# Sanskrit Travelogue: A Large-Scale Unified and Annotated Corpus of Sanskrit Texts

**Giacomo De Luca, Danilo Croce, Roberto Basili**
University of Tor Vergata
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**

We present *Sanskrit Travelogue*, to our knowledge the largest open, unified and richly annotated Sanskrit corpus. Aggregating eight digital libraries, it comprises 12,394 texts, ∼73.1M tokens and >9M segments after de-duplication. A reproducible pipeline standardizes transliteration to IAST, reconciles heterogeneous metadata, preserves structural semantics (verse markers, chapter hierarchies, textual apparatus) and adds automatic annotations. We provide corpus-scale morphosyntactic annotation combining two systems: the ByT5-Sanskrit model for compound and sandhi splitting, and the process-sanskrit library for inflection removal and morphological tagging through a hybrid deterministic-statistical cascade. For each segment we materialize synchronized representations: cleaned, analyzed (sandhi/compound split), stemmed, diacritic-normalized and morphologically tagged. These representations are indexed jointly for retrieval. Both approaches achieve high accuracy (84.61% sentence-level exact matches for ByT5 segmentation, 92.37% correct root extraction for compounds, 95.94% on the Yoga Sūtra). Manual evaluation on the Yoga Sūtra showed 98% correct root extraction when combining both methods, outperforming individual approaches. These annotations enable searching across orthographic sandhi and within compounds, robust lemma-level retrieval despite rich inflectional variation, and provide training material for segmentation and lemmatization while maintaining ambiguity for downstream modeling. We release the annotated corpus as TSV shards, code for corpus acquisition, processing and annotation, a query normalizer, all under a Creative Commons non-commercial license.

**Keywords:** Sanskrit, corpus, sandhi, compounds, morphology, lemmatization

## 1. Introduction

Sanskrit has been in continuous use for roughly three millennia; it underwent an early grammatical crystallization with Pāṇini's *Aṣṭādhyāyī* (Gillon, 2007; Cardona, 1988), while its most ancient layer, Vedic Sanskrit in the *Ṛgveda*, displays even richer morpho-phonology (Jamison and Brereton, 2014). Beyond its cultural centrality across philosophical, religious and literary traditions, Sanskrit provides a stable morpho-syntactic system for diachronic semantic investigation over an exceptionally long time span.

From a computational perspective, Sanskrit remains challenging due to the interaction of free word order, frequent compounding, and orthographically realized sandhi. Written sandhi, while useful for spelling, obscures word boundaries and lexemes, complicating segmentation, dictionary lookup and search. Recent work has advanced Sanskrit word segmentation (SWS) from rule-based pipelines (Huet, 2009) to neural models (Hellwig and Nehrdich, 2018; Krishna et al., 2017), hybrid approaches (Sandhan et al., 2022), byte-level transformers (Nehrdich et al., 2024), and hybrid cascading systems deployed in practical tools (De Luca, 2025). Yet progress in modeling is hampered by the absence of large, unified corpora with consistent metadata and linguistically informed annotations.

This paper introduces *Sanskrit Travelogue*[1], a new open resource that consolidates eight major digital libraries into a single corpus of about 13k works (≈84M tokens before de-duplication, ≈73M tokens across 12.4k de-duplicated texts, >11M segments before deduplication, for $> 9M$ unique segments). A reproducible pipeline harmonizes transliteration and metadata, normalizes markup, and preserves structural features such as verse boundaries and textual apparatus. More importantly, each segment is enriched with morphosyntactic annotation: sandhi and compound resolution, lemmatization with morphological features through multiple synchronized textual views (cleaned, analyzed, stemmed, diacritic-normalized and morphological analysis). Ambiguities are not collapsed but retained, so that philological inspection remains possible and robust training data can be derived.

These aligned representations make the corpus useful well beyond information retrieval: they provide training and evaluation material for segmentation and lemmatization, facilitate lexicographic and semantic resource building (e.g. linking to dictionary headwords as done in the *Sanskrit Sembank* (Hellwig and Biagetti, 2025)), and enable digital humanities research such as critical edition

[1] https://github.com/SanskritVoyager/SanskritTravelogue



*Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026)*, pages 6722–6730
11–16 May 2026. 

support, parallel-text alignment and text reuse detection across large collections. The annotations also support efficient retrieval and navigation: multi-indexed search over the synchronized views allows queries across sandhi boundaries, matching of all inflectional variants of a lemma, and tolerance to transliteration differences. In practice, this enables scalable exploration in systems such as *Sanskrit Voyager* or *DharmaNexus* [2], where navigation relies on corpus-wide annotations.

Our contributions are threefold: (i) a large, unified and provenance-aware corpus of Sanskrit, (ii) a scalable annotation pipeline extending prior hybrid approaches to corpus scale, and (iii) a public release (TSV shards, database dump, query normalizer) under a permissive license. Evaluation shows high annotation quality (90–96% exact match depending on textual complexity). We argue that making annotations available at scale is crucial: they render Sanskrit corpora not only searchable, but also computationally usable for training, evaluation, and linguistic research.

The rest of the paper reviews related resources (Section 2), details corpus construction (Section 3), reports evaluation (Section 4), and concludes with perspectives for reuse (Section 5).

## 2. Existing resources for Sanskrit

A number of digital collections of Sanskrit texts have been developed over the past decades, each with specific strengths but also with limitations in terms of consistency, coverage, or usability. *Sanskrit Travelogue* brings these resources together into a single reconciled corpus. Here we briefly survey the main sources that were integrated, as well as some that were intentionally excluded.

The Göttingen Register of Electronic Texts in Indian Languages (GRETIL)(Grünendahl, 2001) offers more than 800 works in IAST transliteration, distributed in TEI or HTML. It acts as a register for texts gathered from different initiatives, including discontinued projects such as *Gandharva Nāgaram*. While coverage is wide, encoding conventions differ substantially across works, and even within a single file markup may be inconsistently applied.

The SARIT corpus[3], by contrast, is small (about 80 works) but carefully curated. All texts follow TEI P5 encoding, with a consistent schema and detailed documentation. Its limited size is compensated by its high editorial reliability.

The Digital Sanskrit Buddhist Canon (DSBC)[4] focuses on Buddhist texts, including Mahāyāna sūtras, commentaries, and treatises. The project currently lists nearly 1,000 items, of which about 800 are unique texts. The texts are split between Devanāgarī and IAST, with many duplicated in both transliteration schemes. Formatting is minimal, but the collection is indispensable for Buddhist studies.

Another large contribution comes from the Muktabodha Research Institute[5], which provides access to more than 2,000 photographically digitized works from Śaiva traditions and related schools (Vīraśaiva, Pāñcarātra, Śrīvidyā, Śākta, Nātha, Yoga). About 500 of these are distributed as plain text in IAST, usually with minimal metadata and only basic internal segmentation.

A smaller but domain-specific resource is Yogavaiśāradī[6], a web application dedicated to the *Yoga darśana*, which includes 21 works in IAST transliteration. The collection focuses on historical commentaries on the Yoga-sūtra. The commentaries are marked up using CSS classes, allowing them to be distinguished from the root text, though other metadata are sparse.

The DharmaNexus corpus[7], now affiliated with Dharmamitra, is the largest single aggregation, comprising more than 1,700 Sanskrit works. It includes several sources integrated in our project, like GRETIL, DSBC, Muktabodha, and other repositories such as Wisdom Library[8], the Clay Sanskrit Library [9] and Sacred Texts [10]. The quality of the texts is heterogeneous: some files contain metadata or English annotations embedded in the text, and segmentation is often unreliable. Nonetheless, it offers additional pre-analyzed versions and scripts for database integration, making it a valuable technical resource despite inconsistencies.

Not all available resources were included. The Sanskrit Library, Wisdom Library, and TITUS collections largely overlap with GRETIL and DharmaNexus, and were therefore redundant. Similarly, the Sanskrit Wikisource[11] presents just a fraction of the Vedic portion of the Gretil library. The University of Hyderabad corpus contains high-quality formatted texts, also with manual sandhi splitting, but most are modern (20th-century), which fall outside the focus on classical Sanskrit. The Bodleian Library, in collaboration with the Clay Sanskrit Library is in the process of digitising 100 Sanskrit texts, which are still unreleased [12].

[2] https://dharmamitra.org/nexus
[3] http://sarit.indology.info/
[4] https://www.dsbcproject.org/
[5] https://muktalib7.com/DL_CATALOG_ROOT/digital_library.htm
[6] https://kymyogavaisharadi.org/
[7] https://github.com/dharmamitra/DharmaNexus-sanskrit
[8] https://www.wisdomlib.org/sanskrit
[9] https://claysanskritlibrary.org/
[10] https://sacred-texts.com/index.htm
[11] https://sa.wikisource.org/
[12] https://digital.bodleian.ox.ac.uk/collections/sanskrit-manuscripts-clay/

The Hamburg Centre for Tantric Studies hosts a small set of 12 tantric works, but with restrictive copyright claims despite the texts being in the public domain[13]. For these reasons, such corpora were not integrated into the present release. Some small collections present additional challenges. The Kyoto Sangyo University [14] hosts a small collection of Sanskrit texts, including notable treatises on medicine and astronomy. Those texts employ a transliteration convention which precedes the Harvard-Kyoto or the IAST encoding. Similarly, the University of Tsukuba hosts a valuable collection of texts by Dharmakirti[15] in a unique transliteration scheme which is not supported by libraries such as Indic Transliteration[16] or Akshara-mukha[17]. We additionally release two small conversion scripts to IAST that support those two non-standard transliteration schemes. To our knowledge, no existing transliteration library handles this scheme. In sum, these collections provide wide coverage but in highly uneven form. Their reconciliation, including transliteration, markup, and metadata, is essential to create a resource that is both philologically reliable and computationally usable.

## 3. Building Sanskrit Travelogue

The construction of *Sanskrit Travelogue* required a multi-step pipeline: (i) acquisition and consolidation of texts from heterogeneous public collections, (ii) reconciliation of formats and markup conventions, (iii) morphosyntactic annotation through a hybrid cascade, and (iv) storage in a relational schema with multiple synchronized textual views. This section details the first two phases, showing how heterogeneous digital libraries were integrated into a single resource with harmonized provenance, metadata, and encoding. Subsequent sections describe annotation and evaluation.

### 3.1. Aggregating existing corpora

We consolidated eight major public collections (GRETIL, Muktabodha, SARIT, DCS, DSBC, Sanskrit Documents, DharmaNexus/BuddhaNexus, and Yoga-vaiśāradī) through source-appropriate acquisition. Static archives were downloaded from official distribution points, dynamic sites were captured to resolve client-side content, and database exports (e.g., DCS) were ingested from publicly available dumps. All harvesting employed rate-limited requests[18].

To ensure provenance, each item was registered in a manifest that records the source identifier, canonical work label (author, title), original URI, harvest timestamp, licensing status, and file format. Source-specific encodings were preserved at acquisition time to avoid early information loss[19]. For dynamic repositories, the capture included resolving lazy-loaded content and expandable annotations to obtain the underlying text resources[20].

Because several collections overlap, we performed a manually supervised multi-stage de-duplication process both across and within sources; 693 texts were ultimately removed. To avoid removing variant readings and texts from different manuscripts, we applied stringent de-duplication criteria. While this conservative approach likely resulted in some duplicates remaining in the corpus, it minimized the risk of inadvertently removing legitimate textual variants. When multiple witnesses were found, we retained the highest-quality version (richer structure, cleaner text, clearer licensing) and recorded cross-references to alternatives. This aggregation step yields a unified raw store of 12,394 items with harmonized provenance and licensing metadata. In the next subsection we describe format reconciliation and the conversion of all items into a uniform TSV interchange format for downstream annotation and release.

### 3.2. Corpus reconciliation and normalization

Our objective is to turn heterogeneous Sanskrit sources into a structure-preserving, machine-usable corpus while retaining philologically relevant cues. We adopt three principles: (i) neutral acquisition followed by conversion to IAST; (ii) provenance-aware, reversible transformations; and

[13] https://www.gutenberg.org/help/no_sweat_copyright.html

[14] https://www.cc.kyoto-su.ac.jp/~yanom/sanskrit/

[15] https://web.archive.org/web/20121103081809/https://www.logos.tsukuba.ac.jp/~nagasaki/dharmakirti/e-text.html

[16] https://github.com/indic-transliteration/indic_transliteration_py

[17] https://github.com/virtualvinodh/aksharamukha

[18] Acquisition relied on standard web tooling: HTTP clients and HTML parsers (e.g., `requests`, BeautifulSoup), plus headless browser automation (Selenium WebDriver) for client-side rendering. A breadth-first traversal with URL-level deduplication, a 2 s politeness delay, and per-source logs (visited, downloaded, errors) were enforced.

[19] For example, Sanskrit Documents texts were acquired in `.itx` (ITRANS) while preserving directory structure; DCS materials were ingested from MySQL exports and later converted to TSV for downstream processing.

[20] For instance, Yoga-vaiśāradī required executing JavaScript to expand AJAX-driven commentary panels prior to extraction.

(iii) linguistically grounded segmentation that respects verse/prose organization.

Source encodings are detected using Akshara-mukha and converted to IAST for processing; standard Unicode normalization and a consistent punctuation policy are applied, while the original script version is preserved for release[21].

Divergent markup is mapped to a canonical schema (work → part/book → chapter/section → verse or prose unit, with optional *pāda* subdivisions). Marginalia (labels, apparatus, notes) are captured through regular expressions, retained as typed annotations and linked to the nearest structural unit. Where explicit structure is missing, boundaries are inferred from standard delimiters and numbering.[22] Each segment receives a stable identifier based on its position in the work to align multiple textual views.

We regularize structure and metadata with minimal intervention, adapting to each collection as follows:

***GRETIL* (TEI/HTML).** GRETIL is different from SARIT because the TEI encoding is not consistent either across or within texts. For some books, extensive manual intervention was required to restore correct tagging. Taking the text *Viṣṇudharmottara* as an example, we consolidate from a flat paragraph layout into verse-level segments. Label tokens inserted after each *pāda* are merged into verse-level annotations and chapter/verse structure is reconstructed so that each *śloka* is a coherent segment.[23]

***SARIT* (TEI P5).** Encoding is consistent; we retain TEI hierarchy and map it to our canonical schema with minimal loss.

***Muktabodha* (txt).** Hash-delimited bibliographic blocks are promoted to structured metadata while preserving the IAST text.

***Sanskrit Documents* (ITX).** Metadata embedded in comment lines (`%`) are parsed to recover title, author, category and update fields; text boundaries are extracted using collection-specific heuristics[24]. Verses are segmented from numbering conventions (double pipes `||2||`, single-pipe suffixes `|80`, range notations `||1–2||`). Page numbers are likewise extracted using multiple common conventions ("p. 180", "p 32", "page: 140", etc.). [25]

For each work, we produce synchronized artifacts: (i) a segment-level TSV with IAST text, hierarchy coordinates (including explicit verse numbers where applicable) and typed annotations; (ii) a complete IAST plain-text view; and (iii) a metadata record (author, title, source, licensing, acquisition, checksums). These feed the annotation pipeline and the multi-index search layer.[26]

Normalization preserves authorial structure where present and introduces conservative defaults otherwise. Ambiguities (e.g., inconsistent verse numbering, orphaned labels) are flagged in per-work reports for downstream curation. No interpretive emendation is attempted; edits target format, alignment and interoperability for large-scale NLP while keeping philological traces visible in the released data. The output consists of two TSV files, one for the metadata (including title, author, original collection, acquisition details, license, and checksums) and one for the textual segments. The *segments* TSV includes the `book_id`, the `segment_number` inside the book, the `original_text` and multiple aligned textual views (`cleaned_text`, `analyzed_text`, `stemmed_text`, `no_diacritics_text`, together with structural coordinates (chapter/verse hierarchy) and attached annotations (labels, apparatus notes, paratext). This format makes the corpus directly usable for both information retrieval and downstream NLP training.

### 3.3. Annotating Sanskrit morphosyntax

We introduce a novel two-stage architecture that combines neural segmentation with deterministic morphological analysis, extending the hybrid cascade of (De Luca, 2025) to corpus scale while incorporating the ByT5-Sanskrit model (Nehrdich et al., 2024). This design addresses a key limitation: while ByT5 achieves 84.61% sentence-level accuracy for segmentation on the DCS corpus, its performance drops to 61.27% when jointly predicting segmentation, lemmatization, and morphosyntactic tags. Conversely, the process-sanskrit library excels at stemming and morphological analysis but makes errors during non-deterministic sandhi resolution. Our pipeline leverages the complementary strengths of both approaches. Given a normalized

[21] Normalization follows NFC; policy covers anusvāra/visarga, avagraha, danda/double danda, and quotation marks.

[22] Heuristics use danda/double-danda, recurrent headers, and numbering patterns; ambiguous cases are flagged.

[23] Original files insert label tokens after each *pāda*, e.g., `nārāyaṇaṃ namaskṛtya__Vdha_Maṅg1 ...`, fragmenting verse units; consolidation restores full-verse continuity.

[24] For instance, in the case of Sanskrit Documents detection is via a list of several common end-of-text markers (e.g., “Written by”, “Send corrections to”)

[25] Regex extractors cover frequent variants; edge cases are flagged for manual review.

[26] Implementation uses Python with XML/HTML parsing and pattern libraries; code and configuration files will be released for replication.

| Original | Cleaned | Analyzed | No Diacritics | Stemmed | Morphology |
|---|---|---|---|---|---|
| eṣā dhautiḥ ... | eṣā dhautiḥ ... | eṣā dhautiḥ ... | esa dhautih ... | etad dhauti ... | etad_Nom\|Fem\|Sg ... |
| nābhimagnajale ... | nābhimagnajale ... | nābhi-magna- ... | nabhi- ... | nābhi majj ... | nābhi_Cpd ... |
| karābhyāṃ ... | karābhyāṃ ... | karābhyām ... | karabhyam ... | kara kṣālay ... | kara_Ins\|Masc\|Dual ... |
| tāvat prakṣālya ... | tāvat prakṣālya ... | tāvat prakṣālya ... | tavat ... | tāvat ... | tāvat_ind ... |

Table 1: Sample entries from the annotated Sanskrit corpus showing segmentation, morphological analysis, and textual metadata.

IAST segment, processing occurs in three main stages. (1) **Pre-processing**, which entails transliteration detection and conversion to IAST, removal of annotations, verse references, non-alphabetic characters and whitespace normalization. (2) **Neural Segmentation:** ByT5-Sanskrit handles the non-deterministic task of sandhi splitting and compound segmentation. On an NVIDIA A100 with TF32 precision and *torch.compile*, the model processes approximately 3,000 segments per minute in batches of 500. (3) **Stemming and Morphological Analysis:** The segmented output is passed word-by-word into the process-sanskrit library, which uses first a deterministic approach to stem and apply morphosyntactic tags using dictionaries and inflection tables from the University of Cologne [27]. It has been shown (De Luca, 2025) that the majority of the inflected Sanskrit terms can be processed correctly deterministically with just SQL queries and rewriting rules that account for prefixes and suffixes. The deterministic approach works even better after the ByT5 model has removed the ambiguous sandhi cases. When the deterministic approach fails (the term is not in the dictionaries or in the inflection table even with rewriting), the library first cascades in a statistical method through the sanskrit-parser library, which uses a smaller neural model. The results of the previous step are ranked according to morphology and length. If the score is below a confidence threshold, such as in the case of long or technical compounds, the terms are passed to conservative recovery heuristics (right-truncation with targeted rewriting; lexicon-constrained greedy decomposition) to improve coverage without oversplitting. Because most tokens are resolved at the first stage, the pipeline is predominantly CPU-bound and delegates only difficult cases to the neural model. On consumer hardware (M3 MacBook Pro, 8-core) approximately 200 segments per minute are processed, making corpus-scale annotation accessible without specialized GPU infrastructure while reducing energy costs relative to an 'always-on' neural solution.

For each segment we materialize five coordinated textual views, aligned by a stable segment identifier and in addition to the original display text: (1) *cleaned_text*, preserving diacritics while removing non-Sanskrit punctuation and other annotations, including line number references (e.g. "(2.2)"/DCS or "AgBhaist:_2"/GRETIL); (2) *analyzed_text*, the sandhi and compound split version of the segment using ByT5-Sanskrit; (3) *stemmed_text*, reducing surface forms to lemmata and verbal roots using the process-sanskrit library; (4) *no_diacritics_text*, an ASCII-compatible view, tolerant of transliteration variations; (5) *morphology*, containing the morphological tags for each word in the segment[28].

Processed segments are stored in a relational schema that links collections, works and segments. The *segments* table contains the original display text, the five textual views and the (possibly multiple) morphosyntactic analyses; *book_metadata* and *categories* hold bibliographic and collection-level fields. Full-text search is implemented as a composite weighted index over the synchronized views: cleaned receives the highest weight; analyzed receives an intermediate weight; no_diacritics and stemmed receive lower weights.[29] The index supports corpus-scale retrieval with filtering by author, collection, category and text identifiers. Importantly, the same normalization and analysis functions are also applied at query time, so that corpus segments and user queries are represented in a comparable space. Queries undergo transliteration scheme detection, optional conversion to IAST, and cleaning (removal of verse references, page numbers, non-alphabetic and non-apostrophe characters, whitespace normalization). Three retrieval modes are provided: *exact* (structure-preserving), *analyzed* and *stemmed*, similarly to how the segments are preprocessed. Long inputs (e.g., strings $>$40 characters) are routed to a materialized folio-level view that aggregates contiguous segments to maintain latency on paragraph-length text[30]. Python code that uses SQL Alchemy as ORM is

[27] https://github.com/sanskrit-lexicon/csl-inflect

[28] Column names as implemented: `original_text` (display only), `cleaned_text`, `analyzed_text`, `stemmed_text`, `no_diacritics_text`, `morphology`

[29] Weighted full-text index implemented as `tsvector+GIN`; weights correspond to A (cleaned), B (analyzed) and C (no_diacritics/stemmed); queries use `to_tsquery`.

[30] We use a materialized view `folio_search_minimal` for folio-level aggregation; the rerouting threshold is 40 characters.

| Corpus size | |
|---|---|
| Texts (works) | 12,394 |
| Segments | 9,092,023 |
| Tokens (words) | 73,089,921 |
| **Length statistics** | |
| Avg. words per segment | 8.04 |
| Avg. characters per segment | 71.9 |
| Avg. segments per text | 733.52 |
| Avg. words per text | 5896.73 |
| Avg. characters per word (no spaces) | 8.048 |
| **Characters** | |
| Total characters (with spaces) | 653,722,830 |
| Total characters (without spaces) | 588,282,427 |

Table 2: *Sanskrit Travelogue*: corpus statistics (post de-duplication).

released to perform this query normalization and to wrap the corresponding database calls, so that external tools can query the resource without re-implementing preprocessing; at runtime, the dictionary layer maps query tokens to candidate lemmata and inflectional families, enabling compact lemma-based expansions and reducing the need for expensive fuzzy matching[31].

### 3.4. Corpus analysis

We report aggregate statistics for the deduplicated, IAST-normalized release. A *segment* ideally corresponds to a verse (*śloka*) or minimal prose unit, as defined in Section 3. Different collections use different ways to catalog segments. The current release maintains the format used by the collections, splitting the segments only when numerical identifiers (e.g. `|| 2 ||`) are encountered. Counts refer to the post-normalization inventory and exclude duplicates and empty fragments.

The resulting corpus is large in both breadth and depth: more than 12k works and 73M tokens drawn from eight major libraries, harmonized into over 9M consistently annotated segments. Segment lengths are short on average (Table 2), reflecting the predominance of verse quarters and finely split prose. This granularity is crucial: it enables tight alignment among the synchronized textual views (cleaned, analyzed, stemmed, diacritic-normalized, stopword-filtered), supports robust retrieval across orthographic and morphological variation, and provides ML-ready instances for segmentation, lemmatization and semantic tasks at scale.

Beyond offline analysis, the data can be explored interactively in systems that exploit these aligned views. In particular, the accompanying SQL schema is compatible with *Sanskrit Voyager*[32], a navigation and search front-end that (i) queries across sandhi boundaries and within compounds; (ii) performs lemma-based retrieval over all inflectional variants; (iii) tolerates transliteration differences; and (iv) offers faceted browsing by author, collection and work, as well as per-work aggregation (e.g., verse-level hits grouped by text). These functions rely directly on the multiple synchronized representations released with the corpus.

The public release consists of two layers. First, *metadata* TSVs for each work, recording stable identifiers, author/title, source collection, licensing, provenance timestamps and checksums. Second, *segment* TSVs aligned to a stable `segment_id`, containing hierarchy coordinates (work/part/chapter/verse or prose unit), `original_text` (for display), `cleaned_text`, `analyzed_text` (sandhi/compound split), `stemmed_text` (lemmata/roots with features), `no_diacritics_text`, and—when available—lemma/POS/morphological features with dictionary links.

Each row in the segment TSV thus corresponds to a stable segment identifier and contains all aligned textual views and annotations. Files can be queried directly or loaded into the provided PostgreSQL schema for indexed search and interactive navigation. An illustrative excerpt of the TSV structure is shown in Table 1. All data is released on Hugging Face[33] under a permissive Creative Commons non-commercial license, together with processing scripts.

## 4. Evaluating the Annotated Corpus

We evaluate the corpus along two dimensions that correspond to its intended uses: the intrinsic quality of the automatic morphosyntactic analyses (sandhi/compound segmentation and lemmatization), and the effectiveness of the multi-view representations for retrieval. Our aim is not to propose yet another segmentation benchmark, but to assess whether the hybrid cascade adopted here provides reliable annotations for large-scale use and whether the resulting resource is effective for search and downstream NLP.

Byte-level models for Sanskrit segmentation, such as ByT5-Sanskrit, have reported high sentence-level exact matching accuracy on benchmarks derived from the Digital Corpus of Sanskrit

[31] Client library detects/transliterates schemes, applies the same regex cleaners used at indexing time, and issues parameterized queries; most normalization is performed in Python, while PostgreSQL handles matching and ranking.

[32] The software is accessible at https://www.sanskritvoyager.com, and its main features are demonstrated in a short video: https://www.youtube.com/watch?v=FCKlW4NKJEc

[33] https://huggingface.co/SanskritVoyager

(DCS)—90.11% on the 2018 DCS snapshot (Hellwig and Nehrdich, 2018), 93.83% on SIGHUM (Krishna et al., 2017), and 94.29% on the Hackathon set (Krishnan et al., 2020). Manual inspection by the authors, however, has revealed that a substantial proportion of errors reflect problems in the gold data (source mistakes or alternative valid readings), making these figures difficult to interpret. In addition, the stemming policy used in the DCS reduces many nominal forms to verbal roots (e.g. *dagdha* $\rightarrow \sqrt{dah}$), which is problematic for retrieval and lexical research. Following (De Luca, 2025), our pipeline adopts a dictionary-aware policy: if a form is a dictionary headword, it is retained as such, with the verbal root recorded in parallel for linguistic queries. This design choice shapes both our evaluation and the intended uses of the annotations. Intrinsic evaluation of the process-sanskrit library was conducted in two complementary scenarios. First, on the complete *Yoga Sūtra*, measuring segment-level exact match against gold analyses. Second, on a stratified randomised sample of 1,218 compounds from GRETIL, grouped by length (Medium 10–20 characters, Long 20–40, Very Long $>$40), measuring exact decomposition into constituent stems. Gold analyses were created against standard lexica, adjudicating in favor of a single reading in ambiguous cases. Results are summarized in Table 3. The cascade reaches 95.94% exact match on the *Yoga Sūtra* and 92.37% average accuracy on compound decomposition, with graceful degradation on very long forms. To replicate part of the evaluation presented in (De Luca, 2025), we evaluated our enhanced two-stage pipeline on the entire text of the *Yoga Sūtra*. Manual validation reveals that the combined approach achieves 98.05% accuracy (13 errors out of 665 stems to be identified), significantly improving over the 95.94% achieved by process-sanskrit alone.

### 4.1. Error Analysis

As can be seen in Table 4, about half of the errors come from ByT5 oversplitting words that end with *"tā"*, while the remaining half come from the process-sanskrit library making erroneous sandhi predictions or missing particular forms like *"bhāvanātaḥ"* in the inflection table. Since the *"tā"* errors are easily mappable, the annotated dataset could be further cleaned and revised to form a gold-quality annotation. Residual errors tend to cluster in rare or technical compounds, in genuinely ambiguous sandhi contexts (where multiple readings remain plausible), and in prose passages requiring long-distance dependencies. Examples include multi-component compounds in philosophical texts and ambiguous sandhi boundaries in verse commentaries. These limitations are expected, and ambiguity is preserved in the released resource by providing $n$-best alternatives. Errors arise as well in out of vocabulary words, particularly forms ending with the suffix *-tva* (generating abstract terms from nouns, such as *emptiness* from *empty*), which is frequently used but often not recorded in dictionaries. The process-sanskrit library relies on a term lexicon compiled by merging six dictionaries (Monnier Williams, Grassman, Apte, Edgerton, Cappeller and MacDonnell) from the University of Cologne (Cologne University, 2024), plus the *Concise Pali dictionary* (Pali terms are often used in the Pali-Sanskrit hybrid Buddhist literature) totaling 246,955 unique words.

| Type | Total | Errors | Err% | Acc% |
|---|---|---|---|---|
| Very Long | 654 | 58 | 8.87 | 91.13 |
| Long | 377 | 22 | 5.84 | 94.16 |
| Medium | 187 | 13 | 6.95 | 93.05 |
| **Total** | 1,218 | 93 | 7.63 | 92.37 |
| Yoga Sūtra | 665 | 27 | 4.06 | 95.94 |

Table 3: Intrinsic accuracy. Compound decomposition is exact-match by length bucket; *Yoga Sūtra* is segment-level exact match.

Apart from the errors outlined in (Nehrdich et al., 2024), we report a different kind of error with the ByT5-Sanskrit model. Some sentences, such as *"kṣaṇatatkramayoḥ saṃyamād vivekajaṃ jñānam"* [34] are not parsed at all by the model. It is not that the model misclassifies them in the output, but rather that it fails to produce any output for the input sentence. Those segments remain unclassified even when using the Dharmamitra Chrome plugin. The number of segments for which this happens is minimal. In a sample of 1978 segments taken from the Yogācārabhūmi and the Yoga Sutra Bhasya, the number of null outputs is 41, so 2.07%. Currently, in case of undefined, our pipeline passes the untagged segments directly to the process-sanskrit library to parse them in full.

## 5. Conclusion

We present *Sanskrit Travelogue*, a large-scale, unified and automatically annotated corpus that consolidates heterogeneous sources into a single resource with consistent metadata, morphosyntactic analyses, and multiple synchronized textual views. The size and diversity of the corpus extend the coverage of existing resources such as the Digital Corpus of Sanskrit or GRETIL, while the hybrid annotation pipeline provides sustainable accuracy across sandhi, compound resolution, lemmatization and morphological tagging.

The resource supports a range of applications. For computational linguistics, it offers training and

[34] Also: *nirodhasthitikālakramānubhavena nirodhacittakṛtasaṃskārāstitvam anumeyam*

| Incorrect Segmentation | Correct Segmentation |
|---|---|
| pratyayāḥ+lambanā | pratyayāḥ+alambanā |
| bhāvana+ātaḥ | bhāvanātaḥ |
| ātma tā | ātmatā |
| tāna tā | tānatā |
| vāhi tā | vāhitā |
| ekāgra tā | ekāgratā |
| anya tā khyāti | anya tā |
| kramān+yatvam | krama+anyatvam |
| āloka+nyā+asāt | ni+āsāt |
| ana+vacche+adāt | anavacchedāt |
| kṣayaḥ+adayau | kṣayaḥ+udayau |
| aneka+id | aneka+iṣām |
| ana+vadhaḥ+raṇam | avadhāraṇa |

Table 4: Examples of segmentation errors in the automatic annotation of the Yoga Sūtra

evaluation data for segmentation, morphological analysis, and lemmatization, as well as large-scale inputs for representation learning. For corpus-based philology, it provides a basis for diachronic and comparative studies across traditions, genres, and periods. For information access, the multi-view indexing layer enables efficient search that is robust to compounding, sandhi, inflectional variation, and transliteration differences. Error patterns of the automatic analyses are not arbitrary: they tend to generate either unattested surface forms or morphologically inconsistent parses, making dictionary-based validation and constraint checking viable strategies for automated quality control. This opens a path to semi-automatic curation workflows and the progressive construction of gold-standard subsets.

All data, processing code, and indexing schemas will be released under a permissive Creative Commons license. The dataset will be distributed on Hugging Face, accompanied by documentation and a lightweight client library compatible with the *Sanskrit Voyager*[35] platform for interactive exploration. We hope that this resource will foster reuse across computational linguistics, digital philology, and the broader study of Sanskrit as a diachronic and cross-cultural vehicle of knowledge.

## 6. Limitations

The annotation pipeline is entirely automated. While manual evaluation on selected texts demonstrates high accuracy (98% on the *Yoga Sūtra*), comprehensive validation across the full corpus and literary genres remains incomplete. Our de-duplication strategy prioritizes preservation of textual variations over aggressive duplicate removal. This conservative approach retains legitimate variants but may include some redundant copies. Complete de-duplication would require manual inspection to distinguish genuine textual variations from identical duplicates. Some source libraries, particularly GRETIL, exhibit inconsistent encoding both within and across texts. We developed a systematic semi-automatic transformation pipeline that converts presentational markup to semantic tags, extracts chapter structure from bracketed notation, and normalizes verse numbering patterns. Due to time constraints, these corrections were applied only to a subset of the GRETIL corpus. Complete normalization of remaining texts would require extending this pipeline to handle additional edge cases. Similarly, automatic extraction of structural metadata (page numbers, verse markers, chapter divisions) from non-annotated sources like DSBC occasionally fails due to format inconsistencies in these sources.

[35] https://www.sanskritvoyager.com

## 7. Bibliographical References


George R Cardona. 1988. *Pāṇini : his work and its traditions*. Motilal Banarsidass, Delhi, India.

Cologne University. 2024. Cologne digital Sanskrit dictionaries. Online resource. Accessed on February 19, 2025.

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.

Brendan S Gillon. 2007. Pāṇini's" aṣṭādhyāyī" and linguistic theory. *Journal of Indian philosophy*, 35(5/6):445–468.

Oliver Hellwig and Erica Biagetti. 2025. The sanskrit sembank. *Language Resources and Evaluation*. Advance online publication (2025-07-16).

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.

Stephanie W Jamison and Joel P Brereton. 2014. *The Rigveda: 3-Volume Set*. Oxford University Press.

Amrith Krishna, Pavan Kumar 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, Vancouver, Canada. Association for Computational Linguistics.

Sriram Krishnan, Amba Kulkarni, and Gérard Huet. 2020. Validation and normalization of dcs corpus using sanskrit heritage tools to build a tagged gold corpus.

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*.

## 8. Language Resource References

Grünendahl, Reinhold. 2001. *GRETIL: Göttingen Register of Electronic Texts in Indian Languages*. Niedersächsische Staats- und Universitätsbibliothek Göttingen. PID https://gretil.sub.uni-goettingen.de/. Comprehensive repository of machine-readable texts in Sanskrit and other Indian languages.