LDA Topic Modeling for pramāṇa Texts: A Case Study in Sanskrit NLP Corpus Building

Tyler Neill
Leipzig University
Institute for Indology and Central Asian Studies
Schillerstraße 6, 04109
Leipzig, Germany
tyler.g.neill@gmail.com

Abstract

Sanskrit texts in epistemology, metaphysics, and logic (i.e., pramāṇa texts) remain underrepresented in computational work. To begin to remedy this, a 3.5 million-token digital corpus has been prepared for document- and word-level analysis, and its potential demonstrated through Latent Dirichlet Allocation (LDA) topic modeling. Attention is also given to data consistency issues, with special reference to the SARIT corpus.

1 Credits

This research was supported by DFG Project 279803509 "Digitale kritische Edition des *Nyāyabhāṣya*"[1] and by the Humboldt Chair of Digital Humanities at the University of Leipzig, especially Dr. Thomas Köntges. Special thanks also to conversation partner Yuki Kyogoku.

2 Introduction

Sanskrit texts concerned with epistemology, metaphysics, and logic (hereafter: pramāṇa texts) have so far been underrepresented in computational work. Digitized texts are available, but supervised word-level analysis is lacking, and so corpus-level operations remain mostly limited to manual plain-text searching.

In response to this, by building on the knowledge-base of the Digital Corpus of Sanskrit (DCS) (Hellwig, 2010–2019) and looking toward a comparably robust future for pramāṇa studies, a 3.5 million-token corpus of pramāṇa texts has been prepared for word-level NLP, and its potential demonstrated through Latent Dirichlet Allocation (LDA) topic modeling. Attention is also given to data consistency issues, with special reference to the SARIT corpus, and with the goal of continuing to improve existing text corpora, including ultimately with rich annotation.

3 Overview

The process of building the present corpus for use with LDA topic modeling can be idealized as the following sequence of nine steps, in three phases:

Phase	Steps
Obtain Data	(1) Collect E-Texts, (2) Choose Versions, (3) Extract XML to Plain-Text
Prep for LDA	(4) Create Doc IDs, (5) Clean Content, (6) Resize Docs, (7) Segment Words
Implement LDA	(8) Model Topics, (9) Query Topics and Documents

Table 1: Workflow Overview

In reality, Steps 3 through 5 were found to frequently overlap, especially in those cases involving more of the data consistency issues discussed in Section 9.

[1] See also the earlier FWF project out of which this grew: https://www.istb.univie.ac.at/nyaya/.

Nyāya-Vaiśeṣika	Tokens (10^3)	Bauddha	Tokens (10^3)	Other	Tokens (10^3)
Vātsyāyana	45.8	Dharmakīrti	64.5	Jaimini	16.5
Praśastapāda	11.0	Candrakīrti	77.9	Kumārila Bhaṭṭa	50.1
Uddyotakara	117.0	Śāntarakṣita	38.8	Sucarita Miśra	172.8
Jayanta Bhaṭṭa	209.7	Arcaṭa	57.0	Madhva	29.4
Bhāsarvajña	165.5	Kamalaśīla	268.9	Jayatīrtha	364.6
Śrīdhara	95.7	Prajñākaragupta	235.4	(*Yuktidīpikā*)	56.1
Vācaspati Miśra	314.8	Karṇakagomin	161.5	Māṭhara	17.8
Udayana	149.9	Durveka Miśra	120.1	Patañjali	17.1
Gaṅgeśa	34.7	Jñānaśrīmitra	155.3	Siddhasena	27.1
Pravāduka	29.8	Ratnakīrti	48.8	Abhayadeva Sūri	37.4
Vāgīśvara Bhaṭṭa	41.1	Manorathanandin	108.7	Abhinavagupta	45.6
Total	1242.9	Total	1336.9	Total	834.5

Table 2: Corpus Makeup by Well-Represented Authors

4 Obtaining Data

The approximately 70 pramāṇa texts included in the corpus so far — totaling about 3.5 million tokens — were chosen out of a practical need of the aforementioned *Nyāyabhāṣya* project to be able to more effectively cross-reference relevant texts, above all from the voluminous Nyāya-Vaiśeṣika and Bauddha traditions. A representative sample of authors and their cumulative token counts in the corpus so far is presented in Table 2.[2] Many of the corresponding e-texts are incomplete, owing to imperfect editing or digitization. In addition, many more such pramāṇa texts are available not only online (easily over twice as much) but also in private offline collections. Even more textual material awaits basic digitization. Owing to a lack of resources, however, virtually no new material could be digitized here, e.g., through OCR and/or double-keyboarding.

4.1 Collecting Available E-Texts

Among existing digital collections, the open online repositories GRETIL and SARIT emerged as most relevant for Nyāya- and Bauddha-centric pramāṇa studies.[3] All work based on data derived from these sources can therefore be shared without hesitation. In those few cases where exceptions were made for clearly superior text versions in still-private collections of personal colleagues, original and cleaned versions of such texts cannot yet be shared in full.[4]

[2]For more detail on this list, along with nearly all data and tools discussed in this paper, see the associated GitHub page: `https://github.com/tylergneill/pramana-nlp`.

[3]Despite the sophisticated analysis of its other texts, the DCS has few materials directly related to pramāṇa; all are either complete and of small size (e.g. *Viṃśatikākārikā* and *-Vṛtti*) or of large size (e.g. *Prasannapadā*, *Abhidharmakośabhāṣya*, *Nyāyabhāṣya*, *Sarvadarśanasaṃgraha*) and very incomplete (2% or less). Nor do TITUS, The Sanskrit Library, or Muktabodha have significant materials for this genre.
The "Digital Resources" corpus of the University of Hyderabad (`http://sanskrit.uohyd.ac.in/Corpus/`) includes a few such texts (some even sandhi-splitted) but not enough from the Leipzig project "wishlist" to warrant inclusion in this first round of work; a second round would certainly utilize the digitizations of Vāsudeva's *Padapañcikā* on Bhāsarvajña's *Nyāyasāra*, Cinnambhaṭṭa's *Prakāśikā* on Keśavamiśra's *Tarkabhāṣā*, Rucidattamiśra's *Prakāśa* on Gaṅgeśa's *Tattvacintāmaṇi*, and Dharmarājādhvarin's *Tarkacūḍāmaṇi* thereon, among others.
Other digital projects of note for pramāṇa studies are: Ono Motoi's sandhi analysis of Dharmakīrti's works for KWIC-indexation (now housed on GRETIL and included here); R.E. Emmerick's indexation database and programs including bhela.exe (now lost to obsolescence); and Yasuhiro Okazaki's analyzed index of Uddyotakara's *Nyāyavārttika* (not used here; see: `http://user.numazu-ct.ac.jp/~nozawa/b/okazaki/readme.htm#n.con`).

[4]For example, Uddyotakara's *Nyāyavārttika*, Bhaṭṭavāgīśvara's *Nyāyasūtratātparyadīpikā*, and Pravāduka's (a.k.a. Gambhīravaṃśaja's) *Nyāyasūtravivaraṇa*, provided by Prof. Karin Preisendanz in Vienna, as well as Ernst Steinkellner's edition of Dharmakīrti's *Pramāṇaviniścaya* I & II, provided by Hiroko Matsuoka in Leipzig.

4.2 Choosing One E-Text Version Per Work

In comparing and selecting from among digital text versions, data quality, both of edition and digitization, was considered to be of secondary importance relative to two other NLP needs: quantity of text and clarity of structural markup. Only in a few cases was a uniquely available version of a text deemed to be of insufficient quality for inclusion in the analysis presented here.[5] Occasional exceptions to the one-work-one-file rule were made for base texts quoted in commentaries (e.g., Kaṇāda's *Vaiśeṣikasūtra* within Candrānanda's *Ṭīkā* thereon).

4.3 Extracting XML to Plain-Text

As a third, overlapping criterion, special priority was given to the SARIT corpus, nearly half of which (by file size) consists of pramāṇa texts. Along with these texts' relatively good data quality, their hierarchical TEI/XML encoding seemed worth trying to exploit for the current purpose. As a positive side-effect of this inclusion, an XSLT workflow was developed to extract the XML to plain-text. For reasons explored below (Section 9.1), multiple transforms were crafted for each text and then daisy-chained together with Python's *lxml* library. During extraction, rendering of structural elements into machine-readable identifiers was sensitive both to philological understanding of the texts and to the particular NLP purpose at hand.

5 LDA Topic Modeling as Guiding Use Case

LDA topic modeling, as the special purview of the *Nyāyabhāṣya* project's Digital Humanities specialist Dr. Köntges, was chosen on pragmatic grounds as the best means for stimulating potentially useful NLP experimentation on the envisioned corpus of pramāṇa texts.

In machine learning, topic models comprise a family of probabilistic generative models for detecting latent semantic structures (called topics) in a textual corpus. Among these, the relatively recently-developed LDA model,[6] characterized by its use of sparse Dirichlet priors for the word-topic and topic-document distributions,[7] has proven popular for its ability to produce more readily meaningful, human-interpretable results even with smaller datasets and limited computational power. Consequently, the literature on it is already quite vast,[8] and its software implementations are increasingly numerous and user-friendly.[9] In recent years, humanities scholars working in a variety of modern and historical languages have used LDA to support their research[10] in an ever-expanding variety of ways, from studying societal trends reflected in newspapers (Nelson, 2011; Block, 2016), to exploring poetic themes and motifs (Rhody, 2012; Navarro-Colorado, 2018), to direct authorship verification (Savoy, 2013; Seroussi et al., 2014). For Classical Sanskrit, it has also been used to scrutinize authorship, albeit indirectly, by helping to control for significance of other parameters.[11]

[5] For example: GRETIL's versions of Vyāsatīrtha Rāghavendra's *Nyāyadīpatarkatāṇḍava* (transcription error-rate too high), Madhva's Mahābhāratatattvanirṇaya (encoding corrupt), and Śākyabuddhi's *Pramāṇavārttikaṭīkā* (diplomatic transcription of a damaged manuscript).

[6] The original paper is Blei (2003).

[7] These sparse Dirichlet priors "encode the intuition that documents cover only a small set of topics and that topics use only a small set of words frequently" (Anouncia and Wiil, 2018, p. 271).

[8] See, e.g., David Mimno's annotated bibliography: `https://mimno.infosci.cornell.edu/topics.html`.

[9] Used here are open-source tools by Dr. Köntges: (Meletē)ToPān (2018), built on the R libraries *lda* and *LDAvis*, and Metallo (2018). Other options include Java-based MALLET and various Python machine-learning packages like *gensim*.

[10] This subtle point, that digital humanities methods do not supplant, but support traditional humanities approaches, is made nicely by David Blei (2012):

> Note that the statistical models are meant to help interpret and understand texts; it is still the scholar's job to do the actual interpreting and understanding. A model of texts, built with a particular theory in mind, cannot provide evidence for the theory. (After all, the theory is built into the assumptions of the model.) Rather, the hope is that the model helps point us to such evidence. Using humanist texts to do humanist scholarship is the job of a humanist.

[11] Low-dimensional topic models ($k <= 10$) are used by Hellwig (2017) to determine which linguistic features to exclude from authorship layer analysis.

Most important for the present undertaking in corpus building, however, is the basic data requirement in LDA for units at two levels: 1) words and 2) documents.

5.1 Data Need #1: Segmented Words

The first of these, words, is here accepted as equivalent to segmented tokens, namely as provided by the Hellwig-Nehrdich Sanskrit Sandhi and Compound Splitter tool (Hellwig and Nehrdich, 2018), using the provided model pre-trained on the four-million-token DCS corpus.[12] Splitted output from this tool was then modified only slightly, replacing hyphens with space, and these spaces, along with pre-existing spaces, were in turn used to define tokens for this corpus.[13] For example, *kiñcit*, written as such, would be one token, whereas *kiṃ tu* would be two. Efforts should be made to standardize tokenization for this corpus in the future. Similarly, the Splitter's natural error rate increases if orthography is not standardized, as is the case here.[14] Nevertheless, given the tool's ease of use, it was seen as preferable, from the humanities perspective, to work with relatively more familiar, human-interpretable units than to work with, for example, raw character n-grams for the LDA modeling.[15] Moreover, LDA being a statistical method, the relatively large amount of data involved (namely, several million tokens) helps to improve the signal-to-noise ratio.

A further possible concern is that this Splitter, as used here, does not perform any sort of lemmatization or stemming, as have been aimed at by, for example, SanskritTagger or the reading-focused systems, especially Reader Companion and Saṃsādhanī.[16] Thus, *arthaḥ*, *arthau*, *arthāḥ*, *artham*, *arthān*, *arthena*, etc. remain distinct items here rather than all being abstracted to a single word, *artha*. However, whether this is a problem is again an empirical question; such stemming may itself result in the loss of some useful information, such as collocations of certain verbs with certain nouns in certain case endings, or genre-specific uses of certain verb tenses.[17] The current Splitter, therefore, provides a sufficient starting point for experimentation.

5.2 Data Need #2: Sized and Coherent Documents

The second requirement for LDA is segmentation of a corpus into properly sized and suitably coherent documents. Whereas the importance of sizing is generally well-known, the necessity of document coherence, as with the issue of stemming just addressed, may depend on one's specific goals.[18] Toward this end, effort was made by Hellwig to "not transgress adhyāya bound-

[12] Code at `https://github.com/OliverHellwig/sanskrit/tree/master/papers/2018emnlp`. Splitting the entire pramāṇa corpus took only a few hours on the average-strength personal computer used here: a 2017 MacBook Air with a 1.8 GHz Intel Core i5 processor and 8 GB RAM running macOS High Sierra 10.13.6. For another large-scale demonstration of the Splitter's power, see Nehrdich's visualization of quotations within the GRETIL corpus, based on fasttext vector representations of sequences with a fixed length of six tokens, at `https://github.com/sebastian-nehrdich/gretil-quotations`. For a descriptive introduction, see: `http://list.indology.info/pipermail/indology_list.indology.info/2019-February/049348.html`.

[13] This includes the token counts in Table 2 above. The largest pramāṇa text cleaned and splitted so far (but not yet included in the corpus discussed here) was Someśvara Bhaṭṭa's *Nyāyasudhā*, on Kumārila Bhaṭṭa's *Tantravārttika*, sourced from SARIT. It is roughly half a million words long, i.e., one-third the size of the *Mahābhārata*.

[14] The default error rate is summarized on the GitHub page as "~15% on the level of text lines", meaning that "about 85% of all lines processed with the model don't contain wrong Sandhi or compound resolutions." For more on the theoretical accuracy limit, as well as on further limitations related to text genres and orthography, see §5.2 "Model Selection" and §5.3 "Comparison with Baseline Models" in Hellwig and Nehrdich (2018), including sentence-accuracies for non-standardized *Nyāyamañjarī* test sentences, esp. 60.2% for the model "$rcNN^{split}_{short}$". Other immediate drawbacks of using the pre-trained model include: an input limit of 128 characters at a time (compensated for with chunking before splitting) and hyphens indifferently outputted for both intra-compound and inter-word splits (unimportant for LDA).

[15] Not yet tested is the possibility of using n-grams alongside segmented words in a "bootstrapping" effort; cp. Dr. Köntges' upcoming work on LDA bootstrapping with morphological normalization and translation.

[16] Respectively: Hellwig (2009), Goyal et al. (2012), and Kulkarni (2009).

[17] Cp., e.g., the importance of the Spanish preterite form *fue* in an LDA topic concerned with time in Navarro-Colorado (2018). Cp. also use of the Sanskrit imperfect in narrative literature in Hellwig (2017, passim).

[18] For discussion of the importance of size constraints, see Tang et al. (2014), on which the range of words-per-document adopted here is based. For discussion of optimizing topic concentration by using paragraphs to segment documents, as opposed to foregoing all such structural markers (including chapter headings) in favor of simple fixed-length documents for a corpus of 19th-century English novels, see section 6.2 "What is a Document?" in

aries" (2017, p. 145). Here, too, despite the more diverse nature of the śāstric corpus, the challenge of using structural markup was accepted, in part to shed light on encoding issues in this developing body of material. In practice, this meant first seeking out any and all available structural markup — whether in the form of section headers, numbering, whitespace (especially indentation and line breaks), punctuation distinctions like double vs. single daṇḍas, or, in the case of SARIT, XML element types and attribute values — and operationalizing it with unique, machine-readable conventions in plain-text. In addition to basic sections, higher-level groupings thereof were also marked (see Section 6 for details).

These preliminary subdivisions of text, or document candidates, could then be automatically transformed into the final LDA training documents using a two-step resizing algorithm: 1) subdivide document candidates which exceed the maximum length, using punctuation and whitespace as lower-level indicators to guide where a safe split can occur; and 2) combine adjacent document candidates whose length is below the minimum, using the grouping markup as a higher-level indicator to guide which boundaries should not be transgressed. The target size range was set at approximately 50–200 words per document,[19] or 300–1000 IAST characters (pre-cleaning), relying on a conservative average of 7 characters per word.[20] Finally, the resulting training documents each received a unique, machine-readable identifier automatically reformulated from identifiers manually secured during initial cleaning, so as to facilitate meaningful interpretation during analysis (see, e.g., Section 8).[21]

6 Data Cleaning

The above-described need for maximally useful word- and document-segmentation for LDA prompted the development of practical encoding standards as well as tools for enforcing these standards. This cleaning process involved the greatest amount of manual effort, relying heavily on regular expressions.

Content was standardized to IAST transliteration[22] and stored as UTF-8. Orthographic variation, including "optional sandhis", has unfortunately not yet been controlled for, which does result in systematic Splitter errors;[23] this should either be standardized in the future or else the Splitter model should be retrained for orthographic substyles.

Punctuation was standardized in certain respects, especially dashes and whitespace: em-dash was used only for sentential punctuation; en-dash only for ranges; hyphen only for pre-existing manual sandhi-splits;[24] and underscore only for new manual sandhi-splits in rare cases of compounds longer than 128 characters (for the sake of the pre-trained Splitter model). Tab was used only for metrical material; space only for separating words from each other and from punctuation marks; and newline only for marking the start of new sections.[25] In this way, these special characters could more effectively help guide document- and word-segmentation before

Boyd-Graber et al. (2017, pp. 70–71).

[19] Cp. the use of sections each containing "approximately 30 ślokas" and thus "an average length of 404 words (= lexical units)" in Hellwig (2017, p. 154).

[20] Such a proxy is necessary because document resizing occurs before word segmentation in this workflow, since punctuation is used for the former and removed in the latter. It is also assumed here that use of IAST instead of, say, SLP1, with the latter's theoretically preferable one-phoneme-one-character principle, is not problematic, since letters are relatively evenly distributed throughout documents, and since LDA treats words as simple strings.

[21] Cp. use of the Canonical Text Services protocol (`http://cite-architecture.org/`) by the Open Greek and Latin Project (`https://www.dh.uni-leipzig.de/wo/projects/open-greek-and-latin-project/`) for its identifiers. Here, a pragmatic decision was made to opt for simpler, more familiar title abbreviations for now.

[22] Transliteration was performed, for reasons of familiarity and also for included meter detection features, with the author's own small Python library, available on GitHub at `https://github.com/tylergneill/Skrutable`. Other transliteration toolkits, such as that at `https://github.com/sanskrit-coders/indic_transliteration`, should work equally well.

[23] See fn. 14 above.

[24] This occurred mostly in Ono's Dharmakīrti texts, which were in any case mechanically re-sandhified during pre-processing in order to ensure more uniform Splitter results. These texts may eventually also prove useful for comparing manual and automatic splitting of pramāṇa material.

[25] For metrical or sūtra texts with extensive structural markup, these "sections" could be verse-halves or smaller.

ultimately being filtered out in final preprocessing.

Finally, brackets were also allocated structural markup functions: square brackets were used only for identifying the beginnings of document candidates; curly brackets only for marking higher-level groupings of document candidates; angle brackets only for tertiary structural information useful for reading but not needed for the present purpose; and parentheses only for certain kinds of philological notes, for example on related passages, also not needed here. Other philological material, especially variant or unclear readings, whether found in-line or in footnotes, was either deleted from this corpus or flattened into a single, post-correction text. This required a surprising amount of tedious and often haphazard manual work, which should become more avoidable in the future (for more detail, see Section 9.2).

Cleaned Text	Note
<iti pratyakṣasyānumānatvaparīkṣāprakaraṇam>	End of Previous Prakaraṇa
{avayaviparīkṣāprakaraṇam}	Document Group: New Prakaraṇa
[2.1.33]	Document Candidate
("sādhyatvād avayavini sandehaḥ")	Editorial Markup
kāraṇebhyo dravyāntaram utpadyata iti sādhyam etat.	Text Content
kim punar atra sādhyam.	...
kim avyatireko 'thāvayavīti.	...
...	...
ataḥ "sādhyatvād avayavini sandehaḥ" ity ayuktam.	(In-Line Sūtra Quotation)
itaś ca sādhyatvād avayavini sandeha iti na yuktam ...	...

Table 3: Example of Cleaned Text for NV_2.1.33

To more efficiently enforce these standards, a two-part validator script was written in Python, firstly to check for permitted structural patterns as indicated by bracket markup, and secondly to check for permitted characters and sequences thereof. In case of deviations, the script generated a verbose alert to assist in manual correction.

To recap: After e-texts had been collected and most useful versions chosen, usable structure was sought out and highlighted with in-house markup, including during plain-text extraction from XML where needed. Thereafter, structure and content were laboriously standardized for all texts with the help of a custom-built validator tool. Beyond this point, final preprocessing occurred automatically: Extraneous elements were removed, document candidates were resized, final documents were word-splitted, and the results were reassociated with appropriate identifiers in a two-column CSV file for use with the topic modeling software.

7 Modeling Topics with LDA and Visualizing Structure

One application of LDA topic modeling of philological interest is direct interpretation of the automatically discovered topics. This information is contained in the resulting ϕ table describing the word-topic distributions, and it lends itself well to visualization.

For example, using ToPān (Figure 1) to train an LDA topic model on 67 pramāṇa texts segmented into words and documents as characterized above and with near-default settings[26] resulted in fifty topics, all human-interpretable, of which half are presented here, identified both by the respective fifteen top words (adjusted for "relevance")[27] and by an interpretive label based on manual scrutiny of the ϕ table.

[26] $\alpha = 0.02$, $\eta = 0.02$, and seed = 73, but $k = 50$ and number of iterations = 1000. Twelve most frequent function words (indeclinables and pronouns) were also removed as stopwords for training, à la Schofield (2017), summarized at `https://mimno.infosci.cornell.edu/publications.html`. In addition, but only after training, a further eighty-two function words were removed for the sake of more meaningful interpretation of ϕ values.

[27] $\lambda = 0.8$. See Sievert & Shirley (2014), and note log normalization: λ * *log*(p(w|t)) + (1-λ) * *log*(p(w|t)/p(w)).

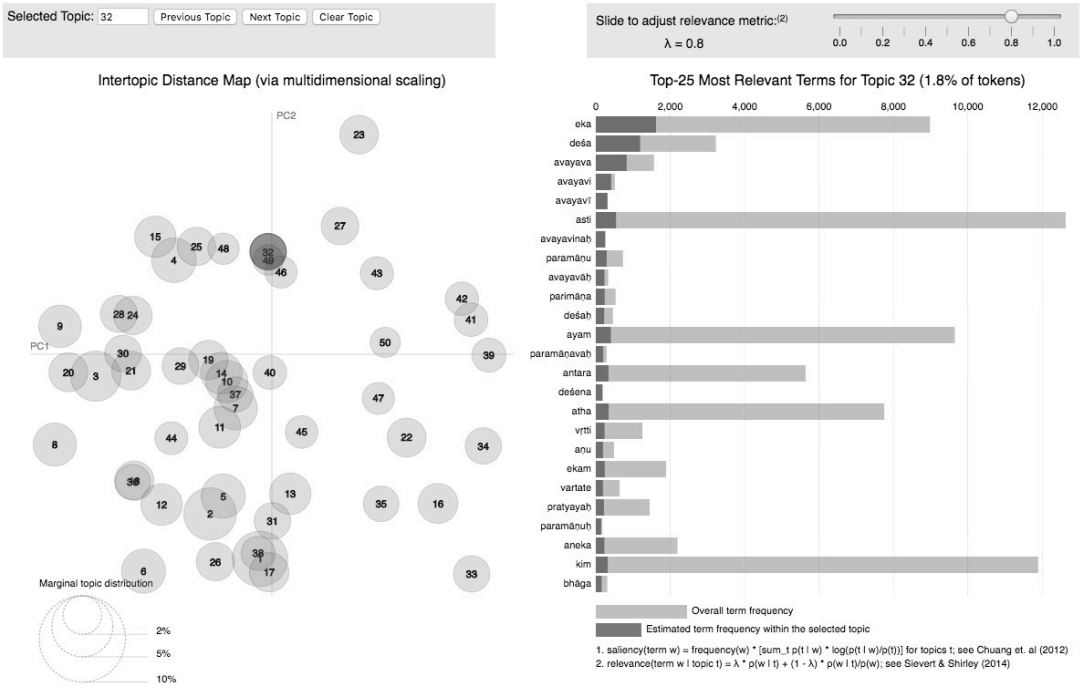

Figure 1: Visualization of Fifty Topics with *LDAvis* in ToPān.
Left: Marginal word-topic probabilities plotted against 2-D PCA of fifty topics.
Right: Top twenty-five words of Topic 32 ($\lambda = 0.8$), with topic and corpus frequencies.

Topic #	Top Fifteen Words	Interpretive Label
4	kārya kāraṇa sahakāri kāryam bīja sāmagrī svabhāva janana aṅkura śakti śaktiḥ eka hetu janaka sāmarthyam	causation
10	prakāśa nīla prakāśaḥ rūpa ātma rūpam grāhya ātmā jñāna grāhaka ākāra saṃvid prakāśate nīlam ābhāsa	Bauddha non-dual perception
11	jñānam jñāna indriya viṣaya pratyakṣam artha jñānasya pratyakṣa viṣayam vijñānam akṣa jam rūpa kalpanā grahaṇam	perceptual cognitive process
14	vikalpa ākāra vastu artha ākāraḥ bāhya vikalpaḥ vāsanā rūpa pratibhāsaḥ pratibhāsa vikalpasya viṣayaḥ sāmānya viṣaya	images and conceptuality
15	bheda bhedaḥ eka bhedāt bhinna abheda bhede abhedaḥ bhedena dharma aneka ekam bhedasya bhedam rūpa	difference
16	brahma mokṣa ānanda bhagavat maya śrutiḥ anna śruti viṣṇu jñāna mukti viṣṇuḥ arthaḥ sadā devānām	Dvaita soteriology
17	nigraha pakṣa sādhana sthānam pratijñā artham sthāna para kathā uttara artha tattva siddhāntaḥ doṣa jalpa	Nyāya method
20	abhāva abhāvaḥ bhāva vastu abhāvasya bhāvaḥ anya rūpa virodhaḥ vidhi niṣedha pratiṣedha abhāvayoḥ virodha niṣedhaḥ	affirmation and negation
22	duḥkha sukha rāga duḥkham sukham ātma tattva doṣa dveṣa saṃsāra nivṛttiḥ avidyā pravṛtti rāgaḥ janma	Nyāya soteriology
23	dravya saṃyoga guṇa vibhāga karma kāraṇa dvi saṃyogaḥ guru ākāśa dravyam mahat samavāyi parimāṇa kāraṇam	Vaiśeṣika ontology

Table 4: Philological Interpretation of Ten out of First Twenty-Five LDA Topics.
Based on ϕ values, relevance-adjusted ($\lambda = 0.8$), excluding eighty-two further stopwords.

Topic #	Top Fifteen Words	Interpretive Label
26	pramāṇa artha pramāṇam pravṛtti jñānam prāmāṇyam prameya niścaya kriyā niścayaḥ phalam viṣaya prameyam prāmāṇya pravṛttiḥ	pramāṇa
27	rūpa sparśa pṛthivī cakṣuḥ gandha indriya śabda rasa guṇa pradīpa śrotra grahaṇam tejaḥ śabdaḥ indriyam	sensation
29	sat asat kāraṇa kāraṇam kāryam kārya sattā asataḥ cit sarvam utpatti prāk sataḥ utpattiḥ sattvam	Sāṃkhya pre-existent effect
32	eka deśa avayava avayavi avayavī avayavinaḥ paramāṇu avayavāḥ parimāṇa deśaḥ paramāṇavaḥ antara deśena vṛtti aṇu	atoms, parts, and wholes
35	phala svarga vidhi phalam karma hiṃsā kāmaḥ vidhiḥ sādhana putra yāga artha vidheḥ yajeta codanā	Vedic sacrifice
36	rajata mithyā bādhaka satya rajatam svapna bādhya sākṣi bādhaḥ sat śukti jñāna asat bhrānti mithyātvam	error
38	prāmāṇyam veda āpta prāmāṇya pramāṇa artha āgama aprāmāṇyam vākya pramāṇam puruṣa doṣa vaktṛ apauruṣeya svatas	trustworthy speech
39	pañca prakṛti vyaktam rajaḥ pradhānam prakṛtiḥ avyaktam vikāra tamaḥ sattva mahat avyakta sargaḥ vṛttiḥ tanmātrāṇi	Sāṃkhya metaphysics
40	smṛti pūrva smṛtiḥ anubhava smaraṇam smaraṇa saṃskāra smṛteḥ anubhavaḥ kāla saṃskāraḥ anubhūta viṣaya jñānam jñāna	experience and recollection
41	karma śarīra śarīram icchā īśvaraḥ īśvara prayatna dharma śarīrasya deha adharma phala karmaṇaḥ cetanā bhoga	karma
42	bhavanti viśeṣāḥ dharmāḥ sarve santi hetavaḥ syuḥ viśeṣa arthāḥ yeṣām kecid śabdāḥ anye teṣu bhāvāḥ	plural words
43	indriya manaḥ ātma manasaḥ śarīra yugapad jñāna sukha viṣaya artha icchā cakṣuḥ jñānam sannikarṣa indriyāṇām	Nyāya prameyas related to the self
45	kriyā kāraka kartṛ karma karaṇa artha vyāpāra vyāpāraḥ dhātu karaṇam arthaḥ bhāvanā kriyām karoti kriyāyāḥ	action
47	aham puruṣa puruṣaḥ buddhi puruṣasya ātmā artham buddhiḥ arthaḥ ātmanaḥ ātmānam buddheḥ prakṛtiḥ mama bhoktā	Sāṃkhya on self and other
48	viśeṣaṇa viśeṣya samavāyaḥ ghaṭa samavāya bhū sambandha ghaṭaḥ viśeṣaṇam viśiṣṭa ādhāra sambandhaḥ paṭa paṭaḥ guṇa	qualification

Table 5: Further Philological Interpretation of Fifteen out of Remaining Twenty-Five LDA Topics.

8 Using Topics for Information Retrieval

Another computational application of interest to philologists, that of calculating similarity among portions of text, can to some extent also be approached directly with these same topic modeling results, namely by vectorizing documents according to their topic distributions and measuring their distance from each other in topic-space.[28] The relevant information for this is found in the θ table describing the topic-document distributions.

For example, using Metallo with default settings[29] to compare documents according to their Manhattan distance in topic-space, one can query topics and documents of interest to a particular research question — here, say, the present author's own dissertation topic: the ontological whole (*avayavī*) in Bhāsarvajña's *Nyāyabhūṣaṇa*. Manual inspection of the fifty discovered topics quickly reveals that Topic 32 (see Table 5 above) will likely be relevant. Metallo then easily generates a list of arbitrarily many documents best exemplifying this topic, or in other words, documents closest to that particular basis vector in the topic-space (see Table 6). It also allows

[28] Ideally, topic distribution would be only one among a number of linguistic features used to characterize documents for information retrieval. The implementation here is therefore mainly for the purpose of demonstration.

[29] Significance parameter = 0.1. Note also that by default, all topics are weighted equally.

for direct querying of any desired document, say, *NBhū*_104,6^1[30] (beginning of the *avayavī* discussion), for arbitrarily many documents closest to it in topic-space, as seen in Figure 2 and Tables 7 and 8.

Rank	Document Identifier	Topic 32
1	*NV*_4.2.7	98.8%
2	*NVTṬ*_4,2.10.1–4,2.10.2^2–4,2.11.1	98.7%
3	*NV*_2.1.31^2	98.4%
4	*NSV*_4.2.7	98.4%
5	*NV*_2.1.32^4	97.2%
6	*NV*_2.1.32^8	95.4%
7	*NBh*_2.1.36.1–2.1.36.2	95.1%
14	*NSV*_4.2.8–4.2.9	90.6%
15	*NSV*_4.2.16	90.3%
20	*NSV*_4.2.11–4.2.13	88.3%
21	*NBh*_2.1.36.3	87.9%
22	*VVṛ*_12	87.8%
24	*VVṛ*_14^2	87.0%
25	*VVṛ*_14^1	87.0%
26	*NBh*_4.2.16.1–4.2.16.3	86.6%
27	*NBh*_2.1.31.3–2.1.31.5	86.4%
35	*NVTṬ*_2,1.32.1^7	82.6%
39	*NM*_9,2.430.325	80.7%
40	*VVṛ*_13	80.6%
43	*NBhū*_106,3	80.0%
46	*NVTṬ*_4,2.7.1	79.3%
48	*NTD*_4.2.7	79.3%
51	*NBhū*_111,24^1	78.8%
52	*NVTṬ*_4,2.25.1^3	78.6%
56	*NTD*_4.2.10	77.0%
65	*PVV*_1.87,0–1.87,1	75.5%
72	*PVin*_1.38.3	74.2%
75	*NK*_59.4^2	74.1%
76	*NSu*_2.2.66cd.3–2.2.66cd.4	74.0%
81	*NTD*_2.1.39	72.9%
86	*NTD*_4.2.15	71.5%
91	*VNṬ*_80,1^2	70.5%
94	*NBhū*_104,6^2	70.1%
97	*NM*_9,2.430.322	69.8%
100	*YŚ*_3.44.5–3.44.6	69.3%

Table 6: Selected Documents in which Topic 32 is Most Dominant.
Top four only shown for *NV*, *NVTṬ*, *NSV*, *NBh*, *VVṛ*, *NTD*. (Sixty-five more not shown.)
All shown for *NM*, *NBhū*, *PVV*, *NK*, *NSu*, *VNṬ*, *YŚ*.

[30] As seen here by the "^1" notation marking a document automatically subdivided in resizing, queriable documents are currently limited to those somewhat artificial ones used in modeling. It is also possible to extrapolate to new data, but this has not yet been done here.

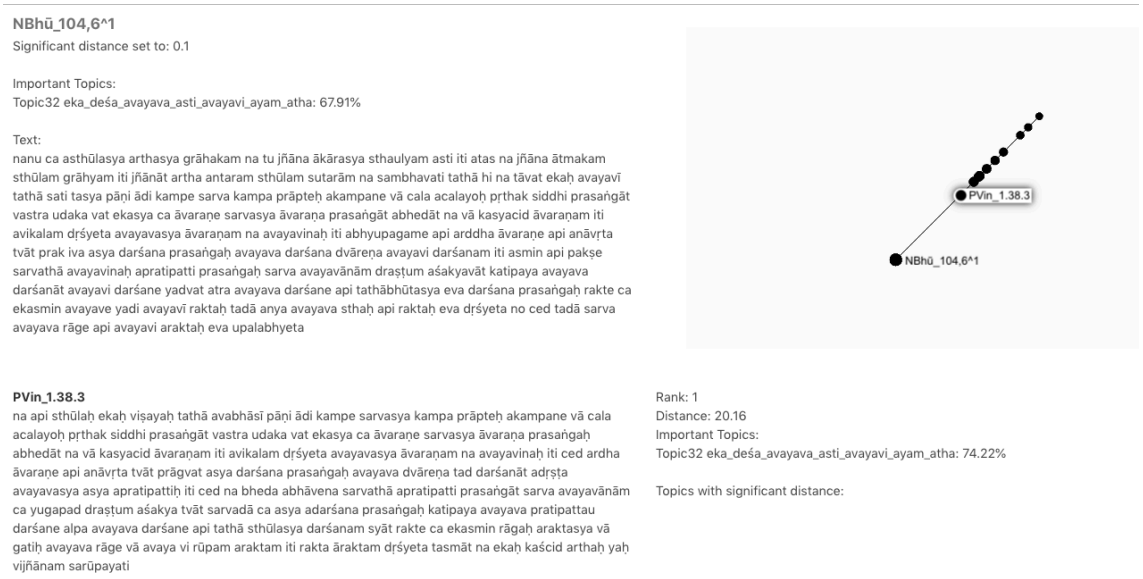

Figure 2: Screenshot of Metallo "view" Query on Document *NBhū*_104,6^1

Rank	*PVin*	*NBh*	*NBhū*	*NV*
0			104,6^1	
1	1.38.3			
7			104,6^2	
13		4.2.24.3		
15			110,12	
17			106,3	
18		4.2.16.1–4.2.16.3		
20		2.1.36.7		
25				2.1.31^10
26				2.1.33^30
27				2.1.32^4
28				2.1.33^31
30		2.1.36.4		
31				4.2.26
34				2.1.36^3
35				2.1.33^33
36				4.2.25^3
37			123,21	
41				2.1.31^3
42				1.1.14^14
43			130,15^2	
45		2.1.36.3		
47		2.1.35.3–2.1.35.4		
49				4.1.13

Table 7: Selected Documents Closest to *NBhū*_104,6^1 in Topic-Space.
Emphasis on: *PVin*, *NBh*, *NBhū*, *NV*.
Not shown: *NM*, *NSV*, *NSu*, *NTD*, *VVṛ*, *NK*, *NVTṬ*, *ĀTV*, *PVV*.

Rank	Document Identifier	Text Preview (Segmented, Unproofread)
0	*NBhū*_104,6^1	... jñānāt artha antaram sthūlam sutarām na sambhavati tathā hi na tāvat ekaḥ avayavī tathā sati tasya pāṇi ādi kampe sarva kampa prāpteḥ akampane vā cala acalayoḥ pṛthak ...
1	*PVin*_1.38.3	na api sthūlaḥ ekaḥ viṣayaḥ tathā pāṇi ādi kampe sarvasya kampa prāpteḥ akampane vā cala acalayoḥ pṛthak siddhi prasaṅgāt vastra udaka vat ...
13	*NBh*_4.2.24.3	... uktam ca atra sparśavān aṇuḥ sparśavatoḥ aṇvoḥ pratighātāt vyavadhāyakaḥ na sāvayava tvāt sparśavat tvāt ca vyavadhāne sati aṇu saṃyogaḥ na āśrayam vyāpnoti ...
18	*NBh*_4.2.16.1–4.2.16.3	... niravayava tvam tu paramāṇoḥ vibhāgaiḥ alpatara prasaṅgasya yatas na alpīyaḥ tatra avasthānāt loṣṭasya khalu pravibhajyamāna avayavasya alpataram alpatamam ...
20	*NBh*_2.1.36.7	... bhavataḥ tena vijñāyate yat mahat tat ekam iti aṇu amahatsu samūha atiśaya grahaṇam mahat pratyayaḥ iti ced saḥ ayam aṇuṣu mahat pratyayaḥ atasmin tat iti pratyayaḥ bhavati ...
7	*NBhū*_104,6^2	vṛtti anupapatteḥ ca avayavī na asti tathā hi gavi śṛṅgam iti laukikam śṛṅge gauḥ iti alaukikam tatas yadi avayavini avayavāḥ varttante tadā ...
15	*NBhū*_110,12	nanu eka avayava kampane api anya avayavānām akampanāt asti cala acala tvam tena bheda siddhiḥ tatas kim aniṣṭam yadi nāma avayavānām cala acala tvena bhedaḥ tatas ...
17	*NBhū*_106,3	itas ca na asti avayavī buddhyā vivecane anupalambhāt na hi ayam tantuḥ ayam tantuḥ iti evam buddhyā pṛthak kriyamāṇeṣu avayaveṣu tad anyaḥ avayavī pratibhāti ...
25	*NV*_2.1.31^10	... atha manuṣe na asmābhiḥ avayavi dravyāṇi kāni cit pratipadyante kim tu teṣu eva parama aṇuṣu paraspara pratyāsatti upasaṃgraheṇa saṃsthāna viśeṣa avasthiteṣu ...
26	*NV*_2.1.33^30	... na tantavaḥ tantūnām avayavāḥ iti viruddhaḥ artha antara pratyākhyānāt ca avayavaḥ avayavī iti etat na syāt yat api idam ucyate ye avayavāḥ avayavinaḥ artha antaram ...
27	*NV*_2.1.32^4	tasmāt ekasmin na kārtsnaḥ vartate iti na api eka deśena vartate na hi asya kāraṇa vyatirekeṇa anye eka deśāḥ santi sa ayam eka deśa upalabdhau avayavi upalabhyamānaḥ na kṛtsnaḥ upalabhyate ...

Table 8: Detail on Ten Documents Close to *NBhū*_104,6^1 in Topic-Space.
In this case, *PVin*_1.38.3, ranked first, is in fact the direct source of the non-verbatim quotation.

9 Data Consistency Issues

These tentative results, encouraging though they may be, stand to be improved not only through more sophisticated application of NLP methods, but also through increased attention to data consistency. Besides systematic tokenization and orthography issues (addressed in Section 5.1) and unsystematic typographical or even editing errors (not yet prioritized here), three additional sets of systematic data consistency issues were revealed through the process of preparing this corpus. These are advanced here as the low-hanging fruit of improving textual data for future Sanskrit NLP work. The first issue applies at the level of documents and relates to being able to effectively manipulate these through meaningful identifiers, while the second and third are concerned with data loss at the level of individual words. In each case, special attention is paid to the SARIT texts so as to further encourage their use for NLP purposes.

9.1 Structural Markup and Identifiers

The essential structural challenge in such corpus-level computational work is to be able to refer to every single piece of text in the corpus with a unique and, if at all possible, meaningful identifier, in order to be able to effectively coordinate retrieval and human use after processing. In the texts used here, however, structural markup for the purpose of creating such identifiers was often less than easily available. Sometimes, only physical features of the edition, rather than logical features of the text, were found to be marked, even when the latter might have been possible (e.g., the digitization of Durveka Miśra's *Hetubinduṭīkāloka* lacking the structure of the underlying *Hetubindu* or *Hetubinduṭīkā*). Sometimes, numerical structural markup was only found mixed in among textual content (e.g., Abhinavagupta's *Īśvarapratyabhijñāvivṛtivimarśinī*). Sometimes, important section information was marked only with the verbal headers or trailers of the printed edition rather than with numbers (e.g., Vinītadeva's *Nyāyabinduṭīkā*).

Of course, some markup issues may reflect citation difficulties within the philological field itself; for example, citation conventions for texts with continuously interwoven prose and metrical (or aphoristic) material may be more varied than for other texts.[31] Similarly, when (or if) creating paragraphs in such prose texts, editors must often make a substantial interpretive departure from the available manuscript evidence. Thus, as the philological understanding of the interrelationships among parts of a given text gradually improves, so too might the corresponding structural markup in digitized texts also be expected to do so.[32]

In other cases, however, it seems that basic encoding work has just been left undone, whether for lack of time or resources, or through a preference for adhering literally to the source edition, which, for better or worse, allows one to postpone further questions concerning structural annotation. Looking forward, insofar as these digitizations can receive more attention, and as more computational projects are attempted with them, the field should continue[33] to gradually move in the direction of the Canonical Text Services protocol. This protocol encourages explicit and usually numerical reference conventions for the sake of unambiguous citation and automatic processing, and its implementation has been admirably exemplified in recent years (also with TEI/XML markup) by the Open Greek and Latin Project (OGL).[34]

Structural Markup and Identifiers in SARIT

The existing SARIT stylesheet transforms proved difficult to understand and adapt for the current purposes, and thus it was decided to utilize the situation as an exercise in understanding the diversity of structures encoded in that corpus. Experimentation quickly revealed that, in contrast to texts in the OGL corpus, where a single XPath expression in the <TEIheader> explicitly identifies the depth at which textual information will be found, the texts in the SARIT corpus varied so much in their use of main structural elements — , <p>, <lg>, <quote>, <q>, etc. — that it was not possible to write and use straightforward XSL transforms that could apply to multiple files, much less to use the XML library of a given programming language (e.g. Python or Golang) to easily unmarshall the structure and expose the textual data.[35] For example, while for some texts, logical structure was encoded using only a single level of elements (e.g., sūtra sections in Vātsyāyana's *Nyāyabhāṣya*), for others, any number of levels of nested s could be used for the same purpose (e.g., Jñānaśrīmitra's *Nibandhāvali* and Prajñākaragupta's *Pramāṇavārttikālaṅkāra*). Meanwhile, still other texts were structured not

[31] Take, for example, Prajñākaragupta's *Pramāṇavārttikālaṅkāra*. It's not always clear whether one should refer to a piece of the prose commentary with the help of a numbered Dharmakīrti verse quoted nearby, or with Prajñākaragupta's own nearby and numbered verses, or simply with the edition page and line numbers.

[32] Cp., e.g., *Nyāyabhūṣaṇa* topical headers and paragraph divisions by editor Yogīndrānanda (1968) with those of S. Yamakami (2002) for the avayavī section at `http://www.cc.kyoto-su.ac.jp/~yamakami/synopsis.html`.

[33] For thoughts so far, see, e.g., Ollett (2014).

[34] See, e.g., the OGL texts in the Scaife Viewer online reading environment: `https://scaife.perseus.org/`.

[35] Cp. such a mass unmarshalling script for OGL texts at `https://github.com/ThomasK81/TEItoCEX`.
Cp. also the simple, two-level, chapter-verse structure of DCS data as exported from the SanskritTagger in XML form, reflecting top-down, NLP-driven decision making from the very beginning. (A version of the Tagger capable of performing this export was secured with the kind help of Oliver Hellwig.)

according to logical structure but rather according to physical structure of the edition. For example, Jayantabhaṭṭa's *Nyāyamañjarī*, printed on the top halves of pages in the book, was therefore encoded as <quote> elements inserted at unpredictable depths, i.e., within <p> or <q> elements, within the supervening modern *Ṭippanī* commentary, following page breaks. This proved especially difficult to understand and deal with from a perspective seeking natural language. Thus, new transforms had to be individually crafted for each of the fifteen SARIT texts used. While this does provide temporary access to the plain-text information, suggestions will be made to modify the SARIT source files so that they adhere to a smaller number of structural patterns that can be explicitly noted in their respective headers.

9.2 Editorial Markup

Also reflecting a still-developing state of editing and understanding, many digitizations of printed editions literally reproduce or add editorial markup — especially variant readings, including additions, deletions, and substitutions of variable length — which can be quite idiosyncratic and not always thoroughly explained in accompanying digitization metadata. For example, see the table below, based on Durveka Miśra's *Hetubinduṭīkāloka* (parenthetical editorial notes turn out to be reporting on the corresponding text in Arcaṭa):

Page	Text (with Editorial Note)	Suggested Change
254	... tadutpattāv eveti(tpattyā veti) vivakṣitam |	replacement
279	a(nya)thā "nirvikalpakabodhena...	insertion
280	anadhigacchann iti (gaṃcchadi)ti |	none?

Table 9: Examples of Inconsistent Editorial Markup

Insofar as it is not possible to automatically flatten such alternatives into a single text, the flow of natural language will be compromised, and words lost. The straightforward solution is to anticipate such flattening — either through XML transforms or simple search-and-replace routines — with consistent use of some unambiguous notation. This does, however, of course require substantial additional investment of time and expertise. Extensive notes taken during the corpus cleaning here should hopefully contribute to such improvements for the future.

Editorial Markup in SARIT

The use of <choice> elements in XML is a perfect way to address this situation, yet the SARIT texts were found to apply this solution only unevenly, leaving many instances of editorial markup uninterpreted as found in the printed edition. For example, as reported in the metadata of Karṇakagomin's *Pramāṇavārttikavṛttiṭīkā*, although many round brackets (i.e., parentheses) and square brackets have been successfully interpreted — as <ref>, <note type='correction'>, and <supplied resp='#ed-rs'> — others have simply been left as is: "All other round brackets (227 occurrences) were encoded as <hi rend='brackets'>" and "All other square brackets (19 occurrences) were encoded as <hi rend='squarebrackets'>". In other cases (e.g., Vācaspati Miśra's *Tattvavaiśāradī*), these editorial notes were left untouched. Such cases require further philological scrutiny in order to allow for consistent extraction of natural language.

9.3 Whitespace

In the printed representation of Sanskrit texts, one can distinguish between two basic conventions, or perhaps styles, of using whitespace between words: 1) maximal use of whitespace, usually associated with Roman transliteration and prioritizing separate phonemes and words, and 2) conservative use of whitespace, usually associated with Indic scripts and prioritizing ligatures as found in the underlying manuscript tradition. Each style has its strengths and weaknesses, e.g., assuming more work on the part of the editor or digitizer and less on the part of the reader (first style) or vice versa (second style). The point of distinguishing these two

styles, however, is not to advocate for one over the other,[36] but rather to distinguish both from outright spacing errors. That is, it should be trivial for an NLP researcher to quickly filter out all markup and obtain a clean, consistent representation of either one style or the other.

In practice, however, this was often found not to be the case, suggesting that whitespace has not yet been conceived of as containing as much information as other character types. To take but one small example from the digitization of Candrakīrti's *Prasannapadā* (prose section preceding 27.19):

... saṃsāraprabandhamupalabhya śāśvata mātmānaṃ parikalpayāmaḥ |

Here, the "conservative" style is found, but with a spurious space. Each such instance represents the effective loss of one or more words in segmentation. Many of these errors do follow certain patterns, such that regular expressions can be part of a standardization solution, but there are limits to what such language-blind methods can detect.[37]

Whitespace in SARIT

For its own part, SARIT experiences this same whitespace consistency issue, but it also introduces novel difficulties with its handling of in-line annotations, i.e., XML node() elements placed within text() elements. For example, consider the following six representative examples in the digitization of Mokṣākaragupta's *Tarkabhāṣā* (transliterated, XML elements simplified):

Space	Proper	Improper
Left	kumbhakārasya <note n="45-1"/>kartṛtvam	pratyakṣa <note n="4-1"/>mabhidhīyate
Right	-mataśrutyai<note n="1-1"/> tarkabhāṣā	balāda<note n="5-2"/> bhyupagatam
None	parokṣatva<note n="18-1"/>pratipādanāya	-pādaiḥ<note n="41-0"/>kāryatvasya

Table 10: Examples of Inconsistent Whitespace in SARIT Texts

It thus becomes impossible to systematically extract the expected result.

Particularly problematic were <lb> (and to a lesser extent <pb>) elements containing the break="no" attribute, as these were not infrequently found to occur adjacent to other <lb> or <pb> elements not possessing this attribute, as well as adjacent to simple whitespace, thereby rendering the attribute ineffective and compromising word segmentation. A particularly dramatic example is found in Jñānaśrīmitra's *Nibandhāvali* (complex whitespace simplified):

... pariṇāma<lb break="no"/> <lb/> <pb n="257"/> <lb/>paramparāparicayasya ...

In such cases, ensuring proper segmentation necessitates removal of competing elements, which can then cause problems of its own, e.g., if line number counts are required for constructing identifiers. On the other hand, this break="no" attribute was sometimes simply not used when it should have been. For example, in Śāntarakṣita's *Vādanyāyaṭīkā* (67,4–5; element simplified) (also observe not one but two whitespaces):

sadādyaviśeṣavi <lb/> ṣayā ...

Fortunately, once identified, fixing such problems is relatively easy with the help of regular expressions and SARIT's recommended Git-based workflow, although again, expertise and time are required. The XSLT workflow described above can also be further modified to help diagnose such issues and assess how much progress has been made in this direction at any given point.

10 Conclusion

This demonstration of working through a certain subset of Sanskrit pramāṇa texts with LDA topic modeling has been of a preliminary character. Nevertheless, it provides a valuable window

[36] From the perspective of NLP, machine-learning-based systems, ever more the rule rather than the exception, can be made to handle both separately, just as OCR systems can be trained for multiple fonts.

[37] E.g., a regex built to find a final consonant migrating to the beginning of the next word, as in the example given, would fail to distinguish between "-m ucyate" and "mucyate", both valid sequences, depending on context.

onto the state of digitization of a large number of e-texts of ever-increasing importance to the scholarly community and shows what potential they have for further computational research. Moreover, issues encountered with LDA and pramāṇa texts in particular should generalize well to many other NLP methods and Sanskrit subgenres. Until a database of supervised word-segmentation, such as found in the DCS, is secured also for such specialized texts, perhaps with the help of a collaborative, online annotation system, the remarks here will hopefully help interested parties continue to improve digitization workflows in ways that anticipate the kind of accessible, citable, machine-actionable text — to be processed, for instance, with an unsupervised segmenter — that will be most needed for a variety of corpus-linguistic and information retrieval applications in the future.

References

S. Margret Anouncia and Uffe Kock Wiil. 2018. *Knowledge Computing and its Applications: Knowledge Computing in Specific Domains*, volume 2. Springer Nature Singapore.

David Blei, Andrew Ng, and Michael Jordan. 2003. Latent Dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022.

David Blei. 2012. Topic modeling and digital humanities. *Journal of Digital Humanities*, 2(1), Winter.

Sharon Block. 2016. Doing more with digitization. *Common-place.org*, 6(2), January.

Jordan Boyd-Graber, Yuening Hu, and David Mimno. 2017. Applications of topic models. *Foundations and Trends® in Information Retrieval*, 20(20):1–154.

Pawan Goyal, Gérard Huet, Amba Kulkarni, Peter Scharf, and Ralph Bunker. 2012. A distributed platform for Sanskrit processing. In *24th International Conference on Computational Linguistics (COLING), Mumbai*.

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, Brussels, Belgium, October-November. Association for Computational Linguistics.

Oliver Hellwig. 2009. SanskritTagger: A stochastic lexical and POS tagger for Sanskrit. In Gérard Huet, Amba Kulkarni, and Peter Scharf, editors, *Sanskrit Computational Linguistics*, pages 266–277.

Oliver Hellwig. 2010–2019. DCS - The Digital Corpus of Sanskrit. http://www.sanskrit-linguistics.org/dcs/index.php.

Oliver Hellwig. 2017. Stratifying the Mahābhārata: The textual position of the Bhagavadgītā. *Indo-Iranian Journal*, 60:132–169, January.

Thomas Koentges and J. R. Schmid. 2018. ThomasK81/ToPan: Rbiter. January. http://doi.org/10.5281/zenodo.1149062.

Thomas Koentges and Jeffrey C. Witt. 2018. ThomasK81/Metallo: HumboldtBonpland. October. http://dx.doi.org/10.5281/zenodo.1445773.

Amba Kulkarni and Devanand Shukl. 2009. Sanskrit morphological analyser: Some issues. *Indian Linguistics*, 70(1–4):169–177.

Borja Navarro-Colorado. 2018. On poetic topic modeling: Extracting themes and motifs from a corpus of Spanish poetry. *Frontiers in Digital Humanities*, 5.

Robert K. Nelson. 2011. Of monsters, men — and topic modeling. *The New York Times*, May.

Andrew Ollett. 2014. Sarit-prasāraṇam: Developing SARIT beyond 'Search and Retrieval'. Posted on Academia.edu. Slides from a talk given in Oxford ('Buddhism and Digital Humanities,' organized by Jan Westerhoff.

Lisa M. Rhody. 2012. Topic modeling and figurative language. *Journal of Digital Humanities*, 2(1), Winter.

Jacques Savoy. 2013. Authorship attribution based on a probabilistic topic model. *Information Processing & Management*, 49:341–354, 01.

Alexandra Schofield, Måns Magnusson, and David Mimno. 2017. Pulling out the stops: Rethinking stopword removal for topic models. pages 432–436, April.

Yanir Seroussi, Ingrid Zukerman, and Fabian Bohnert. 2014. Authorship attribution with topic models. *Computational Linguist*, 40(2):269–310, June.

Carson Sievert and Kenneth Shirley. 2014. LDAvis: A method for visualizing and interpreting topics. In *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, pages 63–70, Baltimore, Maryland, USA, June. Association for Computational Linguistics.

Jian Tang, Zhaoshi Meng, XuanLong Nguyen, Qiaozhu Mei, and Ming Zhang. 2014. Understanding the limiting factors of topic modeling via posterior contraction analysis. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, pages I–190–I–198. JMLR.org.

Svāmī Yogīndrānanda and Bhāsarvajña. 1968. *Nyāyabhūṣaṇam: śrīmadācāryabhāsarvajñapraṇītasya nyāyasārasya svopajñaṃ vyākhyānam*. Ṣaddarśana Prakāśana Pratiṣṭhānam : Prāpti-sthānam Udāsīna Saṃskṛta Vidyālaya, Vārāṇasī.