Chapter 6

Developing Verb Frames for Hindi

A verb frame captures various syntactic distributions in which a verb can be expected to occur in a language. In verb frames, we capture the argument structure of Hindi verbs (for various senses of the verbs). We have also classified the Hindi verbs based on their argument structure. The main objective of this work is to create a linguistic resource which would help the annotators in the annotation of the dependency relations for various verbs. The resource could also prove to be useful in parsing and for other NLP applications. We also hope that this study and resource thus created, would be helpful to the scholars interested in the linguistic study of the Hindi verbs. Two well known studies for English in this direction are VerbNet [101, 100] and Propbank [139]. VerbNet [101, 100] represents the verb argument relations by theta roles and Propbank represents the verb argument relations by Arg0, Arg1, Arg2, etc., depending on the verb. In our study of Hindi verbs, we capture verb argument relations using the dependency relations (mainly karaka relations and also other than karaka relations) from Paninian grammar framework. In the subsequent sections of this chapter, we describe the methodology used in preparing these frames and the criteria followed for classifying Hindi verbs [22]. The study presented here is already published as “Developing Verb Frames for Hindi” [22] in Language Resources and Evaluation Conference (LREC), 2008.

6.1 Verb Frames

Verb plays a critical role in the analysis of a sentence within Paninian Grammatical framework. Thus, information about a verb’s syntactic and semantic behaviour plays an important role both in dependency annotation as well as while parsing. Therefore, studying Hindi verbs and their nature formed a crucial part of the current study.

Thus, the motivation for developing verb frames is:

1) To create a linguistic resource which gives the classification of Hindi verbs.

2) It is helpful for the annotators in deciding various dependency relations for a given verb in the corpus.

3) It is also helpful in preparing demand chart for the Hindi parser [84, 25].

4) It forms a basis for linguistic analysis.
Verb frames are developed following Paninian Grammatical framework. Verb frames provide us the arguments that a particular verb can take in a particular sense, i.e., they show mandatory karaka relations for a verb.

6.1 Related Works

6.1.1 Levin’s verb classes

Levin’s verb classes [105] is an elaborate attempt to investigate English verbs. Drawing from earlier works dedicated to such an investigation, Levin has shown the correlations between the semantic and syntactic behavior of English verbs. Beth Levin's work on “English Verb Classes and Alternations” gives both the syntactic and semantic information. It forms a powerful research tool. The behavior of a verb is to a large extent determined by its meaning. The verb behavior can be used to get an insight of the linguistically relevant aspects of the verb meaning [227].

I. If the members of a set of verbs S share some meaning component M, then the members of S can be expected to exhibit the same syntactic behavior(s), and

II. If the members of a set of verbs S exhibit the same syntactic behavior(s), then the members of S can be expected to share some meaning component(s).

Levin [105] has two major sections. The first section includes a wide variety of syntactic alternations exhibited by the English verbs. The second part contains various syntactically relevant, semantically coherent verb classes of English. For example, the verbs belonging to the class of break verbs participate in Causative/Inchoative Alternation, whereas verbs belonging to the class of cut verbs don’t participate in this Alternation.

- **Causative/Inchoative Alternation**: This alternation involves verbs with transitive and intransitive uses, where the transitive use of the verb V means roughly “cause to V-intransitive”.

Ex-(6.1) (a) *The cup broke.* (inchoative variant)  
(b) *The boy broke the cup.* (causative variant)

Ex- (6.2) (a) *The bread cut.* (inchoative variant)  
(b) *Margaret cut the bread.* (causative variant)
- **Break Verbs:** break, chip, crack, fracture, rip, split
- **Cut Verbs:** cut, clip, hack, saw, scrape, scratch

Using such an alternation one is able to sub-classify a general class into subclasses which can have some semantic nuance, or they may be placed into different subclass purely due to syntactic difference.

### 6.1.1.2 VerbNet (VN)

VerbNet (VN) [101,100] is an on-line verb lexicon for English. It is a hierarchical, domain-independent; broad-coverage verb lexicon which extends Levin’s verb classes [105] and provides the syntactic and semantic information for English verbs. It is an on-line lexicon which has been mapped to other major language resources such as Wordnet [117], FrameNet [16] and PropBank [139]. VN has more than 5,200 verbs and 237 verb classes [101, 100]. VN is organized into verb classes extending Levin classes [105]. Each verb class in VN is described by thematic roles, selectional restrictions on the arguments, and syntactic frames which consist of syntactic description and semantic predicates. Syntactic frames show the possible surface realizations of the argument structure for constructions such as transitive, intransitive, prepositional phrases, resultatives, and a large set of diathesis alternations. Semantic restrictions such as animate, human and organization are used to constrain the types of thematic roles allowed by the arguments. Each frame is associated with explicit semantic information. An example of the verb *eat*, is taken to show its frames in VN:

**Frames:**

**a) Basic Transitive**

Ex-(6.3): "Cynthia ate the peach."

*Syntax:* Agent V Patient

*Semantics:* take_in(during(E), Agent, Patient)

**b) Conative**

Ex-(6.4): "Cynthia ate at the peach."

*Syntax:* Agent V {at} Patient

*Semantics:* take_in(during(E), Agent, Patient)
6.1.1.3 PropBank (PB)

PropBank (PB) [139] is a corpus, annotated with verbal propositions and their arguments. It has recently been extensively used for the semantic role labeling task [CoNLL shared task 2004-05\(^1\), 228, 229]. PB adds a layer of semantic annotation atop the syntactic structures. PB represents the verb argument relations by Arg0, Arg1, Arg2, etc., depending on the valency of the verb [97]. Each set of argument labels and their definitions is called a frameset. The frameset provides a unique identifier for the verb sense, a meaning for that verb sense, and the set of expected arguments giving argument numbers. An example is given below for the verb send:

```
send.01 sense: give
roles: Arg0: sender
    Arg1: thing sent
    Arg2: sent-to
```

Ex-(6.5): They send letters to the card holders.

```
Arg0: They
Rel: send
Arg1: letters
Arg2: to the card holders
```

6.1.1.4 FrameNet (FN)

FrameNet (FN) is an on-line lexical resource for English, based on frame semantics and supported by corpus evidence. FrameNet groups words according to the conceptual structures, i.e., frames that underlie them [16]. This lexical resource is called as FrameNet database. It has three major components [16]:

1) **Lexicon:** It contains entries composed of: (a)Some conventional dictionary-type data, (b)Formulas, (c)Links to semantically annotated example sentences, (d)Links to the FRAME DATABASE and to other machine-readable resources such as WordNet and COMLEX.

---

\(^1\) http://www.lsi.upc.edu/~srlconll/
2) **Frame Database**: It contains descriptions of each frame’s basic conceptual structure and provides names and descriptions for the elements participating in such structures. One of the entries in this database is given below:

- frame (TRANSPORTATION)
- frame_elements (MOVER(S), MEANS, PATH)
- scene (MOVER(S) move along PATH by MEANS)

3) **Annotated Example Sentences**: They are marked up to exemplify the semantic and morphosyntactic properties of the lexical items. Example of a Frame Element Group and an annotated sentence is given below:

Ex- (6.6) [D I] drove [P eastwards along Lake Geneva].  
\( D \rightarrow Driver, P \rightarrow Path \)

All these resources have been extensively used for various NLP applications in English and have proved to be very useful in improving the state of the art for many of these applications. However, there have been hardly any attempts for most of the other languages.

As mentioned above, in the current thesis, Hindi verbs were studied within the Paninian Grammatical framework. More details of the same are given below:

### 6.1.2 Methodology

Methodology was to study Hindi verbs from the corpus and to see the distribution of Hindi verbs in the corpus.

For doing this we used the following resources:
(a) Levin’s verb classes [105]

(b) A Hindi corpus^2
   (i) Raw
   (ii) Dependency annotated

(c) Hindi Wordnet^3 (HWN) [89]

(d) Sahay’s verb classes [154]

The methodology used in developing verb frames is defined in following steps:

A. Selection of the Hindi verbs
B. Extraction and Analysis of the sentences
C. Preparation of Hindi verb frames

6.1.2.1 Selection of the Hindi Verbs

Verbs were selected mostly based on their high frequency and common usage in the language. Selection of the Hindi verbs was done as follows:

(i) Initially, few high frequency verbs were taken (55 verbs).

(ii) Later, most commonly used verbs were extracted from the dependency annotated data of HyDTB.

   ➢ 188 unique verbs.
   ➢ 173 kriyamula and phrasal verbs.

(iii) 250 most frequently used verbs were also taken from a raw Hindi Corpus (75,000 sentences)

(iv) Also referred the verbs from previously prepared verb frames (110 verbs).

---

^2 We use the CIIL (Central Institute for Indian languages) corpus.
^3 Developed by the wordnet team at IIT Bombay, http://www.cfilt.iitb.ac.in/webhwn
Totally, verb frames were created for 687 verbs. I followed a mixed approach in the selection of verbs. These 687 verbs were selected on the following basis: (i) complex nature, (ii) showing interesting patterns, (iii) focus of study in literature, and (iv) rest of them were selected due to their high frequency occurrence in the language. I have created verb frames mainly for the simple verbs and for very few multi-word expressions.

6.1.2.2 Extraction and Analysis of the Sentences

Sentences were extracted from the treebank data as well as from raw Hindi corpus for the above selected verbs, using an extraction tool. After extracting the sentences, each verb and its extracted sentences were studied. First, I studied all the sentences that were extracted for each verb to capture the various senses of a verb and then I analysed the sentences with the same verb sense for each verb sense to capture the multiple frames. By multiple frames, I mean that a verb can take different set of mandatory dependency relations (karaka relations and other than karaka relations) for the same sense of a verb. For example, the verb bheja with the sense ‘send’ has two frames:

Ex- (6.7) maataa-pitaa bachchoM ko skual bhejate haiM
mother-father children Acc. school send be-Pres
‘Parents send their children to school.’

Ex- (6.8) raam sitaa ko saMdesh bhejate haiM
ram sita Acc. message send be-Pres
‘Ram sends message to Sita.’

In the above examples 6.7 and 6.8, the verb bheja has the same sense, i.e., ‘send’. Though they have same sense, but they differ in frames. In the example 6.7, the verb bheja is taking the following dependency relations for its mandatory arguments: karta (k1), karma (k2), and goal (k2p)⁴. In the example 6.8, the verb bheja ‘sense’ is taking the following dependency relations for its mandatory arguments: karta (k1), sampradana (k4), and karma (k2). We can see that there is a difference in the set of dependency relations of the mandatory arguments taken by verb bheja ‘send’ in the above two examples. Thus, they are considered as different frames. Therefore, we can say that the same sense of a verb can take multiple frames. This phenomenon of multiple frames occurs in the case of very few verbs. At the same time, the frames of different senses of a verb also differ. For example, the two senses of the verb aa, i.e., ‘come’ and ‘know’ have different frames:

---
⁴ Goal dependency relation is marked for the nouns denoting goal or destination. Goal is represented by k2p tag.
The senses of the verb *aa* in the above examples 6.9 and 6.10 are ‘come’ and ‘know’ respectively. The verb has different senses as well as different frames in the above examples. In example 6.9, the verb *aa* has the sense ‘come’ and has the following dependency relations for its mandatory arguments: *karta* (*k1*), and *goal* (*k2p*). In the example 6.10, the verb *aa* has the sense ‘know’ and has the following dependency relations for its mandatory arguments: *anubhavkarta* (*k4a*), and *karta* (*k1*). We can see that there is a difference in the set of dependency relations of the mandatory arguments taken by verb *aa* having two different senses in the above two examples. Therefore, with the change in the sense of the verb there is also a change in the frame of the verb, but this is not the case always, i.e., frames can be the same for different senses of a verb.

In the verb frames, along with the mandatory arguments of a verb, I have also captured the arguments which are mostly not present on the surface level of the sentence but are implicit, i.e., *desirable* arguments. An argument which is *mandatory* is compulsory to be present in the sentence as it is obligatory to fulfill the meaning of the verb, without which the meaning of the verb is incomplete. *Desirable* argument is also required to fulfill the meaning of the verb but it is not as compulsory as the *mandatory* argument to be present in the sentence. Even if the *desirable* argument is absent on the surface level, the meaning is conveyed implicitly from the context. The meaning of the verb will be still complete in its absence. We have considered *mandatory* argument as *strong* argument and *desirable* argument as *weak* argument. *Desirable* arguments are *weak* in the sense they can be dropped when compared to *mandatory*. The expectation level in case of *mandatory* arguments is very high whereas the expectation level in *desirable* arguments is very low.

For example, the verb *kaaT* having the sense ‘cut’ takes two mandatory arguments, i.e., *karta*(*k1*) and *karma* (*k2*). *Karta* is the participant who is doing the action of cutting and *karma* is the element that is being cut. *kaaT* ‘cut’ also includes the *instrument* that is used in the action of cutting. The *instrument* argument is considered as an argument which is not as important as the *mandatory* argument to be present in the sentence. For example, *chaaku u* ‘knife’ is the instrument used in the action of *kaaT* ‘cut’, so it becomes the desirable argument in the example 6.11 given below. The dependency relation of the *chaaku u* is *karana* (*k3*).
Ex- (6.11) raam ne chaakuu se (k3) seba kaaTaa
         ram  Erg.  knife  with  apple  cut
         ‘Ram cut an apple with a knife.’

I have also gathered the various vibhaktis (Hindi postpositions/ Hindi case markers) that occur with mandatory and desirable arguments of a verb. While doing this, the TAM (tense, aspect and modality) should be constant in all the sentences as the change in TAM causes change in vibhakti too. Thus, here our focus is to merge the vibhaktis that a particular dependency relation of a particular verb sense takes with a particular TAM. For example, the verb khol having the sense ‘open’ takes two different vibhaktis with karma (k2) with present imperfect TAM taa_hai:

Ex- (6.12) paMDit har saal maMdir kaa darvaazaa (k2) kholte haiM
         priest  every  year  temple  of  door  open. Impf.  be-Pres
         ‘The priest opens the door of the temple every year.’

Ex- (6.13) paMDit har saal maMdir ke darvaaze ko(k2) kholte haiM
         priest  every  year  temple  of  door  Acc.  open. Impf.  be-Pres
         ‘The priest opens the door of the temple every year.’

The noun darvaazaa ‘door’ which is karma (k2) takes two different vibhaktis, i.e., 0 and ko respectively, in the above examples 6.12 and 6.13. In the above examples, the noun is same whereas the vibhaktis are different, i.e., the noun darvaazaa ‘door’ remains the same with the variation in the vibhaktis. However, we also have cases where the noun also changes with variation in the vibhakti for the same dependency relation. For example, the verb aa having the sense ‘come’ takes two different nouns and these two nouns take two different vibhaktis for the dependency relation goal (k2p):

Ex- (6.14) raam chara baje ghar (k2p) aataa hai
         ram  four  o’clock  house  come. Impf  be-Pres
         ‘Ram comes home at 4 o’clock.’

Ex- (6.15) raam chara baje Chat par (k2p) aataa hai
         ram  four  o’clock  roof  on  comes  be-Pres
         ‘Ram comes to the terrace at 4 o’clock.’

In the above example 6.14 and 6.15, the nouns are different on which the dependency relation goal (k2p) is marked and they both are taking different vibhaktis. In the example 6.14, goal (k2p) is taking 0 vibhakti whereas in the example 6.15, goal (k2p) is taking par ‘on’ vibhakti. While merging the vibhaktis that nouns take for a particular dependency relation, the nouns can be same as darvaazaa ‘door’ in the examples 6.12 and 6.13, and nouns can be different as ghar ‘house’ and Chat ‘terrace’ in the examples
6.14 and 6.15 respectively. Hence, in this manner, I have merged all the vibhaktis that a particular dependency relation of a particular verb sense takes with a particular TAM.

I have also captured the POS (part of speech) category of the mandatory and desirable arguments of the verb. The POS category of most of the arguments of the verbs is noun and in very rare cases it is adjectives and clauses too.

6.1.2.3 Preparation of Verb Frames

Given a verb, its senses were taken from Hindi Wordnet, dictionary, etc. Then for each sense, example sentences were taken from the corpus. Verb frames were created for different senses of a verb. Verb frames mainly contain the dependency relations of the mandatory and desirable arguments taken by a verb. For each sense of a verb, multiple frames were created if there were any in case of very few verbs. All the vibhaktis taken by a particular dependency relation of a particular verb sense with a particular TAM were merged together within the verb frame. I have also included the theta role information and POS category information of the verb arguments.

A detailed discussion of the verb frames and procedure followed in creating the verb frames is provided in the sections given below.

6.1.3 Verb entry

The corpus is consulted to get the syntactic distribution in which the verb occurs and the Hindi Wordnet (HWN) is referred to get the required sense information. We define verb frames in terms of dependency relations, i.e., karaka relations [36] and other than karaka relations. The following information is given for each verb entry:

a) Description of the verb

b) Verb Frame

The above two pieces of information are provided in a data file which is referred to as a verb entry. In the verb entry there is a tabular form which is actually a verb frame but we refer to the whole data, i.e., the verb entry (both ‘description of the verb’ and ‘verb frame’) as verb frame when we talk about it.
6.1.3.1 **Description of the Verb**

For each verb we create a verb entry which contains the following description. In the description, we give the following information:

- Name of the verb,
- Its sense id (represented as SID; an id is given according to the number of senses a verb has),
- Hindi Wordnet (HWN) sense id (optional),
- English gloss,
- Example sentence of the verb,
- Theta roles (if necessary),
- Frame id, and
- the *verb frame* (given in a tabular form).

Given below is an example of a verb entry\(^5\) along with the verb frame:

---

\(^5\) Basically we are referring to the verb entry for the file that we have created for entering the data for each verb. But we mostly use the term *verb frame* for *verb entry* also.
In the figure 6.1 given above, the name of the verb is *aa* ‘come’. **SID** stands for sense id and it is represented as *aa% VI% S1*. Sense *id* is a unique id for that particular sense of the verb by which it is identified. In **SID**, I have captured the following information: (i) name of the verb; (ii) type of the verb, and (iii) sense number; all three separated by a percentage symbol. In **SID** given here, *aa* ‘come’ is the name of the verb; (*VI*) stands for verb intransitive which means that the type of the verb is intransitive; and *S1* means sense 1 of the verb. In the **verb frames**, we are using 4 types of verbs: *VI → Intransitive Verb; VT → Transitive Verb; VDT → Ditransitive Verb; VCAUS → Causative Verb*. For numbering the
senses of a verb, we are using the following convention: \textit{S1, S2,...., etc.}, which means \textit{sense 1} and \textit{sense 2}, etc., depending on the number of senses that are associated with a given verb. In the \textit{Verb_Sense} field, I have mapped the sense of a given verb with the senses given in Hindi Wordnet (HWN). Here the sense of the verb \textit{aa} ‘come’ matches with the first sense given in Hindi Wordnet. This is captured with the representation \textit{HWN (1)}. The sense numbers in Hindi Wordnet are not fixed, they keep changing so we are not relying on this information. \textit{Eng_Gloss} field stands for English gloss. Here ‘come’ is the gloss of the verb \textit{aa}.

\textit{Frames} section deals with various \textit{verb frames} which a given verb can have for a particular sense. \textit{Frame_Name_1, Frame_Name_2, ...., etc.}, is the naming convention used for numbering the \textit{verb frames} which means \textit{verb frame 1} and \textit{verb frame 2}, etc., depending on the number of \textit{verb frames} that are associated with a given verb with a particular sense. \textit{Example} field contains the Hindi example sentence for that particular sense of the verb. I have also marked theta roles as additional information to make the resource more richer. \textit{Theta_Roles} field gives the theta roles of the verb arguments. \textit{Theta roles} are taken from the \textit{Verbnet} \cite{101,100}. I hope such a mapping is going to help us make the transition from the dependency level (karaka level (syntactico-semantic) as well as non-karaka) to \textit{thematic} level (semantic) easier. It helps in making certain generalizations.

\textit{Demand_Frame} and \textit{FRAME_ID} field give an unique id to the \textit{verb frame} of a given verb for a particular sense. \textit{Verb frame} (given in a tabular form) contains the verb argument information of the verb. \textit{Frame_Name, Demand_Frame} and \textit{FRAME_ID} all deal with \textit{verb frame numbers} and \textit{verb frame ids}. \textit{Frame_Name_1} says that it is the first frame for that verb. \textit{Demand_Frame} and \textit{FRAME_ID} give an unique id to that particular \textit{Frame_Name}. \textit{Demand_Frame} field contains frame id for that particular frame which is represented as \textit{FRAME_ID_1} which means it is \textit{verb frame 1}. Again we have the field of \textit{FRAME_ID} which contains a unique id which is a combination of \textit{verb sense id} and \textit{frame id}. This unique id will help us in identifying a particular frame of a verb having a particular sense. Here \textit{aa\% VI\% S1\% FID1} is the \textit{verb frame id} of the verb \textit{aa} ‘come’. Frame id is made up of the sense id of the verb, i.e., \textit{aa\% VI\% S1\%} plus \textit{verb frame number}, i.e., \textit{FID1 (FRAME_ID_1 )} both separated by a percentage sign. A verb will have a unique sense id and within that sense it will have unique frame id because a verb can have multiple senses and each sense can have multiple \textit{verb frames}, i.e., syntactic distributions. So sense id helps us in identifying the sense of the verb and verb frame id helps us in identifying the \textit{verb frame} of that particular sense of a given verb. The difference between \textit{Frame_Name}, \textit{Demand_Frame} and \textit{FRAME_ID} is:

\begin{enumerate}
\item \textit{Frame_Name} is just a naming convention for frames, for e.g., \textit{Frame_Name_1, Frame_Name_2};
\item \textit{Demand_Frame} gives an id to the frame, for e.g., \textit{FRAME_ID_1, FRAME_ID_2} ;
\item \textit{FRAME_ID} gives an id which is a combination of both \textit{verb sense id} and \textit{frame id}. For example: 
\begin{enumerate}
\item \textit{verb \% VI\% S1\% FID1},
\item \textit{verb\% VI\% S1\% FID2},
\item \textit{verb\% VI\% S2\% FID1},
\item \textit{verb\% VI\% S2\% FID2};
\end{enumerate}
\end{enumerate}

Here in the example, the first two frame ids denote that there are two \textit{frames}, \textit{FID1} and \textit{FID2} belonging to the same sense \textit{S1}. The last two, frame ids denote that there are two \textit{frames}, \textit{FID1} and \textit{FID2} belonging to the same sense \textit{S2}.

13
Verbs_in_Same_Class field lists the verbs that have same meaning as the given verb. This is to see, if the verbs having the same meaning as the given verb, have the same frame as the given verb. Here we list the verbs having the same meaning as the given verb and also point out the differences between these verbs and the verb given. The differences can be in the terms of arguments, vibhaktis, number of frames, etc. Here the verbs pahuMcha ‘reach’ and padhaara ‘arrive’ have same sense and same verb frames as the verb aa ‘come’. Here the verb padhaara ‘arrive’ has been listed along with its FRAME_ID, i.e., padhaara%VI%S1%FID1 which means verb frame for the verb padhaara ‘arrive’ exists. Verb frame for the verb padhaara ‘arrive’, whose sense and verb frame matches with verb aa ‘come’, has been created. When we say verb frame has been created for a verb, we mean that a verb entry has been created for the verb which contains description of the verb and verb frame (given in tabular form). If a verb that has same meaning as the given verb and if it is listed without any FRAME_ID then there are two possibilities in which it can be interpreted: (i) verb entry has not been created at all for that particular verb or, (ii) verb entry has been created for that verb and somehow not listed in the Verbs_in_Same_Class field of the verb with which it matches in the sense. Here verb pahuMcha ‘arrive’ has no FRAME_ID because verb entry has not been created at all for this verb. By listing a verb under Verbs_in_Same_Class field, whose verb entry has not been created, I am populating a verb frame for that verb. In Verb frames populated field within the Verbs_in_Same_Class field, I am mentioning about the verbs whose verb frames are populated.

By populating a verb frame for a verb, I mean that instead of creating a separate verb entry for that verb, I just list it in another verb entry with which it matches in sense, saying that this populated verb has the same verb frame (tabular form) as the verb in whose verb entry it has been listed. The process of populating verb frames lessens the effort of creating verb entries for all the verbs.

There can be minute differences in verb frames between populated verb (pahuMcha ‘reach’) and the main verb (aa ‘come’) which will be mentioned in the data. They might have same dependency relations but may differ in vibhaktis, POS categories, no. of verb frames, etc.

6.1.3.2 Verb Frame (tabular form)

The actual verb frame is the table given in the verb entry. A verb frame shows the following information:

(a) karaka relations (verb-argument relations)

(b) necessity of the argument, i.e., whether an argument is mandatory (m) or desirable (d).

(c) vibhakti (Hindi postpositions/case markers taken by the arguments)

Desirable arguments are required by the semantics of the verb but they are weak compared to obligatory ones, in a sense that one can omit them without breaking down the communication, they can generally be extracted out of the context; e.g.

- He cuts the apple vs. He cuts the apple with a knife

The semantics of verb ‘cut’ implicitly requires an instrument and even if the instrument is not explicitly mentioned in the sentence, one is able to communicate. Note that this is very different from obligatory arguments and optional arguments.
(d) lexical category of the arguments, i.e., noun, verb and adjective etc. represented by $n$, $v$ and $adj$ respectively.

(e) Position of the argument with respect to the verb, i.e., left or right represented by $l$ and $r$ respectively.

(f) Relation of verb with argument in terms of parent and child represented by $p$ and $c$ respectively.

In figure 6.1, we see that dependency relations for verb $aa$ having the sense ‘come’ is given under the arc-label field. The arguments of the verb $aa$ ‘come’ is $raam$ ‘Ram’ and $haidaraabaad$ ‘Hyderabad’. The karaka relations of $raam$ ‘Ram’ and $haidaraabaad$ ‘Hyderabad’ are $karta$ ($k1$) and $Goal$ ($k2p$) respectively. The necessity of $k1$ ($raam$) and $k2p$ ($haidaraabaad$) is mandatory which is represented as $m$. These two arguments are mandatory to accomplish the semantics of the verb $aa$ ‘come’. $karta$ ($k1$) ($raam$) takes $0$ vibhakti and $goal$ ($k2p$) ($haidaraabaad$) can take either $0$, $ke\_paasa$ ‘near’, $taka$ ‘till’, or $para$ ‘on’ vibhakti depending upon the noun. Here, $haidaraabaad$ ‘Hyderabad’ which is $goal$ ($k2p$) takes either $0$, $ke\_paasa$ ‘near’, $taka$ ‘till’, etc., $vibhaktis$. There is par ‘on’ $vibhakti$ which is not taken by $haidaraabaad$ ‘Hyderabad’ but there can be other nouns that can come in the place of $haidaraabaad$ ‘Hyderabad’ which will be marked as $goal$ ($k2p$) with the verb $aa$ ‘come’ that takes this $vibhakti$. It is not necessary that $haidaraabaad$ ($k2p$) takes all the above mentioned $vibhaktis$. It might take some $vibhaktis$ or all the $vibhaktis$, whereas a different noun argument that can occur in the place of $haidaraabaad$ ($k2p$) might take the rest of the $vibhaktis$ or all the $vibhaktis$ according to the selectional restrictions of the nouns. The total $vibhaktis$ mentioned above can be used by various $Goal$ ($k2p$) arguments taken by the verb $aa$. We are mainly concerned with the relation $goal$ ($k2p$) saying that it takes all these $vibhaktis$ but the lexical item may change.

The $vibhaktis$ of the arguments depends upon the TAM (tense, aspect and modality). The $vibhaktis$ of the arguments may change with the change in TAM. Therefore, I have kept the TAM constant during this whole process of creating verb frames. I have taken present imperfect TAM, i.e., $taa\_hai$. A verb frame is different from another verb frame for a verb with the same sense, when the set of dependency relations marked on the verb arguments are different. An argument of a verb can take various $vibhaktis$ (postpositions) like the $haidaraabaad$ ($k2p$) argument of the verb $aa$ ‘come’. In case of verbs having the same sense as a particular verb, the arguments of these verbs may sometimes take different $vibhaktis$, additional $vibhaktis$, or less $vibhaktis$ than the arguments of the verb with which it is matching in sense.

If we look at the verb frame, i.e., figure 6.1, we see that $goal$ ($k2p$) can take either $0$, $ke\_paasa$ ‘near’, $taka$ ‘till’, $para$ ‘on’, etc., $vibhaktis$. We could have mentioned this fact by forming four verb frames for this sense; one for each $vibhakti$, instead we merge the all the $vibhaktis$ and show it in a single verb frame. The $vibhaktis$ are merged using an ‘or’ operator, represented as a pipe ‘|’ (see fig 6.1). The lexical category of both the arguments ($raam$ ‘Ram’ and $haidaraabaad$ ‘Hyderabad’) of the verb $aa$ ‘come’ is noun which is represented as $n$. This information of lexical category is mentioned in the verb frame under the lex-type field. The relation between verbs and its arguments in terms of parent and child relation is always child which is represented as $c$. The verb is the parent and the arguments are children of the verb. This piece of information is mentioned in the verb frame under the reln field which means relation. There is one more information about the arguments in the verb frame and that is about the position of the argument in relation to the verb, i.e., to which side of the verb does the argument occurs.
Both the arguments (raam ‘Ram’ and haidaraabaad ‘Hyderabad’) of the verb aa ‘come’ occur to the left of the verb so they are marked as left represented by l. This information is mentioned in the verb frame under the posn field which means position. The information of reln (relation) and posn (position) in the verb frame is not required from the point of view of a lexical resource but was required for the purpose of the parser.

The frames are developed based on simple present tense and indicate habitual acts taking it as default TAM. In fact, dependency relations and the vibhaktis (Hindi postpositions / case markers) in the frame reflect the behavior of the verb when it occurs in simple present (‘-taa hai’ in Hindi, eg. khaa-taa hai ‘eats’). This is done to ensure consistency while forming the various frames, as in Hindi, the vibhakti of an argument changes with changes in the TAM (tense, aspect and modality) information of the verb. These changes in the vibhaktis are not syntactic alternations but are transformations due to the change in the default TAM.

There are two figures given below, out of which 6.2 is an example of the verb aa taking a second sense ‘know’, and 6.3 is an example of the verb laad which takes two frames for a single sense ‘load’.
Figure 6.2 Verb Frame for Second Sense of Verb *aa* ‘know’
Verb:: *laad*
SID:: *laad*%VT %S1
Verb_Sense:: HWN (1)
Eng_Gloss:: load
Verbs_in_Same_Class:: Synonyms → cha.Dhaa;
                     Verb frames populated → cha.Dhaa
Frames::

Frame_Name_1::
Example:: *mazaduur traka para| meM boriyaM laadataa haiM*
         workers truck on in bags load.IMPF be.PRES
         “Workers load bags in the truck.”
Theta_Roles:: AGENT LOCATION THEME V
Demand_Frame:: FRAME_ID_1

FRAME_ID:: *laad*%VT %S1 %FID1

<table>
<thead>
<tr>
<th>arc-label</th>
<th>necessity</th>
<th>vibhakti</th>
<th>lextype</th>
<th>posn</th>
<th>reln</th>
</tr>
</thead>
<tbody>
<tr>
<td>kl</td>
<td>m</td>
<td>0</td>
<td>n</td>
<td>1</td>
<td>c</td>
</tr>
<tr>
<td>k7p</td>
<td>m</td>
<td>*para</td>
<td>meM*</td>
<td>n</td>
<td>1</td>
</tr>
<tr>
<td>k2</td>
<td>m</td>
<td>0</td>
<td><em>ko</em></td>
<td>n</td>
<td>1</td>
</tr>
</tbody>
</table>

Frame_Name_2::
Example:: *mazaduur traka ko boriyoM se laadataa haiM*
         workers truck Acc bags with load.IMPF be.PRES
         “Workers load the truck with bags.”
Theta_Roles:: AGENT THEME INSTRUMENT V
Demand_Frame:: FRAME_ID_2

FRAME_ID:: *laad*%VT %S1 %FID2

<table>
<thead>
<tr>
<th>arc-label</th>
<th>necessity</th>
<th>vibhakti</th>
<th>lextype</th>
<th>posn</th>
<th>reln</th>
</tr>
</thead>
<tbody>
<tr>
<td>kl</td>
<td>m</td>
<td>0</td>
<td>n</td>
<td>1</td>
<td>c</td>
</tr>
<tr>
<td>k2</td>
<td>m</td>
<td><em>ko</em></td>
<td>n</td>
<td>1</td>
<td>c</td>
</tr>
<tr>
<td>k3</td>
<td>m</td>
<td><em>se</em></td>
<td>n</td>
<td>1</td>
<td>c</td>
</tr>
</tbody>
</table>

Figure 6.3 Verb Frame of Verb *laad* ‘load’ Having Two Frames for One Sense
Two senses of the verb *aa* are given in the above two figures 6.1 and 6.2 respectively. Each of these senses has one verb frame each. These senses are ‘come’ and ‘know’ respectively. All the senses of a verb are present in a single verb entry. For each sense, a sense id is provided along with the English gloss. For the first sense of the verb *aa* (figure 6.1), the sense id (SID) is *aa% VI% S1* and for the second sense (figure 6.2), the sense id (SID) is *aa%VI%S2*. The Eng_Gloss of the first sense of the verb *aa* is ‘come’ and the second sense is ‘know’. The verb *jaana* ‘know’ matches with the second sense of the verb *aa* ‘know’. The verb entry for the verb *jaana* exists as we can see the FRAME_ID mentioned for the verb *jaana*, i.e., *jaana%VT%S2%FID1* under the field Verbs_in_Same_Class of the second sense of the verb *aa* ‘know’ (figure 6.2). Different senses of a verb match with different senses of other verbs. The first sense of the verb *aa* ‘come’ (*aa%VI%S1*) matches with the first sense of the verb *padhaara* ‘arrive’ (*padhaara%VI%S1*). Both the first sense of the verbs *aa* ‘come’, and *padhaara* ‘arrive’ have single frame which match with each other. So both these verbs match in sense as well as frames i.e., *aa%VI%S1%FID1* and *padhaara%VI%S1%FID1* match with each other. In case of second sense of the verb *aa* ‘know’ (*aa%VI%S2*), the verb *jaana* ‘know’ (*jaana%VT%S2*) matches with it (*aa%VI%S2*). In case of the verb entry *jaana* which is a transitive verb, the sense ‘know’ is the second sense of the verb *jaana* (*jaana%VT%S2*) which is matching with the second sense of the verb *aa* ‘know’ (*aa%VI%S2*). Here we can see that though the second sense of the verbs *aa* ‘know’ (figure 6.2) and *jaana* ‘know’ is same but they are different types of verbs, i.e., *aa* ‘know’ is an intransitive verb (VI) whereas *jaana* ‘know’ is a transitive verb (VT). Since, they are different types of verbs, their frames are also different. Though they differ in frames, still we have listed *jaana* ‘know’ (*jaana%VT%S2*) under the Verbs_in_Same_Class field of the verb *aa* ‘know’ (*aa%VI%S2*) as they have same senses and we clearly make a note there that they don’t match in frames. This gives us a generalization that though semantically they are same: (i) they differ in their verb types; (ii) they differ in their frames, i.e., they take a different set of dependency relations (karaka relations (syntactic-semantic) and other than karakas).

In the Example field of figure 6.2, there are two sentences separated by an ‘or’ operator represented by pipe (‘|’). These two examples differ in the lexical category of *k1* (karta). Here *k1* can have either verb (v) or noun (n) as its lexical category. In one example *banaanaa* ‘cooking’ is *k1* and it is a verbal noun as it is a noun derived from a verb. So *banaanaa* ‘cooking’ is just marked as a verb (v) and no additional information of it being a verbal noun is marked. In the other example, *silaaii-ka.Daaii* ‘stitching’ is *k1* and its lexical category is noun (n). So this information that *k1* can take either verb (v) or noun (n) as its lexical category will be shown in the verb frame as *v|n*. Here we are merging the lexical categories by an ‘or’ operator represented by pipe (‘|’). Generally merging process is done in verb frames for vibhakti and lexical category as arguments of a verb can take multiple vibhaktis in different situations and their lexical category also varies sometimes.

The FRAME_ID of the verb frame of the first sense and second sense are *aa%VI%S1%FID1* and *aa%VI%S2%FID1* respectively. The arguments of the verb *aa* having the sense ‘know’ in the figure 6.2, are *siitaa* ‘Sita’, and *banaanaa / silaaii-ka.Dhaaii* ‘cooking / stitching’. The dependency relations of *siitaa* ‘Sita’ and *banaanaa / silaaii-ka.Dhaaii* ‘cooking / stitching’ are anubhavakarta (k4a) and karta (k1) respectively. The necessity of *siitaa* (k4a) and *banaanaa / silaaii-ka.Dhaaii* (k1) is mandatory i.e., *m. siitaa* (k4a) takes ko vibhakti and *banaanaa / silaaii-ka.Dhaaii* (k1) takes 0 vibhakti. As mentioned above, lexical category of *siitaa* (k4a) is noun, i.e., *n* and the lexical category of *banaanaa / silaaii-ka.Dhaaii* (k1) is verb / noun (v/n). The structure also has thematic role information. The first sense of the verb *aa* ‘come’ (figure 6.1), takes agent and destination theta roles for raam ‘Ram’ and haidaraabaad ‘Hyderabad’ respectively. The second sense of the verb *aa* ‘know’ (figure 6.2), takes experiencer and theme theta roles for *siitaa* ‘Sita’ and *banaanaa ‘cooking’ / silaaii-ka.Daaii* ‘stitching’ respectively.
As mentioned earlier, each verb can have multiple senses and for each sense of the verb there can be a number of possible verb frames. A frame is different from another frame for a verb when the argument relations are different. Only few verbs have more than one frame for its one particular sense. In figure 6.3, we can see an example of a verb having more than one frame for a particular sense. Here the verb is laad having the sense ‘load’. Its SID is laad%VT%S1. For this particular sense of verb laad ‘load’, there are two verb frames. The FRAME_ID of the first and second verb frame is laad%VT%S1%FID1, and laad%VT%S1%FID2 respectively. Here we can see that for the same sense S1 there are two frames FID1 and FID2. The arguments of the verb laad ‘load’ in the first frame are mazaduur ‘servants’, traka ‘truck’, and boriyaaM ‘bags’. The karaka relations of mazaduur ‘servants’, traka ‘truck’, and boriyaaM ‘bags’ are karta (k1), deshahikaran (k7p) and karma (k2) respectively. The necessity of mazaduur (k1), traka (k7p), and boriyaaM (k2) is mandatory i.e., m. mazaduur (k1) takes 0 vibhakti, traka (k7p) takes either para ‘on’ or meM ‘in’ vibhakti and boriyAM (k2) takes either 0 or ko vibhakti. The arguments traka (k7p), and boriyaaM (k2) are taking more than one vibhakti so here we have merged the vibhaktis of traka (k7p), and boriyaaM (k2), i.e., para|meM, and 0|ko respectively. The lexical category of all the arguments is noun i.e. n.

Here the second frame is different from the first frame because the dependency relations are different. As it is a second frame, the number of the verb frame will be Frame_Name_2. The arguments of the verb laad ‘load’ in the second frame are mazaduur ‘servants’, traka ‘truck’, and boriyooM ‘bags’. The karaka relations of mazaduur ‘servants’, traka ‘truck’ and boriyooM ‘bags’ are karta (k1), karma (k2) and karana (k3) respectively. The necessity of mazaduur (k1), traka (k2) and boriyooM (k3) is mandatory i.e., m. mazaduur (k1) takes 0 vibhakti, traka (k2) takes ko vibhakti, and boriyooM (k3) takes se ‘with’ vibhakti. The lexical category of all the arguments is noun i.e. n. Verbs_in_Same_Class field lists a verb i.e., cha.DZaa ‘load’ which has same sense as the verb laad ‘load’. There is no verb entry for the verb cha.DZaa ‘load’ as we can see that there is no frame_id (FRAME_ID) mentioned for the verb cha.DZaa. In case of verb frames, cha.DZaa ‘load’ has only one verb frame which matches with the first frame of the verb laad ‘load’. Verb laad has two frames for this particular sense of ‘load’ whereas the verb cha.DZaa has only one frame for the sense of ‘load’. This information is also provided in the verb entry of laad ‘load’ that; (i) there is a difference between the number of frames of verb laad and cha.DZaa, and (ii) the the single frame of the verb cha.DZaa matches with the first frame of the verb laad.

One of the important features in verb frames is the feature necessity. The necessity of the arguments is divided into 2 categories: (1) mandatory, and (2) desirable. As discussed above, mandatory is compulsory to be present in the sentence to fulfill the meaning of the verb, whereas desirable argument is not compulsory to be present in the sentence. An example of desirable argument is given below:

Ex- (6.16) raam ne chaakuu se (k3) seba kaaTaa
      ram   Erg.  knife with    apple cut
      ‘Ram cut an apple with a knife.’

Ex- (6.17) raam ne seba kaaTaa
      ram   Erg.    apple cut
      ‘Ram cut an apple.’
In the example 6.16, chaakuu se ‘with the knife’ is a desirable argument as it can be dropped and still the meaning can be retrieved from the context. See example 6.17 which is a perfect sentence without the desirable argument chaakuu se ‘with the knife’ which we can makeout even if it is not in the sentence.

Below given is an example of a verb frame taking a desirable argument:

![Verb Frame](image)

**Figure 6.4** Verb Frame of Verb kaata ‘cut’ Having a **desirable** Argument

The above figure 6.4, is the verb frame of the verb kaata ‘cut’ which is taking a desirable argument. The argument chaakuu se ‘with the knife’ is a desirable argument so its necessity is d i.e., desirable.

In our verb frames, we are not considering optional arguments which are not required by the semantics of the verb and are considered as additional information. In example 6.18 given below, the argument shaam ko ‘evening’ is optional.
Below section discusses about the comparison between Fillmore’s NULL complements, Palmer’s Essential roles and desirable and optional arguments of the Dependency structure.

6.1.3.3 Comparison of Null Complements, Essential Roles with desirable and optional arguments:

6.1.3.3.1 NULL Complements

The concept of null complements is related to the deletion of the complements in the sentence and in spite of the deletion the sentence is grammatical and at the same time the meaning conveyed is similar to the one when the complement is present in the sentence. For example, “Suresh volunteered” has a null complement and even then it is correct and has the same meaning as the one with the complement [230].

According to [68], Null complements are divided into two types: (i)Definite Null Complements (DNC) also called as definite null instantiation (DNI) (e.g. [231, 232, 233]), and (ii)Indefinite Null Complements (INC) also called as indefinite null instantiation (INI). Definite NULL complements are those which are definite and can be obtained from the context whereas the Indefinite NULL complements are unidentified or unnecessary. In case of INC, the concentration is on the action [68]. For example: (i)the sentence “Harish found out at home”, expresses that certain definite thing was found out; (ii)in the sentence “Radha sings everyday”, the concern is not on the identification of the song sung [230].

Fillmore also talks about the how the sense of a verb plays an important role in allowing or disallowing null complements. The verb ‘apply’ allows Null complement when its sense is ‘applying for job’ and disallows when its sense is ‘applying something on some surface” [68] [230]. For example:

Sense1: applying for the job
Rohan applied for the job → Rohan applied

Sense2: applying something on some surface
Rohan applied medicine on the finger → * Rohan applied

Indefinite NULL complements depend on the lexical properties of the verb, also called as lexical licensing [234]. Verbs that take INC are clean, drink, read, sing, study, teach and write, etc [105]. The null complements taken by these verbs can be understood as a word which means “stuff” or “something”.

(6.18) \text{ram shaam ko seba khaataa hai}
\begin{align*}
\text{ram} & \quad \text{evening Acc.} \\
\text{apple} & \quad \text{eat.Impf.} \\
\text{be.Pres.} & \quad \text{'Ram eats an apple in the evening.'}
\end{align*}
For example, “He was very hungry, so he ate (stuff) quickly” [234]. Definite null instantiation depends on lexical licensing and the grammatical construction of the predicate and the complement, i.e., the form and meaning of the constituents [68, 231, 233] [234].

According to [68], lexical licensing is a phenomenon where the lexical properties of the predicate play an important role in the licensing of null instantiation, i.e., certain transitive predicates can permit their objects to be omitted and certain predicates cannot. This applies to the verbs that have synonymous meanings, i.e., certain semantically related verbs omit the complements and others don’t omit it [234].

She promised.
*She pledged. [68]

I tried.
*I attempted. [68]

Other than predicate’s lexical properties, DNI is also licensed by the grammatical construction that allows the deletion of some argument position. This type of licensing can be seen in imperative sentences, for example “Please give my book” in which the deletion of the subject is licensed by the grammar. We can also see this in passive sentences, like, “We’ve been cheated” in which the agent phrase is allowed to be deleted [235].

6.1.3.3.2 Essential Roles

The semantic approach introduced by Moerdler et al [236] depends mainly on verb classification and hierarchical structuring inside each category. In verb hierarchies, the top level nodes include the maximum broad information and after that the nodes turn into more and more specific as a hierarchy is traversed. The lower level nodes acquire all the properties from the top level nodes at the time of parsing. More than 90 verbs were analyzed and were categorized into 12 categories. For instance, the verb to get means Transfer of possession pointing out its subject to be recipient or patient (Tom gets $100; Tom is the recipient). The constraints on the verb arguments depend on the features of the noun while parsing. The examples of noun features are: Nouns such as son, daughter, step-son, carry the feature child; Nouns like father, mother, step-father, carry the feature parent. Both the words child and parent carry the feature relative.

The parser, proposed by Moerdler et al [236], handles just the semantically complete sentences. Moerdler et al [236] also shows a method for handling semantically incomplete sentences. While handling this problem, Moerdler et al [236] have used some of Martha Palmer’s methods and terminology [238]. The restriction roles are assigned one of the three categories listed here: (i) obligatory, (ii) essential, and (iii) non essential. Since Obligatory roles are syntactically mandatory so they are always filled. Essential roles are syntactically optional; however they are required to be filled for proper semantic processing. The non essential roles are syntactically as well as semantically optional and thus can be deleted or derived depending on our domain knowledge and previous input. If an essential role is not filled in the tree while traversing the tree, then the system needs to fill it. If the non essential roles are not mentioned then they can be derived by the system [236].
In case of verb *replace*, it takes two objects, **object1** and **object2** and both are ESSENTIAL semantic roles. Even if they are indicated or not indicated explicitly in the sentence, they should be filled. In case, if they are not indicated then reference resolution will preferably fill the role by an entity that has previously been indicated. If there is no previously indicated entity then an entity will be created [237]. For non-essential roles, such as **agent** and **instrument**, if they are not indicated in the sentence for the same verb then they will be left unfilled. The **instrument** is hardly indicated, and the agent can be easily left out in case of the following sentence, *The motor was replaced at 09:00*. In case of other verbs, the agent could be obligatory, and needs to be filled [239].

### 6.1.3.3.3 desirable and optional arguments in Verb frames

In Hindi verb frames, we don’t have NULL complements as all the complements are obligatory. Hindi verb frames can be compared to Essential roles since both have similar concept. In Hindi verb frames, we have mandatory (Obligatory) arguments which are required both syntactically and semantically; *desirable* arguments in Hindi verb frames are similar to Essential roles which are syntactically optional but are required semantically; we are not considering optional arguments in the verb frames but they are again similar to the non essential roles which are both syntactically and semantically optional.

Below, in figures 6.5 and 6.6 are the verb frames of the verbs *aa* ‘cost’ and *mila* ‘get’ respectively. These two verbs match in sense and also have the same number of frames. They both have two frames:
Verb:: *aa*
SID:: *aa%VT%S3*
Verb_Sense:: HWN ()
Eng_Gloss:: cost
Verbs_in_Same_Class:: Synonyms $\rightarrow$ *mila%VT%S1%FID1 - mila%VT%S1%FID2*
Frames::

**Frame_Name_1::**
Example:: *yaha kitaab 125 rupaye meM aatrii hai*
this book 125 Rs in cost-PRES be-PRES
"This book costs Rs.125" or "This book comes for Rs.125"
Theta_Roles:: THEME ASSET V
Demand_Frame:: FRAME_ID_1

**FRAME_ID:: aa%VT%S3%FID1**

<table>
<thead>
<tr>
<th>arc-label</th>
<th>necessity</th>
<th>vibhakti</th>
<th>lextype</th>
<th>posn</th>
<th>reln</th>
</tr>
</thead>
<tbody>
<tr>
<td>kl</td>
<td>m</td>
<td>0</td>
<td>n</td>
<td>l</td>
<td>c</td>
</tr>
<tr>
<td>k7</td>
<td>m</td>
<td>meM</td>
<td>n</td>
<td>l</td>
<td>c</td>
</tr>
</tbody>
</table>

**Frame_Name_2::**
Example:: *yaha kitaab 125 rupaye kii aatrii hai*
this book 125 Rs of cost-PRES be-PRES
"This book costs Rs.125" or "This book comes for Rs.125"
Theta_Roles:: THEME ASSET V
Demand_Frame:: FRAME_ID_2

**FRAME_ID:: aa%VT%S3%FID2**

<table>
<thead>
<tr>
<th>arc-label</th>
<th>necessity</th>
<th>vibhakti</th>
<th>lextype</th>
<th>posn</th>
<th>reln</th>
</tr>
</thead>
<tbody>
<tr>
<td>kl</td>
<td>m</td>
<td>0</td>
<td>n</td>
<td>l</td>
<td>c</td>
</tr>
<tr>
<td>r6v</td>
<td>m</td>
<td>kiikkaa$ke</td>
<td>n</td>
<td>l</td>
<td>c</td>
</tr>
</tbody>
</table>

Figure 6.5 Verb Frame of Verb *aa* ‘cost’ Having Two Frames for One Sense
Figure 6.6 Verb Frame of Verb mila ‘cost’ Having Two Frames for One Sense
The two frames \textit{mila}\textit{VI}S1FID1 and \textit{mila}\textit{VI}S1FID2 that belong to the first sense ‘get’ of the verb \textit{mila}, match with the two frames \textit{i.e.}, \textit{aa}\textit{VI}S3FID1 and \textit{aa}\textit{VI}S3FID2 that belong to the third sense ‘cost’ of the verb \textit{aa}. In the figure 6.5, we can see that the two frame ids of \textit{mila} ‘get’ (\textit{mila}\textit{VI}S1FID1 and \textit{mila}\textit{VI}S1FID2) are listed in the Verbs\_in\_Same\_class field of the verb entry \textit{aa} ‘cost’. The two verb frame ids of \textit{mila} ‘get’ (\textit{mila}\textit{VI}S1FID1 - \textit{mila}\textit{VI}S1FID2) listed there are separated by a ‘hyphen’. Similarly in the figure 6.6, we can see the two frame ids (\textit{aa}\textit{VI}S3FID1 - \textit{aa}\textit{VI}S3FID2) of \textit{aa} ‘cost’ are listed in the Verbs\_in\_Same\_class field of the verb entry \textit{mila} ‘get’.

In figure 6.1, we can see that the verb \textit{aa} ‘come’ and the verb belonging to its class (which have the same sense as the verb \textit{aa}), \textit{i.e.}, \textit{padhaara} ‘arrive’ vary in vibhak\_is with respect to the goal (k2p) dependency relation (see figure 6.1). The verb \textit{padhaara} takes less vibhakti for goal (k2p) than the verb \textit{aa} whereas the verb \textit{pahu.Ncha} ‘reach’ which also belongs to the class of verb \textit{aa} has same no. of vibhaktis as the verb \textit{aa} for the goal (k2p). The verb \textit{aa} and \textit{pahuNcha} take 0, \textit{ke_paas}, ‘near’, \textit{par ‘on’,} and \textit{taka ‘till’} vibhaktis whereas the verb \textit{padhaara} takes 0, \textit{ke_paas} ‘near’, \textit{par ‘on’} vibhaktis. taka vibhakti is not taken by the goal (k2p) argument of the \textit{padhaara} verb. In the example 6.19 given below, we can see that the goal (k2p) with \textit{taka vibhakti} for the verb \textit{padhaara} is ungrammatical. This information about the differences between vibhaktis among verbs having the same sense is provided in the respective verb entries. These differences allows us to come up with generalizations and also classify the verbs based on these features.

(6.19) *\textit{raam} hydaraabaad \textit{taka} (k2p) \textit{padhaare} \\
\textit{ram} hyderabad till arrived  \\
‘Ram arrived till Hyderabad.’

In the above instance, we saw that there is difference in vibhaktis, \textit{i.e.}, the verb (\textit{padhaara} ‘arrive’) listed in the class has fewer vibhaktis than the main verb (\textit{aa} ‘come’). There is another instance where the verb listed in the class has more vibhaktis than the main verb entry. In the verb frame of \textit{bataa} ‘tell’, the relation \textit{k4} (sampradana) takes only \textit{ko} vibhakti whereas the verbs that belong to the \textit{bataa} class, \textit{i.e.}, \textit{kaha} ‘say’ and \textit{bola} ‘tell’ which have the same sense as the verb \textit{bataa} take an additional vibhakti \textit{se} in addition to \textit{ko}. See examples 6.20 and 6.21 given below:

(6.20) \textit{raam} (k1) \textit{siiata ko/*se} (k4) \textit{saab} (k2) \textit{bataataa hai}  \\
\textit{ram} sita Dat./ everything tell.Impf. be.Pres.  \\
‘Ram tells everything to Sita.’

(6.21) \textit{raam} (k1) \textit{siiata ko/soe} (k4) \textit{saab} (k2) \textit{boltaa/kehtaa hai}  \\
\textit{ram} sita Dat./with everything tell/say.Impf. be.Pres.  \\
‘Ram tells/says everything to Sita.’
We also have cases where the main verb and the verb having the same sense as the main verb take totally different vibhaktis for a particular dependency relation. The main verb chaaha ‘love’ takes ko vibhakti with karma (k2) whereas the verb prema kar ‘love’ which belongs to the chaaha class takes se vibhakti with karma (k2) relation. The verb pasaMd kar ‘like’ which also belongs to the chaaha class takes ko vibhakti as the verb chaaha. See the examples 6.22 and 6.23 given below:

(6.22) raam (k1)  siiataa ko (k2)  chaahataa / pasaMd_kartaa hai
ram    sita    Dat.    love.    Impf./    like    do.Impf    be.Pres
‘Ram loves/likes Sita.’

(6.23) raam (k1)  siiataa se (k2)  prem    kartaa    hai
ram    sita    with    love.    do.Impf    be.Pres
‘Ram loves Sita.’

In this section, we have seen cases where the verbs are similar in senses (main verb and the verbs belonging to its class) but there are some variations between them. The variations are as follows:

a) Differ in frames, for example, verb frames of jaana ‘know’ and aa ‘know’ as mentioned in the figure 6.2.

b) All the frames don’t match, i.e., only one frame matches and the other doesn’t, e.g., the verb cha.Dhaa ‘load’ matches with the first frame of laad ‘load’ and doesn’t match with the second frame of the verb laad as mentioned in the figure 6.3.

c) All the frames match, i.e., the two frames of the verb mila get, obtain’ matches with the two frames of the verb aa ‘cost’ as mentioned in the figures 6.5 and 6.6.

d) Fewer vibhaktis, for example, aa ‘come’ and padhaara ‘arrive’. padhaara ‘arrive’ takes fewer vibhaktis in comparison to aa ‘come’ for the dependency relation goal (k2p).

e) More vibhaktis, for example, bataa ‘tell’ and kaha/bola ‘say/tell’. bataa ‘tell’ take fewer vibhaktis in comparison to kaha/bola ‘say/tell’ for sampradana (k4) karaka relation.

f) Totally different vibhaktis, for example, chaaha ‘love’ and prem kar ‘love’. chaaha ‘love’ and prem kar ‘love’ take different vibhaktis for the karma (k2) karaka relation.
6.1.4 Multiword Expressions

I have largely prepared verb frames for simple verbs and prepared verb frames for only very few multi word expressions. But verb frames have been populated for Multi Word Expressions (MWEs) such as conjunct verbs by listing them in the Verbs_in_Same_Class field of the verb entries of the simple verbs with which MWEs match in sense as well as frame.

Baldwin and Su Nam Kim [17] have adopted the following formal definition of multiword expression from [152]:

“Multiword expressions (MWEs) are lexical items that: (a) can be decomposed into multiple lexemes; and (b) display lexical, syntactic, semantic, pragmatic and/or statistical idiomaticity”.

The structure and meaning of MWEs cannot be derived from their component words, as they occur independently. Some of the examples of MWEs in English are: (i) conjunctions like ‘as well as’ (meaning ‘including’), (ii) idioms like ‘spill the beans’ (meaning ‘reveal the secret’), and phrasal verbs like ‘find out’ (meaning ‘search’) etc. Some of the examples of MWEs in Hindi are: (i) adjective-noun pairs like ‘praachiin bhaarat’ (meaning ‘ancient India’), (ii) idioms like ‘svarga sidhaaranaa’ (meaning ‘to die’), and (iii) complex predicates like ‘kshamaa karanaa’ (meaning ‘to forgive’) and expressions like ‘yahaa.N-vahaa.N’ (meaning ‘here and there’) [183].

Though MWEs have been a problematic area in the literature, they have been handled in the verb frames. Example 6.24 given below is an example of conjunct verb. Here pratiikshaa karnaa ‘to wait’ is the conjunct verb where pratiikshaa ‘wait’ is the noun and karnaa ‘to do’ is the verb. The verb frame for the conjunct verb pratiikshaa karnaa ‘to wait’ is given in the figure 6.7.

(6.24) raam siitaa kii pratiikshaa kartaa hai
‘Ram waits for Sita.’
pratiikshaa kar ‘wait’ is a multi-word expression, so the ‘type of verb’ information for these expressions is represented as MWE in the unique sense id, i.e., pratiikshaa_kar %MWE%S1. There are other multi-word expressions such as samajh _meM _aanaa ‘to understand’ and dimaag_meM _aa ‘to strike’. These are also handled as pratiikshaa kar ‘wait’. The only difference is that pratiikshaa kar is made of noun and a verb whereas samajh _meM _aa ‘understand’ and dimaag_meM _aa ‘strike’ are made of noun, postposition, and a verb. samajh _meM ‘in understanding’ and dimaag_meM ‘in mind’ have a pof (Part Of) relation with the verb aa ‘come’.

(6.25) raam ko (k4a) yaha baat (k1) samajh meM (pof) aayii
ram Dat. this fact understanding in came
‘Ram understood this fact.’

(6.26) yaha baat (k1) achaanaka raam ke (r6-k7) dimaag meM (pof) aayii
this fact immediately ram Gen. mind in came
‘It struck Ram immediately.’

The verb frame for example 6.25 is given below:
6.1.5 Verb Frames and their Statistics

The statistics of the verb frames given below in the table 6.1 were arrived at by the following process:

a) The total number of simple verbs (verb entry) for which verb frames were created is 687. Initial verb frames were created by taking only single sense of all the verbs.

b) The total no. of verbs for which verb frames were created based on their multiple senses is 486 (each sense of a verb is counted). Since many verbs had multiple senses, each sense of a verb had their own demands and expectations. It was also observed that the syntactic and semantic patterns or structures of each sense were different from the other sense. In other words, the verb frame of each sense of a verb was different from the other sense. Thus, in the next step, more verb frames were created for the verbs based on their various senses. We can see variation in the verb frames of the different senses of a verb as well as variation in the verb frames of the same sense of a verb (if there occurs another verb frame of the same sense).

c) Once these verb frames were created for multiple senses of each verb, it was also observed that the same verb frame of a particular verb was applicable for some other verbs as well, because
these verbs shared the semantic and syntactic properties of the existing verb frames of certain verbs. Thus the frames were generalized to a class of verbs. Thus, the total no. of additional verbs that were populated in the existing verb frames of certain verbs is 180. Here the populated verbs are counted by just considering their single sense.

d) There were multiple senses of the populated verbs too which shared the existing verb frames of certain verbs. The count of populated verbs based on their multiple senses is 201. Here, I have counted each sense of all the populated verbs.

e) The total no. of verbs belonging to both the categories, i.e., the verbs for which verb frames were created and the populated verbs is 480 (total of both (a) and (c)). Here the verbs are counted by just considering their single sense.

f) The total no. of verb senses in the verb frames is 687 (total of both (b) and (d)). If we take the count of verbs belonging to both the categories, i.e., the verbs for which verb frames were created and the populated verbs then the total is 687. Here, I have counted each sense of all the verbs.

We have extracted the frequencies and percentage of the 687 verbs from the pilot Hindi dependency Treebank (HyDT) (2230 sentences) (see Appendix-I) as well as from the multi-layered treebank (400k words) for Hindi and Urdu (Appendix-II). Some of the statistics related to the verb frames are given below:
<table>
<thead>
<tr>
<th>S.No.</th>
<th>Type of the count</th>
<th>Count</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>Total no. of verbs for which verb frames (verb entry) were created is:</td>
<td>300</td>
<td>Here the verbs are counted by just taking its single sense, i.e., the additional senses of the verbs are excluded out of the count. (list of verbs given in Appendix-III)</td>
</tr>
<tr>
<td>(2)</td>
<td>Total no. of verbs for which verb frames were created based on their multiple senses is:</td>
<td>486 (300 verb entries created+186 multiple senses of the verb entries created)</td>
<td>Here additional senses of the verbs are included, i.e., I have counted each sense of all the verbs. (list of verbs given in Appendix-IV)</td>
</tr>
<tr>
<td>(3)</td>
<td>Highest sense count for a verb:</td>
<td>11</td>
<td>Verb nikala ‘leave’ has 11 senses.</td>
</tr>
<tr>
<td>(4)</td>
<td>Total no. of additional verbs that were populated in the existing verb frames of certain verbs is:</td>
<td>180</td>
<td>Here the populated verbs are counted by just taking its single sense, i.e., the additional senses of the populated verbs are excluded out of the count. (list of verbs given in Appendix-V)</td>
</tr>
<tr>
<td>(5)</td>
<td>Total no. of additional verbs that were populated based on their multiple senses in the existing verb frames of certain verbs is:</td>
<td>201 (180 populated verbs+21 multiple senses of the populated verbs)</td>
<td>Here additional senses of the populated verbs are included, i.e., I have counted each sense of all the populated verbs. (list of verbs given in Appendix-VI)</td>
</tr>
<tr>
<td>(6)</td>
<td>Total no. of verbs for which verb frames(verb entry) were created + Total no. of additional verbs that were populated in the existing verb frames of certain verbs are:</td>
<td>480 Lemmas (300 verb entries created+180 populated verbs)</td>
<td>Here the verbs are counted by just taking their single sense, i.e., the additional senses of the verbs are excluded out of the count in both the categories (By both the categories, we mean verbs for which verb frames are created and the populated verbs). (list of verbs given in Appendix-VII)</td>
</tr>
<tr>
<td>(7)</td>
<td>Total no. of verbs for which verb frames were created based on their multiple senses + Total no. of additional verbs that were populated based on their multiple senses in the existing verb frames of certain verbs are:</td>
<td>687 (480 Lemmas+207 multiple senses of the 480 Lemmas)</td>
<td>Here additional senses of the verbs are included, i.e., I have counted each sense of all the verbs belonging to both the categories (By both the categories, we mean verbs for which verb frames were created and the populated verbs). (list of verbs given in Appendix VIII)</td>
</tr>
</tbody>
</table>

Table 6.1 Verb Frames and their Statistics
It is clear that the entire structure just discussed is very rich. As of now we plan to exploit the frames and the verb classes in parsing. They can also be used for various other applications which require a knowledge base, for example, word sense disambiguation, Machine translation, etc.

These verb frames are used in Hindi Parser [84, 25]. The Output of the Parser is given below for the following examples:

**Ex- (6.27) vaha ghar aataa hai**

he home comes is
‘He comes home.’

**Ex- (6.28) raam seba khaataa hai**

ram apple eats is
‘Ram eats apple.’

The output of the parser for example 6.27 is given below in the figure 6.9. Example 6.28 has two outputs which is given in the figure 6.10 and 6.11 respectively.

```
1  ((   NP   <af=vaha,P,any,s,,0,0,/drel=varg__k1:3/name=1>
1.1 vaha      PRP  <af=vaha,P,any,s,,0,0,>
 ))
2  ((   NP   <af=ghar,n,m,s,,1,meM,/drel=varg__k7:3/name=2>
2.1 ghar      NN   <af=ghar,n,m,s,,1,meM,>
 ))
3  ((   VGF  <af=aa,v,m,s,any,,,taa_hai/name=3>
3.1 aataa     VM   <af=aa,v,m,s,any,,,taa_hai>
3.2 | SYM
 ))
```

**Figure 6.9 Output of the Parser for Example-6.27**
Example 6.28 has two outputs because *raam* ‘Ram’ and *seba* ‘apple’ have 0 (zero) *vibhakti*. In such a situation, two outputs will be generated as both *k1* (karta) and *k2* (karma) take 0 (zero) *vibhakti*. In output-I (figure 6.10), we can see that *raam* is *k2* (karta) and *seba* is *k1* (karta), and in output-II (figure 6.11), *raam* is *k1* (karta) and *seba* is *k2* (karma).

The *verb frames* developed have been used in a Constraint based parser. Constraint based parsing using integer programming [31,25,26] has been successfully tested for Indian languages [37, 35]. The parser uses the syntactic cues available in a sentence and forms constraint graphs (CG) on the basis of the generalizations available. It uses these notions as basic demand frames, i.e., *verb frames* and transformation frames [36] to build the constraint graphs. It translates the constraint graphs into an integer programming (IP) problem. The solutions to the problem provide the possible parses for the sentence. The initial results have shown that the parser gives comparable results with the state-of-the-art data driven Hindi parsers. Small lexicon (linguistic demands of various heads) is one of the main reasons for the
performance of the parser to get affected. The efficiency will automatically increase with the increase in the coverage of this lexicon [31, 25, 26].

After the creation of verb frames for Hindi, I classified the Hindi verbs based on their argument structure given in the verb frames. In the section given below, I discuss about the classification of Hindi verbs.

6.2 Hindi Verb Classification

6.2.1 Earlier Attempts

Earlier attempts on Hindi verb classification have mainly been of three types:

1. There have been efforts to classify the verbs according to their form. Suraj Bhan Singh [165] has made a formal classification of Hindi main verbs based on their form and also compared them with English verbs: (a) Simple root (saral dhaatu), (b) Composite root (saamaasik dhaatu), (c) Complex verb (mishra kriyaa), and (d) Compound verb (saMyukta kriyaa). This internal form or structure of the verb doesn’t show any syntactic and semantic consequences. The other two approaches deal with the syntactic structures.

2. According to Kachru [92], in Hindi there are three sets of inherent properties of verbs which have important syntactic consequences. These are: (a) Stative vs. Inchoative vs. Active, (b) Volitional vs. Non-Volitional, and (c) Factive vs. Non-Factive.

3. Another approach related to syntactic structures is found in Sahay [154] who classifies the Hindi verbs on their karaka requirements. He enumerates different constructions that can be formed using karaka relations and classifies the verbs that participate in such constructions. Some of these constructions are:

(i) karta (agent/theme/force) + kriya (verb)  
(ii) karta + karma (theme) + kriya  
(iii) karta + adhikarana (location) + kriya  
(iv) karta + apaadaan (source) + kriya
All the above classification approaches focus on different aspects of the verbs in Hindi. Singh focuses on word formation, Kachru on inherent properties of verbs having syntactic consequences, and Sahay, on sentence constructions. While classifying verbs, each of these criteria are important. In this section, we present a more holistic approach to classifying Hindi verbs.

### 6.2.2 Our Approach

In this section, I present my approach to classification of Hindi verbs. I have classified the Hindi verbs based on similar *verb frames*, i.e., based on the same argument structure. *Verb frames* of Hindi verbs are the same when they have the same set of dependency relations. Other information in the *verb frame* (tabular form) such as *necessity*, *vibhakti*, *lexical category*, etc., may vary but dependency relations should be the same. The basis of classification was more syntactico-semantic than either totally syntactic or totally semantic. If there exists certain semantic similarities in the verb classes among the verbs which are very rare, then I have grouped the verbs under a semantic class. I also sub-classified the verbs within the main class based on similar *vibhaktis* and other information (*necessity*, *lextype*) in the *verb frame*.

In the major classification, we find different groups of verbs sharing certain semantics. For example, if we look at the *kl+k2p+verb* frame, we see that it contains motion verbs, such as *aa* ‘come’, *pahu.Nca* ‘come’, *padhaar* ‘arrive’, *jaa* ‘go’, *aa* ‘return’, *lauTa* ‘return’, *chala* ‘go’ *dau.Da* ‘run’, *bhaag* ‘run’, *cha.Dha* ‘climb’, *chala* ‘sail’. It also takes raise verbs (increment in the quantity), for example, *cha.Dha* ‘raise’, and *ba.Dha* ‘raise’. In the sub-classification, these groups of verbs sharing certain semantics, form different sub-classes. The motion verbs mentioned below didn’t form a single sub-class but many sub-classes. For instance, *aa* and *pahu.Ncha* form one sub-class, *padhaar* forms another sub-class, and *jaa* forms another sub-class. These motion verbs have more fine-grained semantics which is causing them not to form a single sub-class but many sub-classes. Hence, with in the motion verbs we see many sub-classes whereas raise verbs form single sub-class. There are only two raise verbs. In the sub-classification, mostly there are very few verbs.

All the verbs don’t share the same semantics in the sub-classes. If there are fifteen verbs in a sub-class then we see that five verbs share some semantics, another five share some other semantics and rest of the five don’t share any semantics at all. One sub-class varies from the other in terms of semantics of the verbs, contained in those sub-classes. In *kl+k2+v* frame, one sub-class contains social interaction verbs, such as *prem_kar* ‘like’, *la.Da* ‘fight’, *baat_kar* ‘talk’, etc., whereas the other sub-class contains expression verbs, such as *ha.Nsa* ‘laugh’, *gussaa_kar* ‘to get angry’, etc. In the major classification, we find that certain classes are formed based on some Hindi sentence constructions, such as Causatives. For example, frames, such as ‘*pk1+jk1+k2+v*’, ‘*pk1+jk1+v*’, ‘*pk1+jk1+k4+k2+v*’, ‘*pk1+mk1+jk1+v*’, and ‘*pk1+mk1+jk1+k2+v*’ contain causative verbs. The verb frame *k4a+k1+v* takes all the verbs that take dative subjects.

Our classification is similar to the classification given by Sahay [154] who has classified the Hindi verbs on their *karaka* requirements. The only difference is that our classification has more verb frames on whose basis we have classified the verbs when compared the classification given by Sahay [154]. To some extent our classification is closer to Levin’s as she has shown the correlations between
the semantic and syntactic behavior of English verbs. Though we are working on Hindi language, we are also trying to capture the semantic similarities between the verbs that are classified based on dependency relations.

In the whole process of creating verb frames, I have developed 49 unique verb frames (unique set of dependency relations given in the tabular form) and have classified the verbs based on these verb frames. Below, I have listed all the 49 unique verb frames (dependency relations):

(1) Frame - 1: k1 + k2p + verb
(2) Frame - 2: k4a + k1 + verb
(3) Frame - 3: k4a + k7p + verb
(4) Frame - 4: k1 + k2 + verb
(5) Frame - 5: vmod (sequential) k1 + k2 + verb
(6) Frame - 6: k1 + k7 + verb
(7) Frame - 7: k1 + r6v + verb
(8) Frame - 8: k1 + k7p + verb
(9) Frame - 9: k1 + k5 + verb
(10) Frame - 10: k1 + vmod(simultaneous) + verb
(11) Frame - 11: k1 + vmod(reason) + verb
(12) Frame - 12: k1 + verb
(13) Frame - 13: k7p + k7t + k1 + verb
(14) Frame - 14: k1 + k2 + k5 + verb
(15) Frame - 15: k1 + rd + verb
(16) Frame - 16: k1 + rd + k2 + verb
(17) Frame - 17: k1 + k1s + verb
(18) Frame - 18: k1 + rh + verb
(19) Frame - 19: k1 + rh + k7 + verb
(20) Frame - 20: k1 + k2 + k2g + verb
(21) Frame - 21: k1 + k7t + verb
(22) Frame - 22: k1 + ras + verb
(23) Frame - 23: k1 + rt + verb
(24) Frame - 24: k1 + rt + k2 + verb
(25) Frame - 25: k1 + k2 + k7 + verb
(26) Frame - 26: k1 + k5 + k7 + verb
(27) Frame - 27: k1 + k2 + k7p + verb
(28) Frame - 28: k1 + k2p + k2 + verb
(29) Frame - 29: k1 + k2p + k7t + verb
(30) Frame - 30: k1 + k7t + k5 + verb
(31) Frame - 31: k1 + rt + k5 + k2 + verb
(32) Frame - 32: k7 + k7p + k1 + verb
(33) Frame - 33: k1 + k5 + k7p + k2 + verb
(34) Frame - 34: k1 + k2 + k3 + verb
(35) Frame - 35: k1 + k2 + k4 + verb
(36) Frame - 36: k1 + k2 + k2s + verb
(37) Frame - 37: k4a + k1 + k1s + verb
(38) Frame - 38: pk1 + jk1 + k2 + verb
(39) Frame - 39: pk1 + jk1 + k7p + verb
With the help of verb frames (only dependency relations) given below, I describe the classification process:

6.2.2.1 Verb Frame: \( k1 + k2 + k4 + \text{verb} \) (Based on Verb Frame 35)

Verb Class: (Verbs having the above karakas)

(1)bataa(bataa%VDT%S1%FID1), (2)bola(bola%VDT%S3%FID1), (3)kaha(kaha%VDT%S4%FID1), (4)bataa(bataa%VDT%S4%FID1), (5)bataa(bataa%VT%S6%FID1), (6)bola(bola%VT%S4%FID1), (7)kaha(kaha%VDT%S3%FID1),(8)sunaa,(9)bataa(bataa%VDT%S3%FID1),(10)bataa(bataa%VDT%S2%FID1), (11)bheja (bheja%V?%S1%FID2), (12)bola (bola%VT%S2%FID1), (13)kaha (kaha%VDT%S1%FID1),(14)pahuMchaa(pahuMchaa%VT%S2%FID1),(15)Daala(Daala%VT%S2%FID1), (16)Daala (Daala%VT%S2%FID1), (17)lauTaa (lauTaa%VDT%S1%FID1), (18)vaapasa_kara, (19)parosa (parosa%VDT%S1%FID1), (20)de (de%VDT%S1%FID1), (21)sauMpa (sauMpa%VDT%S1%FID1), (22)Cho.Da, (23)baa.NTa (baa.NTa%VT%S1%FID1), (24)likha (populated vaha raama ko khata likhataa haiM)

In the above verb class, we can see that certain verbs have semantic similarity, such as communicating verbs and giving verbs. We have taken those verbs which have semantic similarities from the verb class and listed them under these semantic classes:

39
**Communicating verbs:** (1)bataa (bataa%VDT%S1%FID1), (2)bataa(bataa%VDT%S2%FID1), (3)bataa (bataa%VDT%S3%FID1), (4)bataa (bataa%VDT%S4%FID1), (5)bataa (bataa%VT%S6%FID1), (6)bola (bola%VDT%S3%FID1), (7)bola (bola%VT%S4%FID1), (8)bola (bola%VT%S2%FID1), (9)kaha (kaha%VDT%S4%FID1), (10)kaha (kaha%VDT%S3%FID1), (11)kaha (kaha%VDT%S1%FID1), (12)sunaa, (13)likha

In the above communication verbs class there are 5 senses of bataa verb, 3 senses of bola verb, and 3 senses of kaha verb.

**Giving verbs:** (1)lauTaa(lautaa%VDT%S1%FID1),(2)vaapasa_kara,(3)parosa(parosa%VDT%S1%FID1), (4)de(de%VDT%S1%FID1), (5)baa.NTa(baa.NTa%VT%S1%FID1)

Verbs in the above verb class belonging to \( k_1 + k_2 + k_4 + \text{verb} \) verb frame, are sub-divided based on the vibhaktis and other information (necessity, lextype, and position) given in the verb frame (tabular form). Main classification of verbs is based on same argument structure (dependency relations) and the sub-classification is based on vibhaktis and other information given in the verb frame (tabular form). Below given are the sub-classes of the \( k_1 + k_2 + k_4 + \text{verb} \) frame:

**I.Frame for the following verbs is given below:**

(1) bataa (bataa%VDT%S1%FID1)

<table>
<thead>
<tr>
<th>arc-label</th>
<th>necessity</th>
<th>vibhakti</th>
<th>lextype</th>
<th>posn</th>
<th>reln</th>
</tr>
</thead>
<tbody>
<tr>
<td>k1</td>
<td>m</td>
<td>0</td>
<td>n</td>
<td>l</td>
<td>c</td>
</tr>
<tr>
<td>k4</td>
<td>m</td>
<td>ko</td>
<td>n</td>
<td>l</td>
<td>c</td>
</tr>
<tr>
<td>k2</td>
<td>m</td>
<td>0</td>
<td>n</td>
<td>cl</td>
<td>r</td>
</tr>
</tbody>
</table>
II. Frame for the following verbs is given below:

(1) bola (bola%VDT%S3%FID1), (2) kaha (kaha%VDT%S4%FID1)

<table>
<thead>
<tr>
<th>arc-label</th>
<th>necessity</th>
<th>vibhakti</th>
<th>lexttype</th>
<th>posn</th>
<th>reln</th>
</tr>
</thead>
<tbody>
<tr>
<td>k1</td>
<td>m</td>
<td>0</td>
<td>n</td>
<td>l</td>
<td>c</td>
</tr>
<tr>
<td>k4</td>
<td>m</td>
<td>ko</td>
<td>se</td>
<td>n</td>
<td>l</td>
</tr>
<tr>
<td>k2</td>
<td>m</td>
<td>0</td>
<td>cl</td>
<td>n</td>
<td>r</td>
</tr>
</tbody>
</table>

III. Frame for the following verbs is given below:

(1)bataa (bataa%VDT%S4%FID1), (2)bataa (bataa%VT%S6%FID1), (3)bola (bola%VDT%S4%FID1), (4)kaha (kaha%VDT%S3%FID1), (5)sunaa, (6)bheja (bheja%V?%S1%FID2), (7)Daala (Daala%VT%S2%FID1), (8)Daala (Daala%VT%S2%FID1), (9)lauTaa (lauTaa%VDT%S1%FID1), (10)vaapasa_kara, (11)parosa (parosa%VT%S1%FID1), (12)de (de%VDT%S1%FID1), (13)sauMpa (sauMpa%VDT%S1%FID1), (14)likha, (15)baa.NTa (baa.NTa%VT%S1%FID1)

<table>
<thead>
<tr>
<th>arc-label</th>
<th>necessity</th>
<th>vibhakti</th>
<th>lexttype</th>
<th>posn</th>
<th>reln</th>
</tr>
</thead>
<tbody>
<tr>
<td>k1</td>
<td>m</td>
<td>0</td>
<td>n</td>
<td>l</td>
<td>c</td>
</tr>
<tr>
<td>k4</td>
<td>m</td>
<td>ko</td>
<td>n</td>
<td>l</td>
<td>c</td>
</tr>
<tr>
<td>k2</td>
<td>m</td>
<td>0</td>
<td>cl</td>
<td>n</td>
<td>r</td>
</tr>
</tbody>
</table>

IV. Frame for the following verbs is given below:

(1)bataa (bataa%VDT%S3%FID1)

<table>
<thead>
<tr>
<th>arc-label</th>
<th>necessity</th>
<th>vibhakti</th>
<th>lexttype</th>
<th>posn</th>
<th>reln</th>
</tr>
</thead>
<tbody>
<tr>
<td>k1</td>
<td>m</td>
<td>0</td>
<td>n</td>
<td>l</td>
<td>c</td>
</tr>
<tr>
<td>k4</td>
<td>m</td>
<td>ko</td>
<td>n</td>
<td>l</td>
<td>c</td>
</tr>
<tr>
<td>k2</td>
<td>m</td>
<td>0</td>
<td>cl</td>
<td>r</td>
<td>c</td>
</tr>
</tbody>
</table>
V.Frame for the following verbs is given below:

(1) bataa (bataa%VDT%S2%FID1)

<table>
<thead>
<tr>
<th>arc-label</th>
<th>necessity</th>
<th>vibhakti</th>
<th>lexttype</th>
<th>posn</th>
<th>reln</th>
</tr>
</thead>
<tbody>
<tr>
<td>k1</td>
<td>m</td>
<td>0</td>
<td>n</td>
<td>1</td>
<td>c</td>
</tr>
<tr>
<td>k4</td>
<td>m</td>
<td>ko</td>
<td>n</td>
<td>1</td>
<td>c</td>
</tr>
<tr>
<td>k2</td>
<td>m</td>
<td>ke_baare_meM</td>
<td>n</td>
<td>1</td>
<td>c</td>
</tr>
</tbody>
</table>

VI.Frame for the following verbs is given below:

(1) bola (bola%VT%S2%FID1), (2) kaha (kaha%VDT%S1%FID1)

<table>
<thead>
<tr>
<th>arc-label</th>
<th>necessity</th>
<th>vibhakti</th>
<th>lexttype</th>
<th>posn</th>
<th>reln</th>
</tr>
</thead>
<tbody>
<tr>
<td>k1</td>
<td>m</td>
<td>0</td>
<td>n</td>
<td>1</td>
<td>c</td>
</tr>
<tr>
<td>k4</td>
<td>m</td>
<td>ko</td>
<td>n</td>
<td>1</td>
<td>c</td>
</tr>
<tr>
<td>k2</td>
<td>m</td>
<td>ko</td>
<td>ke_liye</td>
<td>v</td>
<td>1</td>
</tr>
</tbody>
</table>

VII.Frame for the following verbs is given below:

(1) pahuMchaa (pahuMchaa%VT%S2%FID1)

<table>
<thead>
<tr>
<th>arc-label</th>
<th>necessity</th>
<th>vibhakti</th>
<th>lexttype</th>
<th>posn</th>
<th>reln</th>
</tr>
</thead>
<tbody>
<tr>
<td>k1</td>
<td>m</td>
<td>0</td>
<td>n</td>
<td>1</td>
<td>c</td>
</tr>
<tr>
<td>k2</td>
<td>m</td>
<td>0</td>
<td>ko</td>
<td>n</td>
<td>1</td>
</tr>
<tr>
<td>k4</td>
<td>m</td>
<td>taka</td>
<td>n</td>
<td>1</td>
<td>c</td>
</tr>
</tbody>
</table>
VIII. Frame for the following verbs is given below:

(1) Cho.Da

<table>
<thead>
<tr>
<th>arc-label</th>
<th>necessity</th>
<th>vibhakti</th>
<th>lexttype</th>
<th>posn</th>
<th>reln</th>
</tr>
</thead>
<tbody>
<tr>
<td>k1</td>
<td>m</td>
<td>0</td>
<td>n</td>
<td>1</td>
<td>c</td>
</tr>
<tr>
<td>k4</td>
<td>m</td>
<td>para</td>
<td>n</td>
<td>1</td>
<td>c</td>
</tr>
<tr>
<td>k2</td>
<td>m</td>
<td>0</td>
<td>n</td>
<td>1</td>
<td>c</td>
</tr>
</tbody>
</table>

In the above verb frame, i.e., \( k1 + k2 + k4 + \text{verb} \), we have listed all the verbs that take this verb frame under the verb class field. Certain verbs listed in the verb class have semantic similarities such as Communication verbs and Give verbs. Semantic similarities are not found in the verbs of all verb frames (karaka relations). We have then sub-divided the verbs based on similar vibhaktis and other information given in the verb frame (tabular form). There are 8 sub-classes of verbs based on this sub-division.

6.2.2.2 Verb Frame: \( k1 + k2 + k3 + \text{verb} \) (Based on the verb frame 34)

Verb Class: (Verbs having the above karakas)

(1)bhar (bhar\%VT\%S2\%FID1), (2)jiit (jiit\%VT\%S2\%FID1), (3)kas (kas\%VT\%S1\%FID1), (4)kas, (5)kaaT (kaaT\%VT\%S2\%FID1), (6)katar (katar\%VT\%S1\%FID1), (7)kaaT, (8)pho.Da (pho.Da\%VT\%S1\%FID1), (9)to.Da, (10)dho (dho\%VT\%S1\%FID1), (11)pota (pota\%VT\%S1\%FID1), (12)poMCha (poMCha\%VT\%S1\%FID1), (13)baa.Ndha (baa.Ndha\%VT\%S1\%FID1), (14)bujhaa (bujhaa\%VT\%S2\%FID1), (15)miTaa, (16)laad (laad\%VT\%S1\%FID2), (17)maar (maar\%VT\%S1\%FID1), (18)piiT (piiT\%VT\%S1\%FID1), (19)Tho.ka|Tho.Nka, (20)piTaa\_kara, (21)miTaa (miTaa\%VT\%S1\%FID1), (22)lubhaa (lubhaa\%VT\%S1\%FID1), (23)moha, (24)rijhaa

Above verbs are sub-divided according to vibhaktis and other information given in the detailed frame. Detailed frames are given below:
I. Frame for the following verbs is given below:

(1)bhar (bhar%VT%S2%FID1), (2)kas (kas%VT%S2%FID1), (3)kaaT (kaaT%VT%S2%FID1), (4)katar (katar%VT%S1%FID1),(5)kaaT,(6)pho.Da(pho.Da%VT%S1%FID1),(7)to.Da,(8)dho (dho%VT%S1%FID1), (9)pota (pota%VT%S1%FID1), (10)poM Cha (poM Cha%VT%S1%FID1), (11)baa.Ndha (baa.Ndha%VT%S1%FID1), (12)bujhaa (bujhaa%VT%S2%FID1), (13)miTaa

<table>
<thead>
<tr>
<th>arc-label</th>
<th>necessity</th>
<th>vibhakti</th>
<th>lexttype</th>
<th>posn</th>
<th>reln</th>
</tr>
</thead>
<tbody>
<tr>
<td>k1</td>
<td>m</td>
<td>0</td>
<td>n</td>
<td>1</td>
<td>c</td>
</tr>
<tr>
<td>k2</td>
<td>m</td>
<td>0</td>
<td>ko</td>
<td>n</td>
<td>1</td>
</tr>
<tr>
<td>k3</td>
<td>d</td>
<td>se</td>
<td>n</td>
<td>1</td>
<td>c</td>
</tr>
</tbody>
</table>

II. Frame for the following verbs is given below:

(1) jiit (jiit%VT%S2%FID1)

<table>
<thead>
<tr>
<th>arc-label</th>
<th>necessity</th>
<th>vibhakti</th>
<th>lexttype</th>
<th>posn</th>
<th>reln</th>
</tr>
</thead>
<tbody>
<tr>
<td>k1</td>
<td>m</td>
<td>0</td>
<td>n</td>
<td>1</td>
<td>c</td>
</tr>
<tr>
<td>k3</td>
<td>m</td>
<td>se</td>
<td>n</td>
<td>1</td>
<td>c</td>
</tr>
<tr>
<td>k2</td>
<td>m</td>
<td>0</td>
<td>n</td>
<td>1</td>
<td>c</td>
</tr>
</tbody>
</table>

III. Frame for the following verbs is given below:

(1) laad (laad%VT%S1%FID2)

<table>
<thead>
<tr>
<th>arc-label</th>
<th>necessity</th>
<th>vibhakti</th>
<th>lexttype</th>
<th>posn</th>
<th>reln</th>
</tr>
</thead>
<tbody>
<tr>
<td>k1</td>
<td>m</td>
<td>0</td>
<td>n</td>
<td>1</td>
<td>c</td>
</tr>
<tr>
<td>k2</td>
<td>m</td>
<td>ko</td>
<td>n</td>
<td>1</td>
<td>c</td>
</tr>
<tr>
<td>k3</td>
<td>m</td>
<td>se</td>
<td>n</td>
<td>1</td>
<td>c</td>
</tr>
</tbody>
</table>
IV. Frame for the following verbs is given below:

(1) maar (maar%VT%S1%FID1), (2) piiT (piiT%VT%S1%FID1), (3) Thoka|Tho.Nka, (4) piTaaii_kara,
(5) miTaa (miTaa%VT%S1%FID1), (6) lubhaa (lubhaa%VT%S1%FID1), (7) moha, (8) righaa

<table>
<thead>
<tr>
<th>arc-label</th>
<th>necessity</th>
<th>vibhakti</th>
<th>lextype</th>
<th>posn</th>
<th>reln</th>
</tr>
</thead>
<tbody>
<tr>
<td>k1</td>
<td>m</td>
<td>0</td>
<td>n</td>
<td>l</td>
<td>c</td>
</tr>
<tr>
<td>k2</td>
<td>m</td>
<td>ko</td>
<td>n</td>
<td>l</td>
<td>c</td>
</tr>
<tr>
<td>k3</td>
<td>d</td>
<td>se</td>
<td>n</td>
<td>l</td>
<td>c</td>
</tr>
</tbody>
</table>

In the above verb frame, i.e., $k1 + k2 + k3 + \text{verb}$, we have listed the verbs that take this verb frame under the verb class. In this verb class, we found that there are certain groups of 2 to 3 verbs that have certain semantic similarity. We have not listed the semantic similarities of these verbs as the number of verbs that share the semantic similarity is very small. There are 4 sub-classes of verbs based on the vibhakti and other information.

Syntactic and semantic aspects of the various occurrences of these verbs were studied and generalizations were arrived at. Also, the basis for classification of verbs was considered carefully. The basis for verb classification was more syntactico-semantic than either purely syntactic or purely semantic. We have classified the verbs based on their dependency relations (karaka relations and other than karakas) given in the verb frames. Karaka relations are syntactico-semantic 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]. We again sub-classified the verbs based on the vibhaktis, necessity and lex-type given in the verb frame. We did this because, we saw that verbs had the same dependency relations but they varied in case of vibhaktis. We did sub-classification to capture finer semantics among the verbs. We also considered necessity and lex-type also as a base for classification because they also may help in getting finer classes.

After the classification of verb frames for Hindi, I did the comparison of dependency annotation with Propbank annotation.
6.3 Comparison of Dependency Annotation with Propbank Annotation

6.3.1 Hindi PropBank:

PropBank is a language resource which contains a huge annotated corpus having information related to the argument structure of predicates. It builds a semantic layer of annotation that adds predicate argument structure to syntactic representations [139]. The Hindi PropBank annotations are carried out upon the Hindi dependency Treebank [42, 215]. For each verb, PropBank creates a framefile and adds information about the arguments that the verb takes. The arguments of the verbs are marked with the help of numbered arguments, e.g. Arg0, Arg1, Arg2, etc., which are called as semantic role labels. Below given is the framefile for the verb *khola* open [240]:

```
khol : ‘to open’ (Transitive verb and takes two arguments: Arg0 and Arg1)
```

Ex-(6.29) *raam ne khidakii kholii*  
ram  Erg. window opened  
‘Ram opened the window.’

Arg0 : *raam*  ‘ram’  \(\rightarrow\) Opener
Arg1 : *khidakii*  ‘window’  \(\rightarrow\) Entity opened

The numbered arguments of each verb in its framefile, are linked with fine-grained verb-specific definitions. For instance, Arg0 of the verb *khola* is the ‘agent’ who does the action of opening (‘opener’) and Arg1 is the entity that is affected by opening (‘window’). Whenever the verb *khola* occurs in a sentence then it will have the two arguments defined in its framefile with the above mentioned semantic roles. The same numbered argument should be consistent across various syntactic realizations of the same argument of the verb. For example, *khidakii* ‘window’ should be marked as Arg1 in (6.29) and (6.30) where it is the object and subject of the verb, respectively. *khidakii* gets the same numbered argument label as it has the same semantic role in both the examples because it undergoes the action of motion in both the sentences [240].

Ex-(6.30) *khidakii khulii*  
window opened  
‘The window opened.’

This type of consistency in marking semantic role labeling is very useful in the training of machine learning systems like automatic semantic role labellers. However with the change of sense of the verb, there will be another set of semantic role labels available for that sense. For example, the verb *khil* has two senses [240]:

46
In the above examples, the verb *khil* has two senses: ‘to bloom’ and ‘to look good’. For sense1 in example 6.31, the verb *khil ‘to bloom’* takes ARG1 (*phuul ‘flower’*) ‘entity that undergoes the action of blooming’. For sense2 in example 6.32, the verb *khil ‘to look good’* takes ARG1 (*saadii ‘sari’*) ‘entity that looks good on someone’ and ARG2_LOC (*siitaa ‘Sita’) ‘entity that is looking good’ [240].

Extra information (modifiers) like when and where the action occurred is not considered as the part of the semantic specifications of the verb and hence not included in the framefile. But in case if such information occurs in the sentence then it needs to be annotated. For example, ‘Ram opened the window in the bedroom yesterday’; the phrases ‘in the bedroom’ and ‘yesterday’ which give extra information about time and space are annotated. These types of modifiers are represented with the help of specific set of labels to denote this sort of information specified by these modifiers. These modifiers have the following labels: ArgM-LOC, ArgM-TMP, ArgM-MNR. In the frame file of the verb, these modifiers are not included in the main arguments but if the example contains these additional modifiers then they are annotated using the above mentioned modifier tags [240].

Hindi-Urdu language permits the speakers to delete the arguments of the verb in the discourse. In the example, ‘Ram opened the window’, if ‘Ram’ has been mentioned previously, then the speaker can say ‘opened window’ without uttering ‘Ram’ (subject). There will be many sentences where the arguments of the verbs will be omitted and are retrievable from the context. In case of Hindi-Urdu, PropBank doesn’t add empty arguments to syntactic trees. PropBank inserts the core empty arguments of the verb (subject, object or indirect object) and then assigns semantic role labels to it. The information in the verb framefiles is very helpful as it permits in regaining different kinds of empty arguments before they are marked with semantic role labels. Obligatorily null arguments of a verb are also considered: (i) complement clauses of certain verbs taking null subjects such as *chaaha ‘want’* (*siitaa NULL khaanaa pakaana chaahatii hai ‘Sita wants [NULL to cook food]’); (b) Adjunct clauses taking null subjects (*mai ghar jaakar so gayaa [ NULL going home] I slept’); (c) participial modifiers having gapped arguments (*Kile PUla [NULL] blossomed flowers’); (d) coordinate constructions having omitted arguments (*raam ne khaanaa khaayaa aur paanii piyaa ‘Ram ate food and [NULL] drank water’). An example 6.33 is shown below where a null element is inserted in a sentence with a complement clause [240]:

Ex-(6.33) *siitaa, [PRO, khaanaa pakaanaa] chaahhtii hai*  
  *sita PRO food to cook want is*  
  ‘Sita wants to cook food.’  

In the above example, PRO which is an empty category is inserted and annotated similarly as an overt argument. Here the verb *pakaas ‘cook’* assigns ARG0 semantic role label to PRO [240].
I have also worked on the mapping between Propbank annotation and dependency annotation based on Paninian Grammatical Framework [21, 36]. In this work, I have taken the Propbank annotation and dependency annotation done on the Hindi data and compared both the annotations to get a mapping between them. This mapping will help us in analysing the relation between dependency annotation and propbank annotation.

6.3.2 Initial Comparison of Dependency Annotation with Propbank Annotation

6.3.2.1 Creating Hindi Framesets

The goal of this work is to outline first steps in creating Hindi framesets for verbs taken from a sample of 110 sentences selected from the Webdunia and Jagaran corpora. For each verb in this list, a lexical entry was created consisting of the following information:

1. English translation equivalent (e.g. bataa 'tell')
2. Paninian karaka relations (e.g., kartaa-k1, karma-k2)
3. Theta roles (e.g., Agent, Patient)
4. Propbank roles (e.g. A0, A1, A2)
5. The optionality/obligatoriness of the argument
6. The mapping between the roles in these three frameworks (Paninian, Theta roles, Propbank roles)
7. An example sentence (with an English gloss) in which the karaka relations and Propbank roles are annotated. Example sentences were taken from the corpus or from Hindi WordNet; if the sentences were complex, the minimal clause containing the main verb was chosen to exemplify the semantic/syntactic roles.
8. If the verb had more than one sense, each sense was separately represented with all of information given above in (1)-(7).

A sample verb entry is provided below in the figure 6.12:
In the above figure, the frameset for the Hindi verb *aa* ‘come’ is given. The argument relations are given in three frame works i.e., *karaka* relations, Theta roles, and Propbank. According to the example sentence given, it has the following roles, i.e. *k1* (*karta*), *k2p* (*Goal*), and *k7t* (*kaladhikaran*) in *karaka* relations; *Agent*, *Goal*, and *Time* in Theta roles and; *Arg1: entity in motion ‘comer’, Arg4: end point* and *ArgM-TMP* in Propbank respectively. *k1* (*shraddhaalu ‘devotees’) and *k2p* (*yahaa ‘here’) are mandatory; *k7t* (*saal bhar ‘year long’) is optional.
6.3.2.1.1 Verb Selection

110 verb chunks (main verb + auxiliary verb) were extracted from Webdunia and Jagaran corpora. The main verb tokens were taken from the verb chunks. There were 58 different verbs. They are listed below:

Verbs (wx notation)

<table>
<thead>
<tr>
<th>wx notation</th>
<th>Roman notation</th>
<th>Gloss</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. A</td>
<td>aa</td>
<td>'come'</td>
<td>6</td>
</tr>
<tr>
<td>2. Bara</td>
<td>bhar</td>
<td>'fill' (intr.)</td>
<td>1</td>
</tr>
<tr>
<td>3. Cipa</td>
<td>Chip</td>
<td>'hide'</td>
<td>1</td>
</tr>
<tr>
<td>4. Coda</td>
<td>Cho.Da</td>
<td>'leave'</td>
<td>2</td>
</tr>
<tr>
<td>5. KilA</td>
<td>kilaA</td>
<td>'feed'</td>
<td>2</td>
</tr>
<tr>
<td>6. Koja</td>
<td>khoja</td>
<td>'search'</td>
<td>1</td>
</tr>
<tr>
<td>7. bETa</td>
<td>baiTha</td>
<td>'sit'</td>
<td>1</td>
</tr>
<tr>
<td>8. baDa</td>
<td>ba.Dha</td>
<td>'increase'</td>
<td>3</td>
</tr>
<tr>
<td>9. baca</td>
<td>bacha</td>
<td>'remain'</td>
<td>1</td>
</tr>
<tr>
<td>10. baha</td>
<td>baha</td>
<td>'flow'</td>
<td>1</td>
</tr>
<tr>
<td>11. bjaq</td>
<td>baja</td>
<td>'ring'</td>
<td>1</td>
</tr>
<tr>
<td>12. banA</td>
<td>banaa</td>
<td>'make'</td>
<td>4</td>
</tr>
<tr>
<td>13. banavA</td>
<td>banavaa</td>
<td>'cause to make'</td>
<td>2</td>
</tr>
<tr>
<td>14. bana</td>
<td>bana</td>
<td>'become'</td>
<td>2</td>
</tr>
<tr>
<td>15. basA</td>
<td>basaa</td>
<td>'settle'</td>
<td>1</td>
</tr>
<tr>
<td>16. bawA</td>
<td>bataa</td>
<td>'tell'</td>
<td>2</td>
</tr>
<tr>
<td>17. bazXa</td>
<td>baMdha</td>
<td>'tie'</td>
<td>1</td>
</tr>
<tr>
<td>18. cAha</td>
<td>chaah</td>
<td>'wish'</td>
<td>1</td>
</tr>
<tr>
<td>19. calA</td>
<td>chalaa</td>
<td>'drive'</td>
<td>1</td>
</tr>
<tr>
<td>20. cala</td>
<td>chal</td>
<td>'walk'</td>
<td>2</td>
</tr>
<tr>
<td>21. gUMja</td>
<td>guuMja</td>
<td>'echo'</td>
<td>1</td>
</tr>
<tr>
<td>22. girA</td>
<td>gir</td>
<td>'fall'</td>
<td>1</td>
</tr>
<tr>
<td>23. hE</td>
<td>hai</td>
<td>'be'</td>
<td>2</td>
</tr>
<tr>
<td>24. jA</td>
<td>jaa</td>
<td>'go'</td>
<td>2</td>
</tr>
<tr>
<td>25. KA</td>
<td>khaa</td>
<td>'eat'</td>
<td>1</td>
</tr>
<tr>
<td>26. kaha</td>
<td>kaha</td>
<td>'tell'</td>
<td>2</td>
</tr>
<tr>
<td>27. kara</td>
<td>kara</td>
<td>'do'</td>
<td>2</td>
</tr>
<tr>
<td>28. karavA</td>
<td>karavaa</td>
<td>'cause to do'</td>
<td>4</td>
</tr>
<tr>
<td>29. laA</td>
<td>laa</td>
<td>'bring'</td>
<td>2</td>
</tr>
<tr>
<td>30. lOta</td>
<td>laaT</td>
<td>'return'</td>
<td>1</td>
</tr>
<tr>
<td>31. lagA</td>
<td>lagaa</td>
<td>'wear'</td>
<td>1</td>
</tr>
<tr>
<td>32. laga</td>
<td>laga</td>
<td>'seem'</td>
<td>2</td>
</tr>
<tr>
<td>33. le</td>
<td>le</td>
<td>'take'</td>
<td>3</td>
</tr>
<tr>
<td>34. mAa</td>
<td>maan</td>
<td>'believe'</td>
<td>3</td>
</tr>
<tr>
<td>35. mila</td>
<td>mil</td>
<td>'got'</td>
<td>2</td>
</tr>
<tr>
<td>36. moha</td>
<td>moha</td>
<td>'attract'</td>
<td>1</td>
</tr>
<tr>
<td>37. nikAla</td>
<td>nikaal</td>
<td>'remove'; 'dismiss'</td>
<td>1</td>
</tr>
<tr>
<td>Verb</td>
<td>English Sense</td>
<td>Frequency</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>---------------</td>
<td>-----------</td>
<td></td>
</tr>
<tr>
<td>nikala</td>
<td>'emerge'</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>pI</td>
<td>'drink'</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>paDa</td>
<td>'read'</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>pahuNca</td>
<td>'reach'</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>raKa</td>
<td>'put'</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>raha</td>
<td>'live'</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>suukh</td>
<td>'dry'</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>saNbaal</td>
<td>'look after'</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>savaar</td>
<td>'decorate'</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>sikhaa</td>
<td>'teach'</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>ubhar</td>
<td>'rise, flourish'</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>utar</td>
<td>'climb down'</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>tapa</td>
<td>'scorch'</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>de</td>
<td>'give'</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>deKha</td>
<td>'see'</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>dikh</td>
<td>'visible'</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>dilaa</td>
<td>'cause to give/buy'</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>haNsaa</td>
<td>'laugh'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>khul</td>
<td>'open'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dauD</td>
<td>'run'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nahaa</td>
<td>'bathe'</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

There are 6 verbs for which framesets have not been provided because of not getting proper English sense: (15) basaa "settle", (48) ubhara "rise, flourish", (49) utara 'climb down', (50) tapa 'scorch', (53) dikha 'visible', (54) dilaa 'cause to give/buy'.

### 6.3.2.2 Framesets for Hindi Verbs

I did a comparative study of Hindi verbs in both dependency annotation and Propbank annotation, where I found that karta maps with both Arg0 and Arg1 whereas karma maps with Arg2. In Propbank, missing elements are denoted by trace (t) whereas in dependency annotation we are inserting Null chunks for missing elements which are head nodes and have its arguments in the sentence which are its children. I did create frame files for Hindi Propbank but I didn’t compare it with English Propbank as we didn’t take this as one of the objective of our work.

PropBank framesets are created for 58 Hindi verbs. I created these framesets during my visit to Colorado. Below given are few framesets for Hindi verbs:
(1) **raha** sense1: exist

Roles:

<table>
<thead>
<tr>
<th>Paninian</th>
<th>Theta Roles</th>
<th>Propbank</th>
</tr>
</thead>
<tbody>
<tr>
<td>k1 (karta)</td>
<td>Theme</td>
<td>Arg1: thing existing</td>
</tr>
</tbody>
</table>

Necessity of the Argument: k1 $\rightarrow$ m (mandatory)

Ex:

*yahaa chaibiisoM ghaMTe din rehata hai*

here 24 hours day exist be.Pres

‘Here it is day for 24 hours.’

**REL:**

rehata ‘exist’

- k1 $\rightarrow$ Arg1: *din* ‘day’
- k7p $\rightarrow$ Argm-Loc: *yahaa* ‘here’
- k7t $\rightarrow$ ArgM-TMP: *chaubiisoM ghaMTe* ‘24hours’

**raha** sense2: live

Roles:

<table>
<thead>
<tr>
<th>Paninian</th>
<th>Theta Roles</th>
<th>Propbank</th>
</tr>
</thead>
<tbody>
<tr>
<td>k1 (karta)</td>
<td>Theme</td>
<td>Arg0: alive entity</td>
</tr>
<tr>
<td>k7p (deshadhikaran)</td>
<td>Location</td>
<td>Argm-LOC: medium</td>
</tr>
</tbody>
</table>

Necessity of the Argument: k1,k7p $\rightarrow$ m (mandatory)
Ex:

raam apane dost ke saath usake ghar par rehata hai
ram his friend with his house at live.Impf be.Pres
'Ram stays with his friend at his home.'

REL: raha 'live'

k1 → Arg0 raam 'Ram'
k7p → Argm-Loc usake ghar par 'at his house'
ras → Argn-Mnr apane dost ke saath 'with his friend'
(relation associative)

(2) de sense1:give

Roles:

<table>
<thead>
<tr>
<th>Paninian</th>
<th>Theta Roles</th>
<th>PropBank</th>
</tr>
</thead>
<tbody>
<tr>
<td>k1</td>
<td>Agent</td>
<td>Arg0:giver</td>
</tr>
<tr>
<td>k2</td>
<td>Theme</td>
<td>Arg1:thing given</td>
</tr>
<tr>
<td>k4</td>
<td>Beneficiary</td>
<td>Arg2:entity given to</td>
</tr>
</tbody>
</table>

Necessity of the Argument: k1,k2,k4 → m (mandatory)

Eg:

maine raam ko kitaab dii
I.Erg ram Acc. book give.Past
'I gave book to Ram.'

REL: de 'give'

k1 → Arg0: mai 'I'
k2 → Arg1: kitaab 'book'
k4 → Arg2: raam ko 'to Ram'
(3) kaha sense2:call

Roles:

<table>
<thead>
<tr>
<th>Paninian</th>
<th>Theta Roles</th>
<th>PropBank</th>
</tr>
</thead>
<tbody>
<tr>
<td>k1 (karta)</td>
<td>Agent</td>
<td>Arg0: caller</td>
</tr>
<tr>
<td>k2 (karma)</td>
<td>Theme</td>
<td>Arg1: item being labelled</td>
</tr>
<tr>
<td>k2g</td>
<td>Predicate</td>
<td>Arg2: attribute of arg1</td>
</tr>
</tbody>
</table>

(gauna karma)

Necessity of the Argument: k1, k2, k2g → m (mandatory)

Eg:

loga gaaMdhi jii ko baapuu bhi kehate hai
people Gandhi.hon Acc. bapu also call be.Pres
'People also call Gandhiji as Bapu.'

REL: keha 'call'
k1 → Arg0: loga 'people'
k2 → Arg1: gaaMdhi jii ko 'Gandhi'
k2g → Arg2: baapuu 'Bapu'

6.3.2.3 Issues in the Framesets

6.3.2.3.1 Lexical vs. Constructional Meaning

As mentioned in chapter 4 and 5, in Hindi conjunct verbs are a combination of verb and a noun or verb and an adjective. It is not just the verb, but the combination of the noun/adjective+verb construction which conveys the action.

Ex-(6.34) vaha snaan (pof) kar rahaa hai
he bath do Prog. be.Pres
'He is bathing.'
In the example 6.34 above, the meaning ‘bathe’ is not a sense of the general or ‘light’ verb kar ‘do’, but is associated with the N+V combination as a whole. In the case of noun+verb conjuncts, the question arises as to whether the noun is an argument of the verb, or whether it should be treated as part of a complex predicate. That is, is snaan kar an intransitive (complex) predicate that takes one argument, the “bather”, or is it a transitive predicate, the second argument “bath” being treated as an argument of the verb at one level but as part of the predicate meaning at a different level. Earlier, for the purposes of propbanking, it seems that the latter option was chosen for equivalent cases in English (cf. he took a bath). Now Hwang et. al., [87] has given a method for the annotation of light verbs across Hindi, Arabic, Chinese, and English PropBanks. This annotation method comprises of 3 passes: (i) manually identifying the light verb constructions, (ii) annotating the arguments based on light verb’s Frame File, and (ii) deterministic merging of the above two passes. The above process becomes easy in Hindi, with the help of the dependency label ‘pof’ that identifies a light verb [180].

1) Identify N+V sequences.

2) Annotate predicing expression with ARG-PRX.

   REL: kar ‘do’
   ARG-PRX: snaan ‘bath’

3) Annotate the arguments and modifiers of the complex predicate with a nominal predicate frame.

4) Automatically merge the nominal with the light verb.

   REL: snaan_kiyaa ‘bathe’
   Arg0: raam ‘Ram’

6.3.2.3.2 The Indirect Causer

Hindi has a morphological affix vaa that indicates indirect causation and (sometimes) adds an additional argument:

Ex-(6.35) maaz ne (pk1) mohan se (mk1) raam ko (jk1) gaanaa(k2) sikhvaayaa.
   mother Erg. mohan by ram Dat/Acc song caused-to-learn
   Mother made Mohan to teach songs to Ram. [123]

The secondary causer (mediator-causer) mohan ‘Mohan’ in the example above corresponds to “ARGA” in Prophbank. In Prophbank, earlier there was no role for the outermost causer argument in Hindi i.e., for maaz “mother” in the example given above. It was an issue for the mapping of dependency and propbank annotation for Hindi. But, now an additional role has been created for causer, i.e., ARGC in Prophbank.
Below given is the table which contains the mapping (we have listed some of the mappings from the framesets that we have created for Hindi verbs) between karaka roles, theta roles, and propbank roles. The table also contains the verb and its sense information which has these mappings.

<table>
<thead>
<tr>
<th>karaka</th>
<th>Prophank</th>
<th>Theta role</th>
<th>Verb</th>
<th>Sense</th>
</tr>
</thead>
<tbody>
<tr>
<td>k1(karta)</td>
<td>Arg1: substance</td>
<td>Theme</td>
<td>baha</td>
<td>sense1: flow</td>
</tr>
<tr>
<td></td>
<td>Arg1: thing existing</td>
<td></td>
<td>raha</td>
<td>sense1: exist</td>
</tr>
<tr>
<td></td>
<td>Arg0: alive entity</td>
<td></td>
<td>nikala</td>
<td>sense2: emerge</td>
</tr>
<tr>
<td></td>
<td>Arg1: thing increasing</td>
<td></td>
<td>raha</td>
<td>sense2: exist</td>
</tr>
<tr>
<td></td>
<td>Arg1: thing grown</td>
<td></td>
<td>ba.Dha</td>
<td>sense1: increase</td>
</tr>
<tr>
<td></td>
<td>Arg1: Logical subject, patient, thing rising</td>
<td></td>
<td>ba.Dha</td>
<td>sense2: grow</td>
</tr>
<tr>
<td></td>
<td>Arg0: walker</td>
<td></td>
<td>nikala (synonym of 'uga')</td>
<td>sense1: rise</td>
</tr>
<tr>
<td></td>
<td>Arg1: entity in motion/goer</td>
<td></td>
<td>chala</td>
<td>sense1: walk</td>
</tr>
<tr>
<td></td>
<td>Arg1: entity in motion / 'comer'</td>
<td>Agent</td>
<td>jaa</td>
<td>sense1: go</td>
</tr>
<tr>
<td></td>
<td>Arg0: agent, doer</td>
<td></td>
<td>aa</td>
<td>sense1: come</td>
</tr>
<tr>
<td></td>
<td>Arg0: Taker</td>
<td></td>
<td>kara</td>
<td>sense1: do</td>
</tr>
<tr>
<td></td>
<td>Arg0: buyer</td>
<td></td>
<td>le</td>
<td>sense1: take</td>
</tr>
<tr>
<td></td>
<td>Arg0: creator</td>
<td></td>
<td>le</td>
<td>sense2: buy</td>
</tr>
<tr>
<td></td>
<td>Arg0: believer</td>
<td></td>
<td>banaa</td>
<td>sense1: make</td>
</tr>
<tr>
<td></td>
<td>Arg0: reacheser</td>
<td></td>
<td>maan</td>
<td>sense1: believe</td>
</tr>
<tr>
<td></td>
<td>Arg0: giver</td>
<td></td>
<td>pahu.Ncha</td>
<td>sense1: reach</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>de</td>
<td>sense1: give</td>
</tr>
<tr>
<td>Role</td>
<td>Argument</td>
<td>Aspect</td>
<td>Role</td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------</td>
<td>-----------------------</td>
<td>---------------</td>
<td></td>
</tr>
<tr>
<td>Arg0</td>
<td>entity leaving</td>
<td>Cho.Da</td>
<td>sense1:leave</td>
<td></td>
</tr>
<tr>
<td>Arg0</td>
<td>entity doing the dropping</td>
<td>Cho.Da</td>
<td>sense2:drop</td>
<td></td>
</tr>
<tr>
<td>Arg0</td>
<td>Speaker</td>
<td>bataa</td>
<td>sense1:tell</td>
<td></td>
</tr>
<tr>
<td>Arg0</td>
<td>Speaker</td>
<td>kaha</td>
<td>sense1:tell</td>
<td></td>
</tr>
<tr>
<td>Arg0</td>
<td>regarder, respecter, agent</td>
<td>Experiencer</td>
<td>sense2:respect</td>
<td></td>
</tr>
<tr>
<td>Arg1</td>
<td>thing changing</td>
<td>Patient</td>
<td>sense2:turn</td>
<td></td>
</tr>
<tr>
<td>Arg2</td>
<td>seeming to</td>
<td>laga</td>
<td>sense1:seem</td>
<td></td>
</tr>
<tr>
<td>Arg0</td>
<td>meeter</td>
<td>Actor1</td>
<td>sense1:meet</td>
<td></td>
</tr>
<tr>
<td>Argm</td>
<td>direction or location or extent</td>
<td>Direction</td>
<td>sense1: flow</td>
<td></td>
</tr>
<tr>
<td>Argm</td>
<td>direction or location or extent</td>
<td>Location</td>
<td>sense1: flow</td>
<td></td>
</tr>
<tr>
<td>Argm-LOC: medium</td>
<td></td>
<td>raha</td>
<td>sense2: live</td>
<td></td>
</tr>
<tr>
<td></td>
<td>nikala (synonym of 'uga')</td>
<td></td>
<td>sense1: rise</td>
<td></td>
</tr>
<tr>
<td>Argm</td>
<td>direction or location or extent</td>
<td>Source</td>
<td>sense1: flow</td>
<td></td>
</tr>
<tr>
<td>Arg2</td>
<td>entity taken from, or prepositional complement of arg1</td>
<td>le</td>
<td>sense1: take</td>
<td></td>
</tr>
<tr>
<td>Arg2</td>
<td>seller</td>
<td>le</td>
<td>sense2: buy</td>
<td></td>
</tr>
<tr>
<td>Arg1</td>
<td>thing/place exited from</td>
<td>nikala</td>
<td>sense2: emerge</td>
<td></td>
</tr>
<tr>
<td>Arg3</td>
<td>start point</td>
<td>jaa</td>
<td>sense1: go</td>
<td></td>
</tr>
<tr>
<td><strong>k2p</strong> (goal)</td>
<td><strong>Argm:</strong> direction or location or extent</td>
<td><strong>Goal</strong></td>
<td>baha</td>
<td>sense1: flow</td>
</tr>
<tr>
<td>----------------</td>
<td>------------------------------------------</td>
<td>---------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>Arg4: end point</td>
<td></td>
<td>Goal</td>
<td>aa</td>
<td>sense1: come</td>
</tr>
<tr>
<td>Arg2: goal</td>
<td></td>
<td></td>
<td>Cho.Da</td>
<td>sense2: drop</td>
</tr>
<tr>
<td>Arg4: end point, end state of arg1</td>
<td>jaa</td>
<td>sense1: go</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>k2</strong> (karma)</th>
<th><strong>Arg1:</strong> thing done</th>
<th><strong>Theme</strong></th>
<th>kara</th>
<th>sense1: do</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arg1: thing taken</td>
<td>le</td>
<td>sense1: take</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arg1: thing bought</td>
<td>le</td>
<td>sense2: buy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arg1: belief</td>
<td>maan</td>
<td>sense1: believe</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arg1: entity respected</td>
<td>maan</td>
<td>sense2: respect</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arg1: thing given</td>
<td>de</td>
<td>sense1: give</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arg1: place left</td>
<td>Cho.Da</td>
<td>sense1: leave</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arg1: Logical subject, patient, thing falling</td>
<td>Cho.Da</td>
<td>sense2: drop</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arg2: substance</td>
<td>bhara</td>
<td>sense1: fill</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arg1: creation</td>
<td>Product</td>
<td>banaa</td>
<td>sense1: make</td>
<td></td>
</tr>
<tr>
<td>Arg1: Utterance</td>
<td>Topic</td>
<td>bataa</td>
<td>sense1: tell</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>kaha</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>pk1</strong> (prayojak-karta)</th>
<th><strong>Arg0:</strong> causal agent</th>
<th><strong>Causer</strong></th>
<th>khilaa</th>
<th>sense1: feed</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgA: causative agent</td>
<td></td>
<td>Causer</td>
<td>khilaa</td>
<td>sense1: cause to walk</td>
</tr>
<tr>
<td>Role (Hindi)</td>
<td>Arg2: Action</td>
<td>Arg3: Agent</td>
<td>Sense 1</td>
<td>Sense 2</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
<td>-------------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>jk1 (prayojya-karta)</td>
<td>eater</td>
<td>Cause</td>
<td>khilaa</td>
<td>feed</td>
</tr>
<tr>
<td></td>
<td>walker</td>
<td></td>
<td>calaa</td>
<td>cause to walk</td>
</tr>
<tr>
<td>k7</td>
<td>price paid</td>
<td>Asset</td>
<td>le</td>
<td>buy</td>
</tr>
<tr>
<td></td>
<td>source of joy</td>
<td>Cause</td>
<td>hamsa</td>
<td>laugh</td>
</tr>
<tr>
<td>k4 (sampradan)</td>
<td>entity given to</td>
<td>Beneficiary</td>
<td>de</td>
<td>give</td>
</tr>
<tr>
<td></td>
<td>Hearer</td>
<td>Recipient</td>
<td>bataa</td>
<td>tell</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>kaha</td>
<td></td>
</tr>
<tr>
<td>k1s</td>
<td>end state</td>
<td>Product</td>
<td>bana</td>
<td>turn</td>
</tr>
<tr>
<td>k2g</td>
<td>attribute of arg1</td>
<td>Predicate</td>
<td>kaha</td>
<td>call</td>
</tr>
<tr>
<td>k4a (anubhav karta)</td>
<td>thing seeming</td>
<td>Experiencer</td>
<td>laga</td>
<td>seem</td>
</tr>
<tr>
<td>k3 (karana)</td>
<td>instrument, string</td>
<td>Instrument</td>
<td>barmha</td>
<td>to be tied</td>
</tr>
<tr>
<td></td>
<td>instrument, if separate from attractor</td>
<td></td>
<td>moha</td>
<td>attract</td>
</tr>
<tr>
<td>k4 (sampradana)</td>
<td>benefactive</td>
<td>Benefactive</td>
<td>le</td>
<td>buy</td>
</tr>
</tbody>
</table>

**Table 6.2** Mapping between karaka roles, Theta roles, and Propbank roles

### 6.3.3 Comparison of the Hindi Dependency Structure (HDT) with the Hindi Propbank (HPB)

In this section, we are trying to capture the similarities between Hindi dependency labels and PropBank semantic arguments. PropBank (PB) resource is created for a new language, i.e., Hindi. PropBank annotation is done on syntactic tree structures. In case of English and Chinese PropBanks, the syntactic structure is the phrase structure whereas, Hindi PropBank annotation is done on Hindi dependency structure (DS). There are few advantages when PropBank annotation is done on Hindi
dependency trees. Hindi Treebank has a big set of dependency labels which are rich in semantic information. Hindi PropBank (HPB) contains 26 labels consisting of arguments and modifiers. Verbs have numbered semantic arguments ranging from Arg0 to Arg4 whereas modifiers like ArgM-LOC (location), ArgM-TMP (time), ArgM-PRP (purpose), etc., are common to all the verbs which give additional information [179].

Ex-(6.36)рам ne shyam ko kitaabeM dii
Ram Erg. shyam Acc. books gave
‘Ram gave books to Shyam.’

In the above example 6.36, raam is k1 (agent or doer), shyam is k4 (recipient), and kitaabeM is k2 (patient) of the action. These dependency relations map with PropBank semantic arguments: Arg0, Arg2, and Arg1 respectively. Below given is the table containing 26 labels from HPB [179]:

<table>
<thead>
<tr>
<th>HPB Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARG0</td>
<td>Agent</td>
</tr>
<tr>
<td>ARG1</td>
<td>Patient, theme, undergoer</td>
</tr>
<tr>
<td>ARG2</td>
<td>Beneficiary</td>
</tr>
<tr>
<td>ARG3</td>
<td>Instrument</td>
</tr>
<tr>
<td>ARG2-ATR</td>
<td>attribute</td>
</tr>
<tr>
<td>ARG2-LOC</td>
<td>location</td>
</tr>
<tr>
<td>ARG2-GOL</td>
<td>ARG2-SOU</td>
</tr>
<tr>
<td>ARG2-MNS</td>
<td>Intermediate causer</td>
</tr>
<tr>
<td>ARGA</td>
<td>causer</td>
</tr>
<tr>
<td>ARGA-MNS</td>
<td>Intermediate causer</td>
</tr>
<tr>
<td>ARG0-GOL, ARG0-MNS</td>
<td>causees</td>
</tr>
<tr>
<td>ARGM-VLV</td>
<td>Verb-verb construction</td>
</tr>
<tr>
<td>ARGM-PRX</td>
<td>Noun-verb construction</td>
</tr>
</tbody>
</table>

Table 6.3 Argument Labels in HPB
### 6.3.3.1 Mapping between DS and PB

The tagset of DS (HDT) is to some extent very much similar to PB. The dependency tags specific to verb ranges from k1 to k5; verb modifier tags are k7p, k7t, rh etc; Non-dependency tags are pof, ccof for complex predicates and co-ordination. The HDT tagset has 43 labels. Dependency labels and the dependency trees are in some way useful for deriving PropBank annotations. We can map dependency karaka labels (HDT) with Hindi PropBank labels (HPB) very easily. This kind of mapping is very useful since it speeds up the annotation task, improves the inter-annotator agreement task and also helps in the semantic role labeling task. Below given table shows the mapping between tagset of HDT and HPB [179]:

<table>
<thead>
<tr>
<th>HDT label</th>
<th>HPB label</th>
</tr>
</thead>
<tbody>
<tr>
<td>k1 (karta); k4 a (experiencer)</td>
<td>Arg0</td>
</tr>
<tr>
<td>k2 (karma)</td>
<td>Arg1</td>
</tr>
<tr>
<td>k4 (beneficiary)</td>
<td>Arg2</td>
</tr>
<tr>
<td>k1s (attribute)</td>
<td>Arg2–ATR</td>
</tr>
<tr>
<td>k5 (source)</td>
<td>Arg2–SOU</td>
</tr>
<tr>
<td>k2p (goal)</td>
<td>Arg2–GOL</td>
</tr>
<tr>
<td>k3 (instrument)</td>
<td>Arg3</td>
</tr>
<tr>
<td>mk1 (causer)</td>
<td>ArgC</td>
</tr>
<tr>
<td>pk1 (secondary causer)</td>
<td>ArgA</td>
</tr>
</tbody>
</table>

Table 6.5 Mapping between argument labels of HDT and HPB
Automatic mapping of HDT to HPB is done with the help of rule based, probabilistic system. For this mapping, two types of resources are used: (i) Annotated corpus [Treebank+ PropBank] and (ii) Frame files having mapping rules. There are three types of rules which are used to do automatic mapping: (i) Deterministic rules where dependency labels are directly mapped to PropBank labels. For example, when a child node is marked with POF label (complex predicates) then it gets ARGM-PRX label at the PropBank side, (ii) Empirically derived rules where corpus statistics linked with dependency and PropBank labels are used. The feature tuple contains the following: Predicate ID, Passive or Active Voice (given in HDT), Dependency label. For example, (aa ‘to come’, active, k1). A tuple (aa ‘come’, active, k1) can be linked with Arg0 and Arg1 PropBank labels. If the probability of Arg0 is higher than Arg1, i.e., Arg0 appears more frequently with this feature tuple then Arg0 will be selected, (ii) Linguistically motivated rules is useful for the predicates which are not there in training data. The mapping existing in the frame files is used. This rule is applied only for numbered arguments [179].

In the evaluation, we found out that HPB labels are coarse-grained since they map to 4 separate labels of HDT. It is also fine-grained in the sense, 2 HPB labels (ARGM-MNS ‘means’, ARG-M-CAU ‘causes’) map to one label in HDT [179].

### 6.3.3.2 Linguistic issues

#### 6.3.3.2.1 Causatives

In PropBank frame files, for causatives, the following relation is represented: (i) The same frame file having different rolesets for base and causative verbs, and (ii) This relation is also represented by Frame labels [179].

In Hindi verb frames, different verb frame files are created for base verb and causative verbs [179].

<table>
<thead>
<tr>
<th>HDT label</th>
<th>HPB label</th>
</tr>
</thead>
<tbody>
<tr>
<td>sent-adv (epistemic adv)</td>
<td>ArgM-ADV</td>
</tr>
<tr>
<td>rh (cause/reason)</td>
<td>ArgM-CAU</td>
</tr>
<tr>
<td>rd (direction)</td>
<td>ArgM-DIR</td>
</tr>
<tr>
<td>rad (discourse)</td>
<td>ArgM-DIS</td>
</tr>
<tr>
<td>k7p (location)</td>
<td>ArgM-LOC</td>
</tr>
<tr>
<td>adv (manner adv)</td>
<td>ArgM-MNR</td>
</tr>
<tr>
<td>rt (purpose)</td>
<td>ArgM-PRP</td>
</tr>
<tr>
<td>k7t (time)</td>
<td>ArgM-TMP</td>
</tr>
</tbody>
</table>

Table 6.6 Mapping between modifier labels of HDT and HPB
\textbf{Ex-(6.37)} bachche ne (Arg0/k1) paanii (Arg1/k2) piyaa
\begin{quote}
child\hspace{0.5cm}Erg\hspace{0.5cm}water\hspace{0.5cm}drank
\end{quote}
‘Child drank water’

\textbf{Roleset id: KA.01 to eat}

\begin{tabular}{|l|l|}
\hline
Arg0 & eater \\
\hline
Arg1 & the thing being eaten \\
\hline
\end{tabular}

\textbf{Ex-(6.38)} raam ne (ArgA/pk1) bachche ko (Arg0-GOL/jk1) paanii (Arg1/k2) pilaayaa
\begin{quote}
ram\hspace{0.5cm}Erg\hspace{0.5cm}child\hspace{0.5cm}Dat\hspace{0.5cm}water\hspace{0.5cm}drank-caus
\end{quote}
‘Ram made child to drink water’

\textbf{Roleset id: KilA.01 to feed}

\begin{tabular}{|l|l|}
\hline
ArgA & feeder \\
\hline
Arg0-GOL & eater \\
\hline
Arg1 & the thing being eaten \\
\hline
\end{tabular}

\textbf{Ex-(6.39)} Mohan ne (ArgA/pk1) ram se (ArgA-MNS/mk1) bachche ko (Arg0-GOL/jk1) paanii (Arg1/k2)
mohan\hspace{0.5cm}Erg\hspace{0.5cm}ram\hspace{0.5cm}instr\hspace{0.5cm}child\hspace{0.5cm}Dat\hspace{0.5cm}water
pilaayaa
\begin{quote}
cause someone to drink
\end{quote}
‘Mohan made Ram to cause the child to drink water’

\textbf{Roleset id: KilvA.01 to cause to be fed}

\begin{tabular}{|l|l|}
\hline
ArgA & Causer of feeding \\
\hline
ArgA-MNS & feeder \\
\hline
Arg0-GOL & Eater \\
\hline
Arg1 & the thing eaten \\
\hline
\end{tabular}
The ARG0 label which is used for agent, is divided into subtypes to represent causative forms: ARG0-GOL: affected agent; ARG0-MNS: non-affected agent; ARGA: causer. Other intermediate causers are represented using: ARGA-MNS [179].

6.3.3.2.2 Empty Argument Insertion

Vaidya et al. [178] have studied four types of empty arguments, *PRO*, *REL*, *GAP*, *pro*, and have come up with a way to annotate these arguments. In order to have an entire representation of predicate argument structures consisting of dropped arguments too then, the Hindi PropBank annotation should include semantic role labelling and also insertion of empty arguments. The four types of empty arguments that are being inserted in the Hindi PropBank are core arguments. They can be subjects, objects or indirect objects. These arguments will always be marked as ARG0 (agent), ARG1 (patient), or ARG2 (recipient) in PropBank, [179]. The four types of empty arguments are discussed below (for more details, Bhatia et al. [41]):

(i) PRO: It is the subject of a non-finite clause and is absent in the sentence. It is either controlled by its subject or object.

Ex-(6.40) shyaami [PRO chaay piinaa] chaantraa hai
shyam (he) tea to drink wants be-Pres
‘Shyam wants to drink tea.’

In the above example, PRO is the subject of the the verb piinaa ‘to drink’ and it is controlled by the subject ‘shyam’ of the main clause. Hence PRO is coreferenced with ‘shyam’.

(ii) REL: It is the subject or object of a participial relative clause and indicates to the noun modified by this clause. Here REL is the argument of the verb bhaag ‘run’.

Ex-(6.41) [REL bhaagtaa huuaa] ladkaa has raha thaa
(who run-INF-be boy laugh-Prog-be-Past
‘The REL running boy was laughing.’

(iii) GAP: It is a dropped argument in a co-ordination structure. Here GAP is the argument of the verb so ‘sleep’ and is co-referenced with Shyaam.
(iv) pro: It is an argument which is dropped for discourse-pragmatic reasons. pro is the subject of the verb pii ‘drink’ and is retrievable from the context.

```
Ex-(6.43) pro subah-subah chaay piitaa hai
(he) morning-morning tea drinks be-Pres
‘pro drinks tea early in the morning.’
```

The PropBank annotation of GAP is done as follows:

```
Ex-(6.44) shyam ne (k1) chaay (k2) pii aur GAP (k1) so gayaa
shyam Erg tea drank and (he) sleep go-Perf
‘Shyam drank tea and GAP slept.’
```

In the above example, GAP is the dependent of the verb so ‘sleep’, and is marked with the semantic role ARG0. The co-reference feature is also captured between Mohan and GAP.

```
Ex-(6.45) subah-subah pro chaay piitaa hai
morning-morning (he) tea drinks be-Pres
‘pro drinks tea early in the morning.’
```

The verb pii ‘drink’ assigns ARG0 label to pro.
The Hindi Dependency Treebank does not cover these type of dropped arguments but covers other empty categories such as empty nouns, e.g., ellipsis, empty verbs, e.g., gapping, empty conjunctions, etc. [41]. In Hindi verb frames, we haven’t captured the empty arguments.

In the next chapter we discuss about causatives and conjunct verbs in detail which have been just briefly discussed in chapter 5.