Preliminary Design of a Sanskrit Corpus Manager

Gérard Huet
Inria Paris Center
Gerard.Huet@inria.fr

Idir Lankri
Université Paris Diderot
lankri.idir@gmail.com

Abstract

We propose a methodology for the collaborative annotation of digitalized Sanskrit corpus tagged with grammatical information. The main features of the proposal are a fine grain view of the corpus at the sentence level, allowing expression of inter-textuality, sparse representation allowing non-necessarily sequential acquisition, and distributed collaborative development using Git technology. A prototype Sanskrit Corpus Manager has been implemented as a proof of concept, in the framework of the Sanskrit Heritage Platform. Possible extensions and potential problems are discussed.

1 Introduction

Several digital libraries for Sanskrit corpus have been developed so far. We may mention the GRETIL site of Göttingen's University,[1] with a fair coverage, under various formats. The Sarit site,[2] developed by Dominik Wujastyk, Patrick McAllister and other Indology colleagues, contains a smaller corpus, but it follows a uniform format compliant with the Text Encoding Initiative (TEI) standard, and has a nice interface. Furthermore it benefits from a collaborative acquisition framework using Git technology. The Sanskrit Library[3] developed by Peter Scharf and colleagues, also follows the TEI, and benefits from the tagging services of the Sanskrit Heritage Platform, since individual sentences link to its segmentation cum tagging service. DCS[4] developed at Heidelberg University by Oliver Hellwig, is the most advanced from the point of view of linguistic analysis, since it is fully annotated with morphological tags indexing a lexicon of stems. Its development involved several iterations of deep learning algorithms (Hellwig, 2009; Hellwig, 2015; Hellwig, 2016). Covering at present 560,000 sentences, it is today the closest analogue for Sanskrit of the Perseus Digital Library for Greek and Latin corpus.[5]

Several other efforts are currently under development, although unfortunately with little standardization effort. Not all digital libraries are publicly available. For instance, the TITUS Thesaurus of Indo-European text[6] is accessible only to scholars participating in the acquisition effort.

There exist now several computational linguistics tools that process Sanskrit text in order to parse it under a grammatical representation that can be considered an approximation to a formal paraphrase of its meaning. Typically, a sentence will yield a stream of morphological tags. The DCS analyser of Oliver Hellwig, based on statistical alignment on a data base of lemmas trained from a seed of human-annotated tags, has the advantage of being fully automatic. The Sanskrit Heritage Platform under development at Inria Paris offers a service of segmentation with tagging (at two levels, inflection and morphology of stems), linking into a choice of two dictionaries. It also has a surface parser using *kāraka* analysis that can be used for learners on simple sentences, but is not sufficient for corpus processing (Huet, 2007). It also

[1] Göttingen Register of Electronic Texts in Indian Languages http://gretil.sub.uni-goettingen.de/gretil.htm

[2] Search And Retrieval of Indic Texts http://sarit.indology.info

[3] Sanskrit Library http://www.sanskritlibrary.org

[4] Digital Corpus of Sanskrit http://kjc-sv013.kjc.uni-heidelberg.de/dcs/

[5] Perseus http://www.perseus.tufts.edu/hopper/

[6] TITUS http://titus.uni-frankfurt.de/index.htm

links with Amba Kulkarni's Saṃsādhanī analyser,[7] that helps produce a dependency graph (Kulkarni, 2013). This structure captures the semantic role (*kāraka*) analysis of a sentence, provided it is not too dislocated. Furthermore, an auxiliary tool helps the annotator to transform a dislocated sentence into its prose order by proper permutation of its segments.

Thus it seems that the time is ripe to consider establishing a common repository that would store digital Sanskrit libraries in annotated form, either automatically, or with the help of competent annotators using interactive tools. We present here a preliminary proposal for the design of a Sanskrit corpus manager concept, that could serve as a seed repository for the collaborative editing of texts, and that could support navigation and search through appropriate further tools. We have developed a simplified implementation of the concept, using technology available off-the-shelf as free software. We shall conclude by listing problems in the managing of a joint corpus repository.

2 Specificities of Sanskrit corpus

Processing Sanskrit by computer is in some sense easier than processing other natural languages, at least if we restrict ourselves to the classical language. It benefits of the grammatical tradition [*vyākaraṇa*] dating back from hoary times, since the grammar of Sanskrit was fixed by Pāṇini 25 centuries ago in his Aṣṭadyāyī, which was initially descriptive, but later became prescriptive. Classical Sanskrit was not the vernacular local prakrit, which is used only in the theater. It was the language of the educated [*śiṣṭa*]. And thus, it was assumed grammatically correct, which means that we may align our segmentations to a precise recursive definition.

Granted, there are many non-Paninian forms in epics literature, and there are many corrupted texts. But we may record exceptions, and corrupted texts may perhaps be amended. Of course philologists will shudder at the thought of amending texts, but they must excuse my irreverence, considering that in my professional trade, programs with bugs must be corrected, and only secondarily treated as historical artifacts in the version-maintaining repository. The main merit of mistakes is to trace possible filiations of versions, since scribes often copied without amending their sources, and thus errors would be transmitted. But this assumption is not always met, and thus the classical phylogenetic tradition is challenged (Hanneder, 2017). In any case, I am making the assumption that the corpus recorded in the global repository has been edited to the point of being grammatically correct. Possibly as a result of the interactive use of grammatical tools, in as much as they may be used as editing assistants.

Actually, the Sanskrit language is not that regular. Even seemingly regular processes such as compounding pose problems in the proper analysis of written texts, since compounding is not associative, and accent is not marked in writing. Furthermore, there are many different styles, not just prose and poetry. The grammatical *sūtra* style is very concise, closer to algebraic rules, with phonemes used both for linguistic and meta-linguistic notation. The *śāstra* style of scholastic Sanskrit (Tubb and Boose, 2007) is also highly artificial. The Indian tradition of formal debate (*vāda*) (Tripathi, 2016) produced texts that are layers upon layers of commentaries, with counter-arguments (*pūrvapakṣa*) alternating with upheld theses (*uttarapakṣa*, *siddhānta*). Poets indulged in obscure constructions, rare lexemes, very long compounds, and dislocated sentences. Furthermore, the inherent ambiguity of phonetic enunciations where word boundaries are blurred by sandhi gave rise to a whole new genre of *śleṣa* – double entendre – where ambiguous segmentation yields possibly opposite meanings (Bronner, 2010). For instance, consider *nakṣatrapathavartinā rājñā* from Daṇḍin's Kāvyādarśa. It may mean a glorious king "following the path of the stars" (*nakṣatra-patha-vartinā rājñā*), or a despicable king, "not following a noble path" (*na kṣatra-patha-vartinā rājñā*), playing on the oronyms *nakṣatra* and *na kṣatra*. Here specific philological apparatus is needed in order to display the two readings, it is not just a matter of choice between segmentations, since both readings are intended. But if linear text is given up in benefit of graphical display, we may visualise the mixed two readings as shown in our Reader tool, see Figure 1.

Other difficulties in interpreting Sanskrit text are the absence of distinctive sign for proper names (like capitals in Roman script), making e.g. *kṛṣṇa* ambiguous between the divine hero and the black color, and the ambiguity of *prādi* compounds such as *nirvācya*, that may mean "what should not be talked about"

[7] Saṃsādhanī `http://scl.samsaadhanii.in`

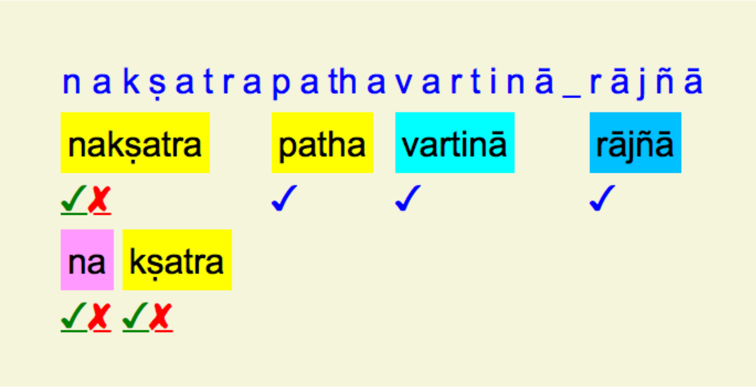


Figure 1: Daṇḍin's prototype *śleṣa*

(with *nis* preposition acting as negation) as well as "what should be explained" (now compositional future participle of verb *nirvac*). Another problem, at the discourse level this time, is indirect speech, whose ending is marked with particle *iti*, but whose beginning must be guessed from the context. All these reasons show that editing a text in order to express several possible meanings with distinct morphological annotations, explained through distinct grammatical analyses, is a much more difficult task than simply listing raw sentences in sandhied form.

Finally, Sanskrit literature abounds in inter-textuality features. Mantras are quoted, stories are retold in a manifold manner, bards adapt orally transmitted tales, learned commentaries pile up on each other, numerous anthologies of poems and maxims (*subhāṣita*, *nyāya*) share a lot of material, *mahākāvya*s expand episodes of epics, etc.

Considering all these difficulties, we propose a set-up for progressive computer-aided tagging of selected portions of corpus, with documented intertextuality, as an alternative to TEI-style full digitalization of corpus in raw form. Thus one of the important requirements is that the (partial) corpus be represented at a low level of granularity, typically a *śloka* for poetry, or a sentence for prose.

3 Available technology

The main paradigm of the proposed annotation scheme is that it should be a distributed service, not just available from a server for consultation of readers, but itself the locus of collaborative annotation activity. This is in line with the recommendation of Peter Robinson (Robinson, 2009): "The most radical impact of the digital revolution is to transform scholarly editing from the work of single scholars, working on their own on single editions, to a collaborative, dynamic and fluid enterprise spanning many scholars and many materials".

In the software development domain, now Git technology (Chacon and Straub, 2014) is the de facto standard for such collaborative development. Originally designed to serve as versioning cum distribution for the Linux effort, it quickly replaced all previous software management systems. It has several implementations, one managing the GitHub site, popular for open-source development. The GitLab software offers additional functionalities, notably in terms of security.

A Git project consists of branches evolving with time, each branch carrying a hierarchy of files. The hierarchy corresponds directly to the structure of the file system of the host operating system. The files typically contain source code of software, and of its documentation. But they may be of whatever format. Collaborators of the project have a local copy of the relevant branch on their own computer station. So they may not only compile and install locally the software, but they may modify it and add to it. After local testing, the developer may request the supervisor of the branch to update the global site with his modifications. On approval, the software merges the modifications with the current version, a possibly complex operation.

Git is a major change of paradigm in the collaborative development of massive efforts. It is now used for the dynamic management of documents of various nature. This is a mature technology, with the finest available algorithms in distributed computing, alignment, compaction, cryptography. It looks like the ideal collaborative tool for developers of a digital library.

The other obvious technological decision is to use Web technology for the user interface. HTML and XML support Unicode for presenting all writing systems. Web services are now the absolute standard for distributed services.

4 Implementing a prototype as a proof of concept

A 2-months effort in summer 2017 was defined as a student Master project. The second author, in the Master program of University Paris Diderot, and an Ocaml expert, was offered an internship at Inria for its implementation. He familiarized himself rapidly with the sources of the Sanskrit Heritage Platform, put at this occasion on Inria's GitLab site for distributed development under Git. At the same time, a second Git project was launched as the Sanskrit Heritage Resources, to distribute the lexical resources used by the Platform machinery, as well as the Sanskrit morphology XML databanks that it produces.

The requirement was to implement a corpus manager as a Web service, using the Sanskrit Heritage Platform as interactive tagging service, and producing progressively an annotated corpus as a sub-branch of the Sanskrit Heritage Resources Git project. The hierarchical structure of the corpus is directly mapped on the directory structure of the UNIX file system.

4.1 The Sanskrit Heritage Corpus Manager

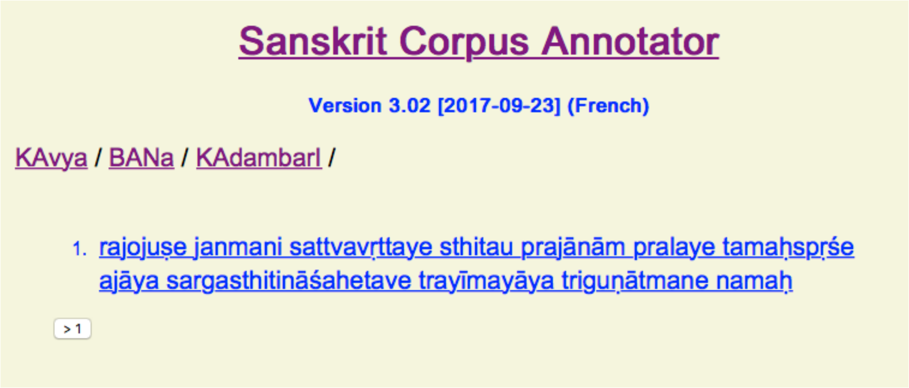
Three levels of capabilities have been defined. The Reader capacity is available to any user. As its name indicates, he is only allowed to read the library, but not to modify it. The Annotator capacity allows addition and correction to the corpus files. The Manager capacity allows addition and correction to the directory structure. These three capacities are mapped respectively to permissions of the UNIX file system, and to roles in the Sanskrit corpus project, initially located as a component of the Sanskrit Heritage Resources Git project.

Texts are available as leaves of the directory structure, such as "KAvya/BANa/KAdambarI/". In Manager mode, one may add to this structure, or edit it. In Reader mode one may navigate through it, through a simple Web interface with scrolling menus. In Annotator mode you may add to the text, or give corrections. For instance, let us assume that, in Annotator mode, we input a sentence in the initially empty directory "KAvya/BANa/KAdambarI/". We are forwarded to the Sanskrit Heritage Reader page (Goyal and Huet, 2016), where we input in the text window the following string:

rajojuṣe janmani sattvavṛttaye sthitau prajānā.m pralaye tamaḥspṛśe |
ajāya sargasthitināśahetave trayīmayāya triguṇātmane namaḥ ||

The segmenter returns with 155,520 solutions represented on a single HTML page as a graphical display where the annotator, if familiar with the tool, very quickly converges to the desired solution (in 13 clicks). When this is done, a Save button prompts the annotator, who may save this segmentation in the corpus, or abort. On saving he is returned to the corpus interface, which prompts him for the next sentence. The screen he sees at this point is represented in Figure 2. It indicates that now branch "KAvya/BANa/KAdambarI/" supports a text where sentence 1 is now listed (in IAST notation according to local settings, could be Devanāgarī or both).

This sentence 1 is itself a link, to the display of the Heritage reader, as shown in Figure 3. Note that this is what we saved, showing the unique solution selected by the annotator. Note also that what we see is just the usual display of the Reader: you may click on segment rectangles in order to get their morphology. The morphology is itself linked to the dictionary, which can be set either to the Sanskrit-French Heritage dictionary maintained at the site, or to the electronic version of the Sanskrit-English Monier-Williams dictionary. It is not just an HTML static page, it is a dynamic page where all services of the Platform are available. Including backtracking on the annotator choices, via the Undo service! You may also click on the Unique Solution tick and continue the analysis with gender agreement. Or by clicking on the UoH Analysis Mode tick, go further into *kāraka* analysis with Amba Kulkarni's dependency analyser. Thus, it would be easy to extend the service with other displays under various analyses, provided with proper meta-data.



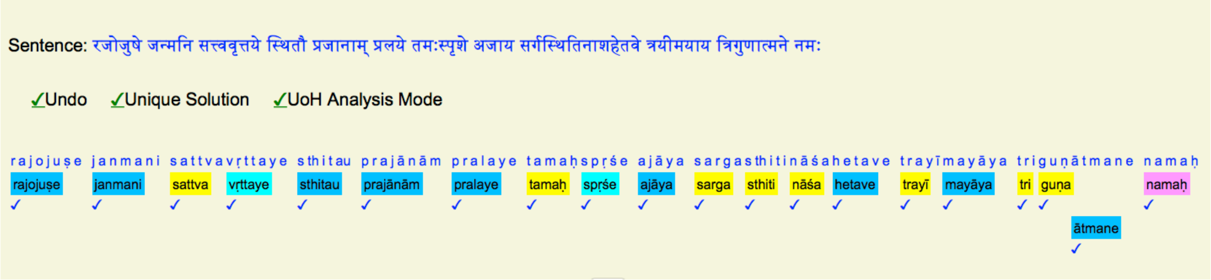
Sanskrit Corpus Annotator

Version 3.02 [2017-09-23] (French)

KAvya / BANa / KAdambarI /

1. rajojuṣe janmani sattvavṛttaye sthitau prajānām pralaye tamaḥspṛśe ajāya sargasthitināśahetave trayīmayāya triguṇātmane namaḥ

Figure 2: After saving sentence 1



Sentence: रजोजुषे जन्मनि सत्त्ववृत्तये स्थितौ प्रजानाम् प्रलये तमःस्पृशे अजाय सर्गस्थितिनाशहेतवे त्रयीमयाय त्रिगुणात्मने नमः

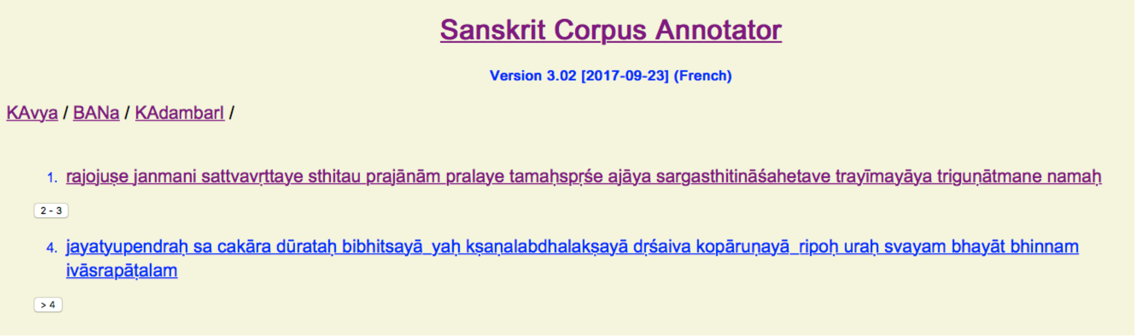
rajojuṣe janmani sattvavṛttaye sthitau prajānām pralaye tamaḥspṛśe ajāya sargasthitināśahetave trayīmayāya triguṇātmane namaḥ

rajojuṣe ✓ janmani ✓ sattva ✓ vṛttaye ✓ sthitau ✓ prajānām ✓ pralaye ✓ tamaḥ ✓ spṛśe ✓ ajāya ✓ sarga ✓ sthiti ✓ nāśa ✓ hetave ✓ trayī ✓ mayāya ✓ tri ✓ guṇa ✓ ātmane ✓ namaḥ ✓

Figure 3: Displaying sentence 1

Now let us return to Figure 2. Please note the "> 1" button. It invites the annotator to continue his tagging task, with another sentence. If you click on it, a scrolling menu prompts you with the sentence number, and an Add button. You may at this point choose not to continue in sequence, for instance choose sentence 4. On pressing Add we are back in the loop with the Reader.

After entering sentence 4, we get the image of this piece of corpus as Figure 4. There are now two mouse-sensitive buttons: one marked "2 – 3" for filling the hole between 1 and 4, the other one, marked "> 4", for entering sentences after the 4th one. This illustrates the partial nature of the annotated corpus. Scholars may concentrate on often quoted parts, or verses they have a special interest in, without having to tag continuously from the beginning of the work. This is important, in view of the non-fully-automatic nature of the annotating task. It also allows work splitting between annotators of the same text. Merging their contributions will be managed by the Git mechanisms.



Sanskrit Corpus Annotator

Version 3.02 [2017-09-23] (French)

KAvya / BANa / KAdambarI /

1. rajojuṣe janmani sattvavṛttaye sthitau prajānām pralaye tamaḥspṛśe ajāya sargasthitināśahetave trayīmayāya triguṇātmane namaḥ

4. jayatyupendraḥ sa cakāra dūrataḥ bibhitsayā_yaḥ kṣaṇalabdhalakṣayā dṛśaiva kopāruṇayā_ripoḥ uraḥ svayam bhayāt bhinnam ivāsrapāṭalam

Figure 4: After annotating sentences 1 and 4

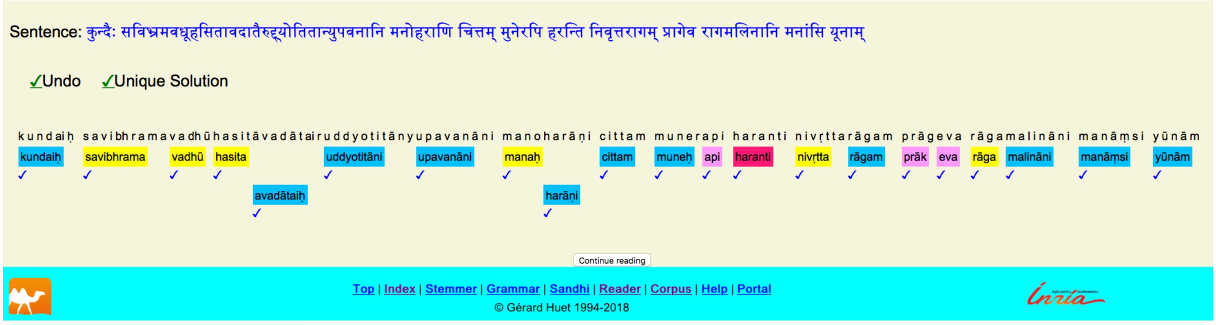
When a Reader browses the corpus, he will see exactly the same display, except that the adding buttons will not appear.

When an annotator has completed some tagging, he may call a routine that will store his annotations in the local repertory of the Corpus Git project. He may then use the `commit` Git command to commit his annotations, with proper documentation. Once in a while he may distribute his contributions to the community by using the `push` Git command in order to merge his work with those of the other annotators, under control of the project Managers.

4.2 Application to citations analysis in the Heritage dictionary

This prototype of a corpus manager has been implemented as an auxiliary service of the Heritage platform segmenter. It is currently being put to use to manage citations in the Heritage hypertext dictionary. Its current 800 citations are being progressively tagged, and entered in a standalone branch "Heritage_citations" of the corpus. The corpus structure is implemented as a sub-project of the Heritage_Resources Git project, and as such is incorporated in the Heritage_platform server data, at installation time. Thus the facility is available for testing to whoever installs the two software packages through Inria's GitLab server, as projects `https://gitlab.inria.fr/huet/Heritage_Resources.git` and `https://gitlab.inria.fr/huet/Heritage_Platform.git` respectively. The public server site `http://sanskrit.inria.fr` has been updated with the corpus manager, available as a "Corpus" service from the site directory at the bottom of its pages. Of course only Reader mode is available at the public site. But the distribution version, available through Git, will allow annotators to develop their own tagged corpus, and possibly merge them in the common Git repository when registered as an official Annotator.

An example of such analysed citation may be viewed in our dictionary at entry *kunda*. Please visit URL `http://sanskrit.inria.fr/DICO/21.html#kunda`. This entry is illustrated by a quotation from śloka 6.25 of Kālidāsa's Ṛitusaṃhāra, underlined as mouse-sensitive. Clicking on it brings you to the corresponding corpus page, where the sentence is displayed as a list of colored segments, as shown in Figure 5. Clicking on a segment brings its lemma, with lexicon access to the root items. Although it has the same look-and-feel as the segmentation tool, it is actually displayed by the corpus manager, navigating in Reader mode in its "Heritage_citations" branch. This can be verified by clicking on the "Continue reading" button, which brings you to this branch directory, where the śloka appears as item 10. This shows the smooth integration of this tool within other services.



Sentence: कुन्दैः सविभ्रमवधूहसितावदातैरुद्द्योतितान्युपवनानि मनोहराणि चित्तम् मुनेरपि हरन्ति निवृत्तरागम् प्रागेव रागमलिनानि मनांसि यूनाम्

kundaiḥ savibhramavadhūhasitāvadātairuddyotitānyupavanāni manoharāṇi cittam munerapi haranti nivṛttarāgam prāgeva rāgamalināni manāṃsi yūnām

Figure 5: Annotated quotation

5 Extending the prototype to other tools

The extreme simplicity of this design makes it easily extensible to other grammatical tools implemented as Web services. All that is needed to incorporate them is to include a save button in the pages that return the result of their analysis, with the functionality of saving their HTML source in the corpus hierarchy. Or, even, in the style of our own implementation, to store the sentence analysis data as parameters for the invocation of a dynamic corpus crawler. Conversely the Add facility of the corpus manager will have to be made aware of the variety of such services, and its display accommodated to show all analyses of the given śloka by the various services. This assumes of course that these services are reachable from the putative annotators, either installed on their station's own Web server, or available at publicly available Internet servers. The Heritage set of services may be used both ways, since it is itself distributed as an open-source system from its Git project repository. Should the concept prove itself useful, it would be easy to separate the Corpus Manager from the Heritage distribution, and make it a stand-alone facility.

It is to be remarked that having several grammatical tools available for displaying corpus in analysed form does not induce any commitment on a standard display, each tool may keep its look-and-feel, and links to its specific functionalities. We are not demanding either to synchronize or align taggings effected by various tools. Annotators using one tool may tag sentences irrespective of whether they have been already processed with some other tool. All we have to agree on is the directory structure and its metadata format (under control by the Git users with Manager capability), and in the designation scheme of individual files representing the analyses.

6 Design of inter-textuality functionalities

This simple prototype provides for the moment a strictly hierarchical view of the corpus. This is too restrictive, since it allows no sharing. For instance, in the skeleton corpus of “Heritage_citations”, we would like to link item 10 to its original in Ṛtusaṃhāra. Of course we could enter its duplicate in its proper branch, say “KAvya/KAlidAsa/Ritusamhara/6/25”. But we would like to document this by recording its “absolute” link in the “Heritage_citations” branch at item 10. This would be an easy extension of the current mechanism. But this is only one simple example of inter-textuality. Some of the citations are not to a full śloka, but perhaps to a portion, or a simplification, or a reordering of some original quotation. Thus we would need to design a notation to document such partial sharing between different branches of the corpus.

6.1 Collating recensions and manuscript segments

We also want to be able to use the tool for recording, and comparing, various manuscripts traditions of the same text. Actually, the idea of this low-granularity corpus representation arose from a presentation by Pr Brockington at a seminar in Paris in december 2016 (Brockington, 2016). He showed there two representations of various manuscripts of Sanskrit epics.

The first one, extracted from traditional phylogenetic methods (Phillips-Rodriguez et al., 2009) represents a tree of manuscripts of Mahābhārata, expressing the growth of the material over time. It has been obtained through phylogenetic analysis performed on sargas 43-47, 51, 59-60 and 64-65 of the Dyūtaparvan by the Supernetwork method in the SplitsTree package. The sigla used are those of the Critical Edition, with J substituted for Ñ and Z for Ś. It is reproduced in Figure 6 below (courtesy Wendy Phillips).

The second one is a Venn diagram of Rāmāyaṇa’s manuscript relationships, reproduced in Figure 7 (taken from (Brockington, 2000), courtesy John Brockington). This Venn diagram representation (possibly completed by the suitable ordering of the verse portions) is a more informative view of relationships between manuscript groups, since it represents the (multi-)set of all ślokas of all manuscripts, each one represented as a subset, possibly intersecting in complex ways with other manuscripts. In other words, the Rāmāyaṇa is there considered as a Boolean expression in terms of its manuscripts segments, a more detailed concept than the phylogenic tree, although not currently producible automatically from recensions in an obvious manner.

This suggests that our śloka-level corpus ought to accommodate notation amenable to express complex sharing relationships between the manuscripts, such as:

$$A = B\,[1 - 250]\,; C\,[5 - 23]\,; B\,[251 - 300]$$

expressing that manuscript A is the same as B with interpolation of a portion of C. Such sharing relationships ought to turn into extra annotations on the corpus data representations, so that navigation through the various versions would be available.

6.2 Paths management on shared corpus

It should be obvious at this point that an extra level of abstraction is needed in order to be able to name contiguous portions of corpus recensions that are shared across manuscript versions, such as $C\,[5 - 23]$

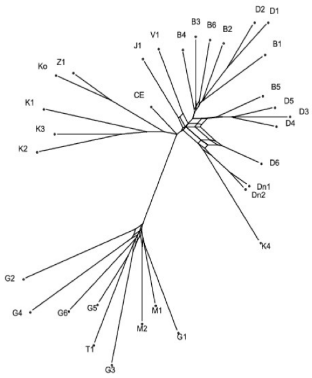


Figure 6: Phylogenetic analysis

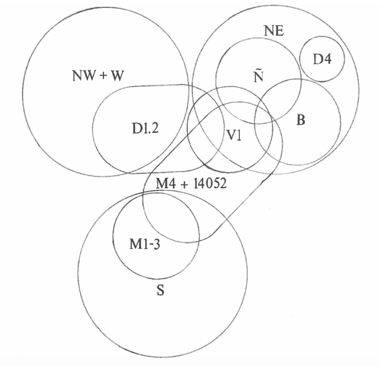


Figure 7: Venn diagram

in the notation above. This path in our corpus tree is shared between recensions A and C. If we want to express this sharing in our corpus structure, and thus avoid the duplication of śloka annotations between A and C, we shall need to introduce the notion of path through a dag[8] of branches, of which our corpus structure is only a specific spanning tree. This induces a need to express the concept of the successor of a śloka node *along a given path*, since in our example node $B.250$ has successor $B.251$ along path B, but $C.5$ instead along path A. Thus we need to record this information in node $B.250$, so that we may later navigate along path A by following the path B until its 250th node, and then continue from node $C.5$, until node $C.23$, which will be followed by node $B.251$ along the A path.

This of course assumes that the numbering of ślokas is now a function of its path, so that e.g. śloka $B.251$ appears at index 269 along path A, since śloka $B.251$ is shared with $A.269$ The same mechanism could allow for instance to assign to index Bhagavadgītā.1.1 the same śloka as Mahābhārata.6.63.23.

The determination of the portions of text that are amenable to sharing is decided by the human corpus managers/annotators, not a fully automatic process, since we do not want to share maximally by identifying all identical ślokas across all texts. For instance, we shall not identify all the evocations of a ritual mantra across all texts, with absurd cluttering of a unique node with all possible successors in all the texts. Furthermore, we do not want that two occurrences of the same verse in one text lead to looping paths.

6.3 Cohabitation of readings

Representing the *padapātha* form of a Sanskrit utterance is the first level of its interpretation. Assigning morphology to its segments is a further level of interpretation. Assigning *kāraka* semantic roles consistent with nominal cases and verbal voices is still a deeper interpretation; linking anaphoric references to their antecedent and cataphoric ones to their postcedent, together with entity-name recognition, brings analysis at the discourse level. Accommodating these various levels of analysis of a text will need adaptations to our corpus representation structure. The basic idea is that a piece of corpus represents more than the raw text as a stream of phonemes, and that paths through the fine-grain structure represent not just a list of phonetic productions, but a specific *reading* of this text.

Thus we must admit paths that represent different glosses of a given text, possibly contradictory. For instance, we would need different path assignments for Bhagavadgītā according to Śaṅkara and to Madhva respectively, so that e.g. BhG$\{24.2.17\}$ appears as *nāsatovidyatebhāvonābhāvovidyatesataḥ* on the first path, and *nāsatovidyate'bhāvonābhāvovidyatesataḥ* on the second.[9] Note that in this example, the use of *avagraha* does disambiguate the two readings, but as stream of phonemes they are the same. This is a case showing that we need two different nodes in our corpus representation for common phonetic material, since their meanings are not compatible. Note that the two readings are oronyms, but this is

[8] directed acyclic graph

[9] communicated by Pr. Madhav Deshpande

not a case of *śleṣa*, where the two meanings are intended. We could talk of XOR-oronyms, contrasted with AND-oronyms (the genuine *śleṣa*s), for which we want to represent the two readings *together* in the same structure. The XOR/AND terminology stems from Boolean algebras in Stone form, such as Venn diagrams.

Genuine *śleṣa*s are often used for expressing simili figures of style, as Bāṇa demonstrated ad nauseum in Kādambarī. Their translation in languages such as French or English necessitates heavy paraphrases weaving the description and its metaphor as coordinated phrases.[10] Giving a notation to represent the two readings without duplication is an interesting challenge: we want to represent minimally the two segmentations while sharing their common segments. Note that the graphical interface of the Heritage Reader gives a possible solution to this problem, since we may keep the two sets of segments, without any duplication, by trimming all segments that appear in neither. See Figure 1.

The design of a proper notation for annotated corpus is beyond the scope of the present paper, and is the affair of professional philologists, but our prototype could provide a test bed for such experiments.

Actually, we could also include in the corpus directories information concerning studies of a particular śloka or portion of text, mentioning bibliographic references to relevant literature. It could also refer to discussions concerning specific grammatical points, respective validity of the various annotations, etc. Each Sanskrit śloka could have its own blog page, and the global corpus structure could evolve into social networking for Sanskrit text!

7 Remaining problems

Our toy corpus manager raises serious issues which will have to be well assessed before scaling up to a durable infrastructure.

First of all, we are suggesting that a common repository of analysed Sanskrit text be agreed upon by both developers of computational linguistics tools and scholars managing digital libraries. This raises issues of a legal and sociological nature. Certain institutions will want to control what they deem is their intellectual property. Certain scholars will refuse to compromise with their freedom of doing things their own way. Even if a critical mass of individuals agree on sharing their work on a common source-free repository, we know from experience that committee work is not always optimum to design a technical artifact. Apparently simple issues such as the naming of branches may reveal complex problems, the solutions of which may not be easy to agree on.

Another important issue is durability. Our proposal assumes that the analyzing tools will be perennial, in as much as their proper availability is necessary for the display of their analyses. This is implicit from the fact that we are not restricting ourselves to displaying static XML or HTML pages, but allow the execution of Web services (cgi-bin executables in the Web jargon) which demand availability of programming languages and their compilers over the life span of the digital library. Thus robustness and maintainability of the satellite tools are critical. Versioning is another issue, since our analysis tools are not finished products, but experimental software that keeps evolving, and that may depend on lexical resources that also evolve themselves. Thus non-regression analysis tools will have to be developed, in order to correct taggings that are no longer identified or are no longer unique after a change of version. However, please note that improvements in precision that do not compromise recall often do not require revisiting the analysed corpus, which should be robust to such upward-compatible improvements.

Finally, let us emphasize that our proposal concerns just the foundations of a collaborative framework for the grammatical annotation of Sanskrit text, and has no pretense at providing philological tools such as collating software. Such tools will have to be re-thought over this low-level representation of the corpus.

8 Conclusion

We have presented general ideas concerning a Sanskrit corpus manager, and implemented a prototype with the Sanskrit Heritage Platform to test the main concepts. The main design imperative is that corpus

[10] In French, *śleṣa* is limited to curiosities like the holorime “Gal, amant de la Reine, alla, tour magnanime, galamment de l’arène à la tour Magne à Nîmes” and jokes like “mon frère est ma sœur” playing on the oronyms *ma sœur*/*masseur*

managing ought to be a collaborative effort, allowing text annotation on a variety of grammatical analysis services. The prototype implementation, in a restricted setting, shows that the infrastructure development is actually rather simple, if one uses off-the-shelf technology such as Web services and Git repositories. It is hoped that this proposal will spur interest from philologists and computational linguists, and hopefully contribute to their increased collaboration.

References

John Brockington. 2000. Textual studies in Vālmīki's Rāmāyaṇa. In Greg Bailey and Mary Brockington, editors, *Epic Threads: John Brockington on the Sanskrit Epics*, pages 195–206. Oxford University Press, New Delhi.

John Brockington. 2016. Regions and recensions, scripts and manuscripts: the textual history of the Rāmayaṇa and Mahābhārata. In *Issues in Indian Philology: Traditions, Editions, Translations/Transfers*. (Abstract) Collège de France.

Yigal Bronner. 2010. *Extreme poetry*. Columbia University Press, New York.

Scott Chacon and Ben Straub. 2014. *Pro Git*. Apress (available as `https://git-scm.com/book/en/v2`).

Pawan Goyal and Gérard Huet. 2016. Design and analysis of a lean interface for Sanskrit corpus annotation. *Journal of Linguistic Modeling*, 4(2):117–126.

Jürgen Hanneder. 2017. *To edit or not to edit*. Pune Indological Series I, Aditya Prakashan, Pune.

Oliver Hellwig. 2009. SanskritTagger, a stochastic lexical and POS tagger for Sanskrit. In Gérard Huet, Amba Kulkarni, and Peter Scharf, editors, *Sanskrit Computational Linguistics 1 & 2*, pages 266–277. Springer-Verlag LNAI 5402.

Oliver Hellwig. 2015. Using recurrent neural networks for joint compound splitting and sandhi resolution in Sanskrit. In Zygmunt Vetulani and Joseph Mariani, editors, *Proceedings, 7th Language and Technology Conference*. Springer-Verlag LNAI (to appear).

Oliver Hellwig. 2016. Improving the morphological analysis of classical Sanskrit. In *Proceedings, 6th Workshop on South and Southeast Asian Natural Languages*, pages 142–151. Association for Computational Linguistics.

Gérard Huet. 2007. Shallow syntax analysis in Sanskrit guided by semantic nets constraints. In *Proceedings of the 2006 International Workshop on Research Issues in Digital Libraries*, New York, NY, USA. ACM.

Amba Kulkarni. 2013. A deterministic dependency parser with dynamic programming for Sanskrit. In *Proceedings of the Second International Conference on Dependency Linguistics (DepLing 2013)*, pages 157–166. Charles University, Prague.

Wendy J. Phillips-Rodriguez, Christopher J. Howe, and Heather F. Windram. 2009. Chi-squares and the phenomenon of "change of exemplar" in the *Dyūtaparvan*. In Gérard Huet, Amba Kulkarni, and Peter Scharf, editors, *Sanskrit Computational Linguistics 1 & 2*, pages 380–390. Springer-Verlag LNAI 5402.

Peter Robinson. 2009. Towards a scholarly editing system for the next decades. In Gérard Huet, Amba Kulkarni, and Peter Scharf, editors, *Sanskrit Computational Linguistics 1 & 2*, pages 346–357. Springer-Verlag LNAI 5402.

Peter Scharf. 2018. TEITagger: Raising the standard for digital texts to facilitate interchange with linguistic software. In Gérard Huet and Amba Kulkarni, editors, *Computational Sanskrit & the Digital Humanities*. D.K. Publishers, New Delhi.

Radhavallabh Tripathi. 2016. *Vāda in Theory and Practice*. D.K. Printworld, New Delhi.

Gary A. Tubb and Emery R. Boose. 2007. *Scholastic Sanskrit*. Columbia University, New York.