Later Nyāya Logic: Computational Aspects

Amba Kulkarni

Contents

Abstract

In this article we describe the computational aspects of the Technical Language of Navya-Nyāya. Navya-Nyaya is an off-shoot of the early Nyāya philosophy. It deviates from the Nyāya philosophy in three major ways. First, instead of prameyas the discussions are centered around the pramāṇas. Second, the Navya-Nyāya has adapted the Vaiseṣika ontology. And finally it has introduced a few concepts expressed through an unambiguous technical terminology that brings in a clarity in the communication removing the inherent ambiguities of a natural language. In this article the syntax of expressions involving this technical terminology is described, followed by a scheme based on Conceptual Graphs of Sowa for their graphical rendering. Finally a computational algorithm is described that renders the graphs corresponding to the Navya-Nyāya expressions semi-automatically.

A. Kulkarni (✉)
Department of Sanskrit Studies, University of Hyderabad, Hyderabad, Telangana, India

S. Sarukkai (ed.), *Handbook of Logical Thought in India*,
https://doi.org/10.1007/978-81-322-1812-8_12-1

Introduction

The words in any language are countably infinite. The concepts they correspond to, however, form a continuum. Naturally the words – both content words and function words – are prone to be over-loaded. This leads to ambiguity in a Natural language. In spite of having ambiguity, human beings do not find much difficulty in language communication. This is mainly because of the shared background knowledge. However, the scientific work and philosophical discussion demand precision. Indian logicians who were engaged in philosophical debates realized the need for expressing communications in an unambiguous way. Their efforts culminated in a new school *Navya-Nyāya* "Neo-Logic" which emerged as an off-shoot of the Nyāya school of philosophy. Seeds of *Navya-Nyāya* (NN) School are found in Udayana's work (eleventh century). Gaṅgeśa (twelfth century) in *Tattvacintāmaṇi* introduced a few technical terms and provided a well-defined scheme to express a cognitive structure using these technical terms. Later on Raghunātha (sixteenth century), Jagadīśa (late sixteenth century) and Gadādhara (seventeenth century) made significant contributions to this field enriching it further.

The philosophers and the logicians in the West also needed precision. This need gave rise to the development of formal languages. These formal languages have a very limited vocabulary with precise meaning. Propositional calculus, for example, uses only three words viz. "and," "or," and "not." Later the vocabulary was enhanced by introducing universal and existential quantifiers, the modal logic extended these further to include modal operators and Montague introduced generalized quantifiers. But in spite of all these developments, we hardly see any use of these formal languages in the field of Humanities and Social sciences. On the other hand the technical language of Navya Nyāya is built on top of the classical Sanskrit. In this language, technical words of Navya-Nyāya are intermixed with the original words in such a way that the resulting expression is unambiguous. Thus, a Navya-Nyāya expression has the complete power of expressibility of a Natural language at its disposal. And probably, it is because of this reason that we find use of technical language of NN in various fields of Humanities such as Vyākaraṇa "grammar," Sāhitya "literature," Mīmāṃsā "exegesis," etc. The main purpose behind the development of such a language was to describe the cognitions either perceived through sense organs or through the verbal utterances, and to deduce inferences from the valid cognitions.

Bhattacharya (1990, 130) rightly observes.

"Thus this language could be used in every sphere where cognition, belief, doubt, and other epistemic and doxastic factors play an essential role. This explains why this language could be used universally in the humanities, where the epistemic factors predominate."

In addition to describing the cognitive structures, because of the precision it provides, this language was also used to provide the definitions (lakṣaṇa) of scientific terms in various disciplines. Heavy usage of these technical terms to define various grammatical concepts and to bring in precision in the communication by the sixteenth-century grammarians such as Bhaṭṭoji Dikṣita and later by Kauṇḍabhaṭṭa

and Nāgeśa distinguishes modern grammatical texts (Navya-vyākaraṇa) from the old ones. Good understanding of this technical language of NN is thus necessary to understand the texts of Indian origin in various disciplines.

Another important reason for understanding the technical language of NN arises because of its relevance in the age of information technology. With the advent of computers and the age of information revolution, in the West there was an upsurge in the field of Knowledge Representation (KR). Several schemes for knowledge representation were designed and proposed. Woods (1975) defines the properties of a Knowledge Representation language as

A KR language must unambiguously represent any interpretation of a sentence (logical adequacy), have a method for translating from natural language to that representation, and must be usable for reasoning.

The technical language of NN is designed precisely for representing the cognition – both perceptual and the verbal one. The well-defined description of the cognition gives a scheme to translate from natural language to the technical language of NN, and it is also used very extensively, in the Nyāya school of Indian logic, for reasoning. Briggs (1985) points out, with examples from the śābdabodha (verbal cognition), how the NN expressions are close to a typical KR scheme such as the Semantic Net.

Two major problems we envisage in understanding the technical language of NN. The first problem is due to the mode of presentation. Traditional learning in India was through oral communication. Sanskrit is very rich in compound formation. This feature of Sanskrit has been utilized to its full extent by the Indian logicians in describing cognitive structures using the technical language of NN. Such expressions are typically exceptionally long, many-a-times one compound running into pages. While the oral transmission of knowledge and all serious debates could sustain these long compounds, modern scholars not trained in oral tradition found it difficult to understand these long expressions. As a result, since as early as twentieth century we find use of diagrams to represent NN expressions. The second problem is related to the syntax and semantics of the technical language of Navya Nyāya. While Ingalls (1951), Matilal (1977), Shaw (1980), and Mohanty (2000), to name a few, worked towards understanding the concepts and comparing them with the western logic counterparts, Bhattacharya (1990) and Ganeri (2008) explored the underlying syntactic structure and the grammar as well.

In what follows, we first describe the syntax of the NN expressions and then give an overview of pictorial representation of NN. Finally we describe how this knowledge of syntax of NN expressions and the computational tools for analyzing Sanskrit texts are put together to mechanically render the NN expressions graphically.

Syntax of Navya Nyāya Expressions

An NN expression involves a small number of technical terms together with a nonlogical vocabulary (Matilal 1968). Ganeri (2008) in the informal description of the NN classifies these technical terms into six categories.

1. **Primitive Terms**
   Nouns such as *ghaṭa* "pot," *bhūtala* "ground," *gandha* "smell," etc., are the primitive terms.
2. **Abstract Functor**
   A derivational suffix "tva" or "tā" (-ness or -hood) that maps a noun to an abstract noun is termed as an abstract functor.
3. **Relational Abstract Expressions**
   Relational abstract expressions are derived from relation-denoting terms by adding a "tva" or "tā" (-ness or -hood) suffix. For example, *pitṛ* "father" is a relation-denoting term. By adding "tva" suffix, it changes to *pitṛtva* "father-hood," a relational abstract expression.
4. **Conditioning Operator**
   The conditioning operator *nirūpita* "determined by" operates on a relational abstract expression to form a term. For example, *X-nirūpita-pitṛtva* "father-hood determined by X."
5. **Sentence-Forming Operator**
   Terms such as *niṣṭha* "resident in" and *avacchinna* "delimited by" combine a relational term with another term to form a sentence.
6. **Negation Functor**
   The term *abhāvaḥ* "negation/absence" is termed as a negation factor.

Bhattacharya (1990) describes the syntactic structure of a cognition expressed by an NN expression.

The canonical form for expressing the cognition in NN is either

$$b \text{ is } a - \text{possessing}, \text{or } b \text{ has } a.$$

Thus, the canonical form for "A cat is on a mat" is "A mat has a cat (on it)" or "A mat is cat-possessing." The stock example in NN is the expression describing the reality "a pot is on the ground," expressed in NN canonical form as either

$$ghaṭavat\, bhūtalam, \text{or } bhūtale\, ghaṭaḥ.$$

A cognition called "qualificative cognition" *savikalpaka jñāna*, according to a Naiyāyika, is of the form *aRb*, where *a* is the "qualifier" *prakāra*, *b* is the "qualificandum" *viśeṣya*, and *R* is the "qualification" *saṁsarga* of the cognition. In case of the cognition arising from *b is a-possessing*, *b* is the qualificandum and *a* is the qualifier. But in case of the cognition of the form *b has a*, *a* is the qualificandum and *b* is the qualifier. Accordingly the description of the cognition differs in both these cases. This structure *aRb* is a complex object, according to a Naiyāyika, where ontologically *a* is considered to be a property viz. super-stratum (*ādheya*) of *b* in the relation *R* and *b* is the property-possessor (substratum or locus) *ādhāra* of *a* in the

relation R. The two relata a and b of the relation R are called the *pratiyogin* and *anuyogin*, respectively.

This cognition has the following structure:

$$\begin{aligned} &R - niṣṭha - saṁsargatā - nirūpita - \\ &a - niṣṭha - R1 - ness - nirūpita - \\ &R_2 - ness - vat - b. \end{aligned} \tag{1}$$

or more elaborately as:

$$\begin{aligned} &R - avacchinna - \\ &a - ness - avacchinna - a - niṣṭha - R_1 - ness - nirūpita - \\ &b - ness - avacchinna - R_2 - ness - vat - b. \end{aligned} \tag{2}$$

And the cognition is described as

$$\begin{aligned} &R - niṣṭha - sa\ \dot{m}sargatā - nirūpita - \\ &a - niṣṭha - R_1 - ness - nirūpita \\ &- b - niṣṭha - R_2 - ness - śāli - jñānam. \end{aligned} \tag{3}$$

Similarly, in Eq. 2 R_2-*ness-vat-b* is replaced by R_2-*ness-śāli-jñānam* when one refers to the cognition. This structure uses three relational terms viz. *avacchinna*, *niṣṭha*, and *nirūpita*, the -ness suffix (which in Sanskrit is represented by -tva), and three relational abstracts viz. R_1-ness and R_2-ness, and saṁsargatā.

For example, the knowledge *nīlo ghaṭaḥ* (the pot is blue) has *nīla* (blue) as the qualifier (prakāra), *ghaṭaḥ* (pot) as the qualificandum (viśeṣya), and the relation of contact *samavāya* (inherence) between them. The structure of this cognition has the form:

$$\begin{aligned} &samavāya - sambandha - avacchinna - \\ &nīla - tva - avacchinna - nīlarūpa - niṣṭha - prakāratā - nirūpita - \\ &ghaṭa - tva - avacchinna - ghaṭa - niṣṭha - viśesyatā - śāli - jñānam \end{aligned} \tag{4}$$

The same structure is used to describe the physical reality as well. For example, the verbal cognition arising from the phrase *ghaṭavad bhūtalam* "the ground with a pot" is described unambiguously as

$$\begin{aligned} &saṁyoga - sambandha - avacchinna - \\ &ghaṭa - tva - avacchinna - ghaṭa - niṣṭha - ādheyatā - nirūpita - \\ &bhūtala - tva - avacchinna - adhikaraṇatā - vat - bhūtalam. \end{aligned} \tag{5}$$

Graphical Representation

Ingalls (1951), Staal (1988), Bhattacharya (1990), and many others have tried to use the Western symbolic notation to represent the NN expressions. But even to translate the NN expressions symbolically, one needs to understand these expressions fully. If the understanding goes wrong, the symbolic translation will be erroneous. Further, when one translates one formal language into another, one has to ensure that the two languages are equivalent in their expressive capability and that there is commensurability between the two. Use of diagrams solved this problem to a large extent. While representing the long linear compounds as a two-dimensional figure, the compounds were broken into its components, and the connection between various components was shown. This made it easy for a student to understand the NN Expressions. Of course, while showing the connection between components, the user had to take the context into account to rule out the various possibilities. Diagrammatic representation also does not pose any commensurability problem, since what is being represented is just a "parse" of the linear string, and there is no attempt to "translate" the original expression into another formal language. We notice the use of diagrams to represent NN expressions since the beginning of twentieth century. Patil (2014) in *Vidyādharī* mentions the use of diagrams by Vāmācaraṇabhaṭṭācārya in the early twentieth century. Though the need for graphical representation was felt, it was not formalized till recently. We find an extensive use of diagrams for understanding Sanskrit texts, especially the NN texts, in Japan. Wada (2007) reports that the first Japanese scholar to use diagrams for interpreting Sanskrit texts was Kitagawa and later Tachikawa, Miyasaka, and Wada (2007) enhanced these diagrams further. V. N. Jha (1987) further systematized them. Concepts, in these diagrams, are represented by rectangular boxes, and relations connecting these concepts are represented by edges. Technical terms *nirūpita*, *avacchedaka*, *niṣṭha*, are represented by different types of arrows. There are only minor differences among these various diagrammatic representations. The differences are more at the stylistic level such as style of the arrow heads (solid, hollow, curved, etc.) and the style of the edges (single, double, solid, dotted, etc.). Figure 1 represents the cognition *ghaṭavat bhūtalam* "the ground with a pot," expressed in brief without mentioning the avacchedakas (From (Jha 1987, 6). Here the plain edges denote the relation *niṣṭha* and the edges with an arrow represent the relation *nirūpita*.).

$$\begin{aligned} saṁyoga - niṣṭha - saṁsargatā - nirūpita- \\ ghaṭa - niṣṭha - prakāratā - nirūpita- \\ bhūtala - niṣṭha - viśesyata - śāli\ \ jñānam \end{aligned} \tag{6}$$

Conceptual Graphs for NN Expressions

Conceptual graph (CG) of Sowa (1985) is a KR scheme originally designed as a semantic representation for natural language. It provides a graphical representation

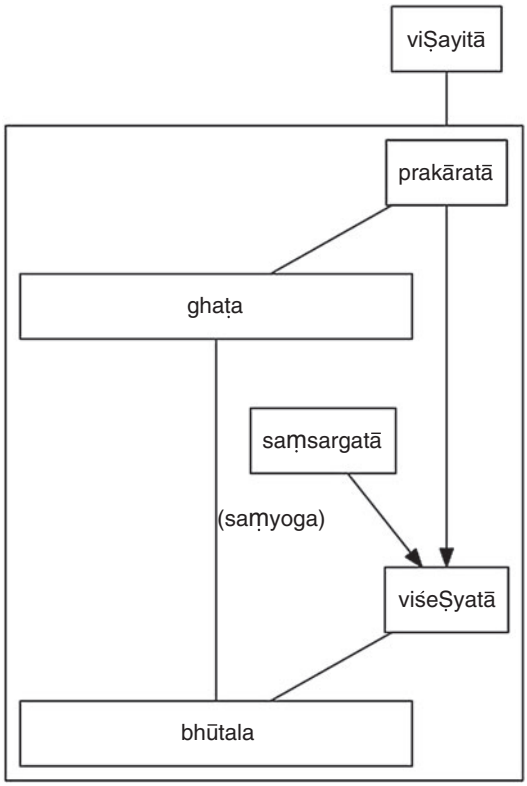

Fig. 1 Description of cognition *ghaṭavad bhūtalam*

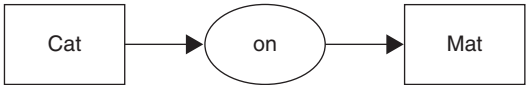

Fig. 2 Conceptual graph of "A cat is on a mat"

that is human readable and at the same time formal for computational purpose. It can represent both the epistemic structure and the ontological structure. Further the representation scheme of conceptual graph is so general that various graphical representation methods such as a parse trees, Petri net turn out to be special cases of the conceptual graph (Sowa 1992). In a CG, the concepts are related through the conceptual relations. Concepts are represented using boxes and relations using ovals. For instance, "A cat is on a mat" is represented in CG as in Fig. 2. Here "cat" and "mat" are the concepts and are represented using boxes and the relation "on" is represented using an oval.

Kulkarni (1994) proposed a scheme for representation of NN using Conceptual Graphs, in an effort towards establishing a bridge between the knowledge representation scheme of NN and the Western paradigms. Over a period of time, the representation scheme was modified further and, with the availability of

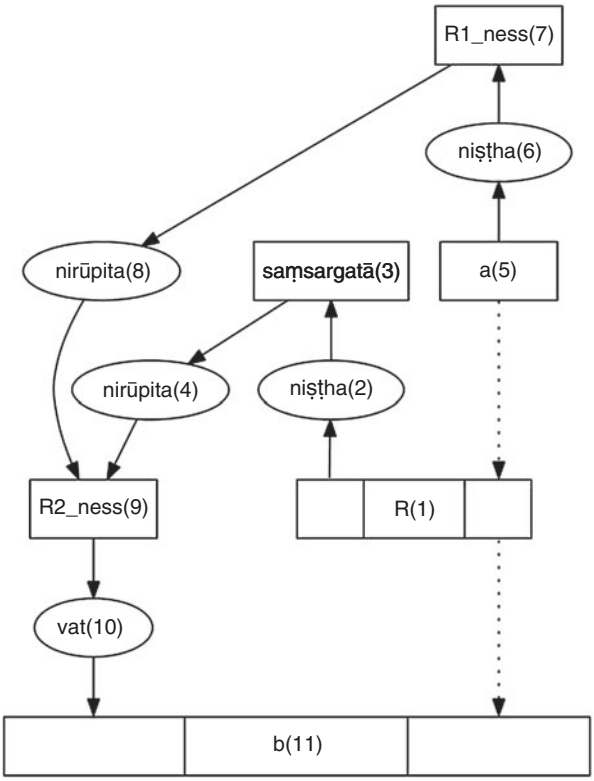

Fig. 3 NN expression as a CG

computational tools to parse a Sanskrit text, now efforts are on at University of Hyderabad to develop a computational tool to render the NNEs through Conceptual Graphs.

Graphical representations of the normal and elaborate descriptions of the cognition *aRb* described in Eqs. 1 and 2, according to this scheme, are shown in Figs. 3 and 4, respectively. The numbers in parenthesis indicate the position of the component in the NNE. Thus, if we navigate the graph sequentially, following the numbers, we get the corresponding NNE. The description of the perceptual cognition *ghaṭavad bhūtalam* "the ground with a pot," given by Eq. 6, is shown in Fig. 5, and the corresponding verbal cognition is described by Eq. 5 is shown in Fig. 6. Note that if we condense the graph by replacing relation nodes by edges with different styles, Fig. 5 reduces to Fig. 7 which is very close to Fig. 1. However, the basic difference is, in Fig. 7, the relation *saṁyoga* is shown as a concept node, while in Fig. 1 it is shown by an edge. Also, Fig. 1 encloses the complete cognition in a box with *viṣayitā* residing in this knowledge. No such box is shown in the CG representation.

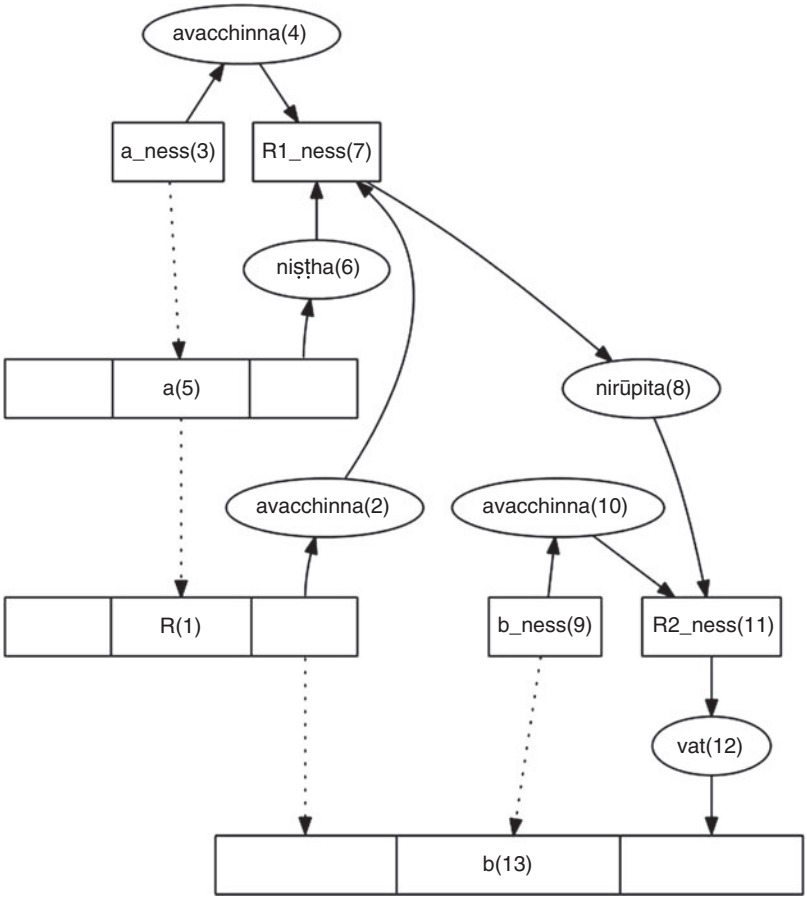

Fig. 4 Elaborate NN expression as a CG

Now we give an example from śābdabodha to illustrate the structure of a verbal cognition arising out of the following sentence.

Sanskrit: *Rāmaḥ hastena brāhmaṇāya dhanam dadāti*.

Gloss: Rama{nom.} hand{instr.} Brahmin{dat.} money{acc.} give{pres., active, 3sg}.

English: Rama gives money to a Brahmin with (his) hands.

The verbal cognition of this sentence according to the grammarian's school is

Sanskrit: *rāma-niṣṭha-kartṛtva-nirūpaka-hasta-niṣṭha-karaṇatva-nirūpaka-brāhmaṇa-niṣṭha-sampra-dānatva-nirūpaka-dhana-niṣṭha-karmatva-nirūpaka-dānakriyā.*

English: An activity of giving characterized by the agent-hood in Rama, the instrument-ness in the hand, the recipient-ness in a Brahmin, and the object-hood in money.

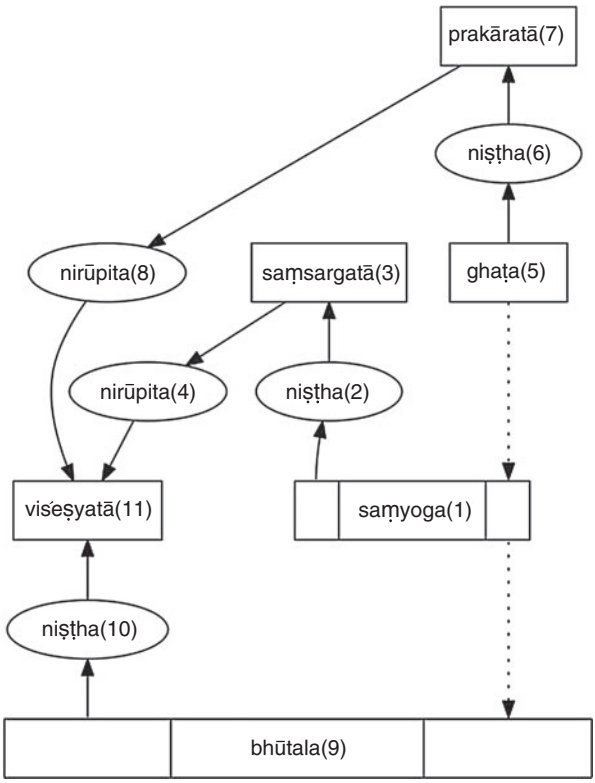

Fig. 5 A CG expressing the cognition

Figure 8 shows the rendering of this expression as a conceptual graph.

Finally let us look at the definition of *pṛthivī* "Earth" expressed as a NN expression. "Earth," according to the Indian school of ontology, is an object which has a characteristic property of having smell which differentiates it from other objects. This is precisely expressed by the NN expression

$$gandhatva - avacchinna - gandha - niṣṭha - \bar{a}dheyat\bar{a} - nir\bar{u}pita - adhikaraṇat\bar{a} - vat\bar{\imath}\ \ pṛthivi. \tag{7}$$

The conceptual graph corresponding to this structure is shown in Fig. 9. Dotted lines show the onto-logical reality viz. that smell-ness is the inherent property of the smell and that the earth has smell as its characteristic property. Solid lines show the connection between the concepts through the conceptual relations expressed in the NNE. Note the taddhita suffix *vat* (possessing) is represented as a relation relating the *adhikaraṇatā* (substratum-ness) with the *bhūtala* (substratum).

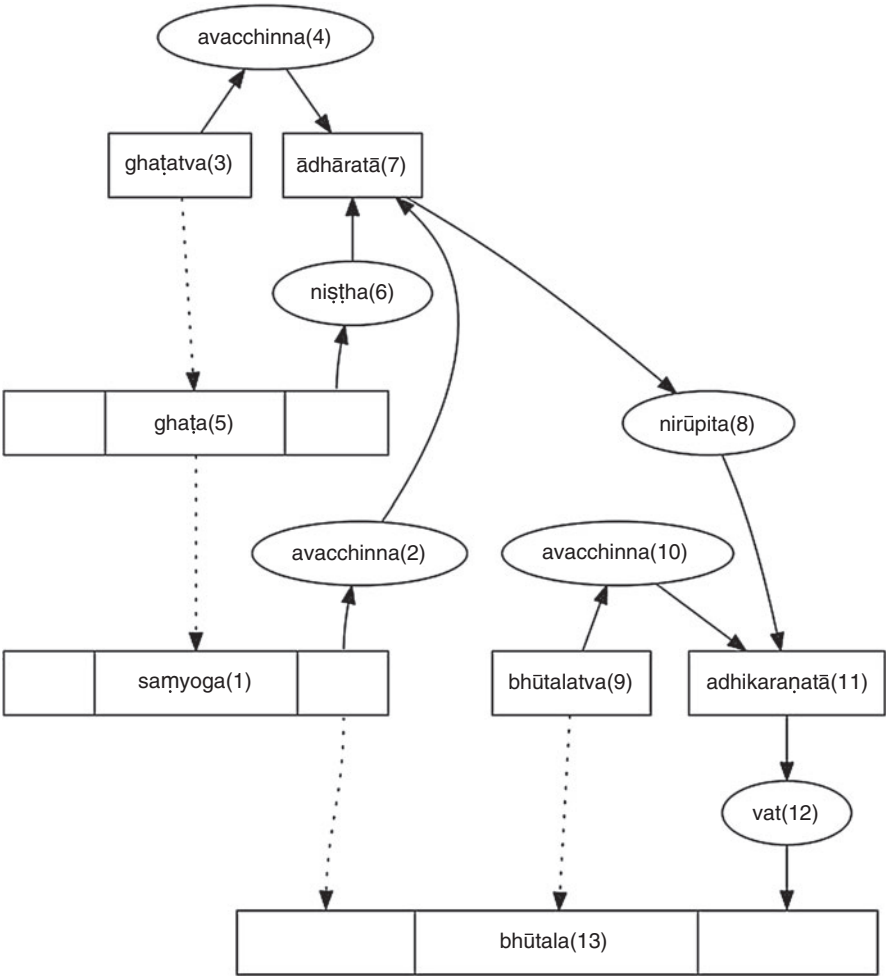

Fig. 6 A CG expressing the reality

Computational Parsing of an NN Expression

An NN expression is a compound. A compound, in Sanskrit, is written as a single word without any gap or hyphen in between the components. These components are joined together following euphonic changes. Compound formation also results in the loss of case markers. Euphonic changes as well as loss of case markers sometimes result in ambiguous compounds. Classical example of an ambiguous Sanskrit compound is *rāmeśvara*. Since the written Sanskrit forms do not mark any accent, this compound

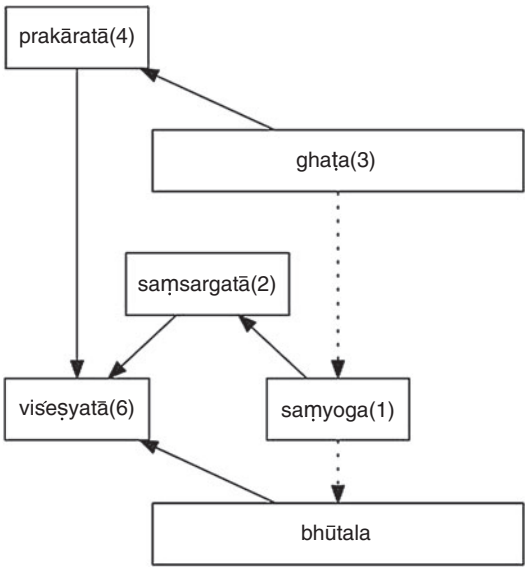

Fig. 7 Condensed Conceptual Graph

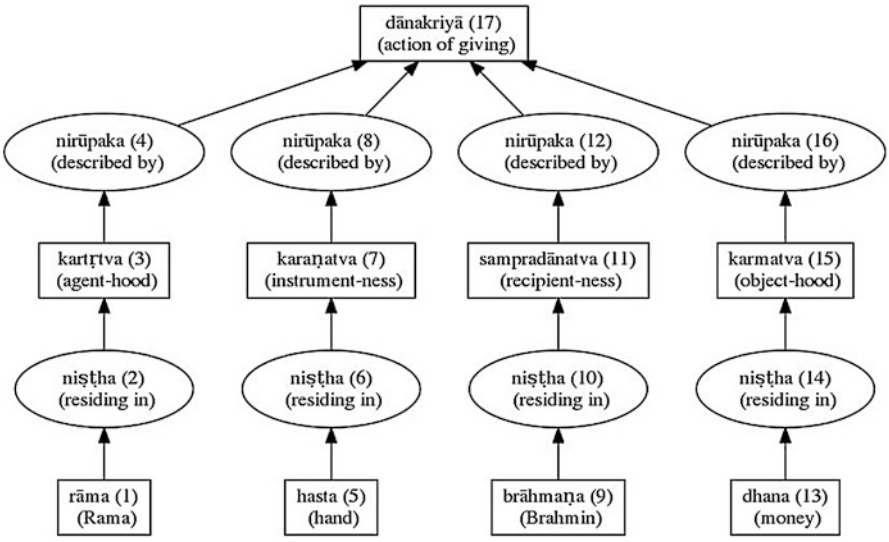

Fig. 8 conceptual graph representing the Śābdabodha

can be analyzed in three different ways: as a coordinating compound "karmadhāraya" (rāma *is* the īśvara), as an endocentric compound "tatpuruṣa" (the īṣvara of rāma), and as an exocentric compound "bahuvrīhi" (the one whose īṣvara is rāma).

Kumar et al. (2010) describe the steps involved in processing Sanskrit compounds and also discuss the associated computational complexity. The steps are

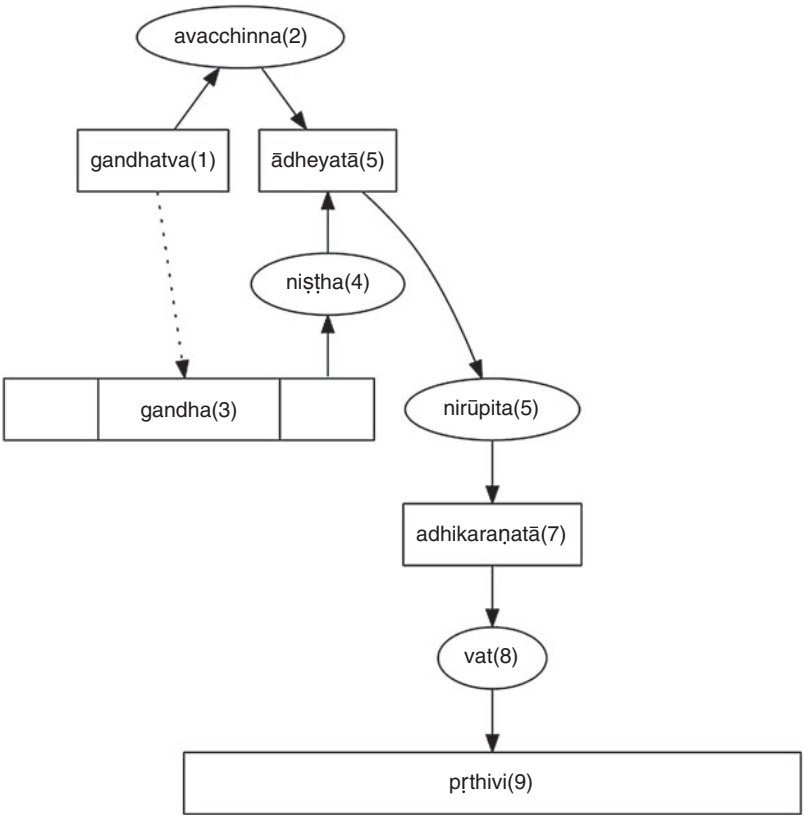

Fig. 9 conceptual graph defining pṛthivi

1. Splitting a compound into components
2. Analyzing its constituent structure
3. Identifying relations between its components
4. Providing a paraphrase for it

We illustrate these steps with an example. Consider an NNE which describes the cognition *ghaṭavat bhūtalam*:

$$\begin{aligned} &\textit{saṁyogasambandhāvacchinnaghaṭatvāvacchinnaghaṭaniṣṭh} \\ &\textit{ādheyatānirūpitādhikaraṇatāvatbhūtalam}. \end{aligned} \tag{8}$$

1. In order to understand this compound it is first split into its components as *saṁyoga-sambandha-avacchinna-ghaṭatva-avacchinna-ghaṭa-niṣṭha-ādheyatā-nirūpita-adhikaraṇatāvat-bhūtalam*

where the components are separated by hyphen.

2. Next a constituency parse of this compound is obtained.
   ((((((saṁyogasambandha-avacchinna)-((ghaṭatva-avacchinna)-((ghaṭa-niṣṭha)-ādheyatā)))-nirūpita)-adhikaraṇatā) ˆ vat)-bhūtalam)
   Note *vat*, a taddhita suffix is separated from its stem, since it is *adhikaraṇatā* (substratum-ness) which is *nirūpita* (delimited) by the *ādheyatā* (superstratum-ness), and not *adhikaraṇatāvat* (super stratum).
3. A graphical rendering of this expression is generated.
   In addition, for the benefit of Sanskrit scholars,
   - The relations between its components are identified.
     ((((((saṁyogasambandha-avacchinna)**T3**-((ghaṭatva-avacchinna)**T3**-((ghaṭa-niṣṭha)**T7**- ādheyatā)**K**)**K**)**K**-nirūpita)**T3**-adhikaraṇ atā)**K** ˆ vat)-bhūtalam)**K**
     where *K*, *T*3, *T*6, and *T*7 stand for *karmadhāraya*, tatpuruṣa with instrumental case suffix, *tatpuruṣa* with genitive case suffix and *tatpuruṣ a* with locative case suffix, respectively. These are all endo-centric compounds, with a requirement of nominative, instrumental, genitive, and locative case suffixes for paraphrasing.
   - And the paraphrase of this compound is provided.
     Sanskrit: *ghaṭatvena avacchinnā, ghaṭe niṣṭhā yā ādheyatā, tannirūpitā yā adhikaraṇatā, tad-vat bhūtalam*
     Gloss: By pot-ness delimited in pot residing which superstratum-ness determined by that which substratum-ness that possessing ground
     English: The ground which has substratum-ness which is determined by the superstratum-ness that is residing in the pot and is delimited by the pot-ness.

   In the following sections, we describe computational modules for (a) segmentation of a given compound into its components undoing the sandhi at the junctures, (b) a human interface for parsing such split compounds semi-automatically, and (c) a graphical renderer for such a parsed NN expression.

Segmenter for NN Expressions

Sanskrit is influenced by an oral tradition, and hence, in an utterance the word boundaries undergo euphonic changes "smoothing" the process of articulation of sounds that otherwise would have required more effort on the part of vocal organs. When a word w_1 is followed by a word w_2 the terminal phonemes in w_1 and the initial phonemes in w_2 undergo a "smoothing" process, which is termed as an operation of *sandhi*. The sandhi rules of Pāṇini are of the form

$$ABC \rightarrow ADC$$

where A and C provide the context under which phonemes B change to D. Thus, we get a continuous string of phonemes. In the process, information of word boundaries is lost. And this brings in nondeterminism during segmentation. For example, the

sequence of phonemes *sālakānanaśobhinī* (Varalakshmi 2013) can be split in two possible ways as

sāla-kānana-śobhīnī, and
sāla-alaka-ānana-śobhīnī

In order to segment a compound into its components, we need a sandhi splitter that splits a string into morphologically valid segments restoring the phonemes that underwent transformations during the sandhi process. Building a morphological analyzer for analyzing compound words is not trivial owing to the rich productive morphology (Kulkarni and Shukl 2009). Further each of the components of a compound can itself be a compound. Because of this recursive nature, it is simply impossible to build a dictionary of all possible compound stems. In the absence of such a lexicon, the only way to analyze a compound is to split it into possible components and analyze each of the components separately. But analysis of such components is further difficult owing to special operations they undergo during compound formation. Here are some such operations.

- **Deletion of the final "n" of a bare stem**: For example, during the compound formation, the final "n" of *rājan* gets deleted as in *rājapuruṣa*. So in order to split it as *rāja-puruṣa*, morphological analyzer should recognize *rāja* as a compounding form of the stem *rājan*.
- **Shortening or lengthening of a vowel**: In some cases, the final vowel of a bare stem is either shortened or lengthened. For example, *iṣṭakā* is changed to *iṣṭaka* in *iṣṭakacitam*.
- **Substitutes**: In some cases, the bare stems are substituted by their allomorphs. For example, *hṛdaya* is substituted by *hṛd* in *hṛllekhaḥ*.
- **Bound morphemes**: There are certain bound morphemes such as *kāra*, *ja*, etc., that occur only as a final component of a compound as in *kumbha-kāraḥ*, *paṅka-jam*, etc. The morphological analyzer should be designed to analyze such words only when they occur as a bound morpheme as a final component of a compound.
- **Change in gender**: In case of certain compounds such as exo-centric or avyayībhāva, the final component assumes an altogether different gender than its original one. The morphological analyzer needs to be equipped to analyze such words.
- **Change in number**: The word *anekān* when split will have *an-ekān* as its components. Now the morphological analyzer should analyze *ekān* as a plural of *-eka*, where *-eka* is the component of a compound.
- **Change in paradigm**: Words such as *kiñcana* when a part of a compound *akiñcana* do not inflect as *kiñcana* but need a separate paradigm.

Thus, in order to analyze the components of a compound, we need to equip our morphological analyzers to handle various phenomena described above. There are significant efforts in this area in the past by Huet (2006), Mittal (2010), Kumar et al. (2010), Natarajan and Charniak (2011), and Huet and Goyal (2013). All these efforts

were centered around general Sanskrit texts only. These segmenters needed some adaption to handle NN texts. One such adaptation is not to split the technical terms of Navya-Nyāya. While it is desirable to split the components and preverbs (*upasargas*) in compounds, in case of technical terms such as *avacchinna*, *nirūpita*, *samānādhikaraṇa*, *vyadhikaraṇa*, etc., it is not desirable to split these as *ava-cchinna*, *ni-rūpita*, *samāna-adhi-karaṇa*, and *vi-adhi-karaṇa*. In such cases, we would rather like to hide the derivation and see such words as a single unit. Arjuna and Huet and Arjuna and Kulkarni (2014) report the development of a segmenter for NN expressions with a special morphological analyzer for the Navya-Nyāya technical terms. Figure 10 shows the interactive user interface which facilitates the user to select the correct split among all possible splits.

Constituency Parser for NNE

The segmented expression needs further analysis to get the underlying constituency structure. For example, a compound with three components *a-b-c* may be analyzed in two different ways viz. (*a*-(*b*-*c*)) and ((*a*-*b*)-*c*). And of these two possibilities, typically for a given compound only one will be meaningful. For example, *eka-priya-darśanaḥ* can be analyzed only as (*eka*-(*priya-darśanaḥ*)) "one who is dear to all," whereas *tapassvādhyāyaniratam* can be analyzed only as ((*tapas-svādhyāya*)-*niratam*) "one who is constantly engaged in penance and self-study." The constituency parsing is similar to the problem of completely parenthesizing $n + 1$ factors in all possible ways. Thus, the total possible ways of parsing a compound with $n + 1$ constituents is equal to a *Catalan number,* C_n (Huet 2009), where

$$C_n = \frac{(2n)!}{(n+1)!n!} \text{ for } n \geq 0.$$

Thus, as n increases, the total number of possible groupings increases exponentially. Correctness of a parse for a given compound is governed by the semantics of the components involved. It is the meaning compatibility (*sāmarthya*) of the components that decides the correct analysis. Kulkarni and Kumar (2011) proposed a statistical constituency parser that uses statistical properties of a tagged corpus to model the *sāmarthya*. Due to unavailability of a tagged corpus for NN, it was not possible to follow this approach for parsing NN expressions. However, an intelligent user-interface that takes advantage of NN syntax was developed (Arjuna and Kulkarni 2016) to help the user to select the correct parse.

Semi-Automatic Parsing

Following observations related to the syntax of NNEs were crucial in designing a constituency parser for NNEs.

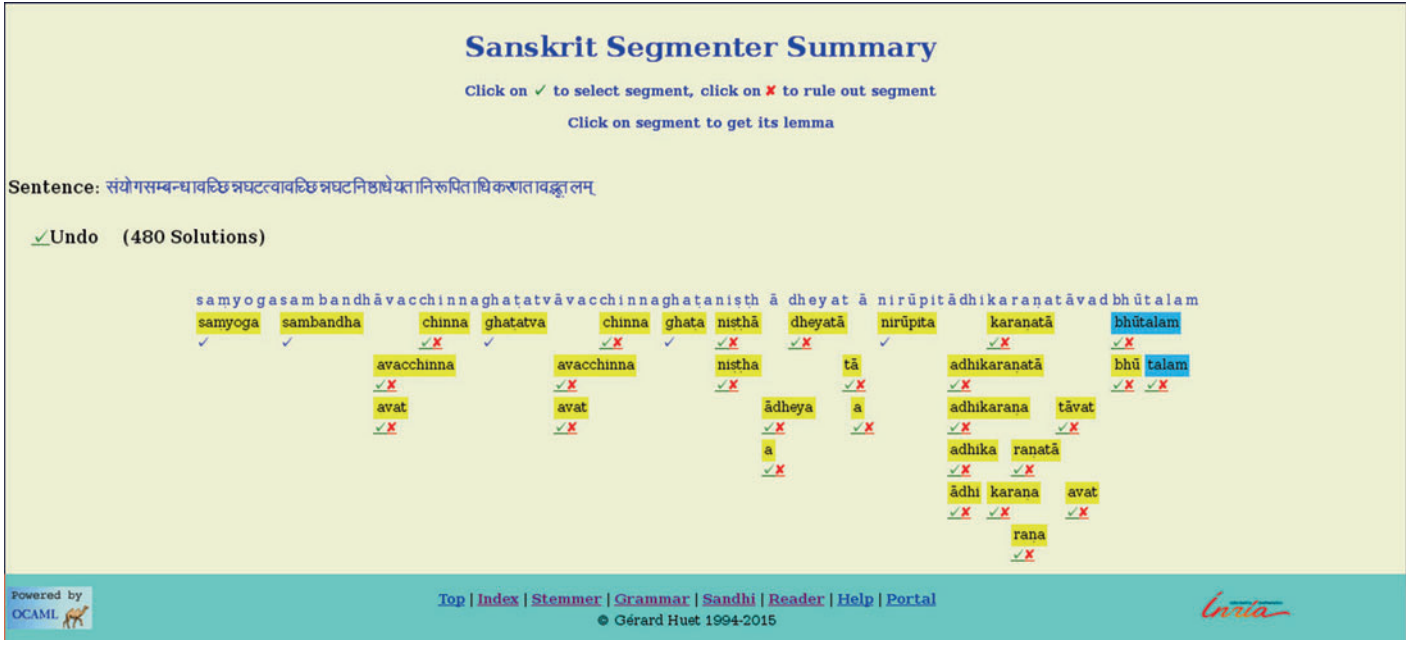

Fig. 10 User interface showing possible splits of a NNE

1. Concepts and Relations alternate in an NNE. For example, consider *gandhatva-avacchinna-gandha-niṣṭha-ādheyatā*. Here the components *gandhatva*, *gandha*, and *ādheyatā* denote the concepts and the components *avacchinna* and *niṣṭha* denote the relations.
2. The pratiyogin of a relation always immediately precedes the relation term. Thus, for example, in the above example, the pratiyogin for *avacchinna* is *gandhatva*, and the pratiyogin for *niṣṭha* is *gandha*. If "R" is a relation which connects two concepts "a" and "b" resulting in a compound "a-R-b," then such a compound thus always will be parsed as $((a - R) - b)$, and never as $(a - (R - b))$. Thus, this constraint rules out almost half of the possible parses. So an NNE with 3 terms "a-R-b" is unambiguous. Now consider an NNE with 5 terms viz. "a-R-b-S-c" where "a," "b," and "c" denote the concept terms and "R" and "S" denote the relation terms. This compound is ambiguous. The ambiguity is with respect to the *anuyogin* of "R," with two possible parses being, $((a - R) - ((b - S) - c))$ and $((((a - R) - b) - S) - c)$. In the first case, the *anuyogin* of "R" is "c," while in the second, it is "b." It is the context that tells us which parse is correct. For example, in *samavāyasambandha-avacchinna-gandha-niṣṭha-ādheyatā*, the *anuyogin* of *avacchinna* is *ādheyatā*, while in *gandhatva-avacchinna-gandha-niṣṭha-ādheyatā*, the *anuyogin* of *avacchinna*, in one reading, can be *gandha*. So, if there are "n" concept nodes after a relation node "R," the *anuyogin* of "R" potentially can be any of these "n" concepts. It is the context that decides which is the correct *anuyogin*.
3. A cue that rules out some more possibilities is the use of co-relative terms in Navya-Nyaāya. *Anuyogitā* and *pratiyogitā* are the co-relative terms, similarly, *ādheyatā* and *adhikaraṇatā* are the co-relative terms. And the relation-terms *nirūpita* and *nirūpaka* always combine two co-relatives.
4. Then there is of course, a well-nested-ness constraint. The resulting constituency structure should be well bracketed, without any crossings. In other words, if the *anuyogin* of a relation at k^{th} position is at "j," then the *anuyogin* of any relation lying between "k" and "j" cannot be beyond "j."

Thus, the three conditions, viz. (a) the *pratiyogin* is always to the immediate left of a relation node, (b) *nirūpita* and *nirūpaka* always connect two co-relative terms, and (c) the well-nested-ness condition, reduce the search space to a considerable degree. Since it is not clear what other factors are responsible for the correct choice of the *anuyogin*, a human being is involved who is well versed in Navya-Nyāya to mark the correct *anuyogin*s in the cases of ambiguities. The design of the interface takes care of the above three conditions and dynamically reduces the search space with every choice.

For instance, the input *samavāyasambandha-avacchinna-gandhatva-avacchinna-gandha-niṣṭha-ādheyatā-nirūpita-adhikaraṇatāvat-vastu* will be parsed as shown in Fig. 11.

After selecting the *anuyogi*s for all relations (see Fig 12), the constituency parse as a linear bracketed expression is

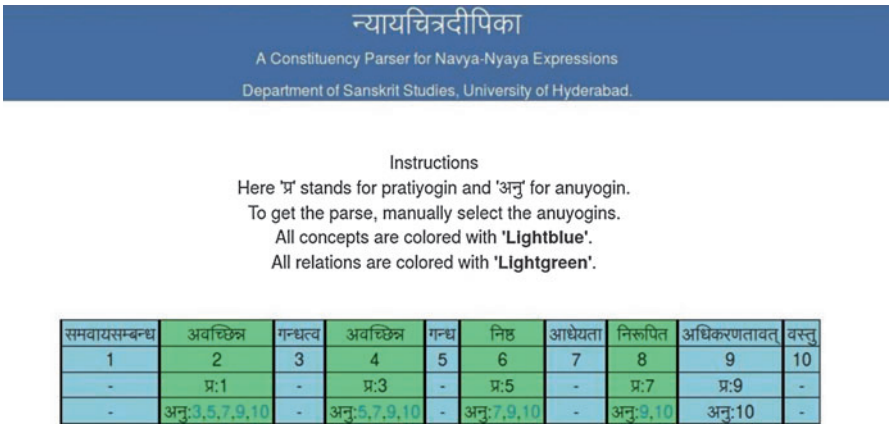

Fig. 11 A screenshot of interface of NN-Parser

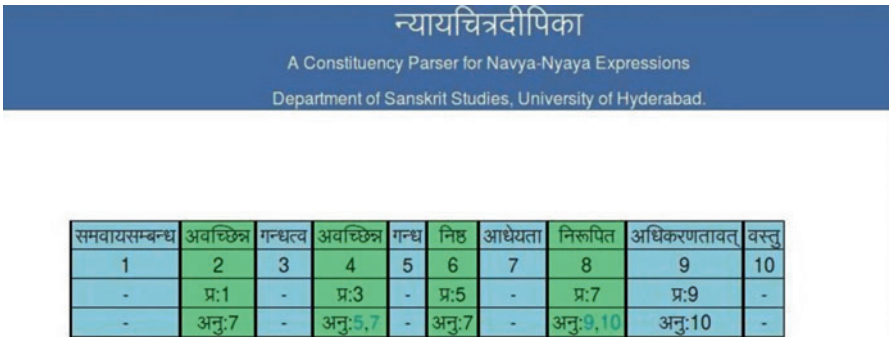

Fig. 12 A screenshot of interface after user-selection

$$((((((samav\bar{a}yasambandha - avacchinna) - (gandhatva - avacchinna) - (gandha - niṣṭha) - \bar{a}dheyat\bar{a}))) - nir\bar{u}pita) - adhikaranat\bar{a})\hat{}vat - vastu)$$

Sometimes NNEs do not specify the relation between the concepts explicitly. For example, the expression *ghaṭa-abhāva-vat-avṛttitvam* has two concepts *ghaṭa* and *abhāva* as consecutive nodes. In such cases these are treated as a compound with an un-specified relation and is parsed as (((*ghaṭa-abhāva*)-*vat*)-*avṛttitvam*).

Translating NN Expressions into Conceptual Graphs

Formal grammar *G* for such a parsed structure is defined below (Arjuna and Kulkarni 2016)

Let $G = (N, T, P, N\,N\,E)$, where

N: Set of nonterminal symbols
= {Compound Concept, Compound Relation, Rel term, Concept term},
T: Set of terminal symbols = {relation and concept},
NNE: The start symbol, and
P: Production rules as described below.

NNE	:	Compound_Concept ↑.*head* = ↓. *head*`
	;	
Compound_Concept	:	'(' Compound_Relation '-' Concept ↑. *head* = Concept term.head establish an edge between the head of the Compound Relation to the head of the Concept term
	;	
Compound_Relation	:	'(' Concept term '-' Rel term ')' ↑. *head* = Rel term.position draw a relation node for Rel term. establish an edge between the head of the Concept term to the relation node.
	;	
Concept_term	:	NNE ↑. *head* = ↓. *head*
	|	CONCEPT ↑. *head* = ↓.position draw a concept node
	;	
Rel_term	:	RELATION ↑. *head* = ↓. *head*
	;	

Concepts are the nouns, relational abstract expressions, the negation functor, and the terms derived with *tva* suffix from nouns. Relations are (a) the sentence forming operators *niṣṭha* and *avacchinna*, (b) the conditioning operator *nirūpita*, and (c) along with their inverse relations viz. *vṛtti* (or *āśraya*), *avacchedaka*, and *nirūpaka*, respectively. In NN the relations are always binary (dviṣṭaḥ sambandhaḥ.). Every relation node needs two relata. Thus, in order to draw a CG corresponding to an NN relation, (i) node labels, (ii) node types, and (iii) the two relata corresponding to the given relation are needed. With each rule of this grammar, a semantics in terms of an attribute grammar is associated which then translates an NNE into a CG. The node labels and the node types correspond to the **intrinsic** attributes of the terminal nodes *concept* and *relation*, which are available from the lexer. The two rules in the

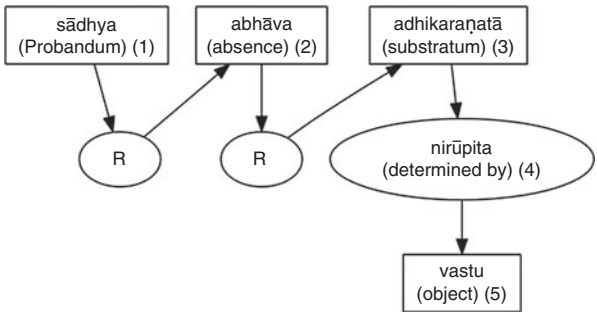

Fig. 13 CG with unspecified relation "R"

grammar above corresponding to *Compound Relation* and *Compound Concept* provide the links between a relation and a concept term.

It was observed that, in actual usage, if a compound is not ambiguous, it is not expanded with NN structure. Thus typically NNEs were found to be heterogeneous mixtures of classical Sanskrit and well-structured descriptions as stated above. In order to handle such heterogeneous structures, an unspecified relation "R" is established between the two consecutive conceptual nodes. For example, the expression

$$sādhyābhāvādhikaranatānirūpitavastu \tag{9}$$

contains only one NN technical term *nirūpita* and the remaining part of the expression is an ordinary classical Sanskrit compound with 4 components *sādhya*, *abhāva*, *adhikaraṇatā*, and *vastu* denoting concepts. Figure 13 shows the CG for this expression. A dummy relation "R" is introduced between two consecutive concept nodes. Similarly, when two relation terms follow each other, a dummy concept node "vastu" is introduced in between in order to faithfully render such an expression with a CG.

Nyāyacitradīpikā combines all the three modules described above and presents a platform for a user to help him understand an NNE (http://sanskrit.uohyd.ac.in/scl/NN/segmenter.).

Conclusion

Advances in computational linguistics and availability of computational tools for the analysis of Sanskrit texts has made it possible to produce conceptual graphs for NN expressions mechanically. The only subjectivity involved in this rendering is involvement of the user in the constituency analysis of the compound. It is also possible to generate the śābdabodha of a given sentence, using the parser for Sanskrit (Kulkarni 2013).

This way of analysis has its own limitations as well. For example, the nañ-tarpuruṣa compounds such as *avṛttitva* or *asāmānādhikaraṇya* are represented in the present scheme as a single concept node. But unless such compounds are

expanded elaborating the underlying structure, even the conceptual graphs will be as difficult as the original NN expressions. In the manual diagrammatic representations of Jha (1987) and Wada (2007), for example, we find such terms are analyzed further, and proper interpretation is provided.

Next natural step is to use the technical language of NN for natural deduction. Some initial work has been carried out by Srinivas Varakhedi in his doctoral thesis, in this direction. It is necessary to take that work one step ahead and build an inference engine that understands NN expressions.

References

Arjuna, S.R., and Amba Kulkarni. 2014. Segmentation of Navya-Nyāya expressions. In *Proceedings of international conference on NLP*, Goa.

Arjuna, S.R., and Amba Kulkarni. 2016. Analysis and graphical representation of navya-nyāya expressions – *Nyāyacitradīpikā*. In *Sanskrit and computational linguistics*, ed. Amba Kulkarni, 21–52. Pragun Publication, A D K Publishers Distributors Enterprise, New Delhi.

Bhattacharya, Sibajiban. 1990. Some features of the technical language of Navya-Nyaya. *Philosophy East and West* 40 (2): 129–149.

Briggs, Rick. 1985. Knowledge representation in Sanskrit and artificial intelligence. *AI Magazine* 6 (1).

Ganeri, Jonardon. 2008. Towards a formal regimentation of the Navya-Nyāya technical language-I. In *Logic, Navya-Nyāya & applications*, London:College Publications. 109–124.

Huet, Gérard. 2006. Lexicon-directed segmentation and tagging of Sanskrit. In *Themes and tasks in old and middle indo-Aryan linguistics*, 307–325. Delhi: Motilal Banarsidass.

Huet, Gérard. 2009. Formal structure of Sanskrit text: Requirements analysis for a mechanical Sanskrit processor. In *Sanskrit computational linguistics 1 & 2,* LNAI 5402, eds. Gérard Huet, Amba Kulkarni, and Peter Scharf. Springer, Berlin Heidelberg.

Huet, Gérard and Pawan Goyal. 2013. Design of a lean interface for Sanskrit corpus annotation. In *Proceedings of international conference on NLP*, Noida, eds. Dipti Mishra Sharma, Rajeev Sanghal, Karunesh Kr.Arora, and B.K.Murthy, 177–186.

Ingalls, Daniel H.H. 1951. *Materials for the study of Navya-Nyāya logic*. Cambridge, MA: Harvard University Press.

Jha, V.N. 1987. *Viṣayatāvāda of Harirāma Tarkālaṅkāra*. Pune: University of Pune.

Kulkarni, Amba. 1994. Navya-Nyāya for Scientists and Technologists: A first step. MTech dissertation, IIT, Kanpur.

Kulkarni, Amba. 2013. A deterministic dependency parser with dynamic programming for Sanskrit. In *Proceedings of the second international conference on dependency linguistics (DepLing 2013)*, 157–166, Charles University in Prague, Matfyzpress, Prague.

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

Kulkarni, Amba, and Devanand Shukl. 2009. Sanskrit morphological analyser: Some issues. *Indian Linguistics* 70 (1–4): 169–177. in the Festscrift volume of Bh. Krishnamoorty.

Kumar, Anil, Vipul Mittal, and Amba Kulkarni. 2010. Sanskrit compound processor. In *Proceedings of the fourth international Sanskrit computational linguistics symposium*, LNAI 6465, ed. Delhi. Girish Nath Jha, 57–69. Springer.

Matilal, Bimal Krishna. 1968. *The Navya-Nyaya doctrine of negation*. Cambridge, MA: Harvard University Press.

Matilal, Bimal Krishna. 1977. *Nyāya-Vaiśeṣika*. Harrassowitz, Verlag.

Mittal, Vipul. 2010. Automatic Sanskrit segmentizer using finite state transducers. In *Proceedings of student research workshop*, 85–90. Association for Computational Linguistics. Uppsala.

Mohanty, Jitendra Nath. 2000. *Classical Indian philosophy*. Oxford: Rowman & Littlefield Publishers.

Natarajan, Abhiram and Eugene Charniak. 2011. Sˆ3 – Statistical Sandhi Splitting. In *Proceedings of IJCNLP*, Thailand. 301–308.

Patil, Devadatta. 2014. *Vidyādharī*. Pune: Samarth media center.

Shaw, J.L. 1980. The Nyāya on cognition and negation. *Journal of Indian Philosophy* 8 (4): 279–302.

Sowa, John F. 1985. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, Reading, MA

Sowa, John F. 1992. Semantic networks. *Encyclopedia of cognitive science*. Wiley.

Staal, Fritz. 1988. Chapter: Means of formalization in Indian and Western logic. In *Universals: Studies in Indian logic and linguistics*, 81–87. Chicago: The University of Chicago Press.

Varalakshmi, K. 2013. Ś leṣālaṅkāra: A challenge for testing sanskrit analytical tools. In *Recent researches in Sanskrit computational linguistics fifth international symposium proceedings*, ed. Malhar Kulkarni and Chaitali Dangarikar. New Delhi: D. K. Printworld.

Wada, Toshihiro. 2007. *The analytical method of Navya-Nyāya*. Groningen: Egbert Forsten.

Woods, W. 1975. What's in a link: Foundations for semantic networks, Representation and Understanding: Studies in Cognitive Science, 35–82. Academic Press, New York.