Chapter 3

Creation of Hindi Treebank (HyDT)

A major goal of this thesis was to develop a scheme for annotating syntactic parsed sentences for Indian languages, taking Hindi as an example language. For achieving this goal, it is important to analyse and examine various Hindi constructions to conclude that the scheme is able to account for these in a sound manner. To verify the applicability of the scheme on actual corpus, a treebank has been created. For this, we have followed dependency-based annotation scheme based on Paninian Grammatical Framework [21, 36] which is a dependency grammar [99]. In this chapter, we present the Paninian Grammatical model used in building HyDT [21].

3.1 Paninian Grammatical Model

Indian Languages are morphologically rich and have a relatively flexible word order. For example, in the examples 3.1 (a)-(c) given below, the word order changes retaining the meaning.

Ex-(3.1) (a) *bachchaa phala khaataa hai*
  child fruit eats is
  'Child eats fruit.'

(b) *phala bachchaa khaataa hai*
  fruit child eats is
  'Child eats fruit.'

(c) *phala khaataa hai bachchaa*
  fruit eats is child
  'Child eats fruit.'

For such languages, syntactic subject-object positions are not always able to elegantly explain the varied linguistic phenomena. In fact, there is a debate in the literature whether the notions ‘subject’ and ‘object’ can at all be defined for Indian Languages [121]. Behavioral properties are the only criteria based on which one can confidently identify grammatical functions in Hindi [123]; Marking semantic properties such as thematic roles as dependency relations is problematic too. Thematic roles are abstract notions and require higher semantic features which are difficult to formulate and to extract as well. We therefore need a grammatical model which can account for most of the linguistic phenomena in Indian languages and
would also work well for computational purposes. Panini’s grammar [36], on the other hand, offers a theoretical model which works well for morphologically rich languages and suits well for our requirements. Moreover, it offers a level of analysis which being syntactico-semantic in nature provides us a good combination of syntactic and semantic features for processing natural language. Thus, for our work we chose a computational model of Panini’s Grammar.

Paninian framework is a dependency based analysis [99, 162]. Dependency grammar formalisms have emerged from the work of Tesnière [171]. The basic elements in the dependency grammar are: (i) head word which is a governor, and (ii) its dependent. There exists a binary relation between these two words in a sentence. In dependency annotation, the relation between head and dependent is marked with functional categories. Syntactic annotation in the dependency framework has two types of inter-related decisions: attachment and labeling [190, 79, 78, 168, 172]. If one word attaches with another then it indicates that there is a syntactic relationship between the head word and the dependent word. There is a parent-child relationship between head word and dependent word. Head word will be the parent and dependent word will be the child. The relations will tell the type of the attachment. For example, if the noun is the subject of the verb then the attachment of a dependent noun with the head verb will be marked as relation subject [77].

### 3.1.1 Roots of Paninian Grammar

Panini’s grammar is a generative grammar. Also, “The earliest references to dependency-based analysis are often attributed to Panini’s grammar of Sanskrit, ca. 400-500 B.C., [104]”¹. Panini's grammar is appreciated for its insightful study of Sanskrit. There are 4 parts in Paninian grammar [98]: (i) Ashtadhyayi explains 4000 grammatical rules which deal with the sentence structure; (ii) Sivasutras explain the phonological segments; (iii) Dhatupatha explains 2000 verbal roots, and their morphological and syntactic properties; (iv) Ganapatha is a list of 261 lexical items. The rules of the Ashtadhyayi are divided into three classes [98]. The grammar studies sentences in a hierarchy of four levels which are passed through by three mappings from semantics to phonology [99, 48, 90] [98]. In other words, the derivation of a sentence begins from the level of meaning and ends with its phonological form (Itkonen, 1991). The three classes of rules map preceding levels onto the succeeding levels [98, 157; 158] which assign karakas and abstract tense;

1) Morphological spellout rules;

2) Rules of allomorphy and phonology.

The figure 3.1 given below (taken from [98]) describes the derivation of the sentence:

The derivation of the sentence is as follows: The lexicon verb roots are *dhatu* and nominal stems are *pratipadika*. The derivation is started by choosing items from the lexicon and deciding semantic relations between verb roots and nominal stems [98]. These relations or roles are called *karakas*. According to Panini’s grammar, a sentence contains an action having different participants. These participants are classified into role types called *karakas*. There are six *karakas*: Agent ‘kartr’, Goal ‘karman’, Recipient ‘sampradana’, Instrument ‘karana’, Locative ‘adhikarana’ and Source ‘apadana’. Few *karakas* map to more than one semantic role and vice versa. *karakas* mediate between meaning and morphosyntactic structure [98]. The basic rules that govern the relation between *karakas* and morphosyntactic surface structure are [98]:

(i) Every *karaka* must be “expressed” (*abhihita*) by a morphological element.
(ii) No *karaka* can be expressed by more than one morphological element.
(iii) Every morphological element must express something.

In the section below, we describe the Computational Paninian Grammar framework [36].

(1) Semantic information

Assignment of *kārakas* (Th-roles) and of abstract tense

(2) Morphosyntactic representation

Morphological spellout rules

(3) Abstract morphological representation

Allomorphy and phonology

(4) Phonological output form

**Figure 3.1** Derivation of a Sentence from Meaning to Phonological Form
3.1.2 Computational Paninian Grammar

Computational Paninian Grammar (CPG) forms the basis of the dependency annotation which has been chosen here for the sentence analysis, hence for the tag names as well. It applies well to Indian languages [36]. The main problem that the Paninian approach addresses is to identify syntactico-semantic relations in a sentence. The framework is motivated by Sanskrit language which is an inflectionally rich language and focuses on the role of case markers such as post-positions and verbal inflections [36].

Thus our motivation for following the Paninian approach is:

a) It is inspired by inflectionally rich language (Sanskrit).

b) Is better suited for handling Indian languages, which have a relatively free word order and richer morphology (similar to Sanskrit).

c) The model, not only offers a mechanism for SYNTACTIC analysis, but also incorporates the SEMANTIC information (dependency analysis), i.e., it provides the level of syntactico-semantic interface for parsing.

The Paninian approach treats a sentence as a series of modifier-modified relations. A sentence is supposed to have a primary modified (the root of the dependency tree) which is generally the main verb of the sentence. The elements modifying the verb participate in the action specified by the verb. ‘karakas’ are the roles of different participants in an action. The relations between noun constituents and the verb are called karaka relations. karaka relations are participants directly involved in the action denoted by the verb. According to Patanjali, karaka, is the one which performs an action (karotiti kaarakam ‘The one that does’). For a noun to have a karta relation with a verb, it is required that both the noun and the verb have a direct relation. The appropriate mapping of the syntactic cues helps in identifying the appropriate karakas (‘participants in an action’) [21]. There are six basic karakas, namely; karta ‘agent’, karma ‘theme’, karana ‘instrument’, samprada ‘recipient’, apadana ‘source’, adhikarana ‘location’.

Below given is the rough mapping of karaka roles with theta roles:

1) **karta** (k1) : subject/agent/doer/experiencer/force.  
   ‘the most independent participant in the action’.

2) **karma** (k2) : object/patient/theme/goal/content-of-event/result (of creation).  
   ‘most desired to be attained by the karta’.
3) karana (k3) : instrument
   ‘instrument which helps in accomplishing the action’.

4) sampradana (k4) : beneficiary/recipient
   ‘intended recipient of the object’

5) apadana (k5) : source
   ‘fixed point of departure (or) moving away from a source’

6) adhikarana (k7p/k7t/k7): location in place/time/other

These are sufficient for providing a mapping from karaka relations to semantic relations. karakas provide the maximum necessary information relative to a verb. In our Paninian based approach, the verb is taken as the root of the tree and its argument structure is considered as its children [36]. The labels on the edges between a parent-child pair show the relation-type between them [84].

One must note here that although one can roughly map the last four karakas to their thematic role counterparts, the notion of karta and karma is not equivalent to that of the ‘agent’, and ‘theme’ thematic roles (although they might map to them sometimes) respectively. The reason for this divergence in the two notions (karaka and thematic role) is due to the difference in what they convey. Thematic role is purely semantic in nature whereas the karaka is syntactico-semantic. Example 3.2, illustrates this point; (This has been explained below using examples in Hindi)

Ex- (3.2) chaabii ne darvaazaa kholaa
   key   Erg. door opened
   ‘The key opened the door’

In the above example chaabii ‘key’ is karta, whereas it takes instrument thematic role. Panini defines karta as svatantra karta which can be translated as ‘the participant which is the most independent in a given action’. karta is the one who carries out the action (discussed below in more detail). In 3.2, ‘key’ has such a property. When the speaker uses ‘key’ in 3.2, he/she intends to elevate the role of ‘key’ in the action of opening and does not communicate the actual agent of the action. The speaker uses ‘key’ as the independent participant in the act of opening. Hence, ‘key’ is the karta.

As mentioned above, the thematic roles are Agent, Patient, Instrument, etc [186]. The examples 3.3 to 3.5, given below, show how thematic roles depict in Hindi.
In the example 3.3, the verb *pakaa* ‘cook’ takes two participants, *the one who cooks* (*raam* ‘Ram’) and *the item cooked* (*chaaval* ‘rice’). In the examples 3.3 and 3.4 (a), the role played by the subject *raam* ‘Ram’ is of causing an event to occur, hence it is called as *agent*. Object *chaaval* in 3.3(a), is undergoing some action so it is called as *theme/patient* theta role. *Agent* theta role has a rough semantic definition on the one hand and syntactic properties on the other, i.e., for example, agent theta role is always expressed as the grammatical *subject* rather than the *object* [186]. The theta role labelling helps in distinguishing the different roles of the *subjects* in the transitive and intransitive constructions given in examples 3.4(a) and 3.4(b) respectively [186]. Though the semantic roles of *raam* ‘Ram’ (agent) and *chaaval* ‘rice’ (theme) are different in the examples 3.4(a) and 3.4(b) respectively, they are the *subjects*. In example 3.5, which is a passive construction, *chaaval* ‘rice’ is *theme/patient* and is the *subject* of the sentence. Thus, any theta role such as *agent, theme, or instrument* can occur as a *subject*. Object is generally *theme/patient* (*chaaval* ‘rice’ in 3.3(a)) in a transitive verb.
Although karaka relations appear similar to the thematic relations, they are markedly different from them conceptually. For example, the notions of karta and karma are not translatable to agent and theme or the grammatical roles of subject and object. A karaka based analysis of examples 3.3 to 3.5 shows the difference more clearly. Here, the karaka based analysis of examples 3.3 to 3.5 is shown in the examples 3.6 to 3.8, given below:

Ex- (3.6) raam (subject, agent, karta) chaaval (object, theme/patient, karma) pakaar rahaai hai

ram rice cook Prog. is

‘Ram is cooking rice.’

Ex- (3.7) (a) raam (subject, agent, karta) pakaar rahaai hai

ram cook Prog. is

‘Ram is cooking.’

(b) chaaval (subject, theme/patient, karta) paka rahe hai

rice cook Prog. is

‘Rice is cooking.’

Ex- (3.8) chaaval (subject, theme/patient, karma) raam dvaaraa (agent, karta) pakaaye gaye

rice ram by cooked Pasv.

‘Rice was eaten by Ram.’

If we look at the karaka based analysis of the above examples, we find that karta does not always map either with the theta role agent or the grammatical relation subject. In the above examples, raam ‘Ram’ (subject, agent) in 3.6(a) and 3.7(a), is karta. chaaval ‘rice’ (subject, theme/patient) in 3.7(b) is also karta. chaaval ‘rice’ (subject, theme/patient) in 3.8 is karma. In the example 3.6, the verb pakaar ‘cook’ is taking two arguments, raam ‘Ram’ (karta) and chaaval ‘rice’ (karma) whereas its unaccusative form paka ‘cook’ in example 3.7(b) is taking only one argument, i.e., chaaval ‘rice’. This single argument of the unaccusative form paka ‘cook’ becomes karta. So both subject-agent and subject-theme/patient combinations are karta. This is because of the concept of vivaksha (discussed below) which is the speaker’s intention or what he wants to communicate [36].

Here, to understand the notion of karta and karma better, we need to look at the elements of the semantic model in the Paninian framework [36];

A verbal root (dhaatu) indicates an action comprising of:

(a) an activity (vyapaara) and,

(b) a result (phala)

Activity consists of actions performed by various participants or karakas involved in the action. Result is the condition or state which when reached, the action is complete [36]. Thus every action involves an
activity and a result. Ashraya or locus of the activity is karta and among all the participants in the action, karta is swatantra ‘independent’, i.e., it is the most independent karaka. Ashraya or the locus of the result is called karma (k2).

To take it further, in Paninian grammar, every action is a bundle of sub-actions. For instance, if we take the action of opening a lock, a person inserts a key in the lock and turns it, the key on its part presses the levers and moves them, then the latch moves and the lock opens. Each of the sub-actions in the action of opening a lock, i.e., (i) inserting and turning a key; (ii) the key pressing and moving the levers; (iii) the latch moving and the lock opening; all these sub-actions have their own semantic relations with associated objects. Thus every karaka becomes the karta ‘doer’ of its own action. Among all the participants in the action, karta is ‘swatantra’ (most independent). karta is the one who carries out the action and is conceptually different from the agent theta role as it does not always have volitionality. It is the locus of the activity implied by the verb root, i.e., the activity resides in or springs forth from the ‘karta’ [28]. This opens up the possibility of expressing their various sub-actions from their respective kartas perspective.

If we take the following example:

Ex- (3.9) (a) The boy (k1) opened the lock (k2) with the key (k3) [36]
(b) The key (k1) opened the lock (k2)
(c) The lock (k1) opened

In the above examples, key gets assigned karta (k1), karana (k3) based on the kind of sub-action under focus. In example 3.9 (a), key gets assigned karana and in 3.9 (b), it gets assigned karta based on what the speaker wants to express. The larger action of opening the lock involves following sub-actions (i) action of boy, (ii) action of the key, and (iii) action of the lock. Therefore, ‘key’ and ‘lock’ are karta in examples 3.9 (b) and 3.9 (c) respectively. The grammar talks of two types of karta: (a) primary and (b) secondary. Primary karta has volitionality, whereas the secondary karta does not. Therefore, kartas of 3.9(b) and 3.9(c) above do not have volitionality. Participants are assigned various relations accordingly. This is referred to as vivaksha in Paninian framework. The grammar facilitates analysis of the intended meaning as an 'expression' of what the speaker wants to communicate (vivaksha). Every sentence reflects speaker’s intention. The notion of vivaksha is the realization of speakers’ intention in a sentence. Vivaksha indicates the speaker’s way of thinking or attitude towards the activity [36]. A sentence is not just a statement of an objective activity but also has information about the speaker’s way of thinking. Vivaksha influences the selection of the verb form which, in turn, affects the choice of participants and their relation with or role in the action [36]. Hence we notice that the notion of karta-karma conceptually differs from the semantic notions of agent-theme, etc. and the syntactic notions of subject-object, etc.

We have discussed about karta (k1) above and now we briefly discuss the remaining karakas. Ashraya or the locus of the result is called karma (k2). karana (k3) karaka is an instrument and with vyapara or activity of the karana, phala or result is immediately achieved. sampradana (k4) karaka is beneficiary or recipient of the action. apadaan (k5) karaka is the one which remains fixed in the action.
involving separation. *adhikarana* (कृ) *karaka* is the locus of *karta* or *karma*. It supports *karta* or *karma* in space or time.

In our framework, we follow two levels of analysis:

1) **Syntactico-semantic relations** (*karaka* relations):
   - (i) Direct participants of the action denoted by a verb (*karaka*)
   - (ii) Other relations: *purpose, genitive, reason* etc.

2) **Relation markers** (*Vibhaktis*)

Thus, *karaka* relations are semantic on the one hand and syntactic on the other. *karaka* relations are syntactico-semantic (or semantico-syntactic) relations between the verbs and other related constituents (typically nouns) in a sentence. They capture a certain level of semantics. This is the level of semantics that is important syntactically and is reflected in the surface form of the sentence(s) [36]. What is important to note here is that such a level can be exploited computationally with ease.

*karakas* and other relations in a sentence are expressed through respective markers (Hindi postposition /case markers). These are referred to as *vibhaktis* in Paninian grammar. Thus, *vibhaktis* that occur with the *karakas* help in establishing the relation between nouns and verbs. *vibhakti* is an abstract notion used for denoting the case markings on the nouns and the TAM (tense, aspect and modality) of the verbs. The case markings on noun, i.e., ending in nominal inflection is called *sup* and the TAM of verbs, i.e., ending in verbal inflection is called *ting*. In example 3.10 given below, the *ne* postposition of *ram* ‘Ram’ and the *yaa* TAM of *kaaTaa* ‘cut’ are the *sup* and *ting* respectively. *vibhaktis* play a key role in indicating semantic relationships. They act as syntactic cues in a sentence. These syntactic cues help in identifying the relation types (*karakas*). In example 3.10, *ne* *vibhakti* indicates *karta*, *se* *vibhakti* indicates *karana*, and *0 (zero)* *vibhakti* indicates *karma*.

**Ex- (3.10)**

```
raam    ne     chaakuu se     seba         kaaTaa
   ram  Erg.    knife    with    apple    cut
```

```
karta(doer)    karana(instrument)    karma (theme)
```

‘Ram cut an apple with a knife.’

However, the correspondence between the *karaka* role and the marker through which it is expressed is not one-to-one. For a given *karaka*, different *vibhaktis* may be used to indicate its relation with a verb. This may vary from language to language. For example, in Hindi, *raam* ‘Ram’ is *karta* in the examples 3.11 and 3.12 given below, which has different *vibhaktis*, i.e., *ne* and *ko* respectively.
Ex-(3.11) *raam ne (k1) seb khaayaa*

raam Erg. apple ate

‘Ram ate an apple.’

Ex- (3.12) *raam ko (k1) seb khaanaa pa.Daa*

ram Dat. apple had-to-eat

‘Ram had to eat an apple.’

Similarly, a *vibhakti* can mark more than one relation. In Hindi, *vibhakti* ‘se’ marks *karana* (k3) ‘instrument’ as well as *apadana* (k5) ‘source’. Similarly, *zero vibhakti* in Hindi may express *karta*, *karma*, or *adhikaran* in some cases.

**Another important feature of the framework is that it spells out that one of the *karakas* (either *karta* or *karma*) may be expressed, not by an explicit marker but through the agreement features of the verb. This *karaka* appears with *zero vibhakti*.** Thus a verb’s TAM plays an important role in identifying *karaka* roles sometimes. For example, the main verb can indicate to the *karta karaka* by being in agreement with it. In the example 3.1 (a), the main verb *khaataa hai* ‘eats’ agrees with *bachchaa* ‘child’ in gender, number, and person. When there is a *vibhakti* following a noun then there will be no agreement between that noun and the verb. *vibhaktis* block the agreement between noun and the verb. This concept of agreement is called as *abhihita*. This concept is required to understand the relationship between noun, *vibhakti* and TAM of the verb which helps us in identifying the *karaka*. The concept of *abhihita* serves as a major cue in a sentence for marking the exact *karaka* relation that holds between a noun and a verb.

Two other important notions in the Paninian grammatical tradition are:

1) *aakaankshaa* (Expectation/Demand):

Every word has certain demands to be fulfilled for its meaning to be completed. For Parsing, verb is the most critical element. The verb’s basic demands which make it meaningful is called the *aakaankshaa (Expectation/Demand)* of the verb, in other words, its argument structure. Few words make demands and the other words satisfy them. Verbs demand for their *karakas*, and nouns satisfy them [36]. For example, the verb ‘read’ demands for two entities i.e.; (i) reader (someone who reads), and (ii) thing read (book). Different verbs have different *aakaankshaa*. The demand frames fulfill the demands of a verb, i.e., the verb’s argument structure which complete the meaning of the verb. This concept of *aakaanksha* is useful in the creation of *verb frames* (demand frames) for Hindi verbs (Refer Chapter 6). While creating treebank also we can refer to Hindi verb frames to decide or get an actual idea of the arguments or demands of a Hindi verb. So this concept of *aakaankshaa* given in Computational Paninian Grammar (CPG) plays a major role in analyzing the Hindi verbs which in turn are central to *treebanking* and *verb frames*.
2) yogyataa (Eligibility):

The demands of a word (verb as an example) are fulfilled not by any element, but by an element which has the ‘eligibility’ to do so. yogyataa, therefore, is roughly equivalent to semantic ‘selectional restrictions’ of the verb. If the demand of a verb has to be fulfilled by a noun, then the nouns have to go through the eligibility constraints to become the arguments of the verb. Only those nouns which have the desired parsarg (postposition) and semantic properties are eligible [36]. For example, the verb ‘eat’ demands for an ‘animate person’ who can eat something which is ‘edible’. In example 3.1 (a), phala ‘fruit’ does not have the eligibility to become the karta of the verb khaa ‘eat’. The verb can’t select a noun argument which doesn’t match with it in semantics. Constraints based on yogyataa require semantic knowledge for each lexical item. This knowledge can be obtained from a lexical resource such as a 'WordNet'. The concept of yogyataa helps us in deciding which noun has what relation with the verb according to the verb’s semantic ‘selectional restrictions’.

With these concepts available in the Computational Paninian Grammar (CPG), it is very easy for us to analyse and understand the Hindi data in a larger context on the basis of this framework.

In this chapter, we have explained the linguistic aspect of Paninian framework as applied to modern Indian languages [36]. In the next chapter, we discuss about the dependency annotation scheme [21] used to annotate Hindi dependency Treebank (HyDT).