Sanskrit Voyager: Unified Web Platform for Interactive Reading and Linguistic Analysis of Sanskrit Texts

Giacomo De Luca, Danilo Croce, Roberto Basili
Department of Enterprise Engineering
University of Rome Tor Vergata
Via del Politecnico 1, 00133, Rome, Italy
deluca@ing.uniroma2.it {croce,basili}@info.uniroma2.it

Abstract

Sanskrit Voyager is a web application for searching, reading, and analyzing the texts in the Sanskrit literary corpus. Unlike previous tools that require expert linguistic knowledge or manual normalization, Sanskrit Voyager enables users to search for words and phrases as they actually appear in texts, handling inflection, sandhi, and compound forms automatically while supporting any transliteration. The system integrates four core functionalities: (1) multi-dictionary lookup with morphological analysis and inflection tables; (2) real-time text parsing and annotation; (3) an interactive reader for over 900 digitalized texts; and (4) advanced corpus search with fuzzy matching and filtering. Evaluation shows over 92% parsing accuracy on complex compounds and substantially higher search recall than BuddhaNexus on challenging queries. Source code is publicly available under CC-BY-NC license, resource-efficient, and designed for both learners and researchers, offering the first fully integrated, user-friendly platform for computational Sanskrit studies [1].

1 Introduction

Sanskrit occupies a unique position as an ancient language whose texts remain highly relevant today. Major treatises of multiple religious traditions (Buddhism, Jainism, Vedic religions, Vaishnavism, Shaivism) and foundational texts on yoga are written in Sanskrit. With many authors arguing about the necessity of a dialogue still to come between Western and Indian philosophy (Garfield, 2014; Westerhoff, 2009), access to Sanskrit literature has become increasingly important. However, computational access remains limited: major digital libraries (e.g. GRETIL[2], SARIT[3], Muktabodha[4], DSBC[5], SanskritDocuments[6]) typically provide plain-text corpora with minimal linguistic annotation, while more advanced platforms such as the Digital Corpus of Sanskrit (DCS)[7] (Hellwig, 2007) and BuddhaNexus[8] focus either on annotated subsets or cross-text search but lack unified, interactive reading and analysis environments.

A key technical challenge is posed by Sanskrit's pervasive *sandhi* (euphonic word merging), productive compounding, and rich inflectional morphology, which obscure word boundaries and complicate dictionary lookup or search (Bhardwaj et al., 2018; Huet, 2009; Hellwig and Nehrdich, 2018; Sandhan et al., 2022). While recent advances, including byte-level transformers fine-tuned for Sanskrit (Nehrdich et al., 2024) and new hybrid approaches (De Luca, 2025), have improved automatic segmentation and analysis, available tools still require significant linguistic expertise and often force users to combine multiple platforms for even basic research or reading tasks.

In this paper, we present Sanskrit Voyager, an integrated web application that addresses these limitations through four core functionalities: (1) dictionary search capable of handling inflected, sandhi-affected, and compound forms across multiple dictionaries, also returning grammatical tagging and inflection tables; (2) real-time text analysis with automatic parsing and annotation for user uploaded texts or books; (3) an interactive Book Reader for more than 900 texts with optional machine translation; and (4) Corpus Search with advanced filtering and navigation. Unlike existing tools, which offer

[1] Application: https://www.sanskritvoyager.com
Source code: https://github.com/SanskritVoyager
Demo video: https://youtu.be/FCKlW4NKJEc

[2] https://gretil.sub.uni-goettingen.de
[3] https://sarit.indology.info/
[4] https://muktabodha.org/
[5] https://www.dsbcproject.org/
[6] https://sanskritdocuments.org/
[7] http://www.sanskrit-linguistics.org/dcs/
[8] https://github.com/dharmamitra/dharmanexus

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 244–253
November 4-9, 2025

these features only in isolation, Sanskrit Voyager integrates them into a unified, mobile-compatible platform, making Sanskrit texts accessible both to newcomers and advanced researchers. The system automatically normalizes input (including inflection, sandhi, compounding, and transliteration), enabling both accurate multi-dictionary lookup and morphological analysis for words as they appear in actual texts. This approach contrasts with existing tools, which typically require manual normalization or specialist linguistic knowledge. The platform covers over 900 texts from major digital corpora (including GRETIL and SARIT), providing on-demand word-by-word annotation and exploration. All processing is performed automatically and in real time, removing the need for manual pre-annotation. By bringing together these core functionalities within a single web application, Sanskrit Voyager addresses long-standing usability barriers and supports both comprehensive search and detailed linguistic study.

We review related work in Section 2, describe the system's design and functionalities in Sections 3 and 4, present evaluation results in Section 5, and conclude in Section 6.

2 Related Work

Sanskrit digital humanities have made significant advances in the last decade, yet current tools remain fragmented, with each addressing only a subset of the linguistic and technical challenges.

Sanskrit Word Segmentation and Analysis. The resolution of *sandhi* and compounds, collectively known as the Sanskrit Word Segmentation (SWS) (Krishna et al., 2017) task, has been the primary focus of computational linguistics for Sanskrit. Traditional rule-based systems (Huet, 2009), statistical models (Hellwig and Nehrdich, 2018), and recent hybrid approaches (Sandhan et al., 2022) have all targeted these problems, but the ambiguity and complexity of the language persist. Token-based transformer models also struggle due to pervasive word merging, a direct consequence of *sandhi*. Recently, byte-level transformer models such as ByT5 (Xue et al., 2022), especially those fine-tuned for Sanskrit (Nehrdich et al., 2024), have enabled substantial progress, allowing mature web applications like Dharmamitra to emerge. However, even state-of-the-art models face challenges such as oversplitting: for example, in the *bhagavadgītā* section of the Sandhikosh benchmark (Bhardwaj et al., 2018), correct splitting is considered *bhagavat+gītā*, thus losing the dictionary entry and meaning of the compound title. Our system builds on these advances with a cascading approach that combines fast, rule-based processing for simple terms with statistical models for more complex words, thus enabling efficient and robust analysis for around 70% of the words, while the remaining 30% are delegated to the more computationally expensive methods.

Dictionary Access and Linguistic Tools. The University of Cologne project (Cologne Digital Sanskrit Dictionaries) offers the largest collection of digitized Sanskrit dictionaries (42 in total), available via web or Python PyCDSL library. While these resources are rich, interfaces require users to select transliteration formats and to search using the lemma or exact root form, limiting usability for non-specialists. Features such as multi-dictionary search, diacritic-insensitive queries, or inflected word recognition are present, but only in isolation and not in combination. The Sanskrit Heritage dictionary offers inflected search in French. Inflected form search is also present on the Sanskrit-Dictionary[9] website. The Sanskrit Reader Companion (Huet and Goyal, 2013), also on the Heritage website, separately provides sandhi splitting. Some websites, like Kosha.app[10], provide collections of dictionaries. However, even with these tools, since a large portion of the terms have some form of sandhi or compounding, retrieving dictionary entries still requires significant linguistic knowledge. The recent Dharmamitra platform[11] represents a notable innovation, using ByT5-Sanskrit (Nehrdich et al., 2024) for automatic sandhi and stemming, but supports only a simplified Monier-Williams dictionary and is not by design a text reader interface.

Digital Corpora and Text Readers. Access to the Sanskrit textual tradition is similarly fragmented. GRETIL (Göttingen Register of Electronic Texts in Indian Languages) is the principal repository, offering over 800 texts with inconsistent TEI encoding and minimal formatting. SARIT provides higher-quality consistent XML for a smaller corpus (63 texts), while other collections like Muktabodha and the Digital Sanskrit Buddhist Canon (DSBC) focus on specialized traditions. The TITUS project[12] provides a broad but difficult-to-use corpus, due to its

[9] https://sanskritdictionary.com/
[10] https://kosha.app/
[11] https://dharmamitra.org/
[12] https://titus.uni-frankfurt.de

proprietary transliteration and restrictive licensing. BuddhaNexus (now DharmaNexus) [13] aggregates 1,700+ texts and enables cross-corpus search, but its interface for reading and analysis remains basic. In contrast, platforms for classical Greek, such as Scaife[14], enable integrated, interactive reading and search, something still missing for Sanskrit.

Shortcomings of Current Tools and Opportunities for Sanskrit Voyager. Despite substantial progress, existing platforms remain fragmented: some, like DCS (Hellwig, 2007), provide high-quality manual annotations for approximately 260 texts, offering gold-standard linguistic analysis but with limited corpus coverage and no real-time processing capabilities. Others aggregate large corpora but offer minimal linguistic support, limited search, or interfaces requiring specialist skills. Few systems combine sandhi, inflection, and compound handling in dictionary queries, and almost none provide cross-corpus search or mobile optimization. Formatting inconsistencies, restrictive licenses, and the absence of integrated, interactive reading further hinder accessibility for non-experts. Sanskrit Voyager addresses these gaps by providing, for the first time, a unified, web-based environment that combines real-time linguistic analysis, comprehensive and user-friendly dictionary search (handling inflected, sandhi-merged, and compounded forms in any transliteration), and large-scale interactive corpus reading and search.

3 System Architecture

Sanskrit Voyager is structured as a modular client-server architecture designed to balance computational efficiency, maintainability, and extensibility. The system is composed of a backend deployed on a cloud-based Dokku platform and a web-based frontend hosted on Vercel.

The backend, running on a DigitalOcean droplet under Dokku, is built around a Python application that centralizes all core linguistic operations: transliteration, sandhi and compound splitting, stemming, and dictionary lookup, implemented via the open-source *process-sanskrit* library[15]. Query preprocessing and database management are handled with SQLAlchemy ORM, facilitating modularity and future code reuse. PostgreSQL acts as both the main data storage and the core search engine for the platform. By leveraging advanced full-text indexing and ranking features, the system can efficiently search and retrieve relevant results across large and heterogeneous Sanskrit corpora. Multiple representations of each text (original, normalized, stemmed, etc.) are indexed and weighted differently, so that users can search effectively regardless of transliteration or input format. Redis is integrated as a caching layer for frequent or computationally expensive queries, supporting low-latency interaction even on modest hardware. The entire backend runs efficiently on a DigitalOcean droplet with 2 vCPUs, 4GB RAM, and 80GB disk storage, demonstrating the resource-efficiency of our architecture. All services are deployed as Dokku plugins, allowing for straightforward scaling, backup, and migration. System functionalities are exposed via a RESTful API (built with Flask), decoupling the frontend from the backend and enabling potential future integration with other research tools, workflows, or interfaces. The user-facing frontend is implemented as a single-page React application using Vite for development speed and performance optimization. Mantine UI, customized for accessibility and visual consistency, underpins the UI, making the application usable across devices and screen sizes. The mobile experience features a reimplemented interface with mobile-specific components, custom hooks for touch interactions, and optimized layouts. Comprehensive documentation, written in Docusaurus and visually integrated with the main application, is hosted separately to support reproducibility and onboarding of new users.

The architecture is designed for scalability and sustainability: all components are source-available and cloud-ready, allowing easy deployment, updates, and horizontal scaling. Additional modules, such as new NLP pipelines or dictionary sources, can be integrated with minimal effort, making the system flexible and future-proof.

4 Core Functionalities

Sanskrit Voyager provides four core features in an integrated environment: dictionary lookup for inflected and compound forms, real-time text parsing, interactive reading of digitized texts, and full-corpus search with fuzzy matching. Each module is outlined below.

Dictionary Search. Traditional Sanskrit dictionaries require users to master complex linguis-

[13] https://dharmamitra.org/nexus

[14] https://scaife.perseus.org/

[15] https://github.com/Giacomo-De-Luca/Process-Sanskrit

Figure 1: Dictionary Search interface for the query *pratiprasave*. The system analyzes the input, splits compounds, assigns grammatical features, and shows aggregated multi-dictionary results. Subwords and citations are clickable for further exploration. Sources are exapandable on hover.

tic transformations, such as recognizing inflected forms (e.g., "eṣyāmi" as the future of "i"), resolving sandhi, splitting compounds, and navigating various transliteration schemes. Sanskrit Voyager removes these barriers by allowing users to search for words exactly as they appear in texts—inflected, compounded, sandhi-merged, or in any common transliteration. Queries are automatically normalized (including diacritic-insensitive matching, so "*samskara*" matches "*saṃskāra*") and require no manual settings, making the system accessible even to users with no linguistic expertise.

The dictionary interface aggregates results from six major Sanskrit lexica (Monier-Williams, Apte, Cappeller, Macdonell, Grassmann, Edgerton) and the Concise Pali Dictionary[16], with over 246,000 unique entries. All abbreviations and cross-references are expanded, and Sanskrit terms within entries are interactive: clicking on a subword or citation navigates directly to the relevant entry or retrieves the cited passage via Corpus Search. Query processing relies on a modular, three-stage cascading system (De Luca, 2025). The pipeline starts with deterministic stemming and sandhi splitting

[16] https://buddhistuniversity.net/content/reference/concise-pali-dictionary

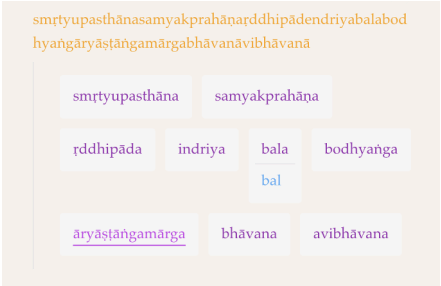

Figure 2: Text Analysis Tool output for a long compound. Each constituent is automatically split and displayed as an interactive card, with clickable access to dictionary entries and grammatical analysis.

via inflection tables and rewriting rules. A statistical parser (`sanskrit_parser` or ByT5-Sanskrit) is invoked for compounds or sandhi blocks, when SQL queries return null (as these are not in the inflection tables). The results from the previous step are scored on morphology and length and sorted by score. A right-truncation heuristic with rewriting is applied if the best score result from the previous step is below a threshold. This modular design allows efficient processing: simple lookups are deterministically processed in under 100 ms, while even the longest compounds complete within 3 seconds. For each matched stem, the system returns grammatical tagging, inflection, dictionary entries for the selected dictionaries and, where available, full inflection tables and a decomposition into sub-components (e.g., "*prati-prasava*" for "*pratiprasava*"). All results are fully interactive and can be further explored. This approach supports a wide range of users, from those interested in understanding terms commonly used in mantras or yoga postures to advanced researchers. It also enables ongoing extension to new dictionaries, parsing modules, or related languages.

Text Analysis Tool. The text analysis tool enables users to input Sanskrit text or to upload books in any transliteration format, which is automatically detected and transliterated. Each segment is transformed into an interactive element which is parsed on-click using the same cascading pipeline as the dictionary search, including sandhi removal, compound splitting, and grammatical analysis, with all possible parses displayed. Roots, stems, and grammatical tags are rendered as interactive elements. Users can click on subwords to instantly access their dictionary entries, and ambiguous forms are shown with all available interpretations. Color coding is used to vi-

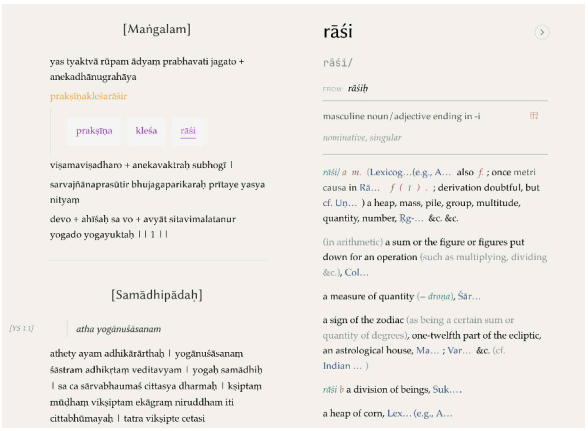

Figure 3: Book Reader: Interactive reading of digitized Sanskrit texts (here: Pātañjalayogaśāstra), with automatic segmentation of compounds (*prakṣīnakleśarāśir*), morphological annotation, and direct dictionary lookup (right panel).

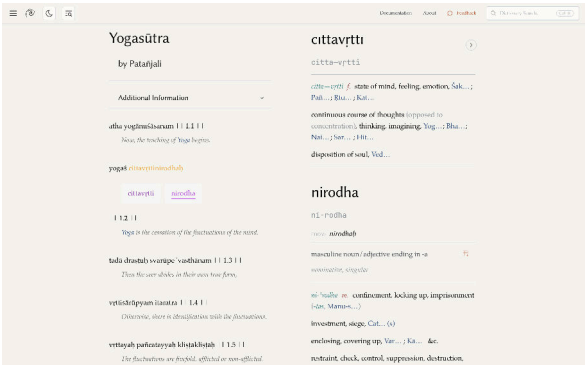

Figure 4: Book Reader Translations: showing line by line automatic translation of the Yogasūtra. The original text and Sanskrit terms in the translations can be clicked for analysis.

sually distinguish verbs, indeclinables, and pronouns, making grammatical patterns immediately clear. Figure 2 demonstrates the tool's output on a complex compound from a Buddhist text. The original compound, which merges several technical terms (e.g., *smṛtyupasthāna*, *samaykpradhāna*, *bodhyaṅga*, *āryāṣṭāṅgamārga*), is automatically decomposed into its constituents, each presented as a clickable card. Selecting a subword (such as *bala*) reveals its root and dictionary entry, and underlined terms link to further analysis or cross-references. This interactive display allows both students and scholars to dissect long compounds and explore their morphology and semantics without manual intervention, greatly simplifying the analysis of real Sanskrit texts. The pipeline is fully modular: each processing stage is implemented as a distinct component in the backend, allowing for easy extension or substitution (e.g., swapping in improved models, new inflection tables, or custom rule sets). This architecture enables both rapid experimentation with new NLP models and straightforward adaptation to other Indic languages or custom annotation schemes.

Book Reader. The Book Reader provides interactive access to over 900 digitized Sanskrit texts from the GRETIL and SARIT collections. Users can select books by searching for titles or authors, with normalization handling various spelling and diacritic forms. All texts were pre-processed and converted from XML into a structured JSON format that preserves metadata, hierarchical segmentation, enables advanced interaction and allows translations to be added. Figure 3 shows the Book Reader interface applied to a classical text. The left panel displays the Sanskrit text segmented by chapter and verse, with complex compounds automatically split and highlighted. Clicking on any segment (such as *prakṣīnakleśarāśir*) brings up grammatical analysis and dictionary lookups for each subword (see the right panel for the analysis of *rāśi*). Metadata, licenses, and annotations are presented in expandable accordions. Inline notes and text variants appear as numbered elements positioned alongside the text that expands on-click. A table of contents is automatically generated from chapter headings for navigation. For a subset of books, which will be gradually expanded until the entire corpus is covered, the Book Reader offers segment-level machine translation (using state-of-the-art LLMs), with Sanskrit terms in the translation remaining clickable for direct dictionary lookup, facilitating learning and research. This integrated, navigable format is designed to support both reading fluency and detailed linguistic study, far surpassing the static, plain-text interfaces typical of other digital Sanskrit libraries.

Corpus Search. The Corpus Search module allows users to search across the entire Sanskrit corpus using both exact and fuzzy queries, with advanced filtering and navigation features. Currently it enables searching and reading within the 1700+ Buddhanexus corpus, which will be expanded in a later release. Unlike traditional interfaces, which typically require users to know the precise lemma or root form, our system supports real-world queries: users can search for any inflected, sandhi-merged, or compounded form, and the system will automatically normalize, stem, and match across multiple

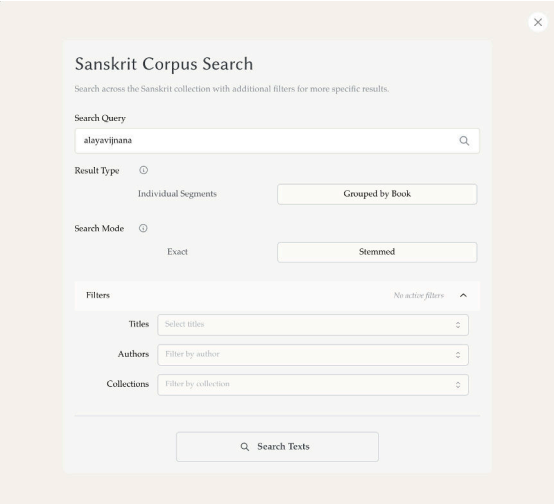

Figure 5: Corpus Search interface: users can specify query type, result grouping, and filters for author, title, and collection.

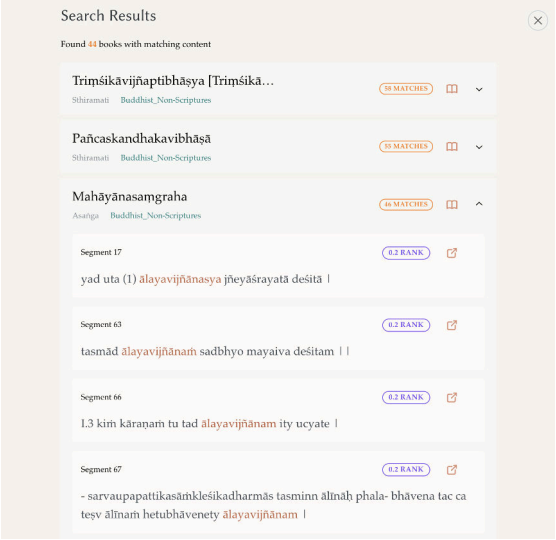

Figure 6: Corpus Search results: books with matches are grouped and ranked, with contextual snippets for each occurrence of the query term (here, "*alayavijñāna*"). Clicking a match opens the text at the passage in the Book Reader.

text representations. Figure 5 shows the Corpus Search interface, where users can enter queries, select result grouping (by individual segment or by book), and set the search mode (exact or stemmed). Additional filters for author, title, and collection are available, and all inputs are normalized to handle variant spellings and diacritics. Internally, each text segment in the corpus is indexed in multiple forms: original, cleaned, sandhi/compound-split, diacritic-free, and stemmed. These representations are stored in a single PostgreSQL column with weighted *ts_vectors* in decreasing order. This unified indexing approach allows queries like "*alayavijnana*" to match "*ālayavijñāna*". PostgreSQL GIN indexes with fuzzy match extensions ensure efficient retrieval even for complex queries. Queries exceeding 40 characters are matched against a folio-level materialized view using the same indexing approach. Figure 6 displays typical results for a query such as "*alayavijñāna*": the left panel groups matches by book (with match counts), while the right panel lists all occurrences of the search term within context. Clicking on any match jumps directly to the corresponding location in the Book Reader, where further analysis and navigation are available. All titles, authors, and collections are normalized and filterable for corpus-wide exploration. Queries take between 150 and 350 milliseconds to complete.

5 Evaluation

We evaluated Sanskrit Voyager's core parsing and search capabilities, which underpin all functionalities of the system.

Parsing Accuracy. Because existing benchmarks like Sandhikosh cover only sandhi and not compounds, and DCS-derived resources use incompatible conventions, we designed a custom evaluation. We randomly sampled 300 compounds from the GRETIL corpus, across three length categories (10–20, 20–40, and 40+ characters, up to 283), and evaluated parsing accuracy. Our cascading architecture achieved over 90% accuracy in all categories: 93.1% for medium, 94.2% for long, and 91.1% for very long compounds (overall: 92.4%). On the entire Yoga Sūtra (665 words), accuracy reached 95.9%, reflecting the system's strength when single words are taken into account.

Corpus Search. We compared Voyager's search module with BuddhaNexus using 10 challenging queries (5 technical terms, 5 multi-word phrases, automatically selected for diversity). Sanskrit Voyager consistently returned more results, e.g., 472 matches for "*ālayavijñāna*" (vs 73 in BuddhaNexus), and robust retrieval for complex phrases like "*sarvabījakam vijñānam*" (18 matches vs 2).

Error Analysis and Usage. Most parsing errors (40%) are due to dictionary gaps, especially for rare or Buddhist technical terms, followed by missing inflection forms (30%), challenging sandhi (20%), and rare morphology (10%). The application has been publicly deployed for two months, with only

Table 1: Selected search queries: Voyager vs BuddhaNexus.

Query	Voyager	BuddhaNexus
ālayavijñāna	472	73
satkāryavāda	152	40
anirvacanīya	409	200+
sarvabījakaṃ vijñānam	18	2
vākyaṃ rasātmakaṃ kāvyam	4 (1 duplicate entry)	2

organic discovery. In June 2025 it was used by 204 users, averaging 9:16 minutes per session. The 81 users active in the preceding the paper completion averaged 11:36 minutes per session, demonstrating strong user engagement and usability of the platform.

To gather direct user impressions, a feedback section was made available on the website, inviting users to evaluate the system's usability and effectiveness through the questionnaire reported in Appendix A. At the time of writing, around ten users have provided responses, consistently assigning the maximum scores for all categories, indicating a highly positive reception with respect to both ease of use and the practical utility of the platform. While these early results are promising and demonstrate strong user appreciation, the current sample remains limited. The feedback form remains open and we plan to conduct a more extensive survey to obtain a broader and more representative assessment of the system's impact as user adoption grows.

6 Conclusion

Sanskrit Voyager makes the Sanskrit literary tradition accessible to a broad audience, enabling search, analysis, and interactive reading over hundreds of digitized texts, with no prerequisite knowledge of morphology, sandhi, or transliteration. By unifying dictionary lookup, text analysis, book reading, and corpus search in a single platform, the system lowers entry barriers for learners and streamlines research workflows for specialists.

Source available, released with CC-BY-CD license and resource-efficient, Sanskrit Voyager demonstrates how modern NLP and thoughtful design can unlock complex classical languages for both scholarship and the wider public.

Limitations

While Sanskrit Voyager currently performs corpus search on the BuddhaNexus dataset, integration of additional Sanskrit digital libraries is in progress, with preliminary versions of a unified "Sanskrit-Literature" dataset already prepared. Some advanced features, such as large-scale machine translation and embedding-based retrieval, are planned for future releases as resource and evaluation cycles permit. Machine translation was tested using multiple LLMs (GPT-4o, GPT-5, Gemini 2.5 Pro, Claude Sonnet 3.7, and the Dharmamitra model[17]), but is currently limited to a subset of texts pending development of appropriate evaluation metrics to determine which model produces the most accurate translations.

The current pipeline achieves strong performance for most literary and technical queries, though rare terms not present in available dictionaries or inflection tables may yield incomplete analysis. Integration with the inflection tables of the Heritage website to increase coverage is planned but not yet in place. Future implementations will also expand dictionary coverage to include traditional lexical resources such as koshas and the Dhātupātha pending the manual formatting required for each dictionary. Broader community and domain expert feedback will further refine system accuracy and coverage as the project evolves.

Ethics and Broader Impact

Sanskrit Voyager aims to democratize access to Sanskrit literature for both academic and general audiences. All included corpora are sourced from open-access or Creative Commons digital libraries; dictionary and software components are distributed under open-source non commercial licenses. No copyrighted or restrictively licensed materials, including proprietary translations, contemporary commentaries, or materials with unclear provenance, were incorporated into the platform. The platform is designed to support linguistic and cultural research, education, and the preservation of textual heritage, with no features enabling commercial exploitation or user profiling. Limitations in dictionary coverage and morphological analysis may impact the accuracy for certain technical or rare terms, and the system does not provide human or cultural context for interpreting sacred or ritual language. Developers are committed to ongoing improvement, community feedback, and responsible handling of cultural material.

[17] https://huggingface.co/buddhist-nlp/gemma-2-mitra-it

References

Shubham Bhardwaj, Neelamadhav Gantayat, Nikhil Chaturvedi, Rahul Garg, and Sumeet Agarwal. 2018. Sandhikosh: A benchmark corpus for evaluating sanskrit sandhi tools. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Cologne Digital Sanskrit Dictionaries. 2025. Cologne digital sanskrit dictionaries.

Giacomo De Luca. 2025. Accessible Sanskrit: A cascading system for text analysis and dictionary access. In *Proceedings of the Second Workshop on Ancient Language Processing*, pages 38–46, The Albuquerque Convention Center, Laguna. Association for Computational Linguistics.

Jay L Garfield. 2014. *Engaging Buddhism: Why it matters to philosophy*. Oxford University Press.

Oliver Hellwig. 2007. Sanskrittagger: A stochastic lexical and pos tagger for sanskrit. In *International Sanskrit Computational Linguistics Symposium*, pages 266–277. Springer.

Oliver Hellwig and Sebastian Nehrdich. 2018. Sanskrit word segmentation using character-level recurrent and convolutional neural networks. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2754–2763.

Gérard Huet. 2009. Sanskrit segmentation. In *Proceedings of the South Asian Languages Analysis Roundtable XXVIII*, Denton, Ohio.

Gérard Huet and Pawan Goyal. 2013. Design of a lean interface for sanskrit corpus annotation. *Proceedings of ICON*, pages 177–186.

Amrith Krishna, Pavankumar Satuluri, and Pawan Goyal. 2017. A dataset for sanskrit word segmentation. In *Proceedings of the Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 105–114.

Sebastian Nehrdich, Oliver Hellwig, and Kurt Keutzer. 2024. One model is all you need: ByT5-Sanskrit, a unified model for Sanskrit NLP tasks. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13742–13751, Miami, Florida, USA. Association for Computational Linguistics.

Jivnesh Sandhan, Rathin Singha, Narein Rao, Suvendu Samanta, Laxmidhar Behera, and Pawan Goyal. 2022. Translist: A transformer-based linguistically informed sanskrit tokenizer. *arXiv preprint arXiv:2210.11753*.

Jan Westerhoff. 2009. *Nagarjuna's Madhyamaka: A philosophical introduction*. Oxford University Press.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. Byt5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

A Appendix: User Evaluation Questionnaire

As part of our user evaluation, we invited participants to complete the following questionnaire regarding their experience with the Sanskrit Voyager system (https://www.sanskritvoyager.com). The questionnaire included the following items:

1. **Usefulness for Reading and Researching Sanskrit Texts**
   How useful do you find the system for reading and researching Sanskrit texts?

   - 1 – Not useful, does not solve any real problem
   - 2 – Slightly useful, solves only very limited problems
   - 3 – Moderately useful, but other tools offer similar functionality
   - 4 – Useful, effectively solves several important issues
   - 5 – Very useful, solves concrete problems that other systems do not address

2. **Ease of Use**
   How do you rate the system's ease of use?

   - 1 – Very difficult and frustrating
   - 2 – Quite difficult, takes time to learn
   - 3 – Moderately intuitive, but room for improvement
   - 4 – Easy, quick to learn
   - 5 – Very easy and intuitive from the start

3. **Dictionary Search Functionality**
   How do you rate the dictionary search functionality?

   - 1 – Poor, does not retrieve correct words and is hard to use
   - 2 – Limited, sometimes retrieves entries, sometimes not
   - 3 – Adequate, finds entries but no improvement compared to other projects
   - 4 – Good, retrieves most entries and additional information is useful
   - 5 – Excellent, easy to use and correctly retrieves entries

4. **Automatic Text Parsing and Analysis**
   How would you rate the automatic text parsing and analysis feature?

   - 1 – Poor, too many errors and unreliable
   - 2 – Fair, works but often incorrect
   - 3 – Acceptable, some errors but still helpful
   - 4 – Good, few errors and generally reliable
   - 5 – Excellent, highly accurate and trustworthy

5. **Corpus Search Functionality**
   How do you rate the corpus search functionality?

   - 1 – Poor, very limited and imprecise
   - 2 – Limited, often misses relevant results
   - 3 – Adequate, finds relevant content with some flaws
   - 4 – Good, easy to find useful results

- 5 – Excellent, very precise and comprehensive

6. **System Response Time**
   Are you satisfied with the system's response time?

   - 1 – No, too slow for practical use
   - 2 – Quite slow, needs major improvement
   - 3 – Moderately fast, but could be better
   - 4 – Fast and convenient to use
   - 5 – Very fast, pleasantly responsive

7. **Recommendation Likelihood**
   Would you recommend Sanskrit Voyager to colleagues or students interested in Sanskrit studies?

   - 1 – Definitely not, not recommendable
   - 2 – Probably not, better tools exist
   - 3 – Not sure, maybe in some specific cases
   - 4 – Yes, I would generally recommend it
   - 5 – Absolutely yes, I would strongly recommend it

8. **Additional Feedback**
   Please provide any additional feedback or comments (optional).

Participants were also provided with a link to the system documentation (https://www.sanskritvoyager.com/docs) to facilitate their evaluation. Responses are used exclusively for system improvement and research purposes.