Dependency Relations for Sanskrit Parsing and Treebank

Amba Kulkarni†, Pavankumar Satuluri‡,
Sanjeev Panchal†, Malay Maity†, and Amruta Malvade†
†University of Hyderabad, India
‡Chinmaya Vishwavidyapeeth, India
ambakulkarni@uohyd.ac.in

Abstract

Dependency relations are needed for the development of a dependency treebank and a dependency parser. The guidelines[1] for the development of treebank for Sanskrit proposed a set of dependency relations. Use of these relations for the development of a sentence generator and a dependency parser for Sanskrit demanded a need for an enhancement as well as a revision of these relations. In this paper, we discuss the revised version of these relations and discuss the cases where there is a possibility of multiple tagging either due to the ellipsis of certain arguments or due to the possible derivational morphological analysis. This led us to arrive at specific instructions for handling such cases during the tagging. A treebank with around 4000 sentences has been developed following these guidelines. Finally we evaluate a grammar based dependency parser for Sanskrit on this treebank and report its performance.

1 Introduction

Sanskrit is one of the oldest languages in the world and has literature at least hundred times that of Greek and Latin together. This literature ranges from scientific disciplines such as Mathematics, Āyurveda, texts dealing with Language Sciences, Ontology, Logic, Metallurgy, Physics, Polity, and Law to Philosophical texts, Epics and several texts of lasting artistic merit. India's contribution to the development of Language Sciences dealing with various branches such as phonetics, phonology, morphology, syntax, semantics, discourse analysis and logic are found to be relevant for Language Technology. Among these, Pāṇini's grammar and the theories of verbal cognition deserve special mention from the Natural Language Processing (NLP) perspective. While the Pāṇini's grammar provides an almost complete grammar for generation, the theories of verbal cognition provide a systematic approach to analyse any text objectively. In this approach attention is paid to the information encoded in a linguistic expression. Division of a word into morphemes, role of some morphemes in connecting other morphemes, deciding the meaning of the morphemes are some of the topics that are discussed in these theories. Pāṇinian grammar provides the detailed description of how the semantic relations are realised through various morphological features, word order, and various other means of information encoding. The theories of verbal cognition use these clues of information encoding and other factors such as expectancy, mututal congruency of word meanings, proximity of the arguments etc. to decide the relations between the words.

The semantic relations used by Pāṇini to describe various relations thus provide a basic set for developing a dependency parser and also for the development of a treebank. This set of relations was enhanced over a period of 2-3 millenia by the grammarians and theoreticians working in the field of verbal cognition. A list of all such relations is compiled by Ramakrishnamacaryulu (2009) and presented as dependency relations for Sanskrit for both inter-sentential as well as intra-sentential tagging. These dependency relations were used as a starting point and the consortium for Sanskrit-Hindi Machine Translation (SHMT) system[2] arrived at a set suitable for the development of Sanskrit treebank. This resulted into the first version of the tagging guidelines for Sansktit treebank[3]. While developing a dependency

[1] http://sanskrit.uohyd.ac.in/scl/GOLD_DATA/Tagging_Guidelines/Tagging_eng_ver1.pdf
[2] funded by Technology Development for Indian Languages, MeiTy, Government of India, 2008-2012
[3] https://sanskrit.uohyd.ac.in/scl/GOLD_DATA/Tagging_Guidelines/Tagging_eng_ver1.pdf

parser, and also a sentential generator for Sanskrit, it was noticed that this set of dependency relations has some limitations and needs further enhancement as well as modifications. In this paper we discuss the revised version of this set. This set of relations is also used to develop a Sanskrit treebank. We present the cases of ambiguities in tagging while developing the treebank. This treebank is also used for the evaluation of the Sanskrit parser. We present the performance of this parser and discuss the limitations of both the parser as well as the dependency relations.

The paper is structured as follows. In the next section, we provide the literature survey of the state-of-art dependency relations and treebanks for parsing. This is followed by the discussion on the modifications to the earlier Sanskrit dependency relations and the enhancement thereupon justifying the necessity. In the fourth section we describe the Sanskrit treebank followed by the evaluation of a grammar based parser on this treebank. This is followed by the conclusion.

2 Brief survey

The last two decades have established the suitability of dependency parse over a constituency parse, even in the case of positional languages, for a wide range of NLP tasks such as Machine Translation, question answering, information extraction. This led to the development of dependency treebanks for various languages. Most of the languages followed an easy path of converting the existing constituency treebanks into dependency treebanks. Therefore the dependency relations used by these treebanks are also more syntactic in nature. At the same time several efforts were on developing a dependency parser for English. For example, the Link grammar, which is closely related to a dependency grammar proposed a set of around 106 relations which were not directional (Daniel and Temperley, 1993). Minipar had 59 relations (Lin, 2003). Caroll et al. (1999) and King et al. (2003) had proposed a set of dependency relations which were used by Marneffe et al. (2006) to convert the Phrase Structure treebanks to Dependency treebanks. This effort also led to some modifications to these relations, largely based on practical considerations. The number of relations proposed by them were 47. Most of these relations were syntactic in nature rather than semantic. These relations were incorporated in the Stanford parser. Thus we see that there was a huge variation between the number of relations used by various research groups, and naturally their semantic content also differed.

For most of the morphologically rich languages like Czech, Hindi, and Finnish manually annotated dependency treebanks were developed. The Prague Dependeny Treebank (PDT) is one of the oldest dependency treebanks (Bejček et al., 2013). This treebank is annotated at both the syntactic as well as semantic (tectogrammatic) level (Böhmovà et al., 2003). AnnCorra, guidelines for annotating dependency relations based on Pāṇinian grammar, was developed for Indian languages, and the treebanks for major Indian languages were developed following these guidelines (Bharati et al., 2002).

The major effort towards bringing in a standard among the dependency relations is by (Nivre et al., 2016) who proposed the Universal dependencies.[4] The Universal dependencies aim for a common annotation scheme for all the languages so that cross-linguistic consistency among the treebanks for several languages is achieved. The Universal dependencies were evolved from the Stanford dependencies (Marneffe and Manning, 2008). Though most of the relations from the Universal dependencies are syntactic in nature, the nsubj relation together with the newly proposed nsubj:pass relation makes this pair equivalent to the concept of *abhihita* of the Pāṇinian dependencies (Bharati and Kulkarni, 2011). Around 90 languages in the world including the three Classical languages viz. Greek, Latin and Sanskrit have dependency treebanks following Universal Dependencies.

Among the classical languages, both Ancient Greek and Latin have dependency treebanks following their own grammars. The ancient Greek dependency treebank consists of 21,170 sentences (309,096 words) from ancient Greek texts (Bamman and Crane, 2011). The Latin dependency treebank (V.1.5) consists of 3473 annotate sentences (53,143 words) from eight texts. The Latin tagset (V.1.3) consists of 20 categories mainly and they are further elaborated into various types. In this tagset, they have explained, with examples, how to annotate specific constructions involving relational clauses, gerunds,

[4] `https://universaldependencies.org/`

direct speech, comparison etc.[5].

All these dependency relations are mostly syntactic in nature. A strong need is also felt for the semantic annotation. Levin and Rappaport (2005) discuss the problems in thematic level annotation. This led to other models for semantic level tagging. Propbank (Palmer et al., 2005) and FrameNet (Fillmore et al., 2003) are the two prominent among them.

Pāṇini's scheme for annotation of relations is syntactico-semantic (Kulkarni and Sharma, 2019). Unlike the semantics dealt with in Propbank or the FrameNet annotations, in Pāṇini's scheme, the level of semantics is precisely the one that can be extracted only from the linguistic expression (Bharati and Kulkarni, 2010).

3 Saṁsādhanī Dependency Relations

Manually annotated data at various levels has become now an essential resource for computational analysis of texts. Such a resource is not only useful for machine learning but also comes handy as a test data for grammar based systems. To extract various kinds of relations between words in a sentence, it is necessary to have a corpus tagged at the level of relations between the words. Pāṇini's grammar provides semantic definitions of various relations between words and also provides rules that tell us how these relations are realised morphologically. The noun-verb relations are called the *kāraka* relations which refer to six different types of participants of an action viz. *kartā* (roughly an agent), *karma* (roughly a goal or a patient), *karaṇam* (instrument), *sampradānam* (recipient), *apādānam* (source) and an *adhikaraṇam* (location). The Indian grammarians further sub-classified and enhanced these relations by introducing a few more relations that deemed to be necessary from analysis point of view. In addition, two other relations viz. *prayojanaṃ* (purpose) and *hetuḥ* (cause) also involve noun-verb relationship. The list of all these relations, with around 100 entries, is collected and classified by Ramakrishnamacaryulu (2009). This list was the starting point in framing tagging guidelines in building treebanks. It was noticed that these relations were very fine-grained, and were neither suitable for a human annotator nor for computer parsing with high accuracy. Taking into consideration both the aspects viz. the manual tagging as well as the automatic parsing, around 31 relations were chosen from this set (Kulkarni and Ramakrishnamacharyulu, 2013). A treebank of around 3,000 sentences was developed following these guidelines.[6] These dependency relations, when, were examined from the sentence generation point of view, it was noticed that this set has several relations that were not semantic in nature, and referred to the morphological requirement or were syntactic in nature. This forced us to look at these relations afresh.

3.1 Enhancements and Modifications

In Sanskrit, there are certain words, in the presence of which a noun gets a specific nominal suffix. This is a morphological requirement, and in Pāṇini's grammar no semantics associated with such morphological requirements is discussed. As an example of such requirement let us consider the following sentence.

(1) *Skt:* *grāmaṃ* *paritaḥ* *vṛkṣāḥ* *santi.*
Gloss: village{sg,acc} surrounding tree{pl,nom} be{pres,pl,3p}.
Eng: There are trees surrounding the village.

In this sentence, the verb 'be' is not a copula, but indicates an existence. The word *paritaḥ* (surrounding) refers to the location and has an expectancy of a reference point, and the word denoting this reference point gets an accusative case marker. Figure 1 shows both the old and the new versions. In the old version, the label was *upapadasambandhaḥ* (literally 'a relation due to an adjacent word') which was a morphosyntactic label. In the new version this has been replaced by a semantic label '*sandarbha_binduḥ*' (reference point). When the word *paritaḥ* (surrounding) is used, there is a natural expectancy: 'surrounding what?'. The answer to 'what' gives a reference point for surrounding. Hence this relation is termed 'reference point' (*sandarbha_binduḥ*).

[5] `http://static.perseus.tufts.edu/docs/guidelines.pdf`

[6] http://tdil-dc.in/san (available for research purpose from TDIL)

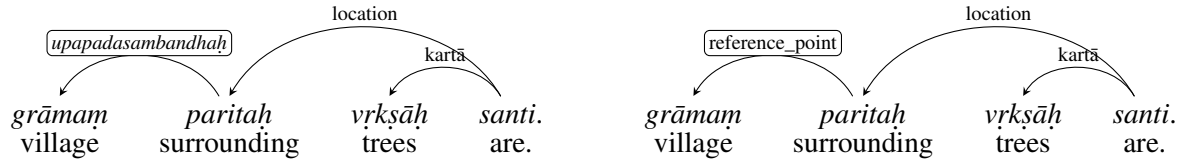

Figure 1: Old and New annotations

Another pair of relations that needed modification was '*anuyogī*' and '*pratiyogī*'. These were the relations used to connect two sentences by a connective. The two words *anuyogī* and *pratiyogī* are from the Indian logic which are used to refer to the two relata of a relation. In the old annotation scheme, some of the relations were not analysed semantically, and hence a general scheme of naming them as relata1 (*anuyogī*) and relata2 (*pratiyogī*) was followed. We illustrate this with an example. Consider the sentence

(2) *Skt:* *ahaṃ gṛhaṃ gacchāmi iti rāmaḥ avadat.*
Gloss: I home{sg,acc} go{pres,1p,sg} thus Rama{nom} say{past,3p,sg}.
Eng: Rama said that he goes home.

In this sentence the relation of the particle *'iti'* (thus) with *gacchāmi* (goes) and *avadat* (said) was marked as *pratiyogī* and *anuyogī* in the earlier version. The embedded sentence being the sentential argument, we propose *vākyakarma* (literally meaning 'sentential object') relation between the heads of the main and the embedded sentence. And *'iti'* serves as a marker for this relation, and hence it is marked as *vākyakarmadyotakaḥ* (literally meaning 'indicator of sentential argument').

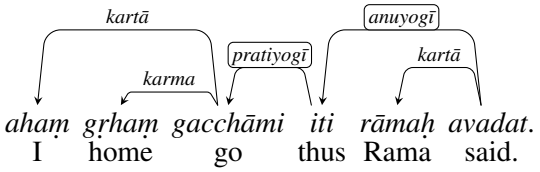

Figure 2: Complementiser: Old version

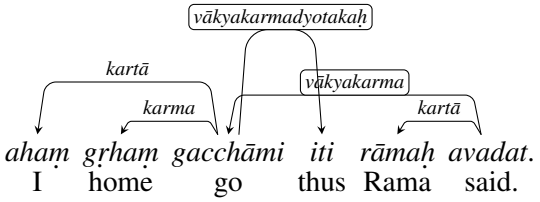

Figure 3: Complementiser: New version

Similarly consider the sentence

(3) *Skt:* *yadā meghāḥ varṣanti tadā mayūrāḥ nṛtyanti.*
Gloss: When cloud{pl,nom} rain{pres,3p,pl} then peacock{pl,nom} dance{pres,3p,pl}.
Eng: When clouds shower then peacocks dance.

In the earlier version the relations were as shown in Fig. 4. The two relations *anuyogī* (relata1) and *pratiyogī* (relata2) and the relation *sambandhaḥ* (literally 'relation') do not provide any semantics other

than that the two words *yadā* (when) and *tadā* (then) are related to each other and they in turn are related to the finite verbs of the respective sentences. But what is the relation between them is not specified. In the revised scheme, these relations are changed as shown in Fig. 5. The modified version clearly marks the relation between co-relatives (when-then), and also marks the semantic relation of each of the co-relative with the verb as a time-location. The revised scheme thus provides a better semantics than the previous one.

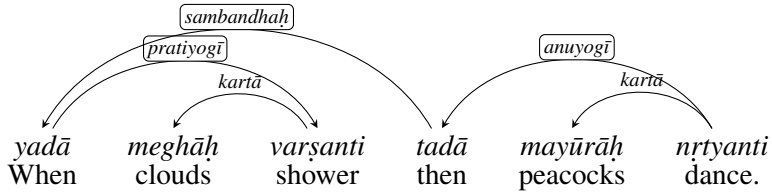

Figure 4: Co-reference: Old version

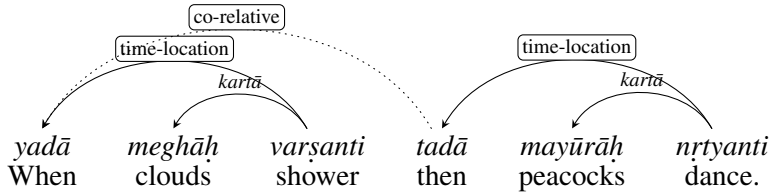

Figure 5: Co-reference: New version

Finally the third major modification was with regards to the co-ordinating conjuncts. In the earlier set of relations the conjunctive particle (*samuccaya-dyotakaḥ*) was marked as the head, connecting the conjuncting co-ordinates by a relation *samuccitam* as shown in Fig 6. This was modified as shown in Fig 7.
Let us look at the following sentence with a conjunct.

(4) *Skt:* *Rāmaḥ* *Sītā* *ca* *vanaṃ* *gacchati.*
Gloss: Rama{nom} Sita{nom} and forest{sg,acc} go{pres,3p,sg}
Eng: Rama and Sita go to forest.

Note here that the verbal form *gacchati* is in singular and not in dual.

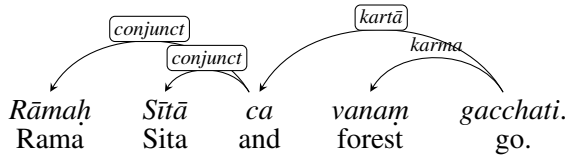

Figure 6: Conjuncts: Old version

In Sanskrit, it is observed that the last conjunct shows concord with the verb (Panchal and Kulkarni, 2019). The conjunctive particle acts as a marker, similar to the case suffix, to mark the relation between the two conjuncts. Hence in the modified analysis, the last conjunct in the phrase is marked as the head, with which the other conjunct is related by a *samuccitam* (conjunct) relation and the conjunctive particle is related to this head by the relation of *samuccaya-dyotakaḥ* (literary 'a marker for conjunction').

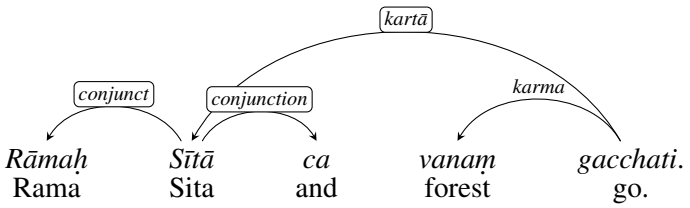

Figure 7: Conjuncts: New version

3.2 Saṁsādhanī Dependency Relations Version 2

The current version has 54 relations (see Appendix A) classified into the following categories.

- Predicate-argument relations
- Non-Predicate argument relations
  - verb-verb relations
  - verb-noun relations
  - noun-noun relations
- Relations due to special words
- Conjuncts and Disjuncts
- Miscellaneous

The predicate-argument relations are known as *kāraka* relations in Pāṇinian terminology. These are six in number with sub-classification of some of them. The six major relations are *kartā* (roughly agent), *karma* (roughly goal or patient), *karaṇam* (instrument), *sampradānam* (recipient), *apādānaṃ* (source) and *adhikaraṇaṃ* (location). If the activity involved is a causative one, then the agent of the basic activity is called *prayojya kartā* and the causative agent is called the *prayojaka kartā*. To account for the arguments of ditransitive verbs, we have introduced two sub categories of *karma* viz. *mukhyakarma* (primary object) and *gauṇakarma* (secondary object). These are something similar to, but not semantically equivalent to, direct and indirect object. As discussed in the previous section, a new tag *vākyakarma* is also introduced to mark a sentential argument to a verb.

Under the non-predicative arguments, the relations are categorised into three sub-categories. The relation of a finite verb with a non-finite verb marking precedence, simultaneity etc. forms the first category. The relation of a verb with a noun marking the cause or the purpose etc. constitutes the second sub-category. The genitive relation between two nouns, the adjectival relation, and the relation due to reduplication are some examples of the relations in the third sub-category. The relations in this category convey only a broad semantics. For example the genitive relation covers various semantic relations such as part-whole relation, kinship relation, and the possessive relation, and many more. Similarly the reduplication may mark a universal quantification, or intensity, etc. The exact semantics depends on the context.

The third category of relations is the set of relations due to certain special words called *'upapada'*s. These words govern the case suffix of the nouns they are in proximity with. Pāṇini has not discussed the semantics of these relations. We found that most of these words are related to the nouns whose case suffix they govern, and they indicate either a reference point or a comparison point. Then there are the relations due to conjuncts and disjuncts and a few miscellaneous relations. The detailed treatment of conjuncts is summarised in (Panchal and Kulkarni, 2019), and we do not discuss these here further. Finally there are relations between sentences. These are typically relations between two full sentences. These relations are marked by cetain indeclinable words such as if then (*yadi-tarhi*), because of (*tataḥ*), hence (*ataḥ*) etc. The relations between them are classified under miscellaneous, since, in the current guidelines we mark them as either relata1 and relata2, or just simply a relation. The terms 'relata1', 'relata2' and 'relation' do not provide any semantics. In Ramakrishnamacaryulu (2009), a semantic classification of inter-sentential relations is provided. The current guidelines need further enhancement to incorporate inter-sentential relations. This is out of scope of this paper and hence is not discussed.

3.3 Saṁsādhanī Parser

During the last decade there is an upsurge in the use of Machine Learning approaches for the development of Dependency parsers. Dependency parsers for several languages including Classical languages such as Latin and Greek are available. Most of these parsers follow the Data Driven approaches. The first parser for Sanskrit was built by Bhattacharyya (1986) using integer programming. Huet (2007) has a shallow parser that uses the minimal information of the transitivity of a verb as a sub-categorisation frame and models it as a graph-matching algorithm. The main purpose of this shallow parser is to filter out non-sensical segmentations. Hellwig et al. (2020) describe a syntactic labeler for manual annotation. This syntactic labeler expects a human being to select the pair of words, and the syntactic labeler suggests a label. This is a first stage towards developing an automatic full syntactic parser.

The first full-fledged parser for Sanskrit is described in Kulkarni (2019). This parser follows the Pāṇinian grammar and the theories of verbal cognition described in the Indian Sanskrit literature. The theories of verbal cognition describe three conditions necessary for verbal cognition. They are *ākāṅkṣā* (expectancy), *yogyatā* (meaning congruity) and *sannidhi* (proximity). Kulkarni (2019) has discussed the computational models of these three factors and describes the design of a parser following the theories of verbal cognition. This parser which is a part of the Saṁsādhanī platform, is implemented as an edge-centric binary join to build a dependency tree, in bottom-up approach, with local and global constraints on the edges and the edge labels. It uses the dependency relations provided in the Appendix A. It differs from the state-of-art parsers in the following aspects.

- It is a grammar based parser and follows the Indian theories of verbal cognition for parsing, while the current trend is to follow data driven approaches.
- It produces all possible parses while a typical parser produces only one parse. There are two reasons for allowing multiple parses. The first reason is, in Sanskrit we come across texts that have multiple readings. These multiple readings may be intended by the author or may be due to different philosophical interpretations. We would like to present all these readings to the reader. The second reason is, and this is purely due to the limitation of the implementation, the mutual congruency (semantic restrictions) between the word meanings is not checked while establishing the relations between words. This leads to over-generation and false positives. It is left to the readers to choose the correct parse from among the possible solutions.
- The solutions are ranked with a cost function which is defined as a sum of product of the cost associated with the relation and the distance between the two relata.
- The parse comes with an intelligent user interface and helps user to select the correct parse if the first parse is not correct.

4 Treebank

The first treebank of dependency analysis for Sanskrit was developed by the Consortium (SHMT-Consortium) executing the project entitled 'Development of Sanskrit Computational Tools and Sanskrit-Hindi Machine Translation System' sponsored by TDIL Programme, Ministry of Information Technology, Government of India, 2008-12. This treebank has 3000 sentences, mostly taken from the modern stories. However, this treebank is not available in public domain, and is available with the TDIL only for research. The second treebank was developed following the Universal Dependencies for a tiny corpus of 230 sentences from a *Pañcatantra* story (Dwivedi and Guha, 2017). The third treebank is the treebank of Vedic Sanskrit of 4004 sentences, which consists of both prose as well as verses, developed by Hellwig et al. (2020). This treebank also follows the Universal Dependencies.

We decided to develop a separate treebank from those described above. Firstly, since the dependency relations used by our parser for tagging are different from the Universal Dependency relations, the second and the third treebanks were not useful for us to evaluate our parser. Secondly we wanted to make the treebank thus developed open. The Saṁsādhanī platform contains three manually annotated texts. The first one is the *Saṅkṣepa Rāmāyaṇam* which has 100 verses. All these verses are tagged manually following the guidelines developed for the SHMT Consortium project. Shukla et al. (2013) reported

a GOLD data of *Śrimad-Bhagavad-Gītā* (BhG), a philosophical text in verse form, consisting of 700 verses. This text was also tagged at various levels - metrical, segmentation, morphological and dependency (Patel, 2018). For the dependency level tagging, the guidelines of SHMT project were followed. The third manually annotated text consists of the first 10 Cantos of a poem *Śiśupālavadhaṁ*[7] which were tagged following the same guidelines.

While these three tagged texts were available under the Saṁsādhanī platform, we noticed that since these treebanks are created by individuals, and are not cross checked, there are a few inconsistencies. Meanwhile, the development of the parser also prompted us to improve upon the dependency relations. So these treebanks need to be modified as per the new guidelines and need to be cross checked as well for consistency in tagging. During the development of a parser, a need was also felt of controlled texts for testing. This led us to develop a new treebank. The sentences for this new treebank are chosen from four different sources. One set is from the grammar books to ensure that the treebank covers various types of constructions and special cases discussed in the grammar books covering various cases of subcategorization etc. The second set contains 284 sentences from a Sanskrit text book for 9^{th} grade by NCERT (National Council for Education, Research and Training). These sentences are not isolated ones, but they constitute complete meaningful paragraphs or stories. The third set of sentences is from various books on Sanskrit learning. These are independent sentences covering wide vocabulary and syntactic constructions for the beginners. The fourth set of sentences is from the modern stories from a story book[8] which is being cross checked by the annotators. The annotation for *Śrīmad-Bhagavad-Gītā* is also being checked and corrected following the new guidelines. The treebank also contains a few verses from the first chapter of this poem. This treebank is available at http://sanskrit.uohyd.ac.in/scl/GOLD_DATA under the creative commons license.

4.1 Ambiguities during annotation

The annotation of all these four sets was checked by two or more of the authors independently. There were a few cases where there was a difference of opinion among the annotators. We discuss here an example of each type of the difference.

There were certain constructions involving non-finite verbs where two different annotations were possible. Here is an example.

(5) *Skt:* *Ṛṣīṇāṃ* *vacanaṃ* *pramāṇaṃ* *asti.*
Gloss: Seer{pl,gen} speech{sg,nom} authentic{sg,nom} be{pres,sg,3p}.
Eng: Seer's speech is authentic.

Here the word *Ṛṣīṇāṃ* is in genitive and hence it can be related to the following word *vacanaṃ* by a genitive relation. However, the word *vacanaṃ* itself is a gerund of the verb *vac* (to speak). Hence the relation of *Ṛṣīṇāṃ* with *vacanaṃ* may be considered to be that of a *kartā* (agent), according to Pāṇini's grammar.[9] In such cases we noticed that the annotators do not have consistency in tagging. This difference in tagging is probably not so important from the translation point of view, but it is important for the tasks such as information extraction, question answering etc. As far as the parser is concerned, it marks the relation as genitive if the genrund analysis is not available. If gerund analysis is available then it produces both the genitive as well as agent relation, giving priority to the agent relation. So the performance of the parser depends on the performance of the morphological analyser. Marking the relation as a genitive leads to loss of information. On the other hand, if the relation is marked as *kartā*, then one can always downgrade it to genitive, for translation purpose. A conscious effort on the part of the annotator is needed to mark such relations, and a good coverage morphological analyser producing

[7] `https://sanskrit.uohyd.ac.in/scl/e-readers/shishu/`

[8] "130 *Sanskrit kathā*", Dr. Narayan Shastri Kankar, Neetha Prakashan, New Delhi, 2007.

[9] *Kartṛkarmaṇoḥ kṛti* A2.3.65 - A *kartā* and a *karma* takes genitive case when the verb is in non-finite form denoting the activity.

analysis of derived stems is needed to get a correct parse.

Let us see another example.

(6) *Skt:* *mārgāḥ* *avaruddhāḥ* *bhavanti.*
Gloss: Road{pl,nom} blocked{pl,nom} be{pres,pl,3p}.
Eng: The roads are blocked.

Here the word *avaruddhāḥ* is a past participle of the verb *rudh* with prefix *ava*. Now this sentence can be analysed in two different ways as follows. The verb *bhū* may mean either 'to happen' or 'to become' and also 'to be'. Accordingly, we have two different interpretations.

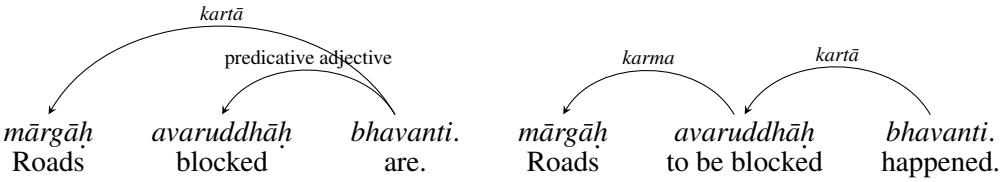

Figure 8: Inflectional Information

Both these analyses are correct. In the first one, the verb acts as a copula. The second one shows the analysis with the verbal meaning 'to happen', and 'being blocked' as its *kartā*. The *mārgāḥ* (roads) is, then, the object of blocking. As in the previous case, the first one is good enough for translation while the second one is better for deeper semantic analysis. In both the above cases, we propose that the manually tagged corpus should produce the analysis that uses the derivational information.

Another observation regarding tagging was with the elliptical sentences. Since Sanskrit is a highly inflectional language, there is no specific position (such as the Subject position in positional languages) that is sacrosanct. This allows Sanskrit to be a pro-drop language as well. Further, even the mandatory arguments such as *kartā* and *karma* may be dropped. For example, in an answer to a question '*rāmaḥ kutra agacchat*' (Where did Rama go?), a simple answer such as '*vanaṃ agacchat*' (went to a forest) is possible where the subject is ellipsed. Here the word *vanaṃ* is ambiguous between a nominative and an accusative analysis with the same stem *vana*. This leads to two parses, one with *vana* as an agent and another with *vana* as a goal. In the absence of any module to deal with meaning congruity between the verb and a noun, the parser fails to select one parse out of the two. The human annotator however marks the correct parse since he knows the meanings of the words. However there are cases where even for a human being the sentence is ambiguous, due to multiple morphological analyses. For example the causative form of the verb *katha* (to tell) is same as its non-causative form. Thus the word *kathayanti* may mean either tell or make somebody tell. So a simple sentence such as

(7) *Skt:* *mitrāṇi* *kathayanti.*
Gloss: friend{pl,nom/acc} tell{pres,pl,3p,[causative]}
Eng: Friends tell / (They) tell friends / Friends make (somebody) tell / (They) make (somebody) tell friends.

is ambiguous between four readings - friends is an agent, friends is a *karma*, friends is the causative agent, and finally friends is the *karma* (object) of the causative verb. This ambiguity is there for a human reader as well, since all the three interpretations are meaningwise compatible. In such cases the annotators are advised to mark all possible readings.

We present the last example where the arguments are shared. Consider an example with one verb in absolutive and the other one in finite form as follows.

(8) *Skt:* *rāmaḥ* *pustakaṃ* *krītvā* *paṭhati.*
Gloss: Rama{nom} book{sg,acc} purchase{abs} read{pres,sg,3p}.
Eng: Rama reads a book after purchasing it.

Here both the *kartā* as well as *karma* viz. Rāma and book are shared between the two verbs purchase and read. Pāṇini has provided a rule for the sharing of the *kartā*, and accordingly, we relate Rāma by the relation of *kartā* with the finite verb read. But, for the sharing of the *karma*, there is no rule in the grammar. Here we fall back to the default word order in prose for deciding which role to mark. If the verb in absolutive were intransitive, then the *karma* would have been always after this absolutive verb and before the final verb, in the default prose word order. Similarly, if the *karma* for both the verbs are different, then the *karma* for the finite verb would be just before it, and that of the one in absolutive would be before it. Taking clues from this, we mark the shared verb as an argument of the verb in absolutive, and then using the rule for sharing of arguments, we share it with the final verb. But if an annotator marks the relation the otherway, we do not want to penalise them. In other words, we provide both possible answers in such cases.

4.2 Evaluation

The sentences in the Saṁsādhanī treebank were run through the Saṁsādhanī parser. Table 1 shows the statistics of the treebank and the performance of the parser on the basis of following parameters: a) exact match, b) totally failed sentences, c) partially correct output, d) Labelled Attachment Score (LAS), and e) Unlabelled Attachment Score (UAS). Totally failed sentences are the ones which the parser fails to parse, either due to Out of Vocabulary words or if any word fails to get connected to any other word in the sentence. Partially correct output are the parses where at least one relation is wrong but not all.

Source	Sentences	Tokens	Exact Match	Failed	Partial Match	LAS	UAS
Grammar	468	1551	343	2 (.4%)	123	89%	97%
9^{th} grade	284	1393	183	15 (.6%)	87	82%	89%
Skt Learner	1070	4987	817	66 (6%)	181	88%	92%
BhG sample(verse)	36	313	7	3 (8%)	26	70%	76%
Average	1858	8244	1350	86 (4.6%)	417	85.5%	91.5%

Table 1: Performance of Parser

Thus we see that the performance of this parser is reasonably good. The percentage of failure is very small. The average LAS is 85.5% and the UAS is 91.5%. We notice that the performance of verse is not good. This is mainly due to some relations such as that of genitive and the adjectival which can move around freely.

The confusion matrix for some of the frequently occuring relations is shown in Table 2. The maximum confusion is with respect to the relation of *kartā* (roughly agent). There are two major reasons for the confusion of any relation with the other one. The first reason is, the relations share the same case marker. For example, both the cause and the instrument always take the instrumental case marker. And in the passive voice, *kartā* also takes the instrumental case marker. Therefore we see the confusion betwen a cause and an instrument and the *kartā*. Similarly the adjective of any of the predicate-argument relation always takes the case of its head noun. Since the relative word order for the adjective and the head noun is not fixed, in the absence of any semantic information about the adjective there is a confusion between which of the two substantives is the head and which one is an adjective. The confusion between a *kartā*

and the predicative adjective is also essentially for the same reason. The second reason for the confusion is due to multiple morphological analyses of a word. For example, in the neuter gender, the accusative and nominative word forms are the same. This results in the confusion between a *kartā* and a *karma* (roughly goal).

machine→ manual↓	kartā (agent)	karma (goal)	adjective	pred adj	instrument	cause	..	Total
kartā (agent)	**1322**	14	10	26	6	6	..	1523
karma (goal)	31	**883**	7				..	1069
adjective	29	12	**260**				..	406
pred adj	23			**114**			..	162
instrument	5				**74**	8	..	99
cause					10	**40**	..	77
..	..	..	..	..	..	..	..	..
Total	1460	952	306	140	99	66	..	6226

Table 2: Confusion Matrix

5 Conclusion

In this paper we have discussed the first publicly available Sanskrit treebank developed following the dependency relations based on the Indian grammatical tradition. The presence of derivational analysis leads to deeper semantic analysis. At the same time it also introduces inconsistency in tagging, since most of the time for frequently used derived words such as *vacanam* (speech), the annotator may take these as underived and provide the dependency relations which do not show up the deeper analysis. Such deeper analysis is useful for certain tasks such as question answering and information retrieval, though might be irrelevant for the machine translation purpose.

We have also discussed the improved version of the dependency relations based on the Indian grammatical tradition. Three major improvements related to the treatment of the complementiser, conjunct and co-relative constructions were discussed. The modified version reflects the associated semantics.

Finally we have tested the dependency parser for Sanskrit on the treebank, and noted that the performance of the parser is reasonably good. The confusion matrix conforms with the grammatical sources of ambiguities. The proper modeling of mutual congruency would help in improving the performance of the parser.

References

V. S. Apte. 1885 [1925]. *The Student's Guide to Sanskrit Composition*. The Standard Publishing Company, Girgaon, Bombay, 9 edition.

David Bamman and Gregory Crane. 2011. The ancient greek and latin dependency treebanks. In Caroline Sporleder, Antal van den Bosch, and Kalliopi Zervanou, editors, *Language Technology for Cultural Heritage*, pages 79–98, Berlin, Heidelberg. Springer Berlin Heidelberg.

E Bejček, E Hajičovà, J. Jajič, P. Jinovà, V. Kettnerovà, V. Kolářová, M. Mikulová, J. Mirovskỳ, , A. Nedoluzhko, J. Panevová, L. Poláková, M. Ševčková, J. Štěpànek, and S. Zikánová. 2013. Prague ependency treebank 3.0. Technical report, Institute of Formal and Applied Linguistics, Charles University.

Akshar Bharati and Amba Kulkarni. 2010. Information coding in a language: Some insights from paninian grammar. *Dhīmahi, Journal of Chinmaya International Foundation Shodha Sansthan*, I(1):77–91.

Akshar Bharati and Amba Kulkarni. 2011. 'Subject' in English is abhihita. In Ashok Aklujkar George Cardona and Hideyo Ogawa, editors, *Studies in Sanskrit Grammars (Proceedings of the Vyakarana Section of the 14th World Sanskrit Conference)*. D.K. Printworld.

Akshar Bharati and Rajeev Sangal. 1990. A karaka based approach to parsing of indian languages. In *Proceedings of International Conference on Computational Linguistics (Vol. 3)*, Helsinki, Association for Computational Linguistics NY.

Akshar Bharati and Rajeev Sangal. 1993. Parsing free word order languages in the paninian framework. In *31st Annual Meeting of the Association for Computational Linguistics*, pages 105–111. acl.

Akshar Bharati, Vineet Chaitanya, and Rajeev Sangal. 1995a. *Natural Language Processing: A Paninian Perspective*. Prentice-Hall, New Delhi.

Akshar Bharati, Ashok Gupta, and Rajeev Sangal. 1995b. Parsing paninian grammar with nesting constraints. In *Proceedings of 3rd NLP Pacific Rim Symposium*, pages 1–6.

Akshar Bharati, Rajeev Sangal, Vineet Chaitanya, Amba Kulkarni, Dipti Misra Sharma, and K. V. Ramakrishnamacharyulu. 2002. AnnCorra: Building tree-banks in Indian languages. In *COLING-02: The 3rd Workshop on Asian Language Resources and International Standardization*.

Pushpak Bhattacharyya. 1986. A system for sanskrit to hindi translation. Master's thesis, IIT Kanpur.

A. Böhmovà, E. Hajičovà, and B. Hladkà. 2003. The prague dependency treebank. 20.

George Cardona. 2007. *Pāṇini and Pāṇinīyas on śeṣa Relations*. Kunjunni Raja Academy of Indological Research Kochi.

George Cardona. 2009. On the structure of Pāṇini's system. In Gérard Huet, Amba Kulkarni, and Peter Scharf, editors, *Sanskrit Computational Linguistics 1 & 2*. Springer-Verlag LNAI 5402.

John Caroll, Guido Minnen, and Ted Briscoe. 1999. Corpus annotation for parser evaluation. In *Proceedings of EACL*.

Harold G Coward. 1983. *Studies in Indian Thought*. Motilal Banarasidas.

Sleator Daniel and Davy Temperley. 1993. Parsing english with a link grammar. In *Third International Worshop on Parsing Technologies*.

Puneet Dwivedi and Easha Guha. 2017. Universal dependencies of sanskrit. *International Journal of Advance Research, Ideas and Innovations in Technology*, 3:479–482.

C J Fillmore, C R Johnson, and M R L Petruck. 2003. Background to framenet. *Intenational journal of Lexicography*, 16(3):235–250.

Brendan S. Gillon. 2002. Bhartṛhari's rule for unexpressed kārakas: The problem of control in classical sanskrit. *Indian Linguistic Studies, Festschrift in Honor of George Cardona*.

Pawan Goyal, Vipul Arora, and Laxmidhar Behera. 2009. Analysis of Sanskrit text: Parsing and semantic relations. In Gérard Huet, Amba Kulkarni, and Peter Scharf, editors, *Sanskrit Computational Linguistics 1 & 2*, pages 200–218. Springer-Verlag LNAI 5402.

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

Oliver Hellwig, Salvatore Scarlata, Elia Ackermann, and Paul Widmer. 2020. The treebank of vedic sanskrit. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 5137–5146. European Language Resources Association.

Oliver Hellwig. 2009. Extracting dependency trees from sanskrit texts. In Amba Kulkarni and Gérard Huet, editors, *Third International Sanskrit Computational Linguistics Symposium*, pages 106–115. Springer-Verlag LNAI 5406.

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

Gérard Huet. 2009. Formal structure of Sanskrit text: Requirements analysis for a mechanical Sanskrit processor. In Gérard Huet, Amba Kulkarni, and Peter Scharf, editors, *Sanskrit Computational Linguistics 1 & 2*. Springer-Verlag LNAI 5402.

Madhusoodan Pai J. 2020. *Sanskrit Sentence Generator: A Prototype*. Ph.D. thesis, University of Hyderabad, Hyderabad.

J J Katz and J A Fodor. 1963. The structure of a semantic theory. *Language*, 39:170–210.

Tracy H. King, Richard Crouch, Stefan Riezler, Mary Dalrymple, and Ronald Kaplan. 2003. The PARC 700 dependency bank. In *Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora (LINC-03)*.

Paul Kiparsky. 2009. On the architecture of panini's grammar. In Gérard Huet, Amba Kulkarni, and Peter Scharf, editors, *Sanskrit Computational Linguistics 1 & 2*, pages 33–94. Springer-Verlag LNAI 5402.

G.-J. M. Kruijff. 2002. Formal and computational aspects of dependency grammar: History and development of dpendency grammar. Technical report.

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 Dipti Sharma. 2019. Pāṇinian syntactico-semantic relation labels. In *Proceedings of the Fifth International Conference on Dependency Linguistics (Depling, SyntaxFest 2019)*, pages 198–208, Paris, France, August. Association for Computational Linguistics.

Amba Kulkarni, Sheetal Pokar, and Devanand Shukl. 2010. Designing a Constraint Based Parser for Sanskrit. In G N Jha, editor, *Fourth International Sanskrit Computational Linguistics Symposium*, pages 70–90. Springer-Verlag, LNAI 6465.

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, Prague, Czech Republic, August. Charles University in Prague Matfyzpress Prague Czech Republic.

Amba Kulkarni. 2016. Application of modern technology to south asian languages. In Hans Henrich Hock and Elena Bashir, editors, *The Languages And Linguistics of South Asia: A comprehensive Guide*, pages 744–747. De Gruyter.

Amba Kulkarni. 2019. *Sanskrit parsing based on the theories of śābdabodha*. Indian Institute of Advanced Study, Shimla and D K Publishers (P) Ltd.

Amba Kulkarni. 2020a. Appropriate dependency tagset for sanskrit analysis and generation. *Acta Orientalia*, forthcoming.

Amba Kulkarni. 2020b. Sanskrit parsing following the indian theories of verbal cognition. *TALLIP*, forthcoming.

B. Levin and M Hovav Rappaport. 2005. *Argument realization*. Cambridge University Press.

Dekang Lin. 2003. Dependency-based evaluation of MINIPAR. In Abeillé Anne, editor, *Treebanks: Building and Using Parsed Corpora*, pages 317–329, Dordrecht. Springer Netherlands.

Marie-Catherine de Marneffe and Christopher D Manning. 2008. Stanford dependencies manual.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*.

Ignor Mel'čuk. 1988. *Dependency Syntax: Theory and Practice*. The SUNY Press, Albany, New York.

J. Nivre, M.-C de Marneffe, F. Ginter, Y. Goldberg, J. Hajič, C. D. Manning, R. T. McDonald, S. Petrov, S. Pyysalo, N. Silveira, R. Tsarfaty, and D. Zeman. 2016. Universal dependencies c1: A multilingual treebank collection.

M. Palmer, D. Gildea, and N. Xue. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Sanjeev Panchal and Amba Kulkarni. 2018. Yogyatā as an absence of non-congruity. In Gérard Huet and Amba Kulkarni, editors, *Computational Sanskrit and Digital Humanities*. D K Publishers.

Sanjeev Panchal and Amba Kulkarni. 2019. Co-ordination in Sanskrit. *Indian Linguistics*, 80(1-2):59–176.

Sanjeev Panchal. 2020. *Modelling Ākāṅkṣā following Pāṇinian Grammar for Sanskrit sentential parsing*. Ph.D. thesis, University of Hyderabad, Hyderabad.

Gopal Dutt Pande. 2004. *Aṣṭādhyāyī of Pāṇini elaborated by M.M.Panditraj Dr. Gopal Shastri*. Chowkhamba Sur bharati Prakashan Varanasi.

Preeti Patel. 2018. *E-teaching capsule for Śrīmadbhagvadgītā*. Ph.D. thesis, University of Hyderabad, Hyderabad.

K V Ramakrishnamacaryulu. 2009. Annotating Sanskrit texts based on Śābdabodha systems. In Amba Kulkarni and Gérard Huet, editors, *Proceedings Third International Sanskrit Computational Linguistics Symposium*, pages 26–39, Hyderabad India. Springer-Verlag LNAI 5406.

N S Ramanujatatacharya. 2005. *Śābdabodha Mīmāṁsā*. Institute Francis De Pondicherry.

Bhā. Va. Rāmapriya and V. Saumyanārāyaṇa. 2001. Saṅgaṇakayantre nyāyaśāstrīyaśābdabodhaḥ. *Journal of Foundation Research*, VI(1–2):61–68.

Phillip Resnik. 1993. Semantic classes and syntactic ambiguity. In *ARRPA Workshop on Human Language Technology*. Princeton.

Preeti Shukla, Amba Kulkarni, and Devanand Shukl. 2013. Geeta: Gold standard annotated data, analysis and its applicationa. In *Proceedings of ICON 2013, the 10th International Conference on NLP*, Noida, India, December.

J. S. Speijer. 1886; Reprint 2009. *Sanskrit Syntax*. Motilal Banarsidass New Delhi.

Veluri Subbarao. 1969. *The philosophy of a sentence and its parts*. Munshiram Manoharlal, Delhi.

Lucien Tesnière, editor. 1959. *Éléments de Syntaxe Structurale*. Klincksieck Paris.

Gary A Tubb and Emery R Boose. 2007. *Scholastic Sanskrit: A Handbook for students*. The American Institute of Buddhist Studies at Columbia University in the City of New York New York.

A Saṁsādhanī Dependency Relations

- **Predicate argument relations**
  - *kartā* (agent)
    - *prayojaka-kartā* (causative agent)
    - *prayojya-kartā* (causee)
  - *karma* (goal/patient)
    - *mukhya-karma* (direct object)
    - *gauṇa-karma* (indirect object)
    - *vākya-karma* (sentential argument)
  - *karaṇam* (instrument)
  - *sampradānam* (recipient)
  - *apādānam* (source)
  - *adhikaraṇam* (location)
    - *kāla-adhikaraṇam* (location of time)
    - *deśa-adhikaraṇam* (location of space)
    - *viṣaya-adhikaraṇam* (locus indicating the subject)
    - *lyapkarma-adhikaraṇam* (*karma* of an ellipsed absolutive verb form marked as a location)
- **Non Predicate argument relations**
  - **Verb-Verb relations**
    - *pūrva-kālaḥ* (precedence)
    - *vartamāna-samāna-kālaḥ* (simultaneity in present)
    - *bhaviṣyat-samāna-kālaḥ* (simultaneity in future) tense
    - *bhāvalakṣaṇa-pūrva-kālaḥ* (simultaneity in the past without sharing of arguments)
    - *bhāvalakṣaṇa-vartamāna-samāna-kālaḥ* (simultaneity in present without sharing of arguments)
    - *bhāvalakṣaṇa-anantara-kālaḥ* (simultaneity in future without sharing of arguments)
    - *sahāyaka-kriyā* (auxiliary verb)
  - **Verb-noun relations**
    - *sambodhyaḥ* (vocative)
    - *hetuḥ* (cause)
    - *prayojanam* (purpose)
    - *kartṛ-samāna-adhikaraṇam* (predicative adjective)
    - *karma-samānādhikaraṇam*
    - *kriyā-viśeṣaṇam*(manner adverb)
    - *atyanta-saṃyogaḥ*(total contact)
    - *apavarga-sambandhaḥ*
    - *pratiṣedhaḥ* (negation)
  - **Noun-Noun relations**
    - *śaṣṭhī-sambandhaḥ* (genitive)
    - *aṅga-vikāraḥ* (body-deformity)
    - *vīpsā* (reduplication)
    - *viśeṣaṇam* (adjective)
    - *sambodhana-sūcakam* (vocative marker)
    - *abhedaḥ* (indifference)
    - *nirdhāraṇam* (determiner)
    - *vākya-karma-dyotakaḥ* (complementiser)
    - *tīvratādarśī* (intensifier)
    - *nāma* (name)
- **Relations due to special words**
  - *sandarbha-binduḥ* (reference point)

- *tulanābinduḥ* (comparison point)
- *udgāravācakaḥ* (exclamatory)
- *saha-arthaḥ* (association)
- *vinā-arthaḥ* (disassociation)

- **Miscelleneous**
  - *anuyogī* (relata1)
  - *pratiyogī* (relata2)
  - *nitya-sambandhaḥ* (co-reference)
  - *sambandhaḥ* (relation)
- **Conjunct-disjunct**
  - *samuccitaṁ* (conjunct)
  - *samuccaya-dyotakaḥ* (conjunction)
  - *anyataraḥ* (disjunct)
  - *anyatara-dyotakaḥ* (disjunction)

Note: The bold entries are the headings and do not indicate relation labels.
We have not provided the gist/translation of these relation tags. The readers are encouraged to refer to the tagging guidelines available at `http://sanskrit.uohyd.ac.in/scl/GOLD_DATA/Tagging_Guidelines/guidelines.html`.