibility, consider incorporating additional design elements, such as shapes or patterns, to convey information.

==========
REVIEWER #2
==========

What is this paper about, and what contributions does it make?

The paper presents a methodology for representing the verbal cognition of UpamÄ alaá¹…kÄ ra using a dependency tree. The work is done in figurative speech.
It identifies and analyses the various named entities such as the vehicle, topic, common property, and the word indicating similitude. This representation can be useful for the computational processing of alaá¹…kÄ ras, thereby making way for efficient approaches to process figurative speech as well as can be used for generation.

What strengths does this paper have?

A hard problem in natural language understanding is taken up for research. The paper presents a detailed analysis of UpamÄ alaá¹…kÄ ra and this will enable to develop the dependency tree without much loss in information. The paper is well written, interesting problem and easy to read.

What weaknesses does this paper have?

Please check grammar and the figures are too clourful and bright.Escpecially the red colour with black letters is difficult for the eye.

Reviewer's Scores

Overall Recommendation: 4.5

--
ICON 2023 - https://softconf.com/icon2023/papers

--
Thanks and regards,
Bhakti Jadhav
Researcher,
HSS, IIT Bombay 400076
(Email id: bhaktij96@gmail.com)

Compound Type Identification in Sanskrit

Sriram Krishnan[1,2], Pavankumar Satuluri[3], Amruta Barbadikar[1], T S Prasanna Venkatesh[4] and Amba Kulkarni[1]
[1] Department of Sanskrit Studies, University of Hyderabad
[2] Central Sanskrit University, Ganganath Jha Campus, Prayagraj [3] IIT Roorkie
[4] Ramakrishna Mission Vivekananda College, University of Madras
sriramk8@gmail.com, apksh.uoh@nic.in

Abstract

Compounds are formed based on the principle of *sāmarthya* (semantic compatibility to compose components). Sanskrit compounds are binary compositions which can be composed further to form a nested constituency of compound structure. Analysing a compound requires us to find the correct structure as well the compound types. Compound analysis has long been approached in two stages: constituency analysis and compound type identification. In this work, we propose a methodology to analyse the compounds by first identifying the compound types based on morphological, syntactic and semantic clues from *Aṣṭādhyāyī*. We implement it within the dependency parser framework of *Saṃsādhanī*, where the types are identified first and then the constituency structure is established using a constraint solver.

1 Introduction

One of the most distinguishing features of Sanskrit is its profound use of compounds (*samāsa*), which allows concise expression of complex ideas. Compounds are formed by combining two or more words into a single lexical unit, often omitting case markers and other grammatical indicators, thus creating a compact structure to convey meaning. These are categorized into four major types on the basis of the semantic relationships between their components. The description of these semantic relations is found in *Aṣṭādhyāyī*, the seminal grammatical text of Sanskrit. The four major types of compounds are *Avyayībhāva*, *Tatpuruṣa*, *Bahuvrīhi* and *Dvandva*. These compound types have further sub-types defined based on both the syntactic and semantic nature of their components. In total, 59 sub-types of these four major types have been observed in Kumar (2012).

Aṣṭādhyāyī meticulously codifies the process of compound generation, providing precise rules for their semantic classification and syntactic construction. The semantic and syntactic compatibility between the components of a compound, referred to as '*Sāmarthya*', ensures that the words forming a compound are meaningfully connected and contribute to a unified sense. The category of the compound-word thus formed can be determined based on its meaning and the grammatical rules that specify the syntactic arrangement of the component words, as well as the case, gender, and number of the final compound. The procedure for generating compounds is clearly outlined, ensuring uniformity in their formation.

When it comes to the analysis of a compound word, there are several challenges due to the inherent complexity of their structure. The first problem we face in analyzing a compound is the segmentation. The compound in Sanskrit is always written as a single word without any space or hyphen between its constituents. Further, the close proximities of the phonemes of two consecutive components also lead to phonological changes in them introducing ambiguities during splitting, thereby worsening the situation further for analysis.

The second problem is the ambiguity at the constituency level when the number of components is more than two. When there are more than two components in a compound, the number of ways in which a compound with $n+1$ components can be subgrouped is a Catalan number C_n

Aesthetics of Sanskrit Poetry from the Perspective of Computational Linguistics: A Case Study Analysis on *Śikṣāṣṭaka*

Jivnesh Sandhan[1]*, Amruta Barbadikar[2], Malay Maity[2], Pavankumar Satuluri[3], Tushar Sandhan[7], Ravi M. Gupta[4], Pawan Goyal[5] and Laxmidhar Behera[6,7]
[1]Kyoto University, [2]University of Hyderabad, [3]IIT Roorkee,
[4]Utah State University, [5]IIT Kharagpur, [6]IIT Mandi, [7]IIT Kanpur
jivnesh@i.kyoto-u.ac.jp,pawang@cse.iitkgp.ac.in

Abstract

Sanskrit poetry has played a significant role in shaping the literary and cultural landscape of the Indian subcontinent for centuries. However, not much attention has been devoted to uncovering the hidden beauty of Sanskrit poetry in computational linguistics. This article explores the intersection of Sanskrit poetry and computational linguistics by proposing a roadmap of an interpretable framework to analyze and classify the qualities and characteristics of fine Sanskrit poetry. We discuss the rich tradition of Sanskrit poetry and the significance of computational linguistics in automatically identifying the characteristics of fine poetry. We also identify various computational challenges involved in this process, including subjectivity, cultural context and lack of large labeled datasets. The proposed framework involves a human-in-the-loop approach that combines deterministic aspects delegated to machines and deep semantics left to human experts.

We provide a human annotated analysis of *Śikṣāṣṭaka*, a Sanskrit poem, from the perspective of 6 prominent *kāvyaśāstric* schools, to illustrate the proposed framework. Additionally, we provide compound, dependency, *anvaya* (prose order linearised form), meter, *rasa* (mood), *alaṅkāra* (figure of speech), and *rīti* (writing style) annotations for *Śikṣāṣṭaka* and a web application to illustrate the poem's analysis and annotations. Our key contributions include the proposed framework, the analysis of *Śikṣāṣṭaka*, the annotations and the web application[1] for future research. We aim to bridge the gap between *kāvyaśāstra* and computational methods and pave the way for future research.

1 Introduction

Sanskrit literature has a rich and diverse tradition that has played a significant role in shaping the literary and cultural landscape of the Indian subcontinent for centuries (Pollock, 2006; Jamison and Brereton, 2014). With its complex grammatical rules and nuanced vocabulary, the Sanskrit language has provided a fertile ground for poets to craft intricate and evocative stanzas that capture the key elements of human experience (Pollock, 1996). From the ancient epics like the Rāmāyaṇa and the Mahābhārata, to the lyrical works of Kālidasa and Bhartṛhari etc. Sanskrit poetry has embodied the key elements of Indian thought and culture, providing inspiration and contemplation for generations of readers and scholars. However, the researchers in the computational linguistics community have not devoted much attention to uncovering the hidden beauty in Sanskrit poetry (Scharf et al., 2015).

The computational linguistic research community could be interested in identifying fine poetry for various reasons, including the need to evaluate machine-generated poetry (Agarwal and Kann, 2020; Van de Cruys, 2020; Li et al., 2020; Hopkins and Kiela, 2017), translating poetry (Yang et al., 2019; Ghazvininejad et al., 2018; Krishna et al., 2019), to gain a deeper understanding of poetic language (Haider, 2021; Hämäläinen and Alnajjar, 2019; Waltz, 1975), and to

*This work was conducted while the author was affiliated with IIT Kanpur.

[1]Link for interactive analysis of *Śikṣāṣṭaka*: https://sanskritshala.github.io/shikshastakam/

Justification for Similarity Index > 10%

Ms Amruta Barbadikar's PhD thesis (20hsph02) reported a similarity index of 20%. Of these, the following three resources, which are co-authored by her, constitute 18% of the similarity index.

1. Following paper from the conference proceedings forms a substantial part of the second chapter.

   A. Automatic Sanskrit Poetry Classification Based on Kāvyaguna
   Amruta Barbadikar | Amba Kulkarni
   Proceedings of the 21st International Conference on Natural Language Processing (ICON)
   ACL Anthology, USA

2. Chapter 5 draws substantially from the journal paper "Symmetry in Bharata's Yamaka: A mathematical analysis".
   Amba Kulkarni, Amruta Barbadikar and Bharati Bhat
   Indian Journal of history of Science.
   Volume 59, Pages 341-350, (2024),
   https://link.springer.com/article/10.1007/s43539-024-00140-5

Amba P.

Supervisor
Department of Sanskrit Studies
School of Humanities
University of Hyderabad
P.O. Central University
Hyderabad-500 046.TS.

Computational Analysis of Figurative Language in Sanskrit with a Focus on Alaṅkāras and Kāvyaguṇas

by Amruta Barbadikar

Submission date: 06-Nov-2025 12:42PM (UTC+0530)
Submission ID: 2805213211
File name: AmrutaB_thesis_fullPL_v7.pdf (1.06M)
Word count: 28480
Character count: 147576

Computational Analysis of Figurative Language in Sanskrit with a Focus on Alaṅkāras and Kāvyaguṇas

ORIGINALITY REPORT

20%	13%	9%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	aclanthology.org Internet Source	11%
2	Amba Kulkarni, Amruta Barbadikar, Bharati Bhat. "Symmetry in Bharata's Yamaka: A mathematical analysis", Indian Journal of History of Science, 2025 Publication	7%
3	gretil.sub.uni-goettingen.de Internet Source	<1%
4	www.wisdomlib.org Internet Source	<1%
5	ambuda.org Internet Source	<1%
6	export.arxiv.org Internet Source	<1%
7	ebin.pub Internet Source	<1%
8	eprints.utm.my Internet Source	<1%
9	Nicolás Andruskiewitsch, Giovanna Carnovale, Gastón Andrés García. "Finite-Dimensional Pointed Hopf Algebras Over Finite Simple Groups of Lie Type IV. Unipotent Classes in Chevalley and Steinberg Groups", Algebras and Representation Theory, 2019	<1%

Amba P.

Amba P.

Supervisor
Department of Sanskrit Studies
School of Humanities
University of Hyderabad
P.O. Central University
Hyderabad-500 046.TS.

	Publication	
10	Submitted to Pacifica Graduate Institute Student Paper	<1%
11	gallium.inria.fr Internet Source	<1%
12	thesis.cust.edu.pk Internet Source	<1%
13	Submitted to Narrabundah College Student Paper	<1%
14	puladesign.info Internet Source	<1%

Exclude quotes	On	Exclude matches	< 14 words
Exclude bibliography	On		