Research Statement

Hrishikesh Terdalkar
Department of Computer Science and Engineering
Indian Institute of Technology Kanpur

hrishirt@cse.iitk.ac.in
https://hrishikeshrt.github.io

The overwhelming majority of Natural Language Processing (NLP) and Computational Linguistics (CL) research focuses on a very small number of very high resource languages such as English and Chinese [Schwartz, 2022]. Excluding a short list of languages: English, Mandarin, German, French, Spanish, Japanese, Portuguese, Italian, Dutch, Arabic and Czech, the remaining, approximately 7000 of the world's languages have limited resources [Bender, 2019]. NLP for low resource languages poses unique challenges, most notable among which is: lack of annotated datasets for training, testing and evaluation of Machine Learning (ML) approaches. The focus of my research has been on *developing knowledge-based systems* for Sanskrit, a classical Indian language. Despite the abundance of existing literature, the availability of labelled datasets in digital form for Sanskrit remains scarce, rendering it a low resource language. In addressing this challenge, my research has encompassed the following areas: (1) Digitization of Sanskrit texts, (2) Development of annotation tools to facilitate dataset creation, (3) Development of Sanskrit question answering framework through the automated construction and querying of knowledge graphs, (4) Creation of software libraries to support and propel the Sanskrit NLP research, and (5) Design of user interfaces utilizing the computational aspects of Sanskrit.

Background: Sanskrit exhibits distinct features such as: morphologically rich structure; generative grammar resulting in a large vocabulary; lack of clear sentence boundaries due to the predominantly poetic nature of the literature; lack of clear word boundaries; relatively free word order; abundance of long compound words and presence of sandhi – a linguistic phenomenon of combining two adjacent words leading to change in pronunciation as well as spelling. As a result, several of the standard methodologies applicable to more well studied languages such as English are not directly applicable to Sanskrit. Further, linguistic tasks such as sentence boundary detection, canonical word ordering and word segmentation are central to Sanskrit.

Progress has been achieved in certain syntactic tasks relevant to Sanskrit NLP. Examples include morphological analysis [Goyal et al., 2012, Gupta et al., 2020], word segmentation [Hellwig and Nehrdich, 2018, Krishna et al., 2016], and semantic tasks like dependency parsing [Kulkarni, 2016, Krishna et al., 2020, Sandhan et al., 2021] and compound type identification [Sandhan et al., 2019, 2022]. Despite these advancements, certain challenges persist without fully satisfactory solutions. These include tasks such as sentence boundary detection, named entity recognition (NER), co-reference resolution, entity-relationship extraction, question-answering (QA), semantic search, and others.

Contributions: My PhD work has revolved around Sanskrit-specific computational linguistics, primarily focusing on question-answering from Sanskrit using knowledge graphs and the development of datasets, annotation tools, and interfaces required for specific tasks. My journey has encompassed software engineering's (SE) core tenets, resulting in the creation of several user-friendly web-based tools and libraries. This diverse compilation is available online at **https://sanskrit.iitk.ac.in/**. The key contributions are:

1. We utilized **computational aspects of Sanskrit prosody** and created a Sanskrit meter identification and utilization system (§1), and investigated its application towards **correction of digital corpora**.
2. We proposed a **Sanskrit question answering framework** (§2) through **automated construction of KGs**. To the best of our knowledge, this was the first such effort. We also performed a thorough error analysis and identified the need for human annotation.
3. We developed two **user-friendly** and **robust annotation tools** (§3) for annotation towards creation of knowledge-graphs and NLP datasets:
   (a) *Sangrahaka* (§3.1), a tool for marking **entities** and **relationships** towards **construction** and subsequent **querying** of KGs using **natural language query templates**.
   (b) *Antarlekhaka* (§3.2), a tool for annotation of **eight categories** of annotation tasks in a **sequential** manner, giving a **comprehensive coverage for NLP annotation**.
4. We have developed a number of **tools** and **web interfaces** (§4) exploring the **computational aspects** of Sanskrit. We have also created several **Python libraries** (§4.1) to aid programmers.

1 Application of Sanskrit Prosody for Correction of Digital Corpora

Sanskrit text corpora have undergone large-scale digitization efforts using optical character recognition (OCR) technology, inadvertently leading to the introduction of various errors. We investigated the capability of Sanskrit prosody towards correcting these errors. The classification of syllables into *short* and *long* forms the core concept of Sanskrit prosody. The classification is related to the amount of time it takes to pronounce a specific syllable. Specific sequences or combinations of *short* and *long* letters result in a particular rhythm or a *meter*. It is observed that majority of Sanskrit text adheres to the rules of Sanskrit prosody [Rajagopalan, 2020]. We created Chandojñānam [**Terdalkar** and Bhattacharya, 2023a], a system for identifying and utilizing Sanskrit meters. The system supports meter identification from text as well as images using OCR engines. Apart from its core functionality of meter identification, the system also enables finding fuzzy matches based on sequence matching, thereby facilitating the correction of inaccuracies in digital corpora. The system supports more input options as well as enhanced functionality as compared to the existing meter identification systems [Rajagopalan, 2020, Neill, 2022]. The Chandojñānam system exhibits tolerance towards erroneous texts and is able to locate the errors as well as make suggestions for fixing them. The error tolerance was evaluated on 14 texts comprising 1038 verses and exhibiting 17 different meters. Chandojñānam was able to identify the correct meter from the erroneous text in 98.2% of the cases, performing better than the extant systems [Rajagopalan, 2020] (91.9%) and [Neill, 2022] (80.3%).
This work appeared in **CDSH, WSC 2023 (ACL)**.

Future Directions: The work establishes the feasibility of Sanskrit prosody for detecting errors in the text. However, the suggestions provided are limited in nature. Further, an error in the text cannot be detected by the Chandojñānam if the error does not result in breaking a known metrical pattern. Therefore, for effective post-OCR correction, Sanskrit prosody isn't sufficient and must be used in combination with other approaches. The usage of syntactic features in the form of metrical signatures along with *seq2seq* models for post-OCR corrections is worth exploring.

2 Sanskrit Question Answering Framework

At the nucleus of my research, was the problem of building a robust QA system by orchestrating KGs for Sanskrit. We developed a framework for answering factual questions through construction of KGs [**Terdalkar** and Bhattacharya, 2019]. We employed heuristics and rule-based approaches to construct KGs through automatic extraction of *(subject, predicate, object)* triples. This is achieved using linguistic features such as morphological information and the semantic connotations associated with it. We constructed a KG capturing kinship relationships from two large Sanskrit texts, Rāmāyaṇa and Mahābhārata. The architecture of QA system is depicted in Fig. 1. The system achieved overall precision of 0.55 and F1 score of 0.47, being able to find answers for roughly 50% of the natural language questions. To the best of our knowledge, there was no other natural language QA system for Sanskrit.

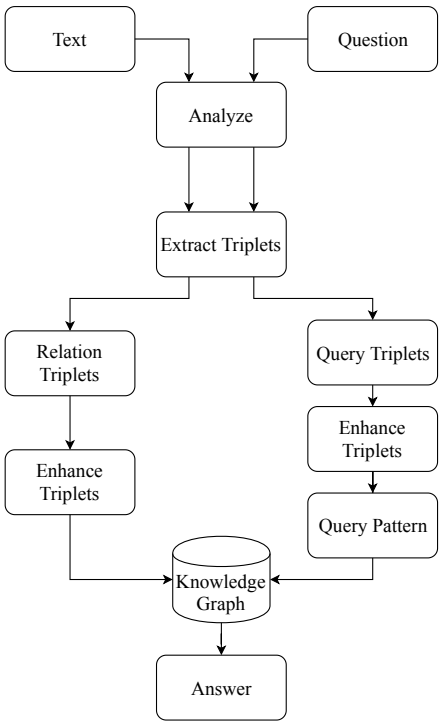

Figure 1: KG-based QA Framework

We processed a technical text from Āyurveda, Bhāvaprakaśanighaṇṭu. Here, we focused on synonym extraction. The task was broken into two subtasks: (1) detection of verses containing synonyms, and (2) identification of synonyms from the aforementioned verses. We used feature engineering, and extracted 42 linguistic features for each verse. These included morphological features and frequency of various parts-of-speech tags. We annotated two chapters from Bhāvaprakaśanighaṇṭu towards synonym identification and trained various classifiers for detection of verses containing synonyms. We achieved F1 score of 0.69 in this task. Further, we could identify synonyms from 70% of synonym groups, achieving F1 score of 0.66.
This work was published in **ISCLS 2019 (ACL)**.

Through these efforts we highlighted the shortcomings and limitations of the state-of-the-art of Sanskrit NLP. The scarcity of appropriate datasets poses a significant challenge in the development and evaluation of automated systems for KG construction. Human annotation plays a pivotal role for the creation of such datasets. We identified the need for annotation tools with task-specific and intuitive interfaces to simplify

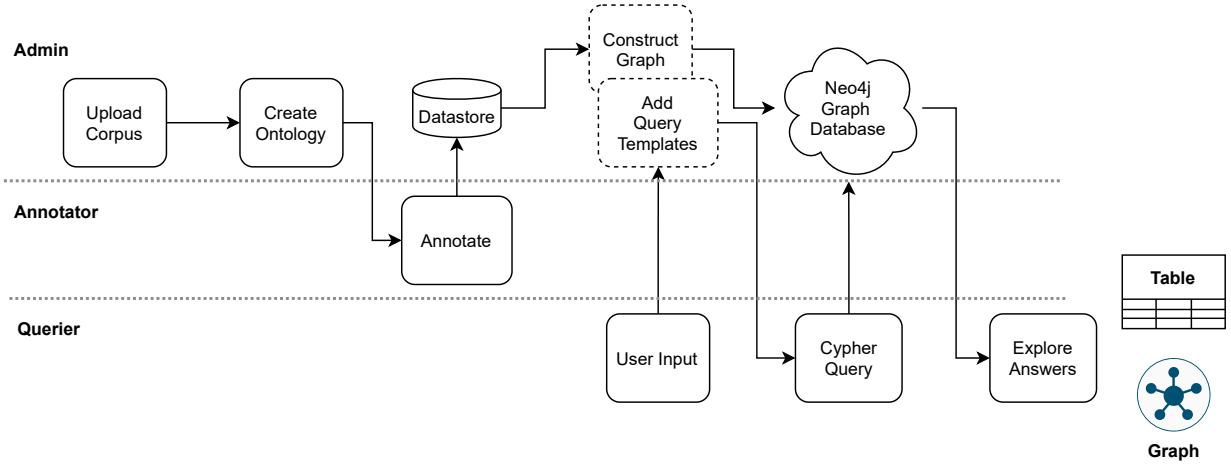

Figure 2: Workflow of Administrator, Annotator and Querier roles and their interaction with each other. Corpus creation, ontology creation, annotation, graph creation, graph querying are the principal components

the tedious task of manual annotation.

Future Directions: We tried extraction of triplets in a limited and domain-specific setting. Automated construction of a knowledge-graph with wider scope needs to be explored. Further, we used rule-based extraction as a starting point. The triplets identified using these methods can be used for training machine learning models towards automated extraction of entities and relationships.

3 Annotator-Friendly Tools for KG and Dataset Creation

Addressing the need for annotation tools, we have developed **two** intuitive, language agnostic, annotation tools: *Sangrahaka* and *Antarlekhaka*, empowering domain experts to enrich KGs and datasets. Both tools support distributed annotation. They are designed to be easily configurable, web-deployable, customizable and with a multi-tier permission system. They are *actively being used* in the real-world annotation tasks. The annotator-friendly interfaces of these tools have received positive feedback from the users, and they outperform other annotation tools in objective evaluation.

3.1 *Sangrahaka*: Annotation towards Construction and Querying of Knowledge Graphs

The first tool, *Sangrahaka* [**Terdalkar** and Bhattacharya, 2021], fuses ontology-driven annotation with query support, amplifying KG construction. The workflow of *Sangrahaka* is illustrated in Fig. 2. We demonstrated the usefulness of the tool through a real-world annotation task [**Terdalkar** et al., 2023a,b]. Through collaboration with domain experts, we created a rich and extensive ontology, with 339 node labels and 396 relationship labels, suitable for the annotation of Bhāvaprakaśanighaṇṭu, a text from Āyurveda detailing medicinal substances, their properties and medical applications. Utilizing this ontology and a custom deployment of *Sangrahaka* we conducted manual annotation on Bhāvaprakaśanighaṇṭu and subsequently constructed a KG (2430 entities and 5216 relationships).

Works related to *Sangrahaka* appeared in **ESEC/FSE 2021** and **CSDH, WSC 2023 (ACL)**. Additionally, an extended abstract based on this work was accepted at **NYCIKS 2023** and received **Best Poster Award**.

3.2 *Antarlekhaka*: Multi-task Annotation

Antarlekhaka [**Terdalkar** and Bhattacharya, 2023b] is a versatile multi-task annotation system. This system proposed a *sequential manner of annotation* (Fig. 3), wherein users annotate small units of text completing multiple categories of NLP tasks one after the other. The system addresses eight generic categories of NLP annotation tasks: sentence boundary detection, canonical word ordering, token annotation, token classification, token graph, token connection, sentence classification and sentence graph, amounting to the support for a much larger set of

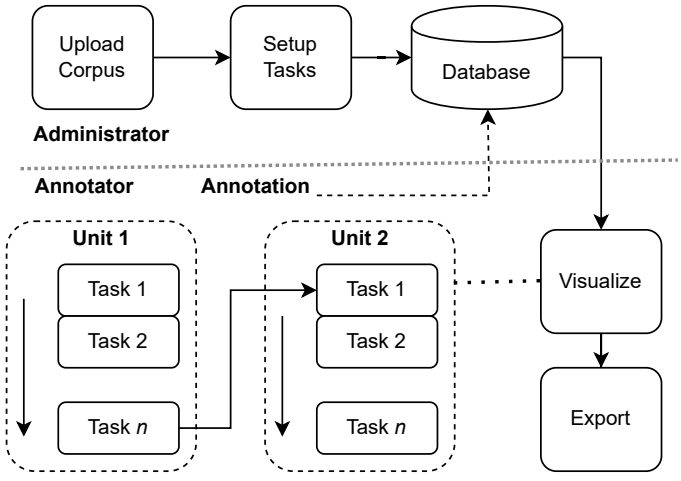

Figure 3: *Antarlekhaka*: Sequential Annotation

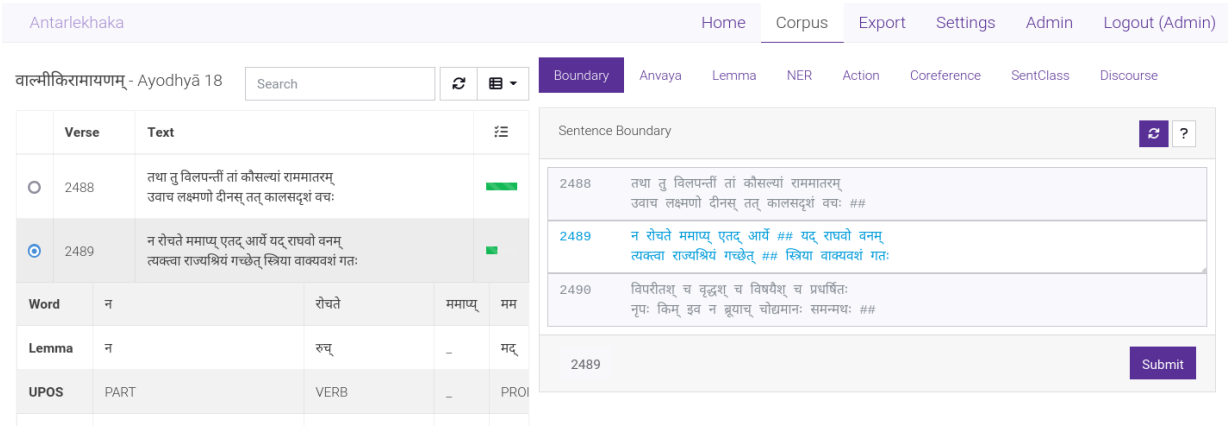

Figure 4: Annotation Interface: Corpus area shows text split into small units, and annotation area highlights various annotation task tabs

NLP tasks. The support for tasks such as identifying sentence boundaries and establishing canonical word order, making it especially useful for Sanskrit and other poetic corpora. Each category of tasks has a unique user-friendly and intuitive interface for annotation, making the tedious task of annotation much more accessible. The tool is being used for a large-scale annotation task, wherein more than 100 annotators are annotating a Sanskrit corpus of Rāmāyaṇa. Fig. 4 displays the overall annotation interface of *Antarlekhaka*. This work is accepted at **NLP-OSS @ EMNLP 2023**.

Both the tools are presented as full-stack web applications making use of state-of-the-art technologies such as *Python*, *Flask* micro-webfraemwork, *SQLite3* relation database and *Neo4j* graph database for backend; and *Jinja2* templating engine, *HTML5*, *JavaScript*, *Bootstrap* styling library for frontend. The tools were evaluated in two ways: (1) subjective evaluation and (2) objective evaluation. The subjective evaluation was conducted by the means of a survey of annotators participating in the annotation tasks. *Sangrahaka* received an overall score of 4.5/5 from a total of 10 annotators, while *Antarlekhaka* was rated 4.1/5 by 16 annotators. Fig. 5 shows the wordcloud of comments received, demonstrating the general positive feedback. Further, the tools were compared in an objective manner using technical, functional and data support related criteria [Neves and Ševa, 2021], assigning a score between 0 and 1. *Antarlekhaka* (0.79) and *Sangrahaka* (0.74) outperformed other tools such as *WebAnno* (0.71), *FLAT* (0.71), *BRAT* (0.64) and *doccanno* (0.55).

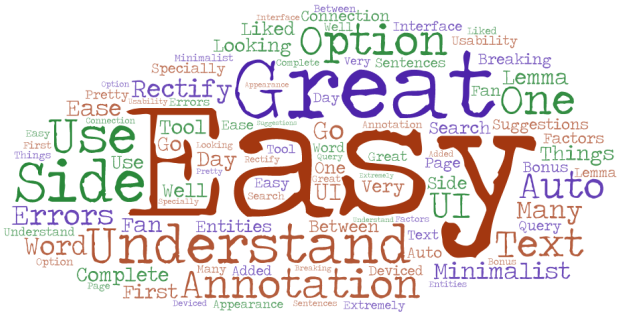

Figure 5: Wordcloud: Survey of Annotation Tools

Future Directions: The annotation tools can be infused with power of state-of-the-art NLP tools in order to make the process of manual annotation interactive. There is already a limited support for these in terms of being able to import and display linguistic informations as well as heuristics as pluggable components. This support can be extended to allow run-time suggestions as well as proactive learning to make the suggestions better over the course of process of annotation.

4 User-Centric Computational Interfaces

As part of our research contribution, we created an extensive range of web-interfaces and tools specifically designed to highlight and utilize the computational aspects of Sanskrit. These include: (1) Saṅkhyāpaddhatiḥ, a web-interface that encompasses three ancient numeral systems, enabling the representation of numbers as text. (2) Varṇajñānam, a utility pertaining to *varṇa*, a phonetic unit of the Sanskrit. (3) a Telegram bot designed to assist learners in comprehending Sanskrit grammar.
These were presented as *demonstrations* at venues such as **ISCLS 2019 (ACL)** and **CDSH, WSC 2023 (ACL)**.

4.1 Software Libraries for Advancement of Sanskrit NLP

We developed a set of software libraries to aid programmers work with Sanskrit corpora. These include: (1) *PyCDSL*, a Python library and a Command Line Interface (CLI) to simplify the processes of downloading, managing, and accessing Sanskrit dictionaries, (2) *Heritage.py*, a Python interface to *The Sanskrit Heritage*

site, (3) *sanskrit-text*, a Python library for the manipulation of Sanskrit alphabet, and (4) *google-drive-ocr*, a Python interface and a CLI to perform OCR on images and PDF documents using Google Drive v3 API.

5 Current Endeavours: Innovations and Neural Networks

Currently, my research endeavors are twofold, encompassing (1) the expansion of prior work, and (2) the pursuit of recent, novel and captivating research directions.

I am continuing our work on the application of annotation tool, *Sangrahaka*, towards completing the construction of a knowledge graph based on Bhāvaprakaśanighaṇṭu. This involves overseeing the progress of annotation, participation in the refinement of ontology, offering advice on annotation and curation, and maintaining as well as improving the existing tool, *Sangrahaka*.

I am working in a project titled *Automated Question-Answering System for Ramayana* under the wings of *Indian Knowledge Systems (IKS) division of All India Council for Technical Education (AICTE), Government of India*, wherein we use *Antarlekhaka* to collect comprehensive annotations on the Sanskrit corpus of Vālmīki Rāmāyaṇa towards construction of an automated QA system.

I am also involved in a project titled *An Interpretable Unified Framework for Text-to-text Translation among Indian Languages using Sanskrit-based Interlingua Representation* sponsored by *Ministry of Electronics and Information Technology (MeitY), Government of India* wherein we are exploring neural machine translation targeting Indian languages such as Hindi, Kannada and Sanskrit.

Additionally, I am participating in a project encompassing the creation of a curated large corpus of Sanskrit language, **pre-training language models** as well as testing their efficacy on several downstream tasks.

6 Future Research

My research lies in the **intersection** of **computational linguistics**, **natural language processing** and **software development**, forging a symbiotic interdisciplinary connection. I plan to leverage upon this to develop tangible solutions specifically pertaining to Indian languages. In particular, I want to target the following research areas:

6.1 High-level Tasks for Low-Resource Languages

NLP confronts distinct hurdles when applied to low-resource languages, encompassing challenges like sparse annotated data, linguistic diversity, and various unique linguistic phenomena. High-level tasks in NLP go beyond basic text processing and require a deeper understanding of language semantics as well as context. These include tasks such as machine translation, text summarization, question answering and so on. The advent of large language models (LLMs) opens new opportunities for such tasks. However, LLMs are still not a viable option for low-resource languages. I plan to expand my work on developing open-domain QA systems through exploration of various approaches such as automatic KG construction, information retrieval, semantic parsing, transformer-based language models, and most importantly **hybrid approaches** that make use of more than one techniques.

I am particularly interested in creating solutions to the following NLP challenges that are pertinent to Indian languages and hold promise of helping in solving high-level NLP tasks:

- **Dataset Creation**: Creating computationally usable datasets for various NLP tasks towards training and evaluating machine learning algorithms.
- **NLP Tasks in Indian Context**: Designing algorithms for standard NLP tasks in the context of Indian languages. For example, *Indian NER*: accurately identify and categorize entities in Indian names, considering the rich variety of naming conventions.
- **Low-Resource Language Modelling**: Developing effective NLP techniques for languages with limited annotated data, paving the way for meaningful analysis and understanding.
- **Cultural and Contextual Sensitivity**: Developing NLP models that respect cultural nuances and context, thus ensuring accurate interpretation and generation of text.

6.2 Multilingual NLP and Cross-Lingual Transfer Learning

A standard approach in dealing with low-resource languages involves harnessing the prowess of extensive language models to bridge the resource gap between languages, enabling knowledge dissemination from data-rich languages to under-represented ones. However, using a foreign language such as English may result in losing nuances specific to various Indian languages. One of the projects I am involved in, *funded by Government of India*, aims to facilitate translation from Indian languages to Indian languages using a Sanskrit-based interlingua. Presence of such an interlingua can also aid cross-lingual transfer learning among Indian

languages. Over the next few years, I am excited about developing a single system for Indian languages that can interact with humans using different languages and utilize knowledge learnt using one language for other languages. I plan to focus on unified grammar for Indian languages as well as tools that can perform multilingual NLP tasks for Indian languages, such as tokenization, lemmatization, morphological analysis, NER and so on. Such a system also holds promise to handle phenomenon of code-switching, which is on the rise through various social media.

6.3 Knowledge-based Systems and Large Language Models

In the rapidly evolving landscape of NLP, large language models have garnered immense attention for their remarkable ability to generate coherent and contextually relevant text. However, their potential applications extend beyond mere language generation. My future research aims to explore the diverse capabilities of large language models, such as advanced question-answering, text summarization, sentiment analysis, and language translation, tailored specifically to the intricate nuances of Indian and low-resource languages.

One promising avenue of exploration involves the development of advanced question-answering systems. By harnessing the power of large language models in conjunction with structured knowledge representations, such as knowledge graphs, these systems can provide nuanced and accurate responses to user queries across a wide range of domains. This approach not only facilitates easier access to information but also aids in preserving and disseminating the vast knowledge and literature embedded in Indian languages, thereby benefiting researchers, scholars, and the broader community.

Conclusion

My research contributions have catered not only to NLP researchers but also to domain experts and language enthusiasts, effectively working towards democratizing access to advanced language processing techniques. My research journey stands as a testament to the intricate interaction between (1) **computation**, (2) **software**, and (3) **linguistics**. I am committed to pushing the boundaries of knowledge representation, linguistic analysis, language processing, and software development driven by the conviction that this **interdisciplinary synergy holds the key to unlocking new frontiers** in linguistics and technology.

References

Emily Bender. The #benderrule: On naming the languages we study and why it matters. *The Gradient*, 2019.

Pawan Goyal, Gérard Huet, Amba Kulkarni, Peter Scharf, and Ralph Bunker. A distributed platform for sanskrit processing. In *Proceedings of COLING 2012*, pages 1011–1028, 2012.

Ashim Gupta, Amrith Krishna, Pawan Goyal, and Oliver Hellwig. Evaluating neural morphological taggers for sanskrit. *arXiv preprint arXiv:2005.10893*, 2020.

Oliver Hellwig and Sebastian Nehrdich. Sanskrit word segmentation using character-level recurrent and convolutional neural networks. In *Conference on Empirical Methods in Natural Language Processing*, pages 2754–2763, 2018.

Amrith Krishna, Bishal Santra, Pavankumar Satuluri, Sasi Prasanth Bandaru, Bhumi Faldu, Yajuvendra Singh, and Pawan Goyal. Word segmentation in sanskrit using path constrained random walks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 494–504, 2016.

Amrith Krishna, Ashim Gupta, Deepak Garasangi, Jivnesh Sandhan, Pavankumar Satuluri, and Pawan Goyal. Neural approaches for data driven dependency parsing in sanskrit. *arXiv preprint arXiv:2004.08076*, 2020.

Amba Kulkarni. Samsaadhanii: A sanskrit computational toolkit, 2016.

Tyler Neill. Skrutable: Another step toward effective sanskrit meter identification. 2022.

Mariana Neves and Jurica Ševa. An extensive review of tools for manual annotation of documents. *Briefings in bioinformatics*, 22(1):146–163, 2021.

Shreevatsa Rajagopalan. A user-friendly tool for metrical analysis of sanskrit verse. *Computational Sanskrit & Digital Humanities*, page 113, 2020.

Jivnesh Sandhan, Amrith Krishna, Pawan Goyal, and Laxmidhar Behera. Revisiting the role of feature engineering for compound type identification in sanskrit. In *Proceedings of the 6th international Sanskrit computational linguistics symposium*, pages 28–44, 2019.

Jivnesh Sandhan, Amrith Krishna, Ashim Gupta, Laxmidhar Behera, and Pawan Goyal. A little pretraining goes a long way: A case study on dependency parsing task for low-resource morphologically rich languages. *arXiv preprint arXiv:2102.06551*, 2021.

Jivnesh Sandhan, Ashish Gupta, **Hrishikesh Terdalkar**, Tushar Sandhan, Suvendu Samanta, Laxmidhar Behera, and Pawan Goyal. A novel multi-task learning approach for context-sensitive compound type identification in Sanskrit. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4071–4083, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics. URL https://aclanthology.org/2022.coling-1.358.

Lane Schwartz. Primum non nocere: Before working with indigenous data, the acl must confront ongoing colonialism. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, volume 2, 2022.

Hrishikesh Terdalkar and Arnab Bhattacharya. Framework for question-answering in Sanskrit through automated construction of knowledge graphs. In *Proceedings of the 6th International Sanskrit Computational Linguistics Symposium*, pages 97–116, IIT Kharagpur, India, October 2019. Association for Computational Linguistics. URL https://aclanthology.org/W19-7508.

Hrishikesh Terdalkar and Arnab Bhattacharya. Sangrahaka: A tool for annotating and querying knowledge graphs. In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2021, page 1520–1524, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450385626. doi: 10.1145/3468264.3473113. URL https://doi.org/10.1145/3468264.3473113.

Hrishikesh Terdalkar and Arnab Bhattacharya. 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*, pages 113–127, Canberra, Australia (Online mode), January 2023a. Association for Computational Linguistics. URL https://aclanthology.org/2023.wsc-csdh.8.

Hrishikesh Terdalkar and Arnab Bhattacharya. Antarlekhaka: A comprehensive tool for multi-task natural language annotation. In *Proceedings of the 3rd Workshop on NLP Open Source Software at the 2023 Conference on Empirical Methods in Natural Language Processing*, NLP-OSS @ EMNLP, pages 199–211. Association for Computational Linguistics, 2023b. doi: 10.18653/v1/2023.nlposs-1.23. URL https://aclanthology.org/2023.nlposs-1.23/.

Hrishikesh Terdalkar, Arnab Bhattacharya, Madhulika Dubey, S Ramamurthy, and Bhavna Naneria Singh. Semantic annotation and querying framework based on semi-structured Ayurvedic text. In *Proceedings of the Computational Sanskrit & Digital Humanities: Selected papers presented at the 18th World Sanskrit Conference*, pages 155–173, Canberra, Australia (Online mode), January 2023a. Association for Computational Linguistics. URL https://aclanthology.org/2023.wsc-csdh.11.

Hrishikesh Terdalkar, Vishakha Deulgaonkar, and Arnab Bhattacharya. Ayurjñanam: Exploring Ayurveda using knowledge graphs. In *National Youth Conference on Indian Knowledge Systems*, Roorkee, India, 2023b. **Best Poster Award**. URL https://sanskrit.iitk.ac.in/ayurveda/.