2017

Concept Maps of Indian Shaastra Texts: Simplifying the Study and Mining of Indic Knowledge

SAI SUSARLA (SAI.SUSARLA@GMAIL.COM)

M.A. LAKSHMITHATHACHAR

M.A. ALWAR

Concept Maps for Indian Shaastra Texts: Simplifying the Study and Mining of Indic Knowledge

Abstract

This paper presents a canonical schema and method for collaborative concept mapping of Indic texts that simplifies the study of Indic texts by modern seekers as well as enables semantic search and indexing tools. We have prototyped our ideas by augmenting a popular text annotation tool and applying it to annotate tarka sangraha text. We demonstrate how our concept mapping method enables sophisticated navigational queries on flat Indic texts, which are necessary to develop user-friendly E-readers for Indic shaastra content.

Introduction

There is growing curiosity and interest among youngsters to explore ancient Indian knowledge for contemporary applications in areas such as sustainable living, holistic wellbeing and knowledge processing. However, several barriers are hindering such exploration by modern seekers:

- lack of widespread, flexible access of indic knowledge sources for western-educated audience,
- unfamiliarity with the language or scripts of those sources,
- paucity of scholarship to interpret them, coupled with dwindling scholars who can cross-correlate multiple sources to glean insights,
- lack of training in the Indic method of inquiry and discourse,
- paucity of tools to search, process, correlate and mine that knowledge at scale,
- lack of standardized data exchange formats and interfaces for interoperability of tools.

The challenge is to bridge the physical and educational divide between traditional experts and modern seekers so that more seekers can get trained to interpret Indic knowledge content and gain mastery to apply it to modern scenarios. One of the chief strengths of traditional shaastra experts is their ability to map out the key concepts of a text, how they relate to each other, and what to emphasize or gloss over based on the objective of the study. In short, a shaastra expert uses his/her familiarity to mentally build a *concept map* of the flat text, and uses it to assist students in navigating the text without getting overwhelmed. The Indic shaastra texts written in Samskrit are even more amenable to such mapping due to their rigorous and precise methodology of exposition.

Mind maps (Beel et al, 2009) are a widely recognized effective tool to assist in cognition and summarization of concepts. However, unless mind maps are closely tied to the original text for fidelity and created with a standardized schema to extract and navigate key concepts, they are only useful as a teaching aid for humans, but not to encode knowledge for automatic processing. The latter capability is crucial for making Indic knowledge exploitable for contemporary uses at scale. Some examples of useful applications include an automated tool for Vaastu rule compliance checking, and a symptom-based herbal cure database.

Hence to effectively exploit Indic scientific knowledge for contemporary uses, we need to reduce the didactic burden of scholars for training large numbers of students and also encode the knowledge contained in Indic texts for machine-assisted cognition and mining. Both the objectives can be achieved together by encoding shaastra text content as a multi-dimensional semantic network with a standard schema for navigation.

This paper presents a methodology for semi-automatic concept mapping of Samskrit shaastra texts that exploits their inherent semantic structure (e.g., tarka sangraha and its commentary). We envision a web-based collaborative editor for experts to collaboratively overlay a shaastra text or its commentary with a concept map representation based on Nyaaya and Mimaamsa principles of shaabda-bodha. Specifically, we prototype a tool that allows users to annotate the text with three levels of tagging: (i) categorize individual sentences into preset types to guide in automated extraction of their semantic meaning, (ii) categorize inter-sentence relationships into a finite set of relation types based on miimaamsa sangati concept, to guide in discourse analysis, and (iii) associate a conceptual summary to individual sentences or a group that capture an expert's paraphrasing of the knowledge therein. The tool is designed to automatically augment the above expert-supplied information with output of automated linguistic analysis such as morphological and karaka analysis. Lastly, the tool enables user correction of automated analysis output. Together, they enable a comprehensive multi-dimensional, machine-processable view of the underlying shaastra text that is validated by experts. When defining sentence types, we draw inspiration from the six Sutra types employed in Paanini's Ashtaadhyaayi. Examples of sentence types include term-defining, exemplary, explanatory, and exception sentences.

Once built, such a map can be used by students to get a conceptual summary of the shaastra content and drill down for details. Another application is to support semantic queries into the shaastra text such as "is concept X a sub-concept of Y?". A third application is auto-generated shaastra-specific dictionary or glossary of terms.

For this paper, we have evaluated the concept mapping methodology by applying it to a popular Nyaaya shaastra commentary text called tarka sangraha and its commentary text called aaloka vyaakhya. Our objective is to validate the definitions of sentence types and sentence relation types and how well they cover the variety encountered in these texts. This paper presents our methodology and experience.

Concept Mapping: Target Use Cases

Our primary objective behind concept mapping is to support navigation of a flat Indic text document based on the concepts discussed in it. A concept map is a network representation of a content's topical structure that facilitates its manual study and automated processing such as indexing, search and visualization.

Intuitively, a text describes a collection of concepts and how they relate to each other. For each concept, it can provide definitions of supporting terms, elucidation of its meaning and illustrative examples to delineate its scope and contrast relative to other concepts. Each of those is accomplished through one or more sentences in the text. Though the number of

distinct types of relations between concepts can be unbounded, we need to identify a finite subset that are essential for text navigation.

Requirements of an E-reader for Indic Texts

To illustrate the nature of assistance required to study an Indic shaastra text, here are the common tasks that a reader often needs to perform when studying a text:

- View a concise **summary** of a given text and drill down to view further details as needed.
- Enumerate and navigate the high-level concepts of a text, or its **table of contents**, skipping over details
- Look up the **definition** of a term or concept,
- Browse the **explanation** of a specific concept and its sub-topics to the desired level of detail,
- View illustrative **examples** of a term or concept to understand it better,
- **References**: View where a term is used or referenced to understand its contextual significance,
- Add **expert notes** to a word, phrase or sentence in the source text to help other readers,
- Browse and **rate others' comments** on a phrase or sentence in the source text.

In the rest of this paper, we discuss a solution that meets these requirements.

Approach to Solution

Supporting the E-reading tasks listed above requires access to the conceptual and semantic structure of the flat text and other texts referenced by it. Hence to support those tasks, the first step is to annotate or *semantically tag* the words, phrases and/or sentences of the text and their relationships. The tags impose a *concept overlay network or map* structure on the flat text. Unlike flat text that only supports sequential navigation through its words / sentences, a concept map provides alternate paths to navigate the text that can be more meaningful semantically (e.g., summary view, detailed view, concept-wise index). Given a concept map, E-reading tasks can be performed via navigational queries as outlined later. Semantic tags can be manually supplied by human experts or inferred by computational linguistics tools. To ensure sanity and to remove ambiguity inherent in natural language processing, human experts must be allowed to make corrections to auto-generated tags, which must be fed back into the tools for improved results.

Concept Map

Formally, a *concept map* is represented as a graph data structure consisting of a set of vertices *V* and a set of edges *E*. Each *vertex* represents a word, phrase or sentence in the source text or additional information derived from it. A vertex can have multiple attributes. Mandatory attributes include the item's location in the source text, its type (e.g, phrase or sentence) and semantic category tag. In case of flat text, location is the text offset and size of the string. Other vertex attributes include grammatical properties of the word, *kaaraka*

tag and user comments / notes. Each *edge* represents a relationship between two vertices, and is labeled by the relation tag.

Given this representation, an example navigational query can be of the form, “show all vertices reachable via the edges labeled “sub-topic”. Its result reveals the nested hierarchy of the topics covered by the text. The text referred by multiple vertices can overlap, e.g., vertices can be defined for the sentence as well as its individual words in which case they are implicitly connected via the syntactic nesting relation.

Design of Semantic Tags

In this section, we discuss the design of semantic tags and its rationale. Semantic tags associate higher-level semantic information with a vertex or edge of a concept map that enables concept-based search and navigation of the source Indic text. To ensure that concept maps are amenable to automated navigation, we impose certain guidelines on the design of tags and their values.

- **Well-known Presets**: A tag should at least have a finite set of predefined values. This enables formulation of automated queries for popular navigation styles.
- **Object-oriented Tag Types**: To ensure independent evolution of tag semantics and the tools using tags, tags should be typed, object-oriented, and versioned. i.e., they should be amenable to inheritance and versioning to support future refinements to tag value categories.

Types of Semantic Tags

In this section, we describe various types of tags needed to support semi-automated navigation of Indic texts. We use a popular entry-level Samskrit text in Nyaaya (Indian logic) called tarka sangraha as a running example to illustrate these tags, because of its highly structured nature of its exposition.

Sentence Categories:

To support E-reading tasks listed earlier, we need to identify specific types of sentences in Indic texts. We identify the following types: term-defining (लक्षण), enumerating, explanatory, illustrative (example-giving), continuation (अनुवर्ति), exception (अपवाद), and jurisdictional (निर्देशक) sentences. The table below provides some examples of these categories.

Sentence Category	**Explanation**	**Example from Tarka Sangraha**
Enumerating	नाम्ना पदार्थ-संकीर्तनम् उद्देशः A sentence that enumerates categories	9.2. सा द्विविधानित्याऽनित्या च ।

Term-defining	असाधारणम् लक्षणम् Definition of a technical term.	9.1. तत्र गन्धवती पृथिवी ।
Exemplifying	Sentence giving an example of a term	9.6. शरीरमस्मदादीनाम् ।
Explanatory	Explanation other than definition	38.2 इन्द्रियार्थसंनिकर्षजन्यं ज्ञानं प्रत्यक्षम् ।
Sub-type of Explanatory: Locating	Identifies where a term / concept is to be found	9.8. तच्च नासाग्रवर्ति ।

Intra-sentence relationships:

Intra-sentence relationships define how the words in a sentence connect with each other. Examples are the Kaaraka sambandhas among words in a sentence. An important relation in a term-defining sentence is that between the term (लक्ष्यम्) and its definition (लक्षणम्) phrase. In the sentence 'तत्र गन्धवती पृथिवी', the word 'पृथिवी' is linked to 'गन्धवती' via the 'defined as' or 'लक्षण' relation.

Inter-sentence relationships:

To extract the semantic structure of an Indic text, it is essential to capture the relations between sentences. Text summarization requires selecting a chain of sentences that together constitute a high-level discourse on a single concept. Concept elucidation requires selecting a chain of sentences that delve deeper into a given concept (e.g., द्रव्य) along with its subtopics, their explanations and examples.

Typically, sentences contain anchor terms that indicate their connection with others. The anchor terms enable us to identify the following types of relations:

- **Sub-topics** of a given topic. Each of the terms listed in an enumerating sentence can be linked to their corresponding defining sentences. For example, in tarka sangraha, sentence 2, namely,
  2. तत्र द्रव्याणि पृथिव्यप्तेजोवाय्वाकाशाकालदिगात्ममनांसि नवैव ॥ २॥
  the term पृथिवी relates to the term-defining sentence 9.1, namely,
  9.1. तत्र गन्धवती पृथिवी ।
  via the 'sub-topic' relation.
- **Linguistic references**: Anchor words in a sentence, especially सर्वनाम words, indicate continuation of a concept from an earlier sentence. In the example above, the word 'तत्र' in sentence 9.1 points to sentence 2 via 'predecessor' relation. In the subsequent sentence 9.2, namely,
  9.2 सा द्विविधानित्याऽनित्या च ।

the word 'सा' is points to the word 'पृथिवी' in sentence 9.1 via the 'referent' relation (समानवस्तु).

- **Lateral Continuation** of a previous sentence (अनुवर्तनम्). Anchor words can also indicate lateral continuation of exposition from a previous sentence (as opposed to sub-topic). For example, in the sentence,
  9.5 पुनस्त्रिविधा शरीरेन्द्रियविषयभेदात् ।
  the word पुनः continues the sub-topic enumeration sentence in 9.2 by giving another type of enumeration. Hence it is a lateral continuation of 9.2.
- **Elaboration** of a previously defined concept (विवरणम्). Sentences that illustrate or explain a concept introduced in an earlier sentence need to be linked as elaborations of that sentence. This relation helps show concise summaries by hiding explanatory text for reader's convenience.

There is an elaborate categorization of relationships between sentences used in miimaamsa texts. In principle, the inter-sentence relation types listed here could be expanded to include them as well. However, the objective of this paper is to outline the broad principles of concept mapping and to motivate the need for taking such relations into account. Hence we have listed a few relation types to illustrate the general direction of the work to be done, and also identified the most essential ones to support E-reading tasks. We leave the incorporation of the other relation types to future work.

Expert Notes:

Another important aid to studying Indic texts is an expert's paraphrasing of a concept and its explanatory notes not part of the source text. An E-reader tool must provide a way for knowledgeable readers to attach notes to an existing word, phrase, sentence or paragraph of the source text that others can comment on. Such notes can then be tagged and linked further using the same mechanisms described earlier.

Using Concept Map for Indic Text Navigation

Given that a concept map of an Indic text is a graph data structure, it is amenable to graph-based traversal and query operations that any standard graph database such as neo4j (Neo4j 2016) supports. For instance, to view the concept-wise index of an Indic text, one needs to extract all sentences tagged as term-defining. Further, to show explanation of a term from the text, one needs to retrieve all vertices of the concept map reachable from its definition vertex via the "continuation", "sub-topic" and "elaboration" relations.

Implementation

We have prototyped the concept mapping methodology described in this paper by leveraging a popular text annotation tool used by natural language processing community, called Brat []. Brat is a web-based tool that allows interactive highlighting and tagging of text in an uploaded text document in a persistent manner. It also supports linking different

segments of text using tagged relation types. Brat supports customization of tag types as well.

We have performed manual tagging of all sentences and relations of the Nyaaya commentary text tarka sangraha in Samskrit. Later, we imported a subset of the text into Brat to build a concept map to validate the ideas presented in the paper. We have also attempted to apply the concepts to tagging its commentary text called aaloka vyaakhyaa for tarka sangraha. Figure 1 and Figure 2 show screenshots of the tool in action. Though several UI improvements are needed to ease navigation, our experience indicates that the tool can simplify Indic text browsing significantly when coupled with our concept mapping methodology.

Discussion

Tarka sangraha text provided an effective context to evaluate several of the concept mapping ideas presented in this paper. The semantic tags for sentences and inter-sentence relations that we presented were found to be adequate to cover the entire text. However, the difficulty of browsing Indic texts is more acute when studying detailed and verbose commentaries (such as aaloka vyaakhya of tarka sangraha) where summarization, drill-down navigation and integrated language help are essential. Moreover, the adequacy of the semantic tag types proposed in this paper needs to be evaluated in the context of more diverse Indic texts, which is an area for future work. Our experience with tarka sangraha and its commentary indicates that our basic concept mapping framework goes a long way in simplifying Indic text browsing.

E-readers should integrate language and semantic analysis with human-assisted disambiguation of meaning along with concept maps to be effective at simplifying user experience. This is an area for future work.

Related Work

The value of mind maps to organize and present knowledge has been well-studied (Beel et al 2009). The proliferation of mind mapping tools (both free and commercial) is a testimony to their effectiveness. However, our unique contribution is in devising a standard schema for semantic networking of Indic text content that allows automated navigational queries. Semantic web is a methodology to represent and organize web content to support sophisticated knowledge processing and visualization. Our approach The natural language processing community has developed numerous tools (Goyal 2012, Hellwig 2009, Kulkarni 2016) for automatically inferring semantic linkages among multiple sentences in a paragraph (Bama Srinivasan 2011). Such tools can be leveraged to infer some of the semantic tags identified in this paper, and hence are complementary. Brat (Brat 2016) is a text annotation tool that provides a user-interface for concept mapping, but leaves the higher-level annotation and tagging strategy to the user.

Conclusions

In this paper, we addressed the problem of simplifying the study of Indic texts by novice seekers via facilitating the creation of concept overlay maps for easy navigation and comprehension. We presented a methodology of concept mapping that aids manual study as well as enables sophisticated semantic navigational queries on text content. We accomplish this through careful design of semantic tags that are standardized yet extensible. Our initial experience using this technique on a popular Indic text indicates that it is feasible and effective at simplifying the navigation of such texts.

Future Work

Our larger goal is to develop a full-fledged collaborative E-reader cum editor for cross-referenced Indic texts at scale. To reach there, several missing functions need to be provided beyond text annotation, including storing concept maps in a scalable graph database, integration with sophisticated Language analysis tools such as Kaaraka (Kulkarni 2016) and discourse analysis (Srinivasan 2011), automated text analytics workflows and improved user interface for both users and experts. We are developing a scalable Indic document analytics platform that addresses several of these architectural challenges as outlined elsewhere (Susarla 2016).

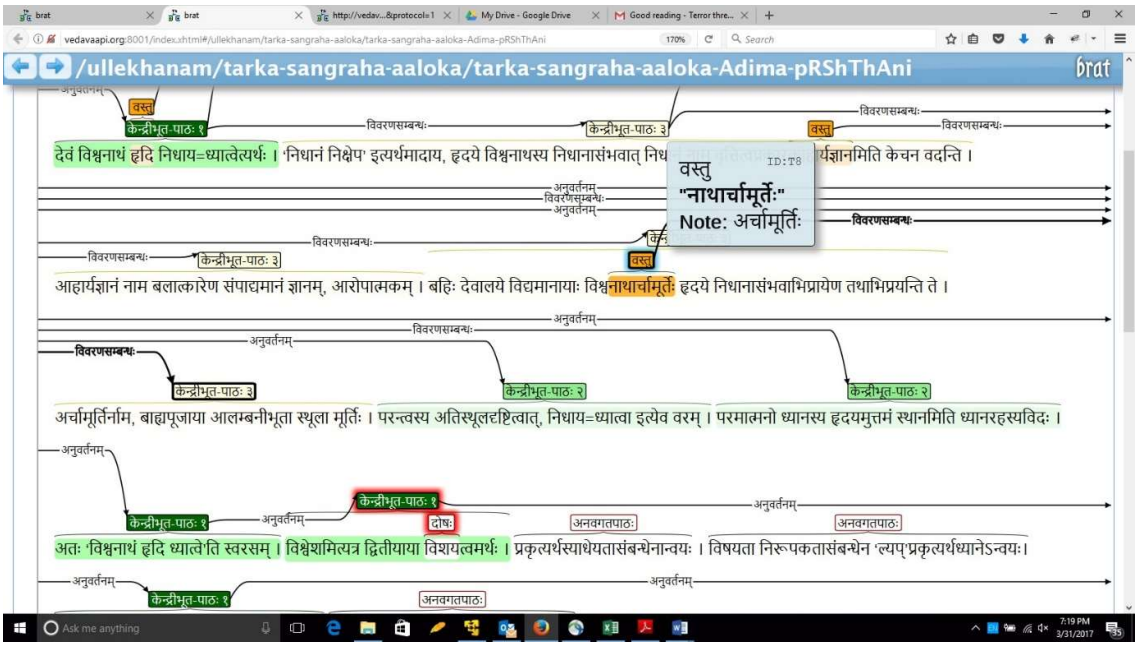

Figure 1: A screenshot of the text annotation of Tarka sangraha's aaloka vyaakhya commentary using Brat tool. This illustrates individual sentences, inter-sentence relations and user notes.

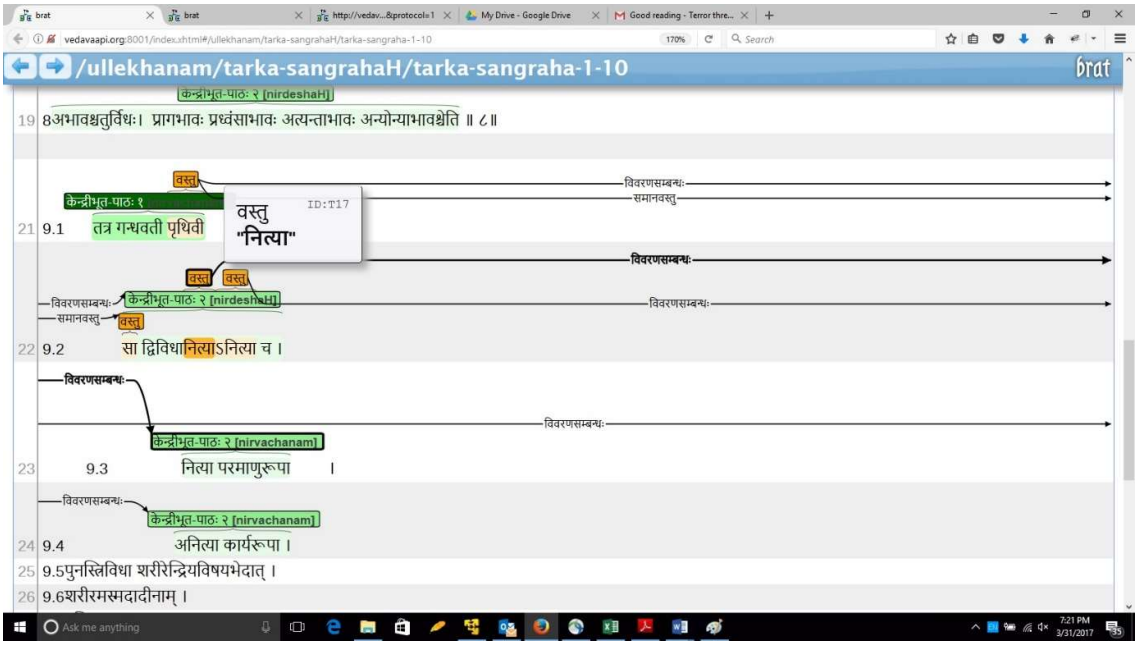

Figure 2: A screenshot of the annotations to tarka sangraha sentences using Brat tool. It shows how individual words can be linked across multiple sentences using semantic tags proposed in this paper.

References

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.

Oilver Hellwig. 2009. Extracting dependency trees from sanskrit texts. Sanskrit Computational Linguistics 3, LNAI 5406, pages 106–115.

Gérard Huet. 2002. The Zen computational linguistics toolkit: Lexicon structures and morphology computations using a modular functional programming language. In Tutorial, Language Engineering Conference LEC'2002.

Amba Kulkarni. 2016. Samsaadhanii: A Sanskrit Computational Toolkit. http://sanskrit.uohyd.ac.in/.

Anil Kumar. 2012. Automatic Sanskrit Compound Processing. Ph.D. thesis, University of Hyderabad.

Brat. 2016. Brat: brat rapid annotation tool. http://brat.nlplab.org/.

Bama Srinivasan, Ranjani Parthasarathi. 2011. Mimamsa Inspired Representation of Actions (MIRA), Fifth Indian International conference on Artificial Intelligence (IICAI) 2011.

Joeran Beel, Bela Gipp, and Jan Olaf Stiller. 2009. Information Retrieval on Mind Maps – What could it be good for? In Proceedings of the 5th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom'09), pages 1–4, Washington (USA), November 2009.

MongoDB. 2016. MongoDB NoSQL Database. http://www.mongodb.com/.

Neo4j. 2016. neo4j: The World's leading Graph Database. http://www.neo4j.com/.

Sai Susarla, Parag Deshmukh, K. Gopinath. 2016. Architectural Considerations for Scalable Indic Document Analytics. In ICON Workshop on Sanskrit Computational Linguistics, ICON SCL 2016.