Geeta: Gold Standard Annotated Data, Analysis and its Application

Preeti Shukla
Dept. of Sanskrit Studies,
University of Hyderabad
Hyderabad, India
shukla.preetidev@gmail.com

Amba Kulkarni
Dept. of Sanskrit Studies,
University of Hyderabad
Hyderabad, India
ambapradeep@gmail.com

Devanand Shukl
MSRVVP,
Ujjain, India
dev.shukl@gmail.com

Abstract

Importance of gold standard data in the field of NLP is well-established. In this paper we describe the development of one such gold standard for Sanskrit annotated at various levels of linguistic analysis. We describe how such a domain specific gold standard data, in addition to being useful for training and evaluation, is also useful for teaching. With the help of a suitable interface of anusāraka we demonstrate its usability for a linguist, and also for a learner.

1 Introduction

The last decade has seen egress of several computational tools for Sanskrit [1]. Most of these tools (Huet, 2003; Goyal et al., 2012; Kulkarni et al., 2010; Kumar et al., 2010; Kumar, 2012) are based on grammar based methods, and use statistics only for ranking the solutions, with an exception of Hellwig's Sanskrit Tagger (Hellwig, 2009) which uses purely statistical techniques using a small manually annotated data for bootstrapping. There are two main reasons for not using pure statistical methods. The first one is the availability of almost complete grammar for Sanskrit in the form of *Aṣṭādhyāyī* of Pāṇini. [2] The users are very critical while using the computational tools for analysis of Sanskrit texts, since at every step of analysis, they would like to have a justification for the answer produced by machine in terms of Pāṇinian rules. This happens especially when machine produces an unfamiliar output which a human being might not have thought of. It is only when the answer is attested with a corresponding rule from the grammar, users are assured that there is no over-generation / over-analysis on the part of the computer. The second reason is non-availability of enough annotated corpus in electronic form. Sanskrit has huge literature in the form of printed books and manuscripts. If the texts in only exact sciences are considered, according to one estimate there are around 10,000 texts and 100,000 manuscripts (Pingree, 1978). With the advent of World Wide Web a lot of these are also appearing on the web. Such a corpora is in raw form, but still useful for constructing statistical models of grammar, investigating prosodic phenomena, etc. If such a corpus is also annotated at various levels, then it serves as a gold standard data for evaluation, and is also useful for comparison of various tools, judging their adequacy, etc. It is always beneficial to have the same chosen corpus annotated at various levels, such as Tree bank annotation, Discourse level annotation, etc. The advantage of this is: one can use the gold standard annotated data as an input for the evaluation of various modules. This avoids the cascading effects and one can measure the absolute performance of various tools. Another advantage of such data, especially the domain specific one is to use data-driven models to tune the machine for better performance in a chosen domain (Plank and van Noord, 2010). In case of English, Penn Tree bank is such a corpus (Marcus et al., 1993), annotated at various levels such as POS tagging, sentential parsing in the form of Tree bank and also the discourse level tagging. With an aim to develop such a gold standard data for Sanskrit text, we chose Śrīmad Bhagvad

[1] http://sanskrit.uohyd.ernet.in/scl
http://sanskrit.inria.fr
http://sanskrit.jnu.ac.in/index.jps
http://kjc-fs-cluster.kjc.uni-heidelberg.de/dcs

[2] This grammar text of eight chapters consists of 3,959 sūtras (rules) regarding morphology, syntax and semantics.

Gītā (BhG for short).

Śrīmad Bhagvad Gītā – the divine song of the Bhagvān is a part of the epic Mahābhārata (section 25 to 42 of the Bhiṣma parva). It consists of 18 chapters and 700 verses (ślokas). It is an important text which summarizes the Upanishadic teachings and is commented upon and interpreted by various schools of Indian philosophies. Being an important text, it has been translated into almost all major languages of the world, and also commented numerously. Thus annotators in doubt can always refer to these commentaries for correct annotation. This scripture being coherent and complete in itself, can be used for higher level analysis such as discourse analysis, topic identification, anaphora resolution, and so on. Further, this also being part of the Mahābhārata, later on if necessary, it can be used as an initial training data for boot-strapping for automatic annotation of complete critical edition of Mahābhārata (with around hundred thousand verses).

In the next section, we describe various levels of tagging and the stages of annotation, the tagging guidelines that were followed and in the third section the methodology we followed in the creation of such an annotated text. In the fourth section we describe various statistical facts about such an annotated text. In the concluding fifth section, we discuss various usages of such a corpus, one such application being a teaching cum accessing tool.

2 Levels of tagging

Common modules needed for any NLP application are parser, discourse level analyser, anaphora resolution module, Named Entity recognizer, and so on. So we find gold standard data being created for evaluating all such modules. In addition to all the levels of tagging mentioned above, Sanskrit, owing to its special nature, requires a few more levels of annotations viz., tokenizer, compound analyser and verse to prose word order generator.

2.1 Annotation of Sandhi

Sanskrit is largely influenced by the oral tradition and this is reflected in its writing style as well. As a result we come across Sanskrit texts as a continuous string of characters without any space in between the word boundaries. The words at the word boundaries undergo euphonic changes (called sandhi) as well. Thus, for analyzing any Sanskrit text, we need a tokenizer which will tokenize an input sentence into possible tokens – the meaningful words. Thus the first level of tagging needed for Sanskrit is the marking of word boundaries undoing the sandhi. At this level we split two types of sandhis – sandhi between two words and also sandhi between the components of a compound. We indicate these two sandhis by two different symbols. The sandhi between two words is indicated by a '+' sign while the sandhi between the components of a compound is indicated by a '-' sign.

For example the fortieth verse from the second chapter of BhG is :

nehābhikramanāśo'asti pratyavāyo na vidyate|
svalpamapyasya dharmasya trāyate mahato bhayāt||

Eng: In this path there is no loss of effort, nor is there any adverse result. Even a little practice of this discipline protects one from great fear (of birth and death).

This verse is tokenized as shown below –

na+iha+abhikrama-nāśaḥ+asti pratyavāyaḥ+ na vidyate|
svalpam+api+asya dharmasya trāyate mahataḥ+ bhayāt||

In this verse, there is one compound word *abhikrama-nāśaḥ*, which is joined with other words *na*, *iha*, and *asti* by sandhi to form a single word *nehābhikramanāśo'asti*, similarly *svalpamapyasya* is formed by joining three consecutive words *svalpam+api+asya*. Also there are instances where the last character of the previous word undergo some phonetic changes due to the presence of certain set of characters in the beginning of following word as in *pratyavāyo* and *mahato*.

2.2 Tagging of compounds

Sanskrit compounds with an exception of dvandva (conjunctive) are binary. So when a compound consists of more than one component, it is not enough to show the constituent components alone. For, the way the components combine together

decides the meaning of a compound. A compound with 3 components a-b-c may be combined in two different ways viz., <<a-b>-c>, and <a-<b-c>>. In addition, one also needs to know what type of compound it is – whether a copulative (karmadhārayaḥ) or an endo-centric (tatpuruṣaḥ / avyayībhāvaḥ) or an exo-centric (bahuvrīhiḥ) or a conjunctive (dvandvaḥ). Kumar et al. (2010) describe various stages involved in the analysis of a compound which also form the natural modules of a compound processor [3] viz.,

1. Segmentation (samāsapadacchedadh)
2. Constituency Parsing (samāsapadānvayaḥ)
3. Compound Type Identification (samastapadaparicāyakaḥ)
4. Paraphrasing (vigraha-vākyam)

A hierarchical tagset of 55 tags has been designed to tag the Sanskrit compounds.

2.3 Morphological Analysis

Sanskrit is rich in morphology – both the inflectional as well as derivational. In order to get the correct parse, both the inflectional as well as derivational information is needed. Hence at this stage we tag derivational information as well wherever applicable.

2.4 Tagging of sentential relations

Sanskrit grammar texts discuss various relations among words necessary to interpret the meaning of a sentence. Ramakrishnamacharyulu (2009) compiled and classified all these relations and further they were investigated for their suitability for automatic parsing. Out of around 90 relations listed there, only those relations which one can predict based on the syntactico-semantic information available in a sentence are considered for automatic tagging (Kulkarni and Ramakrishnamacharyulu, 2013). There are around 35 of them. Complete BhG has been tagged at syntactic level using this tag set.

2.5 Marking the Prose order

Sanskrit literature is dominated by poetic style. In order to understand a text in verse style, two different methods have been followed in Indian education system viz., Daṇḍānvaya (also known as anvayamukhī) and Khaṇḍānvaya (also known as kathambhūtinī). In the first approach the teacher arranges all the words in prose order. This makes the understanding of a verse easy for a student. In the second approach, on the other hand, the teacher gives the basic skeleton of a sentence and fills in other details by asking questions. [4] These questions are centered around the heads seeking their various modifiers. This approach is close to parsing a sentence showing various dependency relations. The first approach on the other hand assumes that if a user is given the 'default prose order' of the sentence, he 'understands' its meanings. So the question is what this 'default word order' is. The default word order or the 'canonical form' is governed roughly by the following verse:

viśeṣaṇam puraskṛtya viśeṣyam tad-lakṣaṇam
kartṛ-karma-kriyā-yuktam etad anvaya-lakṣaṇam
(samāsacakram kā.verse 10)
gloss: Starting with the adjectives, targeting the headword, in the order of kartṛ-karma-kriyā (subject-object-verb) gives an anvaya.

For each of the verse, we provide the canonical form of it. Sometimes, a single verse may correspond to more than one sentences [5] or more than one verses may form a single sentence.

Below we show one foot of the second chapter's fortieth verse annotated at all these levels.

<l>
<seg type="pāda">
<euphonic-word no="1"> nehābhikramanāśo'sti
<word no="1" prose word order_no="3"> na
<mo_anal> na{ind} </mo_anal>
<syntactic_rel> mod 4 </syntactic_rel>
</word>
<word no="2" prose word order_no = "1"> iha
<mo_anal> iha{ind} </mo_anal>
<syntactic_rel> loc 4 </syntactic_rel>
</word>
<word no="3" prose word order_no="2"> ab-

[3] http://sanskrit.uohyd.ernet.in/samAsa/frame.html

[4] Tubb and Boose (2007) gives a good illustration of these approaches.

[5] A sentence is defined as: *arthaikatvād ekam vākyam sākāṁkṣam ced vibhāge syāt* -mīmāṃsā sūtram 2.1.14.46 (If the entire unit conveys a single unitary sense or purpose and if any sub-unit on separation has syntactic expectancy of words from other sub-unit, it is a sentence. Here sentence is understood as one-meaning unit.)

hikramanāśaḥ
<compound label="T6">
<component no="1"> abhikrama
</component>
<component no="2"> nāśaḥ
<mo_anal> nāśa{masc}{nom;sg} </mo_anal>
</component>
</compound>
<syntactic_rel> subj 4 </syntactic_rel>
</word>
<word no="4" prose word order_no="4" > asti
<mo_anal>
as2{active;pres;nom;sg;parasmaipadī}
</mo_anal>
</word>
</euphonic-word>
</seg>
</l>

3 Methodology

The methodology we follow for tagging is semi-automatic due to following reasons:

- BhG being a popular text, a learned Sanskrit scholar could easily split the sandhi and compound just by looking at the verse content.
- A proper user interface for tagging using the existing tools were under development.
- The existing segmenter produced multiple segments which needed much time and effort to select the correct split in comparison to splitting the text manually.

In what follows, we describe the process below:

1. The verse form is converted into prose form.
2. Initially the sandhi and compound in the verse are segmented manually, following the guidelines developed by the SHMT consortium [6]. Then each compound is tagged for its type, along with the complete constituency mark-up.
3. The segmented words are run in the anusāraka interface [7] for obtaining the multiple morph analysis. The output generated as an xml file, is then manually pruned for choosing the correct morph analysis in the context.
4. The synactico-semantic relations are tagged manually, following the guidelines develped by the SHMT consortium [8].
5. The Hindi and English glosses for each word are given manually. For this we followed (Goyenka, reprint 2007).

4 Quantitative Analysis of Gītā

BhG has 700 verses. Majority of these verses (645) are composed in a metric called *anuṣṭup* [9] and the remaining verses are in *indravajrā* [10], *upendravajrā* [11] *and upajāti* [12] metres. A string of characters separated by spaces may correspond to one or more words. The total number of string sequences separated by spaces are 6426 amounting to 9.18 words per verse. After splitting these strings into words, there were 8884 words [13] In other words after segmentation there was around 13.82% increase in the words. Out of 8884, 1413 words were found to be compounds amounting to 15.9%. This suggests that a segmenter which segments a string into words and the compound words

[6] This is the Consortium of 7 institutes, for 'Development of Sanskrit-Hindi Machine Translation System (sampark)' funded by DIT, Govt. of India

[7] http://sanskrit.uohyd.ernet.in/scl

[8] http://sanskrit.uohyd.ernet.in/scl/Corpus/TaggingGuidelines/kaaraka-tagging-guidelines

[9] Each pāda is known as 'caraṇa' (foot) which has eight characters. A verse has four pādas and hence it has in total 32 characters.
śloke ṣaṣṭhaṃ guruṃ geyaṃ sarvatra laghu pañcamaṃ
dvicatuṣpādayorhrasvaṃ saptamaṃ dīrghamanvayoḥ
(Eng: Each 'foot' has eight characters. The first four characters can be of any mātrā. The sixth character is 'guru' while the fifth is 'laghu'. The seventh character is 'hrasva' in even 'foot' and 'guru' in odd 'foot') as in-
|S S S | S | S

[10] syādindravajrā yadi tau jagau gaḥ -vṛttaratnākara 3.28
(Eng: The characters in each 'foot' is in the order of two 'tagaṇa', one 'jagaṇa' and two 'guru') as in-
SS | SS | |S | SS
tagaṇa tagaṇa jagaṇa two-guru

[11] upendravajrā jatajāstato gau -vṛttaratnākara 3.29
(Eng: The characters in each 'foot' is in the order of 'jagaṇa', 'tagaṇa', 'jagaṇa' and two 'guru') as in-
|S | SS | |S | SS
jagaṇa tagaṇa jagaṇa two-guru

[12] itthaṃ kilānyāsvapi miśritāsu
smaranti jātiṣvidameva nāma me -vṛttaratnākara 3.31
(Eng: When each 'foot' in the verse is of mixed metres, that metre is known as 'upajāti').

[13] According to Pāṇinian terminology, these words are called *pada*s. These are the words which can be analysed showing the stem and the suffix resulting into the inflected form.

into its components is a must for any practical application of NLP to Sanskrit.

Compound words were analysed both semantically as well as syntactically. There are around 84.7% compounds consisting of only two components, followed by around 13.5% words with three components and 1.8% compounds with four components. All these compounds with more than two components were found to be left-branching as expected for any head last phrase. Semantically the largest number of compounds were of endocentric type. They were 55%. The exocentric compounds were around 25%, followed by the conjunctive and copulative each amounting to approximately 9% and only 2% compounds were of type *avyayībhāva*. The distribution of these compound tagsets is given in Table 1.

Compound-type	Freq
Endocentric	994
Exocentric	390
Copulative	163
Conjunctive	144

Table 1: Compound distribution.

The morphological analyser [14] gives the analysis of each word. Then we selected the correct analysis in the context manually. Around 5% of the words were not recognized by the morphological analyser. The reasons for non recognition were two. While for some the stems were missing in the lexical dictionary, for the others, the words were a sort of exceptions to Pāṇini's system, and hence could not be analysed. For all such words, we provided the morphological analysis manually. Sanskrit being inflectionally rich, we expected less ambiguity at the morphological analysis level. However, we found that there were quite a few words with as many as 10 or more analyses. Table 2 gives the frequency of words with multiple analyses.

The average number of analyses also (= 1.90) seems to be a bit on higher size. These different forms do not belong to different Part of Speech categories but typically fall within the same POS category. For example, a neuter noun in singular has the same form in nominative as well as accusative case. There are many such instances of regular clashes of forms in different case-number combination. Hence POS tagger based purely on

[14] http://sanskrit.uohyd.ernet.in/scl

analysis count	words	analysis count	words
1	5280	8	22
2	1570	9	28
3	1082	10	15
4	349	11	10
5	292	12	7
6	99	13	3
7	121	14	6
		Total words	8884
		Average	1.90

Table 2: Morph level ambiguity

the category information for disambiguation is of very little use in Sanskrit. On the other hand a hierarchical tagger with information of various associated features makes sense.

Sanskrit has 8 cases and 3 numbers (singular, dual, plural). Thus every noun has 24 nominal forms. Out of these, it was found that dual forms are very rare. The total number of cases of these 24 types are recorded in table 3.

case	singular	dual	plural
nom.	2463	31	613
acc.	1349	20	251
instr.	266	0	94
dat.	57	0	5
abl.	116	0	12
gen.	335	24	179
loc.	273	3	92
voc.	251	1	0

Table 3: Case-Number distribution

After manual selection of the correct morph analysis in context, the distribution of cases with numbers is shown in table 4.

case	singular	dual	plural
nom.	864	29	501
acc.	733	20	103
instr.	78	0	48
dat.	29	0	5
abl.	53	0	12
gen.	222	24	98
loc.	155	3	48
voc.	80	1	0
total	2014	74	795

Table 4: Case-Number distribution in context

Sanskrit has 10 conjugation classes, and

10 tense-mood parameters (known as lakāras). Among these only some of the lakāras occur more frequently. The distribution of these in BhG is given in table 5.

lakāra	freq
laṭ (Present)	355
loṭ (Imperative)	72
lṛṭ (Second future)	42
vidhiliṅ (Potential)	40
laṅ (Imperfect)	26
liṭ (Perfect)	22
luṭ (First future)	16
luṅ (Aorist)	9
āśīrliṅ (Optative)	6
lṛṅ (Conditional)	1

Table 5: Tense-Mood distribution.

The distribution of various syntactico-semantic relations is shown in table 6.

relation	freq	relation	freq
adjective	1277	kartā (subject)	1256
conjunctive	1155	karma (object)	924
predicative adj	401	locative	358
genitive	357	negation	242
emphatic	275	vocative	237
precedence	194	adverb	194
instrument	130	karma-sāmānidhikaraṇa	113
causal	96	co-relative pronouns	82
source	75	vākya-karma	64
purpose	52	simultaneity	50
dative	15	disjunction	10

Table 6: Distribution of Relations

5 Utility and Conclusion

We discuss below three important usages of this gold data. The first one, obviously, for NLP applications, the second one as linguistic inputs for developing a domain specific primer for learning BhG, and the third one is with the suitable interface, a self-learning tool for BhG.

5.1 GOLD data for NLP applications

This data, as stated above serves as a gold standard for evaluation of various Sanskrit tools. Further this is also useful for developers of NLP tools. For example, we observe that (refer table 6), the cases of adjectives are as high as those of subject. Similarly the cases of conjunction are also on the high side. This information is useful in prioritizing the solutions in the case of ambiguities. The linguists are also benefited with such a highly analysed data. This forms a basis for the theoretical linguists and grammarians to test their theories.

5.2 Input for Domain Specific Primer

As the trend is, typically at a grown up age, an Indian takes up the studies of various scriptures, BhG being prominent among them. Naturally, a person starts taking up courses in basic Sanskrit, in order to understand BhG. Various quantitative analysis we have carried out above is useful to develop a primer for Sanskrit in order to understand BhG. Kulkarni (2013) describes the development and use of a primer designed on the basis of similar analysis of a set philosophical texts on Dvaita-vedānta. The quantitative analysis can help a teacher to decide which aspect of Sanskrit Grammar is more relevant for the study of BhG. For example, any Sanskrit noun declines in 8 cases and 3 numbers. However, as we observe in Table 3 above, the presence of nominal forms in dual is less that 3%. So the teacher can postpone teaching dual forms to a later stage. Similarly, he can use this data further to decide how many and which paradigms of noun declension to concentrate on first. The teacher can use the analysed data of Tense-Modality to decide which lakāras to teach and which conjugation classes to concentrate on first. There are four major semantic categories of a compound, which are sub-classified into 55 categories based on their paraphrase. But we noticed that more than half of them belong to one major category of endo-centric type. Thus the teacher can concentrate on teaching this class first. There are hardly 2% of *avyayībhāva* type, which he may explain when they occur.

5.3 As a Self Learning Tool

Finally, with a suitable interface, this analysed data can be used as a self learning tool for BhG. In order to understand any Sanskrit text one basically goes through the following steps:

1. *padajñāna* i.e., Identifying the word boundaries.

2. *padārthajñāna* i.e., knowing the meaning.

3. *vākyārthajñāna* i.e., syntactic knowledge with relations.

4. With the help of a suitable interface such as that of anusāraka (Kulkarni, 2009) an interested reader can have complete analysis of BhG at various levels. The interface provides –

   (a) User controlled access to various levels of analysis (Fig 1,2,3). The graphs showing the constituency information and the kāraka relations are generated automatically from the manually tagged data.

   (b) Link to various dictionaries for meanings of the head words.

   (c) Graphical display of phrase structure analysis of compounds (Fig 4).

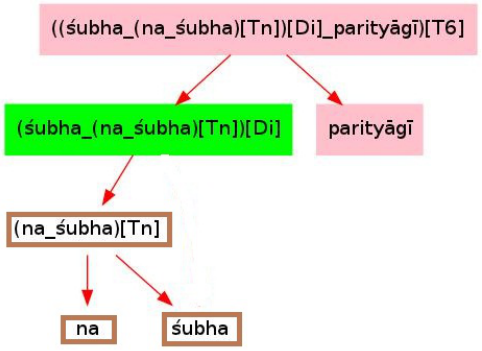

Figure 4: Compound analysis of a word from BhG 12.17

   (d) Graphical display of sentential analysis (Fig 5).

This provides the user a digitized learning and understanding environment. This interface results from the xml tagged version of the BhG. Such an interface comes as a handy tool for a linguist as well (Dimitriadis, 2010).

BhG being in a verse form, one may question the extendibility of this approach to other texts, particularly the prose ones. As is well known, Sanskrit is a free word order language and as far as the dependency analysis is concerned, Kulkarni et al. (2013) have shown that both the prose as well as verse follow the criterion of sannidhi (proximity). So it is not the case that verse is more 'free' than the prose. The degree of 'free-ness' is the same in both the type of texts. What really may matter is the semantics involved with the words. While the

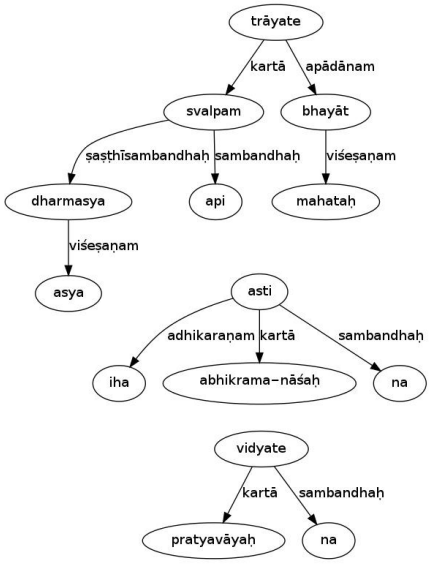

Figure 5: Dependency graph of BhG 2.40

morphology and syntax is more-or-less intact due to the existence of Pāṇinian grammar, the semantics of the content words might have been changed over the period of time, which needs to be investigated further.

References

Hariprasad Bhagirath. 1901. *Samāsacakram*. Jagadishwar Press, Mumbai, India.

Alexis Dimitriadis. 2010. Matching needs and resources: How nlp can help theoretical linguistics. In *Workshop on NLP and Linguistics: Finding the Common Ground, ACL 2010*. Uppsala, Sweden.

Pawan Goyal, Gérard Huet, Amba kulkarni, Peter Scharf, and Ralph Bunker. 2012. A distributed platform for sanskrit processing. In *Proceedings of 24th COLING, Mumbai*.

Jayadayal Goyenka. reprint 2007. *Śrīmad Bhagvad Gītā*. Geeta Press, Gorakhpur, India.

Oliver Hellwig. 2009. Sanskrittagger, a stochastic lexical and pos tagger for sanskrit. In *Sanskrit Computational Linguistics. First and Second International Symposia*, pages 266–277. Springer,Berlin.

Gérard Huet. 2003. Towards computational processing of sanskrit. In *International Conference on Natural Language Processing*.

Daniel Jurafsky and James H. Martin. 2008. *Speech and Language Processing*. Prentice Hall, Upper Saddle River, N.J.

Amba Kulkarni and Anil Kumar. 2011. Statistical constituency parser for Sanskrit compounds. In *Proceedings of ICON 2011*. Macmillan Advanced Research Series, Macmillan Publishers India Ltd.

Amba Kulkarni and K. V. Ramakrishnamacharyulu. 2013. Parsing Sanskrit texts: Some relation specific issues. In Malhar Kulkarni, editor, *Proceedings of the 5th International Sanskrit Computational Linguistics Symposium*. D. K. Printworld(P) Ltd.

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

Amba P Kulkarni, Anil Kumar, and V Sheeba. Sanskrit compound paraphrase generator.

Amba Kulkarni, Sheetal Pokar, and Devanand Shukl. 2010. Designing a constraint based parser for Sanskrit. In G N Jha, editor, *Proceedings of the 4th International Sanskrit Computational Linguistics Symposium*. Springer-Verlag LNAI 6465.

Amba P Kulkarni, Preeti Shukla, Pavankumar Satuluri, and Devanand Shukl. 2013. How much 'free' is the free word order in sanskrit. forthcoming.

Amba P Kulkarni. 2009. *Anusaaraka: An approach for MT taking insights from the Indian Grammatical Tradition*. Ph.D thesis (Unpublished), University of Hyderabad, Hyderabad.

Tirumala Kulkarni. 2013. vedāntasaṃskṛtam. In *5th International Sanskrit Computational Linguistics Symposium*.

Anil Kumar, Vipul Mittal, and Amba Kulkarni. 2010. Sanskrit compound processor. In G N Jha, editor, *Proceedings of the 4th International Sanskrit Computational Linguistics Symposium*. Springer-Verlag LNAI 6465.

Anil Kumar. 2012. *An automatic Sanskrit Compound Processing*. Ph.D thesis (Unpublished), University of Hyderabad, Hyderabad.

M Marcus, M Marcinkiewicz, and B Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19,2:313–330.

David Pingree. 1978. In *Proceedings of the American Philosophical Society*, volume 122.

Barbara Plank and Gertjan van Noord. 2010. Grammar-driven versus data-driven: Which parsing system is more affected by domain shifts? In *Workshop on NLP and Linguistics: Finding the Common Ground, ACL 2010*. Uppsala, Sweden.

K V Ramakrishnamacharyulu, A P Kulkarni, and Anil Kumar.

K V Ramakrishnamacharyulu. 2009. Annotating the sanskrit texts based on the śābdabodha systems. In *Proceedings of 3rd International Sanskrit Computational Symposium*. Springer-Verlag LNAI-5406.

Swami Ramsukhdas. 1998. *Geeta Darpana*. Geeta Press, Gorakhpur, India.

Pt. Alakhdev Sharma. 1989. *Param Laghu Mañjūṣā of Nagesh Bhatt*. Choukambha Amarbharati Prakashan, Varanasi, India.

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

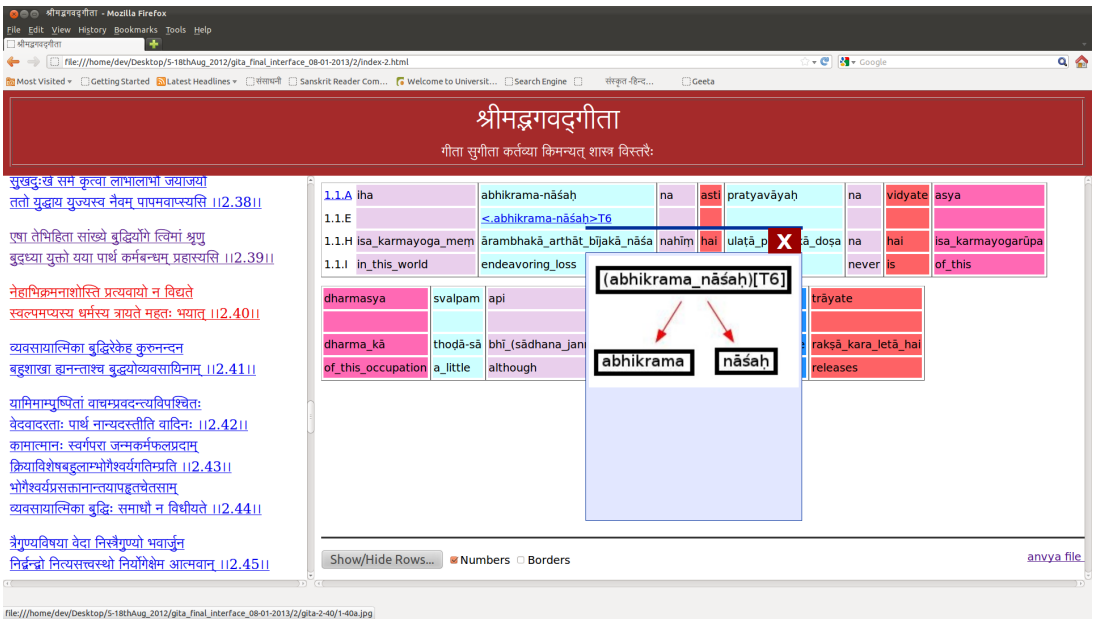

Figure 1: compound analysis of BhG 2.40

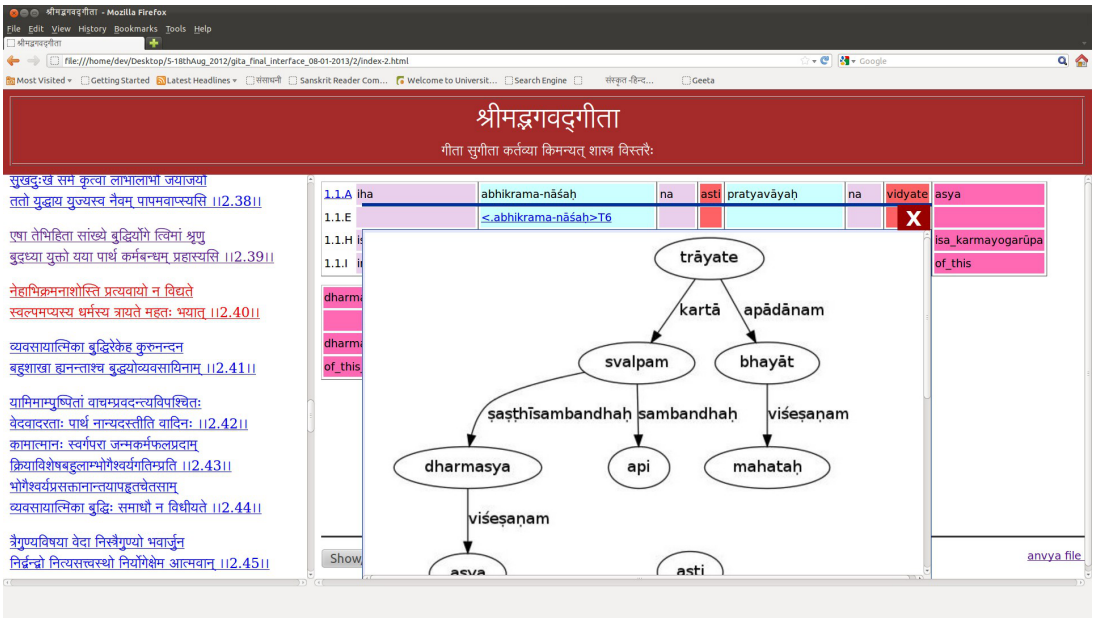

Figure 2: kāraka analysis of BhG 2.40

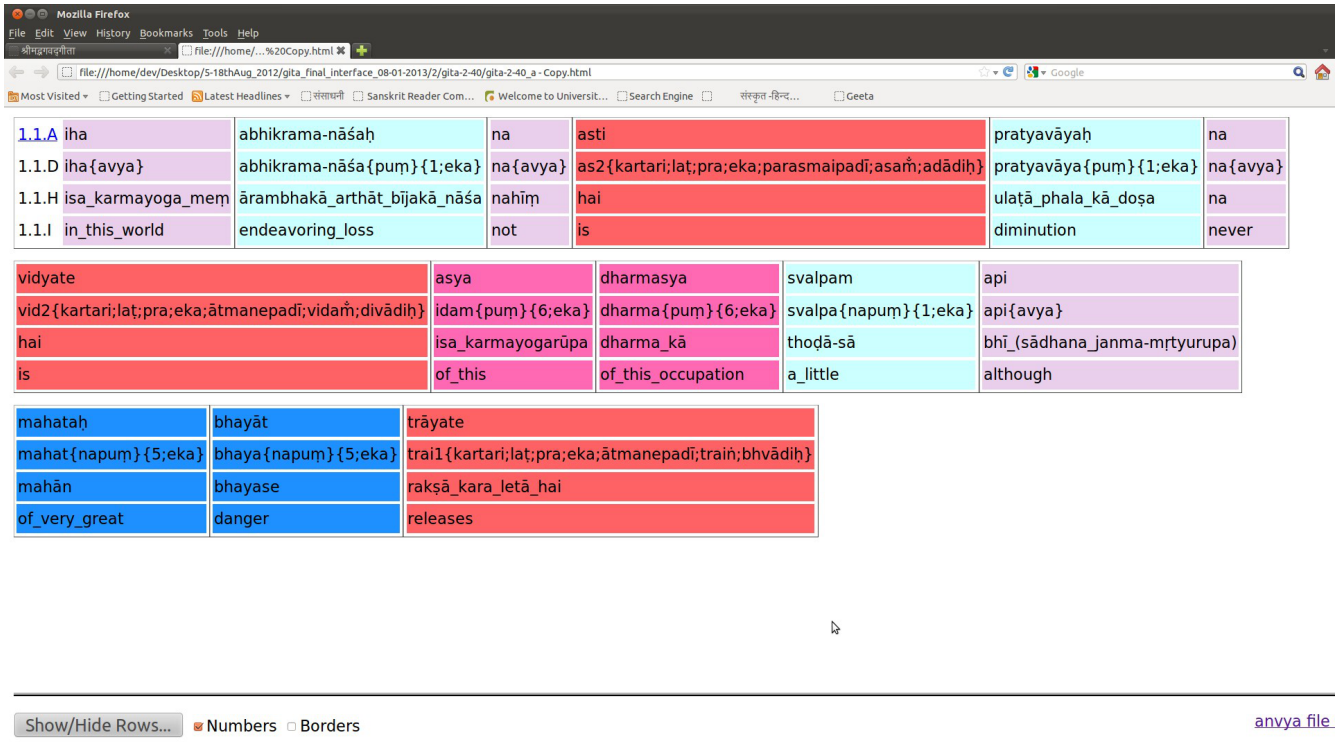

Figure 3: morph analysis of BhG 2.40