Chapter 5

Some Hindi Constructions

For developing scheme and its applicability on the Hindi data, some specific constructions of Hindi were studied in detail. The objective was:

(a) To check whether these constructions could be handled well by Computational Paninian Grammar (CPG) and if yes then,

(b) To come up with an appropriate way to represent them within the scheme.

[21] shows that Computational Paninian Grammar (CPG) is able to account for these constructions well and the scheme is also able to represent them consistently. As the constructions were analysed with the help of dependency annotation scheme, we got a deep understanding of the Hindi data. We found that Computational Paninian grammar (CPG) could quite exhaustively analyse various Hindi constructions. For verifying this, we studied certain complex constructions which pose issues and have been the subject of debate in linguistics and saw that the scheme handles these constructions seamlessly.

Following constructions were studied:

I. It was necessary to test the applicability of the scheme on known sentence types such as Relative clauses, Causatives, Complement clauses, Unaccusatives, Copula, Existential, Gapping, Coordination etc.

II. Different types of constructions having Ellipses were also studied.

III. Conjunct verbs and Discontiguous elements are well known constructions of Indian languages for their complexity and productivity.
5.1. Relative Clauses

A relative clause modifies a noun or a noun phrase. They give additional information about the noun which they modify. It contains a relative pronoun or a relative adverb through which it gets connected to the main clause of the sentence. In Hindi, a relative clause construction has a 'jo' pronoun. The modified element contains a pronoun 'vaha' referred to as correlative. The relative and correlative pronouns are used in pairs which correlate two clauses [169].

There are two types of relative clauses in Hindi:

a) Correlative clauses
b) Participle clauses

5.1.1 Correlative clauses

They have explicit relative markers which appear at the initial positions of the clauses. The correlative clauses are divided into two types: (a) Prenominal, and (b) Postnominal.

5.1.1.1 Prenominal

The relative clause occurs to the left of the head noun and it carries a relative pronoun *jo*.

Ex- (5.1) [jo laDakaa vahaa.N kha.Daa hai ] [ vaha meraa bhaaii hai ] [28].

The boy who is standing there is my brother.

In the above example 5.1, the relative clause [jo laDakaa vahaa.N kha.Daa hai] 'the boy who is standing there' is modifying *vaha* 'he' of the main clause. The pronoun *vaha* 'he' in the main clause is identical to *jo laDakaa* 'who boy' of the relative clause. *jo laDakaa* 'who boy' in the relative clause is the referent of the pronoun *vaha* 'he' in the main clause.

5.1.1.2 Postnominal

The relative clause occurs to the right of the head noun. Here the relative pronoun behaves like a full-fledged pronoun and is not a modifier any more.
The relative clause occurs to the right of the main verb as in Ex- (5.3):

Ex- (5.3) vaha laDakaa meraa bhaaii hai [jo vahaa.N khaDa hai]
he boy my brother is who there standing is
'That boy is my brother who is standing there.'

A relative clause modifies the following elements in the main clause: karta (subject/agent), karma (direct object), samradana (indirect object), karana, adhikarana (oblique object) etc., [28].

Ex- (5.4) jisa chaakuu se mai ne seba kaaTaa [vaha bahut teza thii]
which knife with I Erg. apple cut that very sharp was
‘The knife with which I cut the apple was very sharp.’

5.1.2 Participles

These are non-finite clauses which have a non-finite form of the verb preceding the modified NP. These participle clauses modify head nouns as relative clauses do. These participle clauses don’t have a relative pronoun. There are adjectival and adverbial participle clauses. They take both present and past participle forms. Present appears in ‘verb + -taa + huua’ form and past appears in ‘verb + -aa + huua’ form. Below given is an example of adjectival past participle form:

Ex- (5.5) [mere dvaaraa khaaye gaye] [phala achChe the] [28].
my-obl by eat-perf Pasv. fruits good were
‘The fruits eaten by me were good.’

In the above example ‘khaaye gaye’ ‘eaten’ is the non-finite form of the verb which is present in the past participle clause ‘mere dvaaraa khaaye gaye’ ‘eaten by me’. The non-finite form of the verb is preceding the noun phala ‘fruits’ that it modifies which is present in the main clause ‘phala achChe the’ ‘the fruits were good’.

A participle clause modifies the noun. The modified noun is an argument of the main verb and also an argument of the non-finite verb present in the participle clause. The noun is shared by the main verb and the verb of the participle clause. This noun occurs only once in the sentence but is semantically
related to both the verbs. The adjectival participle clause can’t have the modified noun as its explicit argument. The shared argument syntactically always attaches with the main verb. For the verb in the participle clause, this noun is semantically realized but not syntactically [28].

Here the issue is how to capture the relation between phala ‘fruits’ and the verb of the participle clause ‘khaaye gaye’ ‘eaten’? In the above example 5.5 (figure 5.1), noun phala 'fruits' is an argument of both, the finite verb the 'were' and the non-finite verb ‘khaaye gaye’ ‘eaten’. The relation of phala 'fruits' with the main verb the ‘were’ is karta (k1). The participle clause will attach to the phala 'fruits'. The relation between phala ‘fruits' and the verb of the participle clause ‘khaaye gaye’ ‘eaten’ is nmod__k2inv. nmod in nmod__k2inv indicates that the participle clause is modifying the noun phala 'fruits'. k2inv in nmod__k2inv indicates that phala 'fruits' has k2 relation with the verb of the participle clause ‘khaaye gaye’ ‘eaten’ and inv in k2inv denotes that, here the relation arc is going from child to the parent instead of going from parent to the child. In this way we have marked the relation of phala 'fruits' with the main verb the ‘were’ as well as with the verb of participle clause ‘khaaye gaye’ ‘eaten’.

![Dependency Diagram](image)

**Figure 5.1** Participles

The dependency analysis of sentences having relative clauses, such as example 5.1 given above, can be done in two possible ways. Bharatí et al., [36] provides the following analysis for relative clauses:

### A. Possibility-I

There are two clauses in the above example 5.1 'jo laDakaa vahaa.N kha.Daa hai’ ‘The boy who is standing there’ which is a relative clause and 'vaha meraa bhaaaii hai’ ‘he is my brother’ which is a main clause. The verb hai ‘is’ is the root of the main clause, whereas 'jo laDakaa’ ‘who boy’ is the
root of the relative clause. Possibility-I provides a relation between vaha ‘he’ in the main clause and ‘jo laDakaa’ ‘who boy’ in the relative clause, which means, there is a dependency relation between them. ‘jo laDakaa’ ‘who boy’ directly attaches to vaha ‘he’. The relation between them is captured by nmod__relc tag. This relation between vaha ‘he’ and ‘jo laDakaa’ ‘who boy’ is yat-tat (that-which in Sanskrit) relation. Within the relative clause, the verb ‘kha.Daa hai’ ‘is standing’ has an nmod__k1inv relation with ‘jo laDakaa’ ‘who boy’. Below given are the advantages of Possibility-I:

(i) Provides relation between vaha ‘he’ in main clause and ‘jo ladZakaa’ ‘who boy’ in relative clause.

(ii) Dependency of ‘jo ladZakaa’ ‘who boy’ is on head noun vaha ‘he’.

(iii) Captures the relative - corelative relation well.

The dependency tree for Possibility-I will be as follows:

![Dependency Tree](image)

**Figure 5.2 Relative Clauses Possibility-I**

---

1 nmod in the tag nmod__relc stands for noun modifier and relc stands for relative clause. nmod__relc tag is used where relative clause construction modifies a noun.

2 nmod__k1inv tag is used for the nouns that are modified by a non-finite verb and at the same time it is also the argument of that non-finite verb. jo laDakaa is being modified by the non-finite verb kha.Daa hai and is also the argument of this verb. It has kl relation with the verb and inv in the tag stands for inverse as the relation is inverse (child node is above the parent node).
B. Possibility-II

In Possibility-I, the root of the relative clause is ‘jo ladZakaa’ ‘who boy’, whereas in Possibility-II, the root of the relative clause is ‘kha.Daa hai’ ‘is standing’. Here the whole relative clause ‘jo laDakaa vahaa.N kha.Daa hai’ ‘the boy who is standing there’ is modifying vaha ‘he’ of the main clause. The relation between the relative clause and vaha ‘he’ will be nmod__relc. Within the relative clause, ‘jo laDakaa’ ‘who boy’ attaches with the verb ‘kha.Daa hai’ ‘is standing’. Though the whole relative clause is modifying vaha ‘he’ of the main clause, there is also a coreference relation between ‘jo laDakaa’ ‘who boy’ and vaha ‘he’. The coreference relation between them is captured by the feature coref. Below given are the advantages of Possibility-II:

(i) Allows to capture the relation between jo ladZakaa ‘who boy’ and vaha ‘he’ by the feature coref.

(ii) All the nouns get attached with the respective verbs of their clauses.

(iii) Entire relative clause modifies vaha ‘he’ which occurs in the main clause.

The dependency tree for Possibility-II will be as follows:

![Dependency Tree](image)

**Figure 5.3** Relative Clauses Possibility-II

6
The question before us, while deciding the annotation for relative clauses in our scheme was – which of the two possibilities should be adopted? Both represent the relative clause analysis aptly. Possibility-I gives a more accurate representation of the syntactic dependency between relative-correlative clauses whereas Possibility-II tilts more towards the semantic side.

In Possibility-II, the root of the relative clause is the verb ‘kha.Daa hai ‘is standing’. According to [36], the verb is the primary modified element in the sentence. The noun ‘jo laDakaa ‘who boy’ has dependency relation with the verb ‘kha.Daa hai ‘is standing’ of its clause,. Here the whole relative clause ‘jo laDakaa vahaa.N kha.Daa hai’ ‘the boy who is standing there’ is modifying vaha ‘he’ of the main clause. Though jo laDakaa ‘who boy’ does not attach directly to vaha ‘he’, even then ‘jo ladZakaa’ gets related to vaha through the verb ‘kha.Daa hai ‘is standing’ and it also allows to capture the relation between them by the feature coref.

We chose Possibility-II, as this representation is more consistent with the annotation for other clauses elsewhere, i.e., verb forms the root node of the tree. So for relative clauses, our current decision is Possibility-II, i.e., we are following Possibility-II in marking relative clauses.

5.2 Causative Constructions

Causative verbs mean that somebody makes somebody else do something or causes somebody to be in a certain state [9]. In the example 5.6 given below, maa.N ‘mother’ is causing bachchaa ‘child’ to sleep. Causatives in Hindi are realized through a morphological process. An intransitive or a transitive verb changes to a causative verb when affixed by either an ‘-aa’ or a ‘-vaa’ suffix [28]. Below we can see that verb so ‘sleep’ becomes sulaa ‘put to sleep’ which is a first causal by adding suffix ‘-aa’ and becomes sulvaa ‘cause to put to sleep’ which is a second causal by adding suffix ‘-vaa’.

<table>
<thead>
<tr>
<th>Base verb</th>
<th>First causal</th>
<th>Second causal</th>
</tr>
</thead>
<tbody>
<tr>
<td>so</td>
<td>sul-aa</td>
<td>sul-vaa</td>
</tr>
<tr>
<td>‘sleep’</td>
<td>‘put to sleep’</td>
<td>‘cause to put to sleep’</td>
</tr>
</tbody>
</table>

Ex- (5.6) maa.N ne bachche ko sulaayaa [28]
mother Erg. child Acc. caused-to-sleep
‘Mother put the baby to sleep.’

Hindi verbs are divided into two groups based on their behaviour in causative sentences: affective verbs and effective verbs [91]. The action of affective verbs benefits or affects the agent. Affective verbs
will have both first and second causal forms. **Effective verbs** and ditransitive verbs have only one causal form (Detail discussion in chapter 7).

The dependency analysis of causatives, such as example 5.7 given below, can be done in two possible ways. Below we have listed the two possibilities:

**Ex-** (5.7) **maa.N ne aayaa se bachche ko khaanaa khilvaaya** [28]

mother Erg. maid by child Dat. food caused-to-feed

‘Mother caused the maid to feed the child.’

A. **Possibility-I: Go by syntactic analysis**

The dependency tree for the above example 5.7 will be as follows:

```
khilvaaya `caused-to-feed`
  k1 k3 k4 k2
maa.N ne aayaa se bachche ko khaanaa
`mother Erg. `maid by` `child Dat. `food`
```

**Figure 5.4 Causatives Possibility-I**

In the above example 5.7 (figure 5.4), **khilva** `cause to feed` is the verb root. The arguments of the verb **khilva** `cause to feed` are marked as follows:

(i) **maa.N** `mother` has **ne vibhakti** (Hindi postpositions / case markers) which is a **karta vibhakti** so we mark it as **k1**;

(ii) **aayaa** `maid` has **se vibhakti** which is a **karana vibhakti** so we mark it as **k3**;

(iii) **bachcha** `child` has **ko vibhakti** which is a **sampradana vibhakti** so we mark it as **k4**.
Below given are the Advantages of Possibility-I:

a) Easier
b) Better accuracies
c) Less labels
d) Level of analysis is simpler
e) Syntactic analysis is given

B. Possibility-II

The verb khilvaā ‘cause to feed’ is a causative verb and it is morphologically related to the base verb khaa ‘eat’. Paninian framework provides the following relations [28]:

(i) prayojaka karta ‘causer’ (pk1): It is the causer in a causative construction.
(ii) prayajya karta ‘causee’ (jk1): It is the causee in a causative construction.
(iii) madhyastha karta ‘mediator causer’ (mk1): It is the mediator-causer in a causative construction.

Below given are the Advantages of Possibility-II:

a) Marks causativity
b) Semantically richer
c) Morphological relatedness is captured

So after comparing the advantages of both the possibilities, we prefer Possibility-II over Possibility-I because it has the causativity information. It captures the morphological relatedness between the base verb and the causative verb. It is also semantically richer. If we mark the above dependency roles, i.e., pk1, jk1 and mk1, then the base verb root will be khaa ‘eat’. The justification for selecting Possibility-II is discussed below with the help of examples:

Ex-(5.8) maa.N ne (k1) chammacha se (k3) bacce ko (k4) khaanaa (k2) khilavaayaa
mother Erg. spoon with child Dat. food caused-to-feed
‘Mother caused to feed the child with the spoon.’
Ex-(5.9) maa.N ne (k1) aayaa dvaaraa(?) cammaca se (k3) bacce ko (k4) khaanaa (k2) khilavaayaa
mother Erg. maid by spoon with child Dat. food caused-to-feed
‘Mother caused the maid to feed the child with the spoon.’

In the above example 5.8, the person doing the action of feeding is not present on the surface level but it is implicit. chammacha ‘spoon’ is an instrument used by the person doing the action of feeding. In the above example 5.9, there is an extra argument aayaa ‘maid’ when compared with example 5.8. Now the question that arises is that what will be the relation of aayaa ‘maid’ in the above example sentence 5.9?

By this evidence we can say that due to morphological relatedness present between the base verb khaa ‘eat’ and causative verb khilvaa ‘cause to feed’, we mark prayojak karta (pk1), madhyastha karta (mk1), prayojya karta (jk1) instead of karta (k1), karana (k3), sampradana (k4) respectively in the below examples 5.10 and 5.11. If we mark the above dependency roles, i.e., pk1, jk1 and mk1 then the base verb root will be khaa ‘eat’. The example 5.7 is again repeated here in examples 5.10 and 5.11.

Ex- (5.10) maa.N ne (k1) aayaa se (k3) bache ko (k4) khaanaa (k2) khilavaayaa.
mother Erg. maid by child Dat. food caused-to-feed
‘Mother caused the maid to feed the child.’

Ex- (5.11) maa.N ne (pk1) aayaa se (mk1) bache ko (jk1) khaanaa (k2) khilavaayaa.
mother Erg. maid by child Dat. food caused-to-feed
‘Mother caused the maid to feed the child’.

Both examples 5.10 and 5.11 are causative examples. In example 5.10, we have followed Possibility-I and in example 5.11, we have followed Possibility-II. In Possibility-II, maa.N ‘mother’, aayaa ‘maid’, and bache ‘child’ are marked as prayojak karta (pk1) ‘causer’, madhyastha karta (mk1) ‘mediator causer’, and prayojya karta (jk1) ‘causee’ instead of karta (k1), karana (k3), and sampradana (k4).

By choosing Possibility-II, we are able to capture the difference between aayaa ‘maid’ and chammacha ‘spoon’ by the verb khilavaanaa ‘cause to feed’. This difference is shown in the example 5.12 given below.

Ex-(5.12) maa.N ne (pk1) aayaa dvaaraa (mk1) chammacha se (k3) bacce ko (jk1) khaanaa (k2)
mother Erg. maid by spoon with child Dat. food
khilavaayaa caused-to-feed
‘Mother caused the maid to feed the child with the spoon.’

In this scenario, the dependency tree for the above example 5.7 will be as follows:
In the above figure, by ‘root = khaa’ we mean that the base verb root of the causative verb khilvaa ‘cause to feed’ is khaa ‘eat’. The causative verb khilvaa ‘cause to feed’ is derived from base verb khaa ‘eat’. As there is morphological relatedness between the base verb khaa ‘eat’ and causative verb khilvaa ‘cause to feed’, we mark pk1, mk1, and jk1 instead of k1, k3, and k4 respectively.

So for causatives, our current decision is Possibility-II, i.e., we are following Possibility-II in marking causatives. More detailed analysis of Hindi causative verbs is discussed in chapter 7.

5.3 Dative Subject Constructions (Experiencer Subject)

In these constructions, the subject is marked with a dative postposition ‘ko’. The verbs that take dative subjects denote liking, disliking, states of health or sickness, happiness and unhappiness, feeling, remembering, pity, pain, thirst, hunger, sleepiness, anger, etc. These verbs are divided into following categories [76]:

a) Verbs of sense-perception: These verbs include dikhaaii paDnaa ‘to be seen’ or sunaaii paDnaa ‘to be heard’. The subject of these verbs is an experiencer of the action of seeing or hearing and doesn’t initiate the action from his side.

b) Verbs of mental and emotional experiences: These verbs take nominals that express emotional experiences such as fear, shame, shyness, liking, disliking, anger, laughter, crying, happiness, sorrow, regret, etc. Dative subjects also occur with verbs of cognition such as maaluum honaa ‘to be known’, achChaa lagnaa, ‘to feel good’, pataa honaa ‘to be known’, buraa lagnaa ‘to feel bad’, yaad aanaa ‘to remember’ etc.
c) **Verbs denoting health conditions:** These verbs include *khaa.Nsii aanaa* ‘to cough’, *buKhaar aanaa* ‘to suffer from fever’

Ex- (5.13) *raam ko* (experiencer) *caa.Nd dikhaa* [28]

ram.Dat moon appeared

‘Moon was visible to Ram.’

Perception verbs such as *seems, appear* take a perceiver/experiencer. In the above example 5.13, what is the *karta* (k1) of the verb *dikhnaa* ‘to appear’? *caa.Nd ‘moon’* is the *karta* (k1) as *caa.Nd* is the doer of the action *dikhnaa*. The action of *dikhnaa* is happening in *caa.Nd*, i.e., it is the locus of the activity. *caa.Nd ‘moon’* is in agreement with the verb *dikhnaa* ‘to appear’. Therefore, *caa.Nd ‘moon’* is the *karta* (k1) of the verb *dikhnaa* ‘to appear’.

On the other hand, *raam* is the experiencer of the event of *‘caa.Nd dikhaa’*. Since *raam ‘Ram’* is the experiencer who is not making any effort so it becomes passive agent as it is experiencing the action of *dikhnaa* ‘to appear’. *raam ‘Ram’* is receiving or perceiving the activity carried out by another actor (*caa.Nd ‘moon’*). So, the question is what relation does *raam* have with *dikhnaa*? Since *raam ‘Ram’* is the experiencer, then indirectly it is also a participant in the action of *dikhnaa* ‘to appear’. However, since, *raam ‘Ram’* is passively performing the task of perceiving so it becomes a recipient. Therefore, it can be considered as a subtype of *sampradana* which is different from the *sampradana* (*k4 ‘beneficiary’*). We call it as *anubhava karta* represented by *k4a*. The *anubhava karta* is always takes a *‘ko’ vibhakti* (Hindi postpositions/ case markers). All experiencer verbs show this pattern. *k4a* label is used for annotating such dative subjects. Thus, in the above example *raam* is annotated as *k4a* (see figure 5.6). The dependency tree for the above example 5.13 will be as follows:

```
dikhaa ‘appeared’
   \   / 
  k4a k1
 /      
raam ko caa.Nd
 ‘Ram Dat.’ ‘moon’
```

**Figure 5.6** Dative Subject Construction
Similarly, in the example 5.14 given below, dukh ‘sorrow’ is the karta as syntactically it agrees with the verb hai ‘is’ and mujhko ‘I-Dat’ is anubhava karta (k4a) as it is experiencing the state denoted by the verb hai.

Ex- (5.14) mujhko (k4a) dukh hai [28]
I.Dat. unhappy is
‘I am unhappy.’

5.4 Conjunction (co-ordination and sub-ordination - ccof)

There are two types of conjunctions: (1) co-ordination, and (2) sub-ordination.

5.4.1 Co-ordinate Conjunction

Two or more independent clauses are conjoined with a coordinating conjunction. These clauses denote parallel events or sequential events [91]. The most commonly used coordinating conjunction is aur ‘and’ [9]. Similarly, even disjunctives come under this class. The most commonly used coordinating disjunctives are yaa, athva‘or’ [9]. The elements participating in the coordination can belong to various categories, such as, noun, adjective, adverbs etc; they can also be entire clauses, participles, etc [21]. In the example 5.15 given below, the two independent clauses are conjoined with the coordinating conjunction aur ‘and’ denote parallel events.

Ex- (5.15) (raam ne khaanaa khaayaa) aur (siitaa ne paanii piyaay)
ram Erg. food ate and sita Erg. water drank
‘Ram ate food and Sita ate an apple.’

According to Tesni`ere [171] the relations present in coordination structures form an undirected cycle. For instance, when a noun is being modified by two coordinated adjectives then there will be a (symmetric) coordination relation between the two conjuncts and there will be two (asymmetric) dependency relations between the conjuncts and the noun. According to Lombardo and Lesmo [106], “dependency paradigms exhibit obvious difficulties with coordination because, differently from most linguistic structures, it is not possible to characterize the coordination construct with a general schema involving a head and some modifiers of it”. The representation of Coordinate Structures (CSs) becomes more complicated by the following facts [142]:

(a) Coordinate Structures (CSs) have more than two conjuncts.

(b) Conjuncts have “private” modifiers of as well as shared modifiers shared by, such as “Mary came and cried”.

13
(c) Shared modifiers are coordinated: “big and cheap apples and oranges”.

(d) There are Nested (embedded) coordinations: “John and Mary or Sam and Lisa”.

(e) Many languages have comma or other punctuations that play the role of the main coordinating conjunction.

(f) The coordinating conjunction may be a multiword expression (“as well as”).

Some of the issues listed above were identified by Tesni`ere [171]. The solution given by him is that the conjuncts are connected by vertical edges directly to the head and by horizontal edges to the conjunction (which constitutes a cycle in every CS). The following are the most frequently used models out of the many different models that have been proposed [142]:

I. Mel’ˇcuk style used in the Meaning-Text Theory (MTT)

The first conjunct becomes the head of the Coordinate Structure (CS) and takes the second conjunct as its dependent, third conjunct is attached under the second one, etc. Coordinating conjunction is attached under the penultimate conjunct, and the last conjunct is attached under the conjunction [115].

II. Prague Dependency Treebank (PDT) style

“All conjuncts are attached under the coordinating conjunction (along with shared modifiers, which are distinguished by a special attribute)” [80].

In our approach, in the case of coordinating conjunction like aur ‘and’, the coordinator becomes the root/head of the sentence. It takes the two conjoined elements as children and the relation marked on the edges is ccof (conjunct of). The above example 5.15 is an example of co-ordination of two clauses. The verbs khaayaa ‘ate’ and piyaa ‘drank’, are the two conjoined elements which are conjoined by the coordinating conjunction aur ‘and’. The relation marked between the conjoined elements and the coordinating conjunction is ‘ccof’ (conjunct of). ‘cco’ is not exactly a dependency relation. It is a non-dependency relation. This tag is used for annotating coordinating as well as subordinating conjunctions. The conjunctions will be represented as heads in the dependency trees. Since, conjunctions are classified as function words, they would normally not become a node in the tree. However, they cannot be left out of the tree; they need to be represented in the tree. A function word joins two content words which are present in two different chunks. A chunk generally consists of a content word which is the head of the chunk and the other words that are related to the head word. However, since conjunctions do not reflect dependency, the labels for the arcs coming out from conjuncts do not have a dependency label. Thus, the label ‘cco’ is classified in our scheme under the non-dependency labels category.

In case of coordinating conjunctions, it will have two or more branches, i.e., two or more children, which will be annotated as 'cco'. This analysis captures the fact that neither of the conjoined elements is the head (the head of the two or more conjoined elements lies in the conjunct). Other conjunct
and punctuations which act like conjuncts are annotated similarly [21]. The dependency tree for the above example 5.15 is given below:

![Dependency Tree](image)

**Figure 5.7 Co-ordinating Conjunction**

### 5.4.2 Sub-ordinate Conjunction

Subordinating conjunctions join an independent clause with one or more dependent clauses. The dependent clauses are subordinate clauses whereas the independent clause is the main clause. The subordinate clause contains a subordinating conjunction. The subordinating conjunctions are *agar/yadi* ‘if’, *ki* ‘that’, *jo* ‘who, which’, etc. The subordinating conjunctions, except the complimentizer *ki* ‘that’, normally occur in correlative constructions [91]. The subordinate clause containing a *ki* ‘that’ conjunction is shown in the example 5.16 given below.

**Ex- (5.16)** *raam ne kahaa (ki vo kal aayegaa)*

Ram Erg. said that he tomorrow will-come

‘Ram said that he will come tomorrow.’

A subordinating conjunction would attach to the verb of the main clause and take the clause (subordinate clause) to which it is syntactically attached as its child [28]. It will have only one branch, i.e., one child. In the example 5.16 given above, *‘ki vo kal aayegaa’* ‘that he will come tomorrow’ is the sub-ordinate clause which contains the subordinating conjunction *ki* ‘that’. This clause is dependent on the main verb *kahaa* ‘said’. The subordinating conjunction *ki* ‘that’ is the head of the subordinate clause and takes *vo kal aayegaa* ‘he will come tomorrow’ as its child. The relation between them will be marked as *ccof* (conjunct of). The dependency tree for example 5.16 is given below.
In the above diagram, the complimentizer *ki* ‘that’ present in the subordinate clause attaches to the verb of the main clause, i.e., *kahaa* ‘said’. The relation between the verb of the main clause and the complimentizer is *karma* (k2). The verb of the subordinate clause, i.e., *aayegaa* ‘will come’ attaches to the complimentizer *ki*. The relation between the verb of the subordinate clause and the complimentizer is *ccof*.

### 5.5 Clausal Complements

The verbs that take object complements are: (i) communication verbs such as *kahanaa* ‘to say’, *bolnaa* ‘to speak’, *bataanaa* ‘to tell’, *puuChnaa* ‘to ask’, *likhna* ‘to write’, among others; (ii) verbs of intending, hoping, doubting, believing, deciding, etc., for example, *iraadaa karnaa* ‘to intend’, *vichaar karnaa* ‘to think’, *ummiid karnaa* ‘to hope’, among others; (iii) verbs of agreeing and accepting, such as *maannaa* ‘to agree’, *sviikaar karnaa* ‘to accept’ [91]. These verbs introduce the sentential object by a complimentizer *ki* ‘that’.

**Ex- (5.17)** *raam ne kahaa (ki vo kal aayegaa).*

Ram, Erg. said that he tomorrow will-come
‘Ram said that he will come tomorrow.’

In example 5.17, the clause ‘*ki vo kal aayegaa*’ ‘that he will come tomorrow’ is the object and since it is a clause, it becomes a clausal object. The clausal object will directly attach to the verb *kahaa* ‘said’ as *karma* (k2). The dependency tree for example 5.17 is given below:

![Dependency Tree for Example 5.17]
In Hindi, however, we also have examples such as 5.18, where we have both pronoun and a clause.

**Ex- (5.18)** \textit{raam ne yaha kahaa (ki vo kal aayegaa)}

ram Erg. this said that he tomorrow will-come

‘Ram said that he will come tomorrow.’

The above given example shows the presence of a pronoun \textit{yaha} ‘this’ and a complement clause ‘\textit{ki vo kal aayegaa}’ ‘that he will come tomorrow’. Both, pronoun and the clause mentioned above contend as the \textit{karma (k2)} of the verb \textit{kahaa} ‘said’. Since only one of them can be the \textit{karma (k2)}, we need to decide which of the two would be the \textit{karma (k2)} and how would the other be annotated?

In example 5.18, the clause ‘\textit{ki vo kal aayegaa}’ ‘that he will come tomorrow’, preceded by the subordinating conjunct \textit{ki} ‘that’, forms a single constituent with the pronominal head \textit{yaha} ‘this’ and the resulting complex phrase functions as the \textit{karma (k2)} of the verb \textit{kahaa} ‘said’. Here, this constituent is a discontinuous constituent. The clause ‘\textit{ki vo kal aayegaa}’ is the clausal complement of the pronominal head \textit{yaha} ‘this’ [91]. Since, the pronoun \textit{yaha} ‘this’ is the head of the complex phrase so it will be marked as \textit{karma (k2)} of the verb \textit{kahaa} ‘said’ and the clause ‘\textit{ki vo kal aayegaa}’ which is the complement of the \textit{karma (k2)} \textit{yaha}, attahes to \textit{yaha} as \textit{rs} (\textit{relation samanadhikaran}). Elements (normally clauses) which elaborate on a noun/pronoun are called as \textit{relation samanadhikaran} and are marked as \textit{'rs'}. \textit{samanadhikaran} is attached with one of the \textit{karakas} with which it is associated. Here the \textit{samanadhikaran} is associated with \textit{karma} so it gets attached to \textit{karma}. The dependency tree for example 5.18 is given below.

![Dependency Tree](image_url)
Thus, clausal compliments in all the verb types listed above will be annotated in a similar way.

5.6 Conditionals

The conditionals in Hindi have a correlative construction with paired conditional markers. It has two clauses; one clause beginning with agar/yadi ‘if’ and the other beginning with to/tab ‘then’. agar/yadi ‘if’ clause is the conditional clause and the to/tab ‘then’ clause is the main clause. The subordinating conjunctions agar and yadi ‘if’ are used to introduce the conditional clauses which express condition and to ‘then’ is used to introduce the main clause which express consequence. agar/yadi ‘if’ clause is a subordinating clause and generally comes first.

Ex- (5.19) [agar vaha biimaar na hotii] [to paarTii meM zaruura aatii]
    if she sick not had-been then party in definitely would-have-come
    ‘If she had not been sick then she would have definitely come to the party.’

agar ‘if’ and to ‘then’ are two connectives in example 5.19. In this sentence there are 2 clauses: [agar vaha biimaara na hotii] ‘If she had not been sick’ and [to paarTii meM zaruura aatii] ‘then she would have definitely come to the party’. Here, an issue arises that whether the conditional clause, i.e., agar/yadi ‘if’ clause is a subordinating clause or not. This issue arises because the diagnostics used for subordination does not work well for this agar/yadi ‘if’ conditional clause in Hindi. The well known
diagnostic for identifying the subordinating clause is that the subordinator or the subordinating conjunction present in the subordinating clause can’t be dropped. In English we can not drop the subordinator “if” whereas we can drop “then”. In the English examples given below, ‘then’ is present in 5.20 whereas it is dropped in the example 5.21. Example 5.22 is ungrammatical because the subordinator ‘if’ is dropped in the sentence.

Ex- (5.20) [If it rains today], [then children won’t be able to go to school].
Ex- (5.21) [If it rains today], [children won’t be able to go to school].
Ex- (5.22) *[It rains today], [then children won’t be able to go to school].

In Hindi, we can drop the subordinator agar/yadi ‘if’ present in the conditional clause. Even though agar/yadi is dropped, still it can be implied by to ‘then’ present in the second clause. The presence of to ‘then’ between the two clauses is enough to indicate that it is a conditional sentence. to ‘then’ present in the main clause can’t be dropped and is always required in Hindi. If we compare the examples 5.19 (given above) and 5.23 (given below), we can see that agar ‘if’ is present in 5.19 and it is dropped in 5.23.

Ex- (5.23) [vaha biimaar na hotii] [to paarTii meM zaruura aatii]
she sick not had-been then party in definitely would-have-come
‘If she had not been sick then she would have definitely come to the party.’

Thus, in Hindi, the question arises that if the subordinator of a subordinate clause in the conditional sentences gets dropped then should we consider the conditional clause as a subordinating clause.

We also notice that in Hindi, the conditional clause always precedes the consequential clause (main clause) whereas in English, the conditional clause and consequential clause may be interchanged. Example 5.24 given below is ungrammatical because this interchange of conditional clause and consequential clause is not allowed in Hindi. Example 5.25 is grammatical although the consequential clause has moved before the conditional clause because to ‘then’ which occurs in the consequential clause has not moved along with it.

Ex- (5.24) *[to paarTii meM zaruura aatii] [agar vaha biimaar na hotii]
then party in definitely would-have-come if she sick not had-been
‘Then she would have definitely come to the party if she had not been sick.’

Ex- (5.25) [paarTii meM zaruura aatii] [agar vaha biimaar na hotii] to
party in definitely would-have-come if she sick not had-been then
‘Then she would have definitely come to the party if she had not been sick.’
We can interchange the conditional clause and consequential clause in English. The order of conditional clause and consequential clause given in the example 5.26 (example 5.20 is repeated) is interchanged in the example 5.27.

Ex- (5.26) [If it rains today], [then children won’t be able to go to school].

Ex- (5.27) [Children won’t be able to go to school] [if it rains today].

Since the above mentioned diagnostic to identify the subordinate clause doesn’t seem to work well in Hindi so it opens up the possibility of treating this conditional with paired connectives, where both clauses are placed at same level.

A. Possibility-I: Treat ‘agar-to’ as a complex conjunct which is an Abstract node.

Thus an abstract node [agar-to] ‘if-then’ has to be posited which will have two branches and the branches will have agar ‘if’ and to ‘then’ as their root nodes and they subsequently will have attachments with their clauses. The representation is given below in figure 5.11.

![Figure 5.11 Conditionals Possibility-I](image-url)
The problem with the above representation is that it has an abstract node [agar-to] ‘if-then’ which is not desirable in a dependency tree. So Possibility-I is not possible because in the above figure 5.11, [agar-to] ‘if-then’ is the head of the tree which is an abstract node, i.e. it is not a lexical node.

Although, syntactically, the subordinator agar ‘if’ is dropped in Hindi, semantically, agar ‘if’ clause is the subordinate clause. Also, agar ‘if’ clause can’t occur on its own so it needs another clause to complete the sentence. Thus the other possibility of representing conditionals is Possibility-II.

B. Possibility-II: One clause depends on the other clause

In Possibility-II, the agar ‘if’ clause is dependent on the to ‘then’ clause, i.e., agar clause is the subordinate clause and to clause is the main clause. In the tree representation, the to clause occurs at the top of the tree and agar clause occurs at the bottom of the tree, i.e., the agar clause attches to the to clause. Both agar ‘if’ and to ‘then’ will be the heads of their respective clauses. The verb aatii ‘would have come’ occurring in the to clause will attach to to ‘then’ by ccof relation. Similarly, the verb naa hotii ‘not had been’ occurring in the agar clause attaches to agar by ccof relation. The relation between agar clause and to clause is captured by the vmod relation, i.e., the agar clause attaches to the verb aatii ‘would have come’ of the to clause by vmod relation. The tree representation is given below in figure 5.12.

![Diagram](image-url)

**Figure 5.12** Conditionals Possibility-II
For conditionals, our current decision is Possibility-II, i.e., we are following Possibility-II as Possibility-I is not possible.

5.7 Handling Ellipsis

When one or more element(s) from a sentence are dropped, it is called ellipsis. Hindi is a pro-drop language and it shows a high degree of ellipsis. An issue that comes while annotating sentences in a treebank are whether these missing elements have to be posited for annotation or not? Insertion of missing nodes adds complexity to the task both for the annotators and also for machine learning. Thus, normally it is not preferred. However, non-inclusion of this in the scheme poses a problem for annotating sentences, such as the following:

5.7.1 Missing Coordination

Ex- (5.28) [bachche ba.De ho gaye haiM ] (aur) [kisii kii baat nahiiM sunate]

children big happen is and someone Gen. matter not listen

‘The children have grown up, they don't listen to anyone.’ [28]

The two clauses in the above example 5.28 can be annotated separately as dependency trees. However, since the coordinating conjunct (aur) ‘and’ is missing here, their relation to each other cannot be captured and also this example will then end up as a forest rather than a tree. Thus, to handle such cases, we decided to insert those missing elements which form the root of a tree or a subtree and whose absence leads to a forest. The null elements, in our scheme, are posited as a special token. NULL_NP (Null Noun Phrase), NULL_VG (Null Verb Phrase), Null_CCP (Null Conjunction Phrase), etc., are chunks which mark different kinds of cases of ellipses [28]. In the above example 5.28, there is no explicit conjunct, so we insert a NULL element, i.e., NULL__CCP for aur ‘and’ (missing elements are enclosed in brackets to identify them in the examples and figures) to show the dependencies. Both the clauses, i.e., ‘bachche ba.De ho gaye haiM’ ‘children have grown’ and ‘kisii kii baat nahiiM sunate’ ‘don’t listen to anyone’ depend on the parent node aur. We insert a NULL node only if it is essential to represent the dependencies, i.e., we don’t insert a NULL for a child node and insert only for the parent nodes to show the dependencies.
There are other instances in Hindi (see example 5.29 below) where the absence of a token leads to forests rather than trees.

### 5.7.2 Missing Pronoun

Apart from the ellipses of coordinating conjunct, Hindi also has several cases of empty pronouns which often have modifiers. For example:

**Ex- (5.29)**

\[
\text{tum \ jo bhi kahoge (vo) maiM maan luuMgii}
\]

you whatever will say (that) I will-believe

‘I will believe whatever you say.’

In the above example 5.29, vo ‘that’ is missing which becomes the parent node for the relative clause ‘\text{tum jo bhi kahoge}’ ‘Whatever you say’. So this is an example of elision of head in head-modifier relation. vo ‘that’ is the head and the relative clause ‘\text{tum jo bhi kahoge}’ ‘whatever you say’ is its modifier. Here the issue is how do we show dependency when the head is missing? To show the dependency relation between vo and the relative clause we insert a NULL_NP for vo and attach the relative clause ‘\text{tum jo bhi kahoge}’ to NULL_NP.
The naming convention for NULL chunk is that if an NP (Noun Phrase) is missing which is to be inserted then we say NULL__NP [28].

### 5.7.3 Gapping

Ross [151] introduced the term “Gapping” which refers to the omission of the finite verb in the sentence. It is a type of ellipses where a verb is deleted in its repeat occurrences. In some instances, the arguments of the verb may also be deleted along with the verb [28]. It occurs only in coordinate structures [151]. The example 5.30 given below is the case of gapping.

**Ex- (5.30)** *raam ne siitaak kitaab dii aur aatif ne Tiinaak ko (kitaab) (dii)* [28]

```
ram  Erg. sita  Dat. book  gave  and  aatif  Erg. tina  Dat. (book)  (gave)
```

‘Ram gave a book to Sita and Atif to Tina.’

In the above example 5.30, an argument and a verb is dropped in the second clause. The missing argument is *kitaab* ’book’ and the missing verb is *dii* ‘gave’. This example is a case of gapping where missed elements can be fetched from the previous clause. Here *kitaab* is a child node so we are not
inserting a NULL for it, whereas the verb \textit{dii} is parent node so we need to insert a NULL for it. At the same time, verb \textit{dii} ‘gave’ is also a child node in the example 5.30. The dependency tree is as follows:

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{dependency_tree.png}
\caption{Dependency tree of the sentence ‘Ram and Sita gave book to Atif and Tina’}
\end{figure}

Thus, the chunk annotated for the gapped element, i.e., \textit{dii} ‘gave’ in the above example would look as follows:

```
(( NULL_VGF
   NULL
 ))
```

5.8 Complex Predicates

5.8.1 Conjunct Verbs

Conjunct verbs have the following structure [9]:

\[ \text{Noun/Adjective} \ + \ \text{Verb (Verbalizer)} \]

The most common \textit{verbalizers} in Hindi are \textit{karna} ‘to do’, \textit{honaa} ‘to be’, \textit{denaa} ‘to give’, \textit{lenaa} ‘to take’, \textit{aanaa} ‘to come’, etc. [28]. Identifying a conjunct verb is a very challenging task in Indian languages. They have been exhaustively studied by the researchers in the past. Some notable efforts in this direction have been Greaves [73]; Kellogg [96]; Mohanan [123]; Butt [52]. One of the recent study on Complex
Identifying a conjunct verb is a tough task and then marking a dependency relation between the noun/adjective and verb of the conjunct verb is again a major task [19]. The example below shows a $N+V$ conjunct verb usage:

Ex- (5.31) maine usase eka prashna kiyaa
     I-Erg him-Abl one question did'
     ’I asked him a question.’

In the above example 5.31, prashna kiyaa ‘questioned’ is a conjunct verb and behaves as a single semantic unit. Ideally, the noun/adjective + verb sequence of the conjunct verb is placed in one chunk. Keeping this in mind, example 5.31 above is problematic. Only the noun prashna ‘question’ within the conjunct verb sequence prashna kiyaa is being modified by the adjective eka ‘one’ and not the entire noun+verb sequence. The annotation scheme should be able to account for this relation in the dependency tree. If prashna kiyaa is grouped as a single verb chunk, it will not be possible to mark the appropriate relation between eka and prashna. To overcome this problem, it is proposed to break eka prashna kiyaa into two separate chunks, [eka prashna]/NP and [kiyaa]/VG. In the current scheme of annotation we are using a special tag ‘pof’ to mark the conjunct verbs. The dependency relation of prashna with kiyaa will be POF (‘Part OF’ relation), i.e. the noun or an adjective in the conjunct verb sequence will have a POF relation with the verb. ‘POF’ is again not a dependency relation though there is a parent child relation between them. POF is just a tag or label to mark the arc between the noun and verb of conjunct verbs. We instead use the ‘pof’ tag to mark a relation between two elements and show that these two elements form a multi word expression (MWE) [28]. This way, the relation between eka and prashna becomes an intra-chunk relation as they will now become part of a single NP chunk. What makes such a sequence unique is the fact that the components which make up a conjunct verb are chunked separately, but semantically they constitute a single unit. The proposed scheme has the following advantages [21, 28]:

a) It captures the fact that the noun+verb sequence is a conjunct verb by linking them with an appropriate tag, i.e., POF (Part OF relation); this information is extremely crucial, syntactically.

b) It allows us to deal with the modifier-modified relation between an adjective and its modified noun, as in example 5.31, which is a frequent phenomenon.

The dependency tree for the above conjunct verb example is given below:
The tree above shows the proposed solution, where the adjective *eka* ‘one’ modifies the noun *prashna* ‘question’ instead of the entire *prashna kiyaa* ‘questioned’, which would have been the case, had we not separated *prashna kiyaa* into two separate chunks.

Only verb is considered as the syntactic head of the conjunct verb since noun alone is incomplete unless it joins with the verb and verb carries more meaning and also tense and aspect information. If we consider noun as the head of the conjunct verb then the verb and the modifier of the noun if any attach to that noun and rest of the nouns of the sentence attach to the verb.

### 5.9 Discontiguous Elements

There are certain Hindi sentences in which a postposition, a negative particle or an auxiliary are separated from the NP or VP of which they are a part. They don’t occur in the chunk to which they actually belong. In the example 5.32 given below, the NP chunk ‘*bhaakapaa ke*’ ‘of BKP’ is broken into two chunks by (*maaovaadiii*) ‘maoist’ which is adding extra information about *bhaakapaa* ‘BKP’. The noun (*maaovaadiii*) ‘maoists’ forms a separate NP chunk. Therefore, the whole expression ‘*bhaakapaa (maaovaadiii) ke*’ will be divided into three different chunks, i.e., *bhaakapaa* ‘BKP’ as one chunk, (*maaovaadiii*) ‘maoists’ as second chunk and *ke* ‘of’ as third chunk:

**Ex- (5.32)** *bhaakapaa (maaovaadiii) ke raamabachana yaadava ko girafataar kara liyaa*

BKP (maoist) of rambacana yadav Acc. arrest do reflx-perf

gayaa
go-perf

‘Apart from this, Rambacana Yadav of BKP (Maoist) was arrested.’ [28]
The NP chunk ‘bhaakapaa ke’ ‘of BKP’ is broken into two chunks. The postposition ‘ke’ ‘of’ is separated from its noun ‘bhaakapaa’ ‘BKP’ and is chunked as ‘FRAGP’, i.e., the separated part of the chunk will be chunked as ‘FRAGP’. To show or to capture the information that the post position ‘ke’ is part of the noun chunk ‘bhaakapaa’, the postposition chunk will be attached to the chunk ‘bhaakapaa’, by the ‘fragof’ relation [28]. The non-dependency relation tag ‘fragof’ is used to mark the relation between two members of the same chunk. ‘fragof’ occurs in those cases where an element is inserted between the parts of a single chunk or some times when the main part of the chunk is dropped. Therefore, the expression ‘bhaakapaa (maaovaadii) ke’ would appear as follows in chunks after annotating the value ‘fragof’ for the attribute ‘drel’:

```
(( bhaakapaa NP 
  (( NP 
    ( SYM maaovaadii NN 
    ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
  ) 
```
There are many more constructions in Hindi which are linguistically very interesting but I have constrained myself to discuss only the above constructions here.

So far we have discussed the annotation scheme and the annotation procedure of annotating dependency analysis of various types of Hindi constructions. This prepares us towards developing larger treebanks. However, the task of treebank annotation also requires certain other linguistic resources which help in the smooth and consistent annotation of the treebank. One such resource is verb frames. As part of this work, thus, a resource of verb frames for some Hindi verbs were also created. Next chapters provide a detailed description of this resource which also includes verb frames for causatives and conjunct verbs in Hindi.