Development and Analysis of Verb Frame Lexicon for Hindi

Rafiya Begum[*], **Dipti Misra Sharma**

Language Technology and Research Center, India

Abstract A verb frame (VF) captures various syntactic distributions where a verb can be expected to occur in a language. The argument structure of Hindi verbs (for various senses) is captured in the verb frames (VFs). The Hindi verbs were also classified based on their argument structure. The main objective of this work is to create a linguistic resource of Hindi verb frames which would: (i)Help the annotators in the annotation of the dependency relations for various verbs; (ii)Prove to be useful in parsing and for other Natural Language Processing (NLP) applications; (iii)Be helpful for scholars interested in the linguistic study of the Hindi verbs. In this study of Hindi verbs, the verb argument relations are captured using the dependency relations from Paninian Grammatical Framework (PGF). Analysis of Hindi verbs is the focus of this study since it gives us a good understanding of syntactic and semantic behaviour of verbs which is required for dependency annotation and for parsing. [1]. The preliminary work on this study was published as "Developing Verb Frames for Hindi" [1] in Language Resources and Evaluation Conference (LREC), 2008.

Keywords Hindi Verbs, Verb Frames, Linguistic Resources, Paninian Grammatical Framework, *karaka* Relations, Verb Classification

1. Introduction

Verbs play a major role in interpreting the sentence meaning. Since verbs are important, the study of verb argument structure and their syntactic behaviour provides the necessary knowledge base for intelligent NLP applications. In this work, Hindi verbs were analyzed and then verb frames (capture the argument structure of the verbs) were created for these verbs. Verb frames were created following Paninian Grammatical Framework (PGF) where a verb plays a critical role in the analysis of a sentence. Hindi verbs were also classified based on their VFs.

The justification for following PGF is as follows: Indian Languages (ILs) are morphologically rich and have a relatively flexible word order [2,3]. There is a debate in the literature whether the notions *subject* and *object* can at all be defined for ILs [4]. Behavioral properties are the only criteria based on which one can confidently identify grammatical functions in Hindi [5]; Marking semantic properties such as thematic roles as dependency relations is problematic too. Thematic roles are abstract notions and require higher semantic features which are difficult to formulate and to extract. Therefore, a grammatical model which can account for most of the linguistic phenomena in ILs and would also work well for computational purposes is required. Panini's grammar [3], offers a theoretical model which works well for morphologically rich languages and offers a level of analysis which being syntactic-semantic in nature provides us a good combination of syntactic and semantic features for processing natural language. Since Hindi is an Indian language (IL) and has relatively free word order [6,3], dependency grammar formalism is very well suited for it. In such languages, because of their rich morphology, there is more freedom in word order for expressing syntactic functions [7,8]. Thus, for this work, computational model of Panini's Grammar has been chosen.

Paninian Grammar (PG) is a dependency based grammar [9,10]. Dependency grammar formalisms have emerged from the work of Tesnière [7]. The basic elements in the Dependency Grammar (DG) are: (i)head word, and (ii)its dependent. Syntactic annotation in the dependency framework has two types of inter-related decisions: attachment and labeling [11,12,13,14,15]. If one word attaches with another then it indicates that there is a syntactic relationship between the head word and the dependent word. There is a parent-child relationship between head word (parent) and the dependent word (child). The relations will tell the type of the attachment. For example, if the noun is the subject of the verb then the attachment of a dependent noun with the head verb will be marked as relation subject [16].

Paninian Grammar treats a sentence as a series of modifier-modified relations where a sentence is supposed to have a primary modified (root of the dependency tree) which is the main verb (central binding element) of the sentence. The elements modifying the verb, participate in the action specified by the verb.

Paninian Grammar is followed for creating verb frames since it provides a *karaka* based analysis framework for a sentence where *karakas* are the roles of different participants directly involved in the action denoted by the verb. The relations between noun constituents and the verb are called *karaka* relations which are dependency relations. The *karaka* relations are syntactico-semantic in nature, i.e., they have both syntactic and semantic information [3]. There are six basic karakas, namely; *karta (k1, agent)* 'doer of the action', *karma (k2, theme)* 'one who undergoes the action', *karana (k3, instrument)* 'instrument in accomplishing the action', *sampradana (k4, recipient)* 'reciever of the action', *apadana (k5, source)* 'fixed point of departure', and *adhikarana (k7, location)* '*location in place/time/other'.* 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 a 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 demands (arguments) for the Hindi parser [17,18]; and (4)It forms a basis for linguistic analysis.

The focus of this work has been on identifying a verb's argument structure as it is crucial for parsing and other NLP applications. Verb frames for Hindi provide us the arguments that a particular verb can take for a particular sense, i.e., they show *mandatory* and *desirable* (not mandatory and not optional; but required) arguments for a verb. In verb frames, arguments are annotated using *karaka* relations and other dependency relations (Other than *karaka* relations; these were introduced since *karaka* relations were not sufficient). Many formal theories of grammar talk about the distinction between constituents that are arguments and those that are adjuncts: Arguments are something that lexical heads have [19]. Complements/Arguments are obligatory and Adjuncts are always optional [20]. Adjuncts are not considered in this work as they are optional. Some of the arguments are considered as 'desirable' which means that these arguments are required to fulfill the meaning of the verb and they don't have the compulsion to be present on the surface level of the sentence.

This paper is organized as follows: Section 2 discusses Related Work which talks about resources related to verb argument structure created for English; Section 3 gives a brief overview of the Paninian Grammar and the motivation for following it; Section 4, 5 and 6 talk about verb frames, methodology followed in creating VFs and results related to it, respectively; Section 7 discusses about the comparison between Paninian Dependency Annotation and Propbank Annotation; Section 8 gives the Classification of Hindi verbs based on their frames; Section 9 gives the Conclusion along with the Future Work.

2. Related Work

Some of the well-known linguistic resources related to the verb argument structure created for English, are discussed briefly in this section.

Beth Levin's work on verb classes [21] shows correlations between the semantic and syntactic behavior of the English verbs. The verb behavior can be used to get an insight into linguistically relevant aspects of the verb meaning [22]: (1) 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 (2) 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).

VerbNet (VN) [23,24] is a hierarchical, domain-independent; broad-coverage online verb lexicon which extends Levin's verb classes [21] and provides the syntactic and semantic information for English verbs. It is mapped to other language resources such as *Wordnet* [25,26], *FrameNet,* and *PropBank*. Each Verb class in VN is described by thematic roles, selectional restrictions on the arguments, and syntactic frames [21].

PropBank (PB) [27] 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] and 2008-2009). 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 [28]. Each set of argument labels and their definitions is called a *frameset*. For example, the frameset of the verb *dance* contains *Arg0: dancer, Arg1: dance* and *Arg2: partner* as essential roles. It also has non-essential roles such as *Argm-loc: location* and *Argm-tmp: time.*

FrameNet (FN) [29] is an on-line lexical resource for English, based on frame semantics and supported by corpus evidence. FN groups words according to the conceptual structures, i.e., frames that underlie them [29]. It has three major components [29]: (1) *Lexicon*; (2) *Frame Database* contains descriptions of each frame's basic conceptual structure, and provides names and descriptions for the elements participating in such structures; (3) *Annotated Example Sentences* are marked to exemplify the semantic and morpho-syntactic properties of the lexical items. Each frame contains various participants, i.e., core (core arguments) and non-core (adjuncts or peripheral roles) elements which are considered as semantic roles. For example, core elements of the frame *Getting-up* are *person/animal* getting up from sleep and *place* of sleeping; non-core elements are *time, purpose, etc.*

All these resources talk about the verb argument structure of the English verbs. They provide syntactic and semantic information, and correlation between them. These resources are also mapped to each other to make individual resources

[1] http://www.lsi.upc.edu/~srlconll/

richer. In this work of creating verb frames for Hindi, the verb argument structure is captured using *karaka* relations which capture both syntactic and semantic information of the verbs. A mapping is done between *karaka* relations, *theta* roles and Propbank annotation. It is also mentioned if an argument is mandatory or non-mandatory for a particular verb.

All these resources for English 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. This paper shows the work on Hindi language and presents the study on Hindi verbs which have been analyzed within the Paninian Grammatical Framework. It is believed that this resource of verb frames proves to be helpful for various NLP tasks in Hindi.

3. Paninian Grammar

The main problem that the Paninian approach addresses is to identify syntactico-semantic relations in a sentence. Thus the motivation for following the Paninian approach is: a)The framework is motivated by Sanskrit language which is an inflectionally rich language and focuses on the role of case markers such as post-positions and verbal inflections [3]; b)Is better suited for handling Indian languages, which have a relatively free word order and richer morphology (similar to Sanskrit); c)The model, not only offers a mechanism for SYNTACTIC analysis, but also incorporates the SEMANTIC information (dependency analysis), i.e., it provides the level of syntactico-semantic interface for parsing.

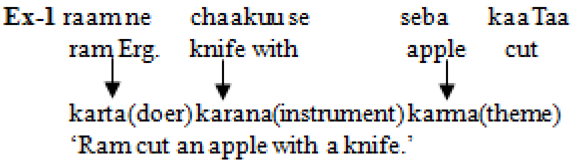
In Paninian based approach, the verb is taken as the root of the tree and its argument structure is considered as its children [3]. The labels on the edges between a parent-child pair show the relation-type between them [17]. Two levels of analysis are followed in Paninian framework: (1) Syntactico-semantic relations (karaka relations): (i) Direct participants of the action denoted by a verb (karaka); (ii)Other relations: purpose, genitive, reason etc; (2)Relation markers (vibhaktis or Hindi postposition/case markers).

The elements of the semantic model within the Paninian framework [3] are explained as follows: A verbal root (dhaatu) indicates an action comprising of (i)an activity (vyaapaara) and, (ii)a result (phala). Activity consists of actions performed by various participants or *karakas* involved in the action. Result is the condition or state reached when the action is complete [3]. Thus every action involves an activity and a result. *Ashraya* or locus of the activity is *karta* and among all the participants in the action, *karta* is swatantra ‘independent’, i.e., it is the most independent *karaka*. Ashraya or the locus of the result is called *karma* (k2). The rough mapping of all *karaka* roles with its theta roles is given below in table 1:

Table 1. Rough Mapping of karaka-Roles with Theta-Roles

karaka-Roles	Theta-Roles
karta (k1)	subject/agent/doer/experiencer/force. ‘the most independent participant in the action’.
karma (k2)	object/patient/theme/goal/content-of-event/result of creation ‘most desired to be attained by the karta’;
karana (k3)	Instrument ‘instrument which helps in accomplishing the action’;
sampradana (k4)	beneficiary/recipient ‘intended recipient of the object’;
apadana (k5)	Source ‘fixed point of departure (or) moving away from a source’
adhikarana (k7p/k7t/k7)	location in place/time/other ‘It supports karta or karma in space or time’

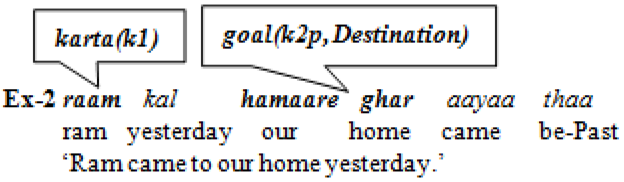
In Paninian grammar, Hindi postposition/case markers are referred to as *vibhaktis* (Hindi postpositions) which are relation markers. A *vibhakti* denotes case markings on the nouns and the TAM (tense, aspect and modality) of the verbs. *Vibhaktis* play a key role in indicating semantic relationships. They act as syntactic cues in a sentence and help in identifying the appropriate *karakas* [30]. In example 1, *ne* vibhakti indicates *karta* (doer), *se* vibhakti indicates *karana* (instrument), and 0 (zero) vibhakti indicates *karma* (theme).

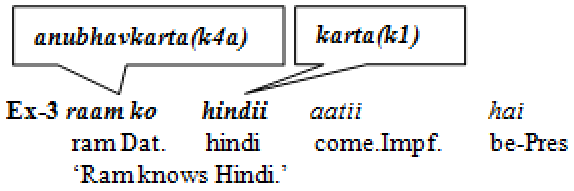

Ex-1 raam ne chaakuu se seba kaaTaa
ram Erg. knife with apple cut
karta(doer) karana(instrument) karma(theme)
‘Ram cut an apple with a knife.’

After discussing PG, a detailed discussion of the verb frames and the procedure followed in creating the VFs is provided in the sections given below.

4. Verb Frames for Hindi

Verb frames were created on the following basis: (1) multiple senses of a verb may lead to change of frame, hence change in syntactic alternation; (2) multiple frames for a verb having the same sense. According to the first basis given above, the frames of different senses of a verb may differ. For example, the two senses of the verb *aa,* i.e., ‘come’ and ‘know’ have different frames, i.e., *karta+goal* and *anubhavkarta+karta*:

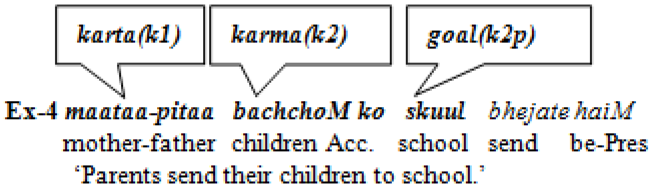

karta(k1) *goal(k2p, Destination)*
Ex-2 *raam kal hamaare ghar aayaa thaa*
ram yesterday our home came be-Past
‘Ram came to our home yesterday.’



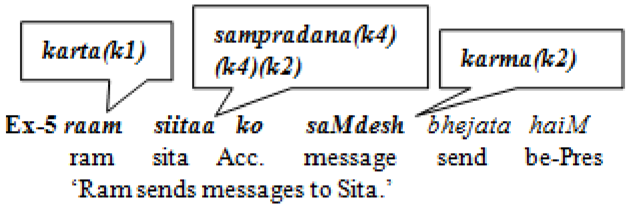
Ex-3 *raam ko hindii aatii hai*
ram Dat. hindi come.Impf. be-Pres
'Ram knows Hindi.'

The senses of the verb *aa in* the above examples *2* and *3 is* 'come' and 'know' respectively. In example *2,* the verb *aa* having the sense 'come' takes the following arguments: *karta,* and *goal*. In example *3,* the verb *aa* with the sense 'know' takes the following arguments: *anubhavkarta,* and *karta.* It can be noticed here that there is a difference in the set of dependency relations of the arguments taken by the verb *aa* having two different senses in the above two examples. Therefore, with the change in the sense of the verb there is also change in the frame of the verb, but this is not always the case, i.e., frames can be same for different senses of a verb.

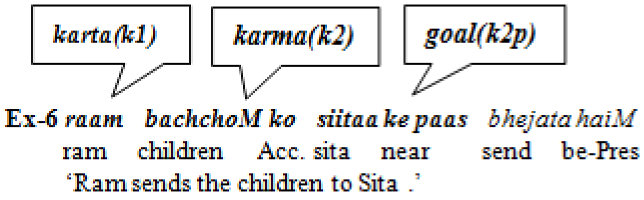
Multiple frames, mentioned in the second basis, means that a verb can take a different set of dependency relations for the same sense of a verb. For example, the verb *bheja* with the sense 'send' has two different frames:



Ex-4 *maataa-pitaa bachchoM ko skuul bhejate haiM*
mother-father children Acc. school send be-Pres
'Parents send their children to school.'



Ex-5 *raam siitaa ko saMdesh bhejata haiM*
ram sita Acc. message send be-Pres
'Ram sends messages to Sita.'



Ex-6 *raam bachchoM ko siitaa ke paas bhejata haiM*
ram children Acc. sita near send be-Pres
'Ram sends the children to Sita .'

In the above Hindi example sentences *4* and *5,* the verb *bheja* has the same sense, i.e., 'send'. In example *4,* the verb *bheja* is taking the following arguments: *karta, karma,* and *goal*. In example *5,* the verb *bheja* 'send' is taking the following arguments: *karta, sampradana,* and *karma.* Here, it can be noticed that there is a difference in the set of dependency relations of the arguments taken by verb *bheja* 'send' in the examples 4 and 5. This shows that the same sense of a verb can take multiple frames. There exists a finer distinction in the sense of *bheja* in examples *4* and *5*, i.e., in example *4*, it is an individual (*bachche* 'children') who is being sent so the verb *bheja* becomes a causative verb here. Whereas in example *5*, it is an object (*saMdesh* 'message') that is being sent so the verb *bheja* becomes a ditransitive verb here. Such a finer distinction in the senses of *bheja* given in the examples *4* and *5* is not captured. Even Hindi Wordnet[2] (HWN) [31] considers the above senses of *bheja* as a single sense. Also, the type of causative type that exists in example *4* is lexical causative. The base verb root of the lexical causative *bheja* 'send' is *jaa* 'go'. The causative structure is as follows:

jaa 'go' (base verb root) → *bheja* 'send' (first causal) → *bhijavaa* 'to cause to send' (second causal).

Since, lexical causatives are very rare in Hindi, the causative nature of the verb is ignored here. Both these usages are ditransitive which take different participants here; hence there is a change in the frame.

In the verb frames, along with the *mandatory* arguments of a verb, other arguments are also captured which are mostly not present on the surface level of the sentence but are implicit. For example, the verb *kaaT having* the sense 'cut' takes two *mandatory* arguments, i.e., *karta* and *karma* in the example 7 given below. It also takes the *instrument* argument that is used in the action of cutting. So the *instrument* is considered as a *desirable* argument which is not strictly required to be present in the sentence. For example, *chaakuu* 'knife' is the instrument used in the action of *kaaT* 'cut', so it becomes the *desirable* argument. The dependency relation of the *chaakuu* is *karana* (k3).

Ex-7 *raam ne* ***chaakuu se*** *seba kaaTaa*
ram Erg. knife with apple cut
'Ram cut an apple with a knife.'

5. Materials and Methods

Hindi verbs were taken from a corpus and studied. Its distribution was taken from the corpus. For doing this, the following resources were used: (1) Levin's verb classes [21]; (2) A Hindi corpus[3] (Raw and Dependency annotated); (3) Hindi Wordnet (HWN) [31]; and (4) Sahay's verb classes [32].

Verb frames (VFs) were created for 300 verbs which are simple verbs (non-complex verbs: combination of noun and verb) and these verbs were selected from a raw Hindi Corpus (75,000 sentences) on the following basis: complex nature, showing interesting patterns, focus of study in literature.

Given a verb, first of all its senses were taken from the corpus. Then for each sense, example sentences were taken from the corpus. VFs were created for different senses of a verb. VFs mainly contain the dependency relations of the *mandatory* and *desirable* (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

[2] Developed by the wordnet team at IIT Bombay, http://www.cfilt.iitb.ac.in/webhwn

[3] We use the CIIL (Central Institute for Indian languages) corpus.

extracted out of the context; e.g. *He cuts the apple* vs. *He cuts the apple with a knife.* Note that this is very different from obligatory arguments and optional arguments) arguments taken by a verb. For each sense of a verb, multiple frames were created if there were any. All the *vibhaktis* (Hindi postpositions/case markers) taken by the arguments of a particular verb sense were merged together within the *VF*. While doing this, the TAM (tense, aspect and modality) was kept constant in all the sentences as the change in TAM causes change in *vibhakti* too. Theta roles are also marked as additional information to make the resource richer. POS categories of the verb arguments are also included.

5.1. Verb Frame File

The corpus was consulted to get the syntactic distribution in which the verbs occur and the Hindi Wordnet (HWN) was referred to get the required sense information. Verb frames are defined in terms of dependency relations, i.e., *karaka* relations [2] and other dependency relations. The following information is given for each verb in its *verb frame file*: (1) Description of the verb, and (2) *Verb Frame.*

The two types of information above are provided in a data file which is referred as *VF file*. In the *VF file* there is a table which is actually a verb frame but the verb frame file of a verb itself is referred as a *verb frame* of that verb.

5.1.1. Description of the Verb

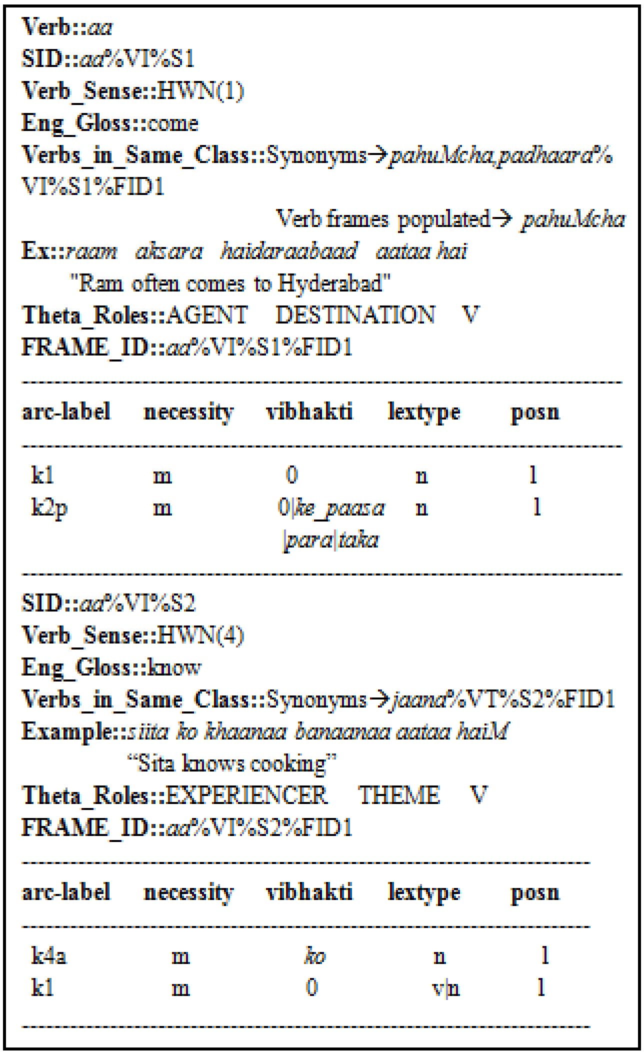
For each verb, a verb frame file is created and it contains the following description for each verb sense: (1) Verb name, (2) Sense-id (SID), (3)HWN sense-id (optional), (4)English gloss, (5)Synonyms, (6)Example sentence, (7)Theta roles, (8)Frame-id, and (9)V*erb frame* (given in a table). Given below is an example of a VF file in figure 1 (Basically 'verb frame file' is used to refer to the file that has been created for entering the data of each verb. But mostly the term '*verb frame'* is used instead of *verb frame file*) including the verb frame for verb *aa* 'come'.

Given below is the frame file of the verb *aa* having two senses. Here, first sense is discussed and then the second sense. In the below *VF* file, the first field is the name of the verb, i.e., *aa* 'come'. Sense-id (SID) is a unique id given to each sense of the verb and here it is represented as *aa%VI%S1*. In *SID,* the following information is captured: (1) verb*,* (2) verb type, and (3) sense number; all three separated by a percentage symbol. In *SID, VI* is the verb type which means *verb intransitive* and *S1* means sense-1. Here*,* 4 types of verbs are used: *VI→ Intransitive Verb; VT→ Transitive Verb; VDT→ Ditransitive Verb;* and *VCAUS→ Causative Verb.* Here the sense of the verb *aa* matches with the first sense given in HWN hence, it is represented as *HWN (1)* in the frame file. The sense numbers in HWN are not fixed, they keep changing hence this information is not considered.

Theta_Roles field gives the theta roles (taken from Verbnet) of the verb arguments. It is believed that such a mapping is going to help in making the transition from the dependency level (*karaka* level (syntactico-semantic) as well as non-*karaka*) to *thematic* level (semantic) easier. It helps in making certain generalizations.

FRAME_ID field gives a unique-id to the verb frame of a verb for a particular sense. *FRAME_ID* is a combination of *verb sense-id* and *frame number*. Here, Frame-id *aa%VI%S1%FID1* is made up of the sense-id, *aa%VI%S1* and verb frame number, *FID1* 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 verb frames, i.e., syntactic distributions. So sense-id helps us in identifying the sense of the verb and frame-id helps us in identifying the verb frame of that particular sense of a given verb. If a verb takes two frames for the same sense then they are represented as: (i)*verb%VI%S1%FID1* and (ii)*verb%VI%S1%FID2* where frame-ids *FID1* and *FID2* denote that there are two *frames,* belonging to the same sense *S1*.

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, also have the same frame as the given verb. The differences between these verbs and the verb given are listed out. The differences can be in the terms of arguments (dependency relations), *vibhaktis,* number of frames, etc. If the sense is same and verb frames (dependency relations) are different, even then the verbs are listed under this field and the details of the differences are given in VFs. Here the verbs *pahuMcha* 'reach' and *padhaara* 'arrive' have same sense and same *VFs* 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* exists. This means that verb frame for the verb *padhaara*, whose sense and VF matches with verb *aa* has been created. 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: (1) VF file has not been created at all for that particular verb or, (2) VF file might have 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 VF file has not been created at all for this verb. By listing a verb under *Verbs_in_Same_Class* field, whose VF file has not been created, a verb frame is getting populated for that verb. In *VFs populated* field, within the Verbs_in_Same_Class field, those verbs are mentioned whose *VFs* are populated. Even conjunct verbs (noun/adjective+verb combination) are included in the populated verbs.



Verb::*aa*

SID::*aa*%VI%S1

Verb_Sense::HWN(1)

Eng_Gloss::come

Verbs_in_Same_Class:: Synonyms→*pahuMcha,padhaara*% VI%S1%FID1

Verb frames populated→ *pahuMcha*

Ex::*raam aksara haidaraabaad aataa hai*

"Ram often comes to Hyderabad"

Theta_Roles::AGENT DESTINATION V

FRAME_ID::*aa*%VI%S1%FID1

arc-label	**necessity**	**vibhakti**	**lextype**	**posn**
k1	m	0	n	l
k2p	m	0|*ke_paasa* |*para*|*taka*	n	l

SID::*aa*%VI%S2

Verb_Sense::HWN(4)

Eng_Gloss::know

Verbs_in_Same_Class:: Synonyms→*jaana*%VT%S2%FID1

Example::*siita ko khaanaa banaanaa aataa haiM*

"Sita knows cooking"

Theta_Roles::EXPERIENCER THEME V

FRAME_ID::*aa*%VI%S2%FID1

arc-label	**necessity**	**vibhakti**	**lextype**	**posn**
k4a	m	*ko*	n	l
k1	m	0	v|n	l

Figure 1. Verb Frame for Verb *aa* 'come'

Populating a VF for a verb, means that instead of creating a separate *VF* file for that verb, it is just listed in another VF file with which it matches in sense. By doing this it is getting conveyed that this populated verb has the same *VF* (dependency relations) as the verb in whose VF file it has been listed. The process of populating *VFs* lessens the effort of creating more VF files.

There can be differences in VFs between populated verb (*pahuMcha* 'reach') and the main verb (*aa* 'come'). They might have same dependency relations but may differ in *vibhaktis*, POS categories, no. of *VFs,* etc.

5.1.2. Verb Frame (table in the *Verb Frame File*)

The actual VF is the table given in the *verb frame file. VF* contains the verb argument information of the verb. A VF shows the following information: (1)dependency relations; (2)*necessity* of the argument, i.e., whether an argument is *mandatory* (m) or *desirable* (d); (3)*vibhakti* (Hindi postpositions/case markers taken by the arguments); (4)lexical category of the arguments, i.e., noun, verb and adjective, etc., represented by *n, v,* and *adj* respectively; (5)position of the argument with respect to the verb, i.e., *left* or *right* represented by *l* and *r* respectively.

In Fig. 1*,* the dependency relations for verb *aa* 'come' is given under the *arc-label* field. The arguments of the verb *aa* is *raam* 'Ram' and *haidaraabaad* 'Hyderabad', and their *karaka relations* are *karta* (k1) and *Goal* (k2p) respectively. These two arguments are *mandatory* to accomplish the semantics of the verb *aa* hence, their *necessity* information is *mandatory* which is represented as *m*. *raam (k1)* takes *0 vibhakti* and *haidaraabaad (k2p)* can take either *0, ke_paasa* 'near'*, taka* 'till' *vibhakti*. There is *par* 'on' *vibhakti* which is not taken by *haidaraabaad* but taken by other nouns that can replace *haidaraabaad.* So basically, all the *vibhaktis* taken by different nouns for the *Goal karaka* relation of the verb *aa* are merged here. The *vibhaktis* are merged using an 'or' operator, represented as a pipe '|' (see Fig. 1). Here, it is getting conveyed that the relation *goal* (*k2p)* takes all these *vibhaktis* but the lexical item may change.

The *vibhaktis* of the arguments may change with the change in TAM (tense, aspect and modality). Therefore, the TAM is kept constant during the process of creating *VFs.*

Present imperfect TAM, i.e., *taa_hai* is taken here. A *VF* is different from another *VF* 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, 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.

The lexical category of both the arguments (*raam* 'Ram' and *haidaraabaad* 'Hyderabad') of the verb *aa* 'come' is *noun* (*n*) and this is mentioned under the *lex-type* field. 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 *posn* (position) in the verb frame is not required from the point of view of a lexical resource (was required for the parser purpose). As the parser is rule-based, one needs some heuristics in order to reduce the possible search space to find various arguments of a verb. Thus, the notion of 'most frequent' position helps in this sense. If the position field is made part of the parser implicit strategy then this makes the rules hard-coded whereas it is required to make sure that such parser behaviour is externalized, therefore the position field was introduced.

The frames are developed based on simple present tense and indicating habitual acts taking it as the default TAM. In fact, dependency relations and the *vibhaktis* (Hindi postpositions/case markers) in the frame reflect the behaviour of the verb when it occurs in simple present ('*-taa hai'* in Hindi, eg. *khaa-taa hai* 'eats'). This is done to bring in consistency while forming the various frames, as in Hindi, the *vibhakti* of an argument changes with the change 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.

After discussing the first sense, the discussion about the second sense of the verb *aa* is given below. As mentioned above, the two senses of the verb *aa,* i.e., 'come' and 'know' are given in Fig. 1. Each of these senses has one *verb frame* each. All the senses of a verb are present in single *verb frame* file. For each sense, a sense id is provided along with the English gloss. For the first sense of the verb *aa*, the sense id (SID) is *aa%VI%S1* and for the second sense, 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 it can be seen that the *FRAME_ID* for the verb *jaana*, i.e., *jaana%VT%S2%FID1* is mentioned under the field *Verbs_in_Same_Class* of the second sense of the verb *aa* 'know'. 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 frame* file *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 it can be seen that though the second sense of the verbs *aa* 'know' 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)*. The difference in their verb type is making an effect on their frames also. Though they differ in frames, still the verb *jaana* 'know' (*jaana%VT%S2)* has been listed under the *Verbs_in_Same_Class* field of the verb *aa* 'know' (*aa%VI%S2)* as they have same senses and a note has been made clearly saying that they don't match in frames. This gives us a generalization that though semantically they are same: (1) they differ in their verb types; (2) they differ in their frames, i.e., they take a different set of dependency relations (*karaka* relations (syntactico-semantic) and other dependency relations).

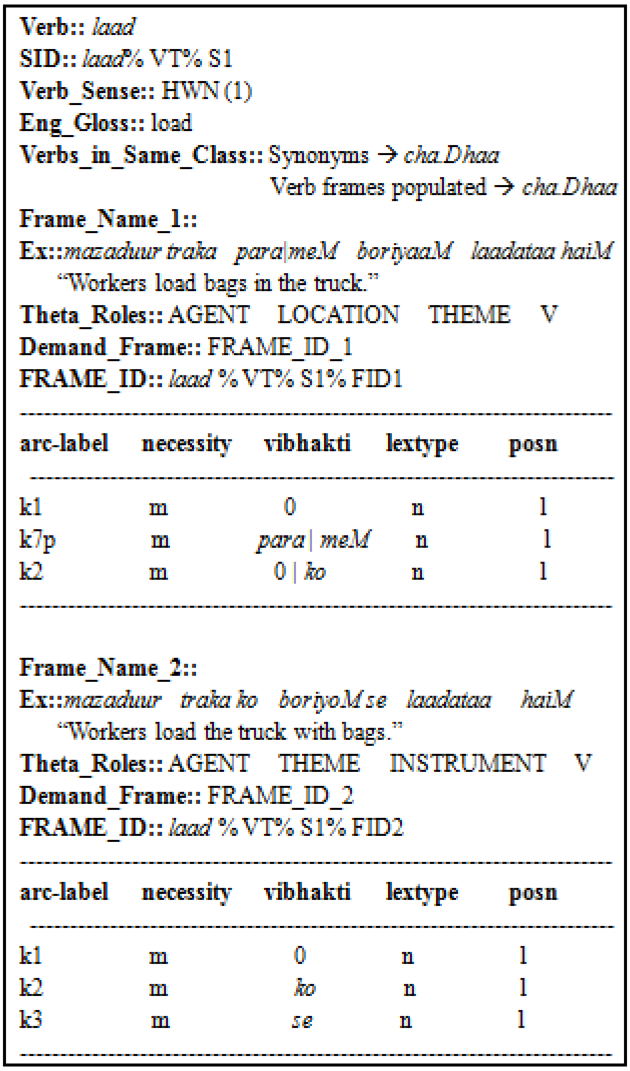
In the *Example* field of sense-2*,* there are two sentences separated by an 'or' operator represented by *pipe* ('|'). These two examples differ in the *lexical category* of *k1* (karta). Here *k1* can have either *verb (v)* or *noun (n)* as its *lexical category.* In one example *banaanaa* 'cooking' 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 the lexical categories are merged 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 is *aa%VI%S1%FID1* and *aa%VI%S2%FID1* respectively. *aa%VI%S1%FID1* means that it is the *verb frame-1* of the first sense of the verb *aa* 'come' and *aa%VI%S2%FID1* means that it is the *verb frame-1* of the second sense of the verb *aa* 'know'. The arguments of the verb *aa* having the sense 'know' 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, experiencer*) 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'*,* takes *agent* and *destination* theta roles for *raam* 'Ram' and *haidaraabaad* 'Hyderabad' respectively. The second sense of the verb *aa* 'know'*,* takes *experiencer* and *theme* theta roles for *siitaa* 'Sita' and *banaanaa* 'cooking'/*silaaii-ka.Daaii* 'stitching' respectively.

5.2. Some Examples of Hindi Verb Frames

In this section, the following *verb frames* are discussed: (1)a verb having two frames for a single sense, (2)verb taking a *desirable* argument, (3)verbs matching in sense as well as number of frames, and (4)causative verb's representation in verb frames.

5.2.1. Verb Frame of Verb *laad* 'load' Having Two Frames for one Sense



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*
Frame_Name_1::
Ex::*mazaduur traka para|meM boriyaaM laadataa haiM*
"Workers load bags in the truck."
Theta_Roles:: AGENT LOCATION THEME V
Demand_Frame:: FRAME_ID_1
FRAME_ID:: *laad* % VT% S1% FID1

arc-label	necessity	vibhakti	lextype	posn
k1	m	0	n	1
k7p	m	*para* | *meM*	n	1
k2	m	0 | *ko*	n	1

Frame_Name_2::
Ex::*mazaduur traka ko boriyoM se laadataa haiM*
"Workers load the truck with bags."
Theta_Roles:: AGENT THEME INSTRUMENT V
Demand_Frame:: FRAME_ID_2
FRAME_ID:: *laad* % VT% S1% FID2

arc-label	necessity	vibhakti	lextype	posn
k1	m	0	n	1
k2	m	*ko*	n	1
k3	m	*se*	n	1

Figure 2. Verb Frame of Verb *laad* 'load' Having Two Frames for One Sense

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. An example of a verb having more than one frame for a particular sense is given in Fig. 2. 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_IDs* of the first and second verb frames are *laad%VT%S1%FID1* and *laad%VT%S1%FID2* respectively. Here it can be seen 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), *deshadhikaran* (k7p, place of action) 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 *boriyaaM (k2)* takes either *0* or *ko vibhakti.* The arguments *traka* (*k7p*), and *boriyaaM (k2)* are taking more than one vibhakti so here these *vibhaktis* of *traka* (*k7p*), and *boriyaaM (k2)* are merged, 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 *boriyoM* 'bags'. The *karaka relations* of *mazaduur* 'servants', *traka* 'truck' and *boriyoM* 'bags'are *karta* (k1), *karma* (k2) and *karana* (k3) respectively. The necessity of *mazaduur* (*k1*), *traka* (*k2*) and *boriyoM (k3)* is *mandatory* i.e., *m*. *mazaduur* (*k1)* takes *0* vibhakti, *traka* (*k2*) takes *ko* vibhakti, and *boriyoM (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 frame file for the verb *cha.DZaa* 'load' as it can be seen 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 frame file of the verb *laad* ' load' that; (i)there is a difference between the number of frames of verbs *laad* and *cha.DZaa,* and (ii)the single frame of the verb *cha.DZaa* matches with the first frame of the verb *laad*.

5.2.2. Verb Taking a *Desirable* Argument

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*. An argument which is *mandatory* is compulsory to be present in the sentence as it is obligatory to fulfil the meaning of the verb, without which the meaning of the verb is incomplete. A *desirable* argument is also required to fulfil the meaning of the verb but it is not compulsory for the *desirable* 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. *Mandatory* arguments are considered as *strong* arguments and *desirable* arguments as *weak* arguments. *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. An example of *desirable* argument is given below:

Ex-8 *raam ne chaakuu se (k3) seba kaaTaa*
ram Erg. knife with apple cut
'Ram cut an apple with a knife.'

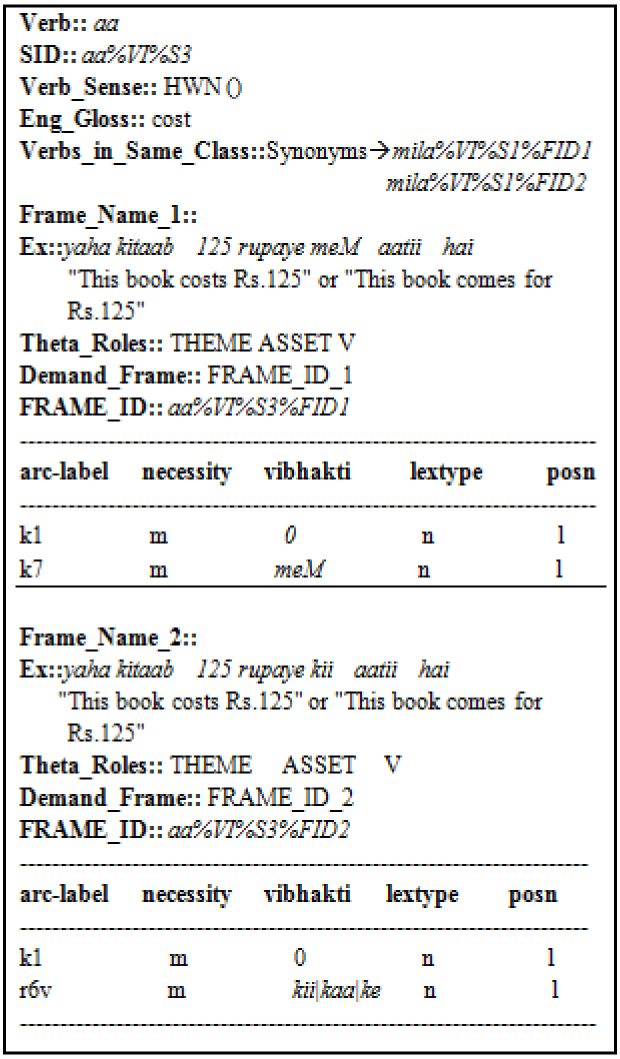
Ex-9 *raam ne seba kaaTaa*
ram Erg. apple cut
'Ram cut an apple.'

In example *8*, *chaakuu se 'with the knife'* is a *desirable* argument as it can be dropped and still the meaning can be retrieved from the context. Example *9*, is a perfect example of the sentence without the *desirable* argument *chaakuu se 'with the knife'* and the desirable argument can be identified even if it is not present in the sentence.

5.2.3. Verbs Matching in Sense as well as Number of Frames

Below given figures *3* and *4* 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.

The two frames *mila%VI%S1%FID1* and *mila%VI%S1%FID2* belonging to the first sense 'get' of the verb *mila,* match with the two frames, i.e., *aa%VI%S3%FID1* and *aa%VI%S3%FID2* belonging to the third sense 'cost' of the verb *aa*. In the Fig. 3*,* it can be seen that the two frame ids of *mila* 'get' (*mila%VI%S1%FID1* and *mila%VI%S1%FID2*) are listed in the *Verbs_in_Same_class* field of the verb entry *aa 'cost'*. The two verb frame ids of *mila* 'get' (*mila%VI%S1%FID1 - mila%VI%S1%FID2*) listed in the field *Verbs_in_Same_class* of the verb entry *aa 'cost'* are separated by a 'hyphen'. Similarly in the Fig. 4, it can be seen that the two frame ids (*aa%VI%S3%FID1; aa%VI%S3%FID2*) of *aa* 'cost' are listed in the *Verbs_in_Same_class* field of the verb entry *mila 'get'*.



Verb:: *aa*
SID:: *aa%VT%S3*
Verb_Sense:: HWN ()
Eng_Gloss:: cost
Verbs_in_Same_Class:: Synonyms→*mila%VT%S1%FID1*
mila%VT%S1%FID2

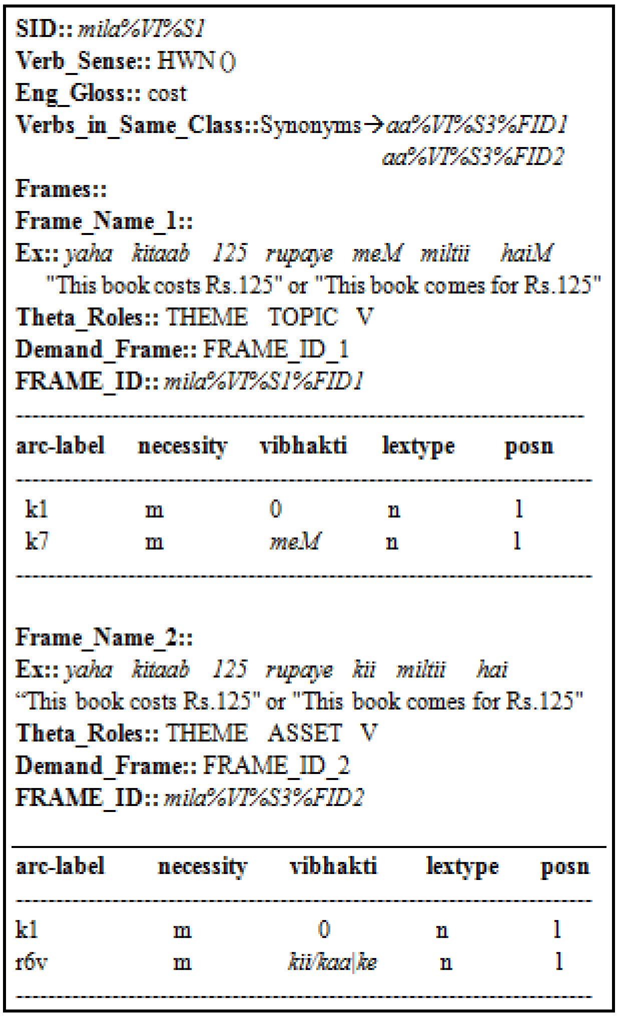
Frame_Name_1::
Ex::*yaha kitaab 125 rupaye meM aatii hai*
"This book costs Rs.125" or "This book comes for Rs.125"
Theta_Roles:: THEME ASSET V
Demand_Frame:: FRAME_ID_1
FRAME_ID:: *aa%VT%S3%FID1*

arc-label	necessity	vibhakti	lextype	posn
k1	m	*0*	n	1
k7	m	*meM*	n	1

Frame_Name_2::
Ex::*yaha kitaab 125 rupaye kii aatii hai*
"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*

arc-label	necessity	vibhakti	lextype	posn
k1	m	0	n	1
r6v	m	*kii|kaa|ke*	n	1

Figure 3. Verb Frame of Verb *aa* ‘cost’ Having Two Frames for One Sense



SID:: *mila%VT%S1*
Verb_Sense:: HWN ()
Eng_Gloss:: cost
Verbs_in_Same_Class:: Synonyms→*aa%VT%S3%FID1*
aa%VT%S3%FID2
Frames::
Frame_Name_1::
Ex:: *yaha kitaab 125 rupaye meM miltii haiM*
"This book costs Rs.125" or "This book comes for Rs.125"
Theta_Roles:: THEME TOPIC V
Demand_Frame:: FRAME_ID_1
FRAME_ID:: *mila%VT%S1%FID1*

arc-label	necessity	vibhakti	lextype	posn
k1	m	0	n	1
k7	m	*meM*	n	1

Frame_Name_2::
Ex:: *yaha kitaab 125 rupaye kii miltii hai*
"This book costs Rs.125" or "This book comes for Rs.125"
Theta_Roles:: THEME ASSET V
Demand_Frame:: FRAME_ID_2
FRAME_ID:: *mila%VT%S3%FID2*

arc-label	necessity	vibhakti	lextype	posn
k1	m	0	n	1
r6v	m	*kii/kaa|ke*	n	1

Figure 4. Verb Frame of Verb *mila* 'cost' Having Two Frames for One Sense

In Fig. 1, it can be seen that the verb *aa* 'come' and the verb belonging to its class (which have the same sense as the verb *aa*), i.e., *padhaara* 'arrive' vary in *vibhakti* in case of goal (*k2p*) dependency relation (see figure 1). The verb *padhaara* takes less *vibhaktis* for goal (*k2p*) than the verb *aa* whereas the verb *pahu.Ncha* 'reach' which also belongs to the class of verb *aa has* same no. of *vibhaktis* as the verb *aa* for goal (*k2p*). The verb *aa* and *pahuMcha* take *0, ke_paas,* 'near', *par* 'on', and *taka* 'till' *vibhaktis* whereas the verb *padhaara* takes *0, ke_paas* 'near', *par* 'on' *vibhaktis*. *taka vibhakti* is not taken by the goal (*k2p*) argument of the *padhaara* verb. In the example *10* given below, it can be noticed that goal (*k2p*) with *taka vibhakti* for the verb *padhaara* is ungrammatical. This information about the differences between *vibhaktis* among the verbs having same sense is provided in the respective verb frame files. These differences allow us to come up with generalizations and also classify the verbs based on these features.

Ex-10 **raam hydaraabaad* ***taka (k2p)*** *padhaare*
ram hyderabad till arrived
'Ram arrived till Hyderabad.'

In the above instance, there is difference in *vibhaktis,* i.e., the verb (*padhaara* 'arrive') listed in the class has less *vibhaktis* than the main verb (*aa* 'come'). There is another instance where the verb listed in the class has more *vibhaktis* than the main verb frame file. In the verb frame of *bataa* 'tell', the relation *k4* (sampradana) takes only *ko vibhakti* whereas the verbs that belong to the *bataa* class, i.e., *kaha* 'say' and *bola* 'tell' which have the same sense as the verb *bataa* take an additional *vibhakti se* other than *ko.* See the examples *11* and *12* given below:

Ex-11 *raam* (k1) *siitaa* ***ko/*se*** **(k4)** *sab* (k2)
ram sita Dat./ everything
bataataa hai
tell.Impf. be.Pres.
'Ram tells everything to Sita.'

Ex-12 *raam* (k1) *siitaa* ***ko/se*** **(k4)** *sab* (k2)
ram sita Dat./with everything
boltaa/kehtaa hai
tell/say.Impf. be.Pres.
'Ram tells/says everything to Sita.'

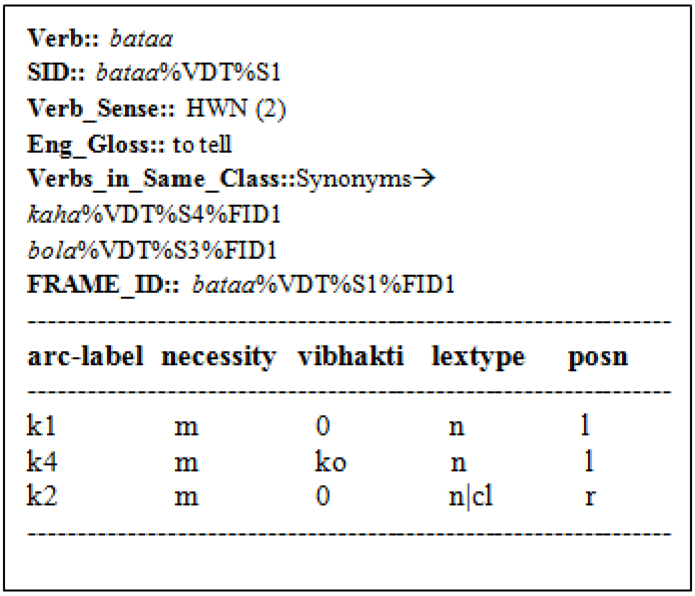
Below given figure 5 shows how the examples 11 and 12 are represented in the verb frame.

There are also cases where 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 *13* and *14* given below:

Ex-13 *raam* (k1) ***siitaa ko*** (k2) ***chaahtaa/pasaMd_kartaa***
ram sita Dat. love.Impf./ like do.Impf
hai
be.Pres.
‘Ram loves/likes Sita.’

Ex-14 *raam* (k1) ***siitaa se*** *(k2)* ***prem kartaa hai***
ram sita with love. do.Impf. be.Pres
‘Ram loves Sita.’

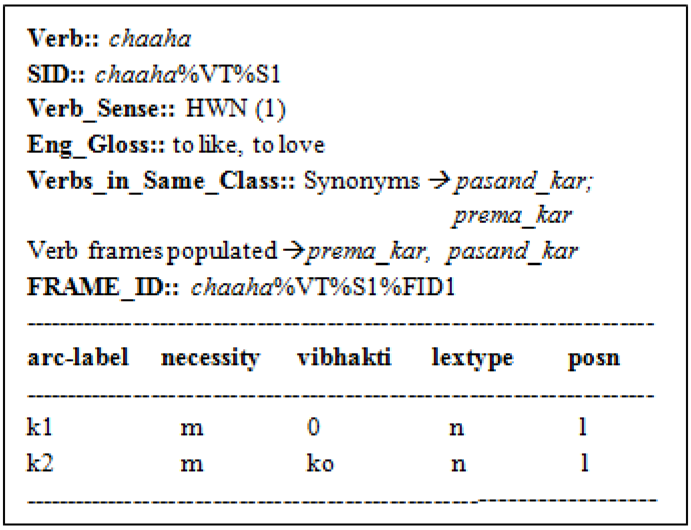
Below given figure 6 shows how the examples 13 and 14 are represented in the verb frame.



Verb:: *bataa*
SID:: *bataa*%VDT%S1
Verb_Sense:: HWN (2)
Eng_Gloss:: to tell
Verbs_in_Same_Class:: Synonyms→
kaha%VDT%S4%FID1
bola%VDT%S3%FID1
FRAME_ID:: *bataa*%VDT%S1%FID1

arc-label	necessity	vibhakti	lextype	posn
k1	m	0	n	l
k4	m	ko	n	l
k2	m	0	n|cl	r

Figure 5. Verb Frame of Verb *bataa* ‘tell’ Showing Differences in *vibhakti* for *k4* relation



Verb:: *chaaha*
SID:: *chaaha*%VT%S1
Verb_Sense:: HWN (1)
Eng_Gloss:: to like, to love
Verbs_in_Same_Class:: Synonyms →*pasand_kar;*
prema_kar
Verb frames populated →*prema_kar, pasand_kar*
FRAME_ID:: *chaaha*%VT%S1%FID1

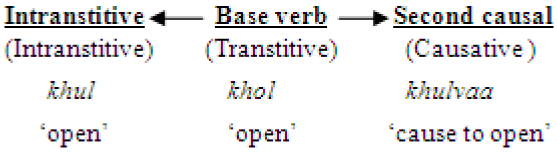
arc-label	necessity	vibhakti	lextype	posn
k1	m	0	n	l
k2	m	ko	n	l

Figure 6. Verb Frame of Verb *chaaha* ‘like’ Showing Differences in *vibhakti* for *k2* relation

There are 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: (1) Differ in frames, for example, verb frames of *jaana* 'know' and *aa* 'know' as mentioned in the Fig. 1. Verb *aa* 'know' takes *k4a+k1* relations. It takes *ko* vibhakti with *k4a* and *0* vibhakti with *k1*. Verb *jaana* 'know' (which belongs to the verb class *aa*) takes different frame, i.e., different *karaka* relations, *k1+k2*. Both the relations take 0 *vibhaktis*; (2)All the frames don't match, i.e., only one frame matches and the other doesn't, *e.g.,* the verb *cha.Dhaa* 'load' (which belongs to the verb class *laad*) matches with the first frame of *laad* 'load' and doesn't match with the second frame of the verb *laad* as mentioned in the Fig. 2. Verb *laad* 'load' takes two frames for the same sense. One is *k1+k7p+k2* where *k1* takes 0 *vibhakti, k7p* takes either *par* or *meM* vibhakti, and *k2* takes either 0 or *ko* vibhakti. The second frame is *k1+k2+k3* where *k1* takes 0 vibhakti, *k2* takes *ko* vibhakti, and *k3* takes *se* vibhakti. The verb *cha.Dhaa* 'load' which has the same sense as *laad* 'load' has only one frame, i.e., the first frame *k1+k7p+k2* of the verb *laad* 'load' and it doesn't have the second frame *k1+k2+k3* of the verb *laad* 'load'*;* (3)All the frames match, i.e., the two frames of the verb *aa* 'cost' (which belongs to the verb class *mila*) matches with the two frames of the verb *mila* get, obtain' as mentioned in the Fig. 3 and 4. The two frames of *mila* are *k1+k7* (*k1* takes 0 vibhakti and *k7* takes *meM* vibhakti) and *k1+r6v* (*k1* takes 0 vibhakti and *r6v* takes *kii/kaa/ke* vibhakti). The verb *aa* 'cost' also takes same two frames as *mila;* (4)Less *vibhaktis*, for example, *aa* 'come' and *padhaara* 'arrive'. *padhaara* 'arrive' (which belongs to the verb class *aa*) takes less *vibhaktis* in comparison to *aa* 'come' for the dependency relation *goal (k2p);* (5)More *vibhaktis*, for example, *bataa* 'tell' and *kaha/bola* 'say/tell'. Verbs *kaha/bola* 'say/tell' (which belongs to the verb class *baat*) take more *vibhaktis* in comparison to the verb *bataa* 'tell' for *sampradana (k4) karaka* relation; (6)Totally different *vibhaktis,* for example, *prem kar* 'love' (which belongs to the verb class *chaaha*) takes totally a different *vibhakti* in comparison to the verbs *chaaha* 'love' and *pasand kar* 'love'. *prem kar* 'love' takes *se* vibhakti for *k2* relation whereas the verbs *chaaha* 'love' and *pasand kar* 'love' take *ko vibhakti* for the *karma (k2) karaka* relation.

5.2.4. Causative Verbs Representation in Verb Frames.

Causatives in Hindi are realized through a morphological process. In Hindi, a base verb root changes to a causative verb when affixed by either an *'-aa'* or a *'-vaa'* suffix. The causativization process of the causative verb *khulvaa* 'cause to open' is given below:



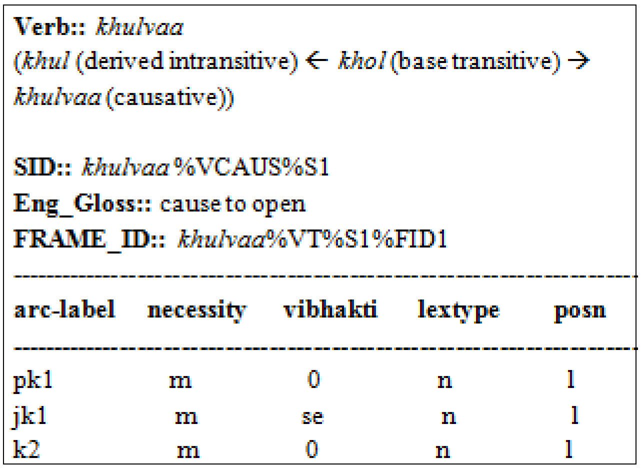
In the above causativization process, *khol* 'open' is taken as the base verb which is a transitive verb. Here, other than morphology, the semantics of the verbs is also taken into consideration. Here *khul* 'open' and *khulvaa* 'cause to open' are derived from the base verb *khol* 'open'. *khulvaa* 'cause to open' is a causative verb which is derived from the base verb by adding suffix *'-vaa'* to it. *khul* 'open' is a derived intransitive form which is derived from the base verb *khol* 'open'. The agent of the base verb *khol* 'open' is not realized on the surface level of the derived intransitive verb *khul* 'open' though it is implied semantically. Here there is both forward and backward derivation. From base verb *khol* to the causative verb *khulvaa* it is a forward derivation which means there is an increment of one argument from base verb to the causative verb. From base verb *khol* to the derived intransitive *khul* it is a backward derivation which means there is a reduction of one argument from base verb to the derived intransitive verb [33,34]. Example sentences (15, 16 and 17) of causatives are given below: [35].

Ex-15 *darvaazaa* ***(k1)*** *khultaa* *hai*
door open.Impf. be.Pres.
'The door opens.'

Ex-16 *raam* ***(k1)*** *darvaazaa* ***(k2)*** *kholtaa* *hai*
ram door open.Impf. be.Pres.
'Ram opens the door.'

Ex-17 *maiM* ***(pk1)*** *raam se* ***(jk1)*** *darvaazaa* ***(k2)***
I ram by door
khulvaataa *huuM*
open.Caus.Impf. be.Pres.
'I cause Ram to open the door.'

Below given is the verb frame file (figure 7) for the causative verb *khulvaa.* The causative derivation process is given in the verb frame file of the causative verb *khulvaa*. The verb frame files of *khul* and *khol* can be referred to get an idea of alternations since these verbs are related to the causative verb in the process of causativization.



Verb:: *khulvaa*
(*khul* (derived intransitive) ← *khol* (base transitive) → *khulvaa* (causative))

SID:: *khulvaa* %VCAUS%S1
Eng_Gloss:: cause to open
FRAME_ID:: *khulvaa*%VT%S1%FID1

arc-label	necessity	vibhakti	lextype	posn
pk1	m	0	n	l
jk1	m	se	n	l
k2	m	0	n	l

Figure 7. Representation of Causative Verbs in Verb frames

6. Results

The frequencies and percentage of the 300 verbs were extracted from the pilot Hindi dependency Treebank (HyDT) (2230 sentences) [30] whose coverage is 67.47%. The statistics related to the *verb frames* discussed above are given below in table 2.

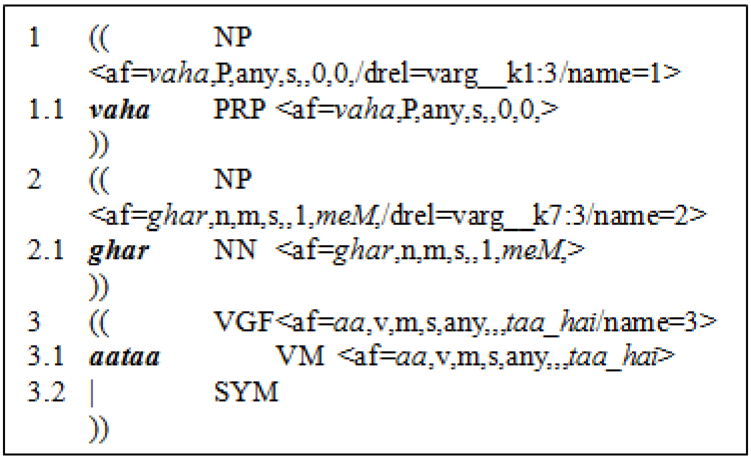
It is clear that the entire structure of the verb frame discussed (refer table 2) is very rich. As of now it is planned to exploit the frames and the verb classes in parsing. They can also be used for various other applications which require a knowledge base, for example, word sense disambiguation, Machine translation, etc.

These *verb frames* are used in Hindi Parser [17,18]. The Output of the Parser is given below for the following examples:

Ex-18 ***vaha ghar meM aataa hai***
he home in comes is
'He comes inside the home.'

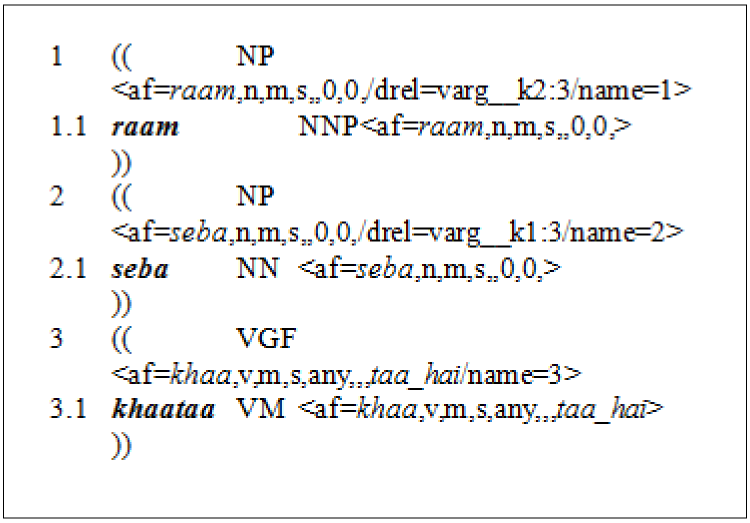
Ex-19 ***raam seba khaataa hai***
ram apple eats is
'Ram eats apple.'

The output of the parser for example *18* is given below in Fig. 8. There will be 2 parse outputs for example 19. They will then go through a ranking process which will rank the parses. The two outputs are given in the Fig. 9 and 10 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 8. Output of the Parser for Example-18

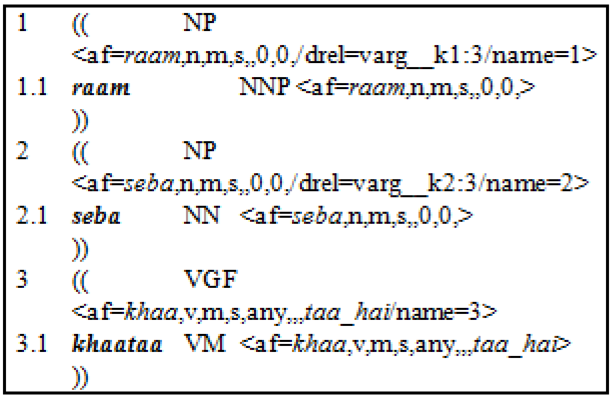


```
1     ((        NP
      <af=raam,n,m,s,,0,0,/drel=varg__k2:3/name=1>
1.1   raam            NNP<af=raam,n,m,s,,0,0,>
      ))
2     ((        NP
      <af=seba,n,m,s,,0,0,/drel=varg__k1:3/name=2>
2.1   seba      NN  <af=seba,n,m,s,,0,0,>
      ))
3     ((        VGF
      <af=khaa,v,m,s,any,,,taa_hai/name=3>
3.1   khaataa   VM <af=khaa,v,m,s,any,,,taa_hai>
      ))
```

Figure 9. Output-I of the Parser for Example-19

Table 2. Verb Frames and their Statistics

Type of the Count and Count	Description
Case-1= Total no. of verbs for which *verb frames* (verb entry) were created is: 300	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.
Case-2= Total no. of verbs for which *verb frames* were created based on their multiple senses is: 486	Here additional senses of the above 300 verbs are included, i.e., each sense of all the 300 verbs has been counted. (The single sense count is 300 + the count of additional senses is 186= 486)
Case-3= Highest sense count for a verb: 11	Verb *nikala* 'leave' has 11 senses.
Case-4= 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 taking its single sense, i.e., the additional senses of the populated verbs are excluded out of the count.
Case-5= Total no. of additional verbs that were populated based on their multiple senses in the existing *verb frames* of certain verbs is: 279	Here additional senses of the populated verbs are included, i.e., each sense of all the populated verbs has been counted.
Case-6 (Case1+Case4)= Total no. of verbs for which *verb frames* (verb frame file) were created + Total no. of additional verbs that were populated in the existing *verb frames* of certain verbs are: 480	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 (Both the categories mean verbs for which *verb frames* are created and the populated verbs).
Case-7 (Case2+Case5)= 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: 765 (In other words, total no. of verbs in the *verb frames* is:)	Here additional senses of the verbs are included, i.e., each sense of all the verbs belonging to both the categories (Both the categories means verbs for which *verb frames* were created and the populated verbs) has been counted



```
1    ((       NP
     <af=raam,n,m,s,,0,0,/drel=varg__k1:3/name=1>
1.1  raam          NNP <af=raam,n,m,s,,0,0,>
     ))
2    ((       NP
     <af=seba,n,m,s,,0,0,/drel=varg__k2:3/name=2>
2.1  seba     NN  <af=seba,n,m,s,,0,0,>
     ))
3    ((       VGF
     <af=khaa,v,m,s,any,,,taa_hai/name=3>
3.1  khaataa  VM  <af=khaa,v,m,s,any,,,taa_hai>
     ))
```

Figure 10. Output-II of the Parser for Example-19

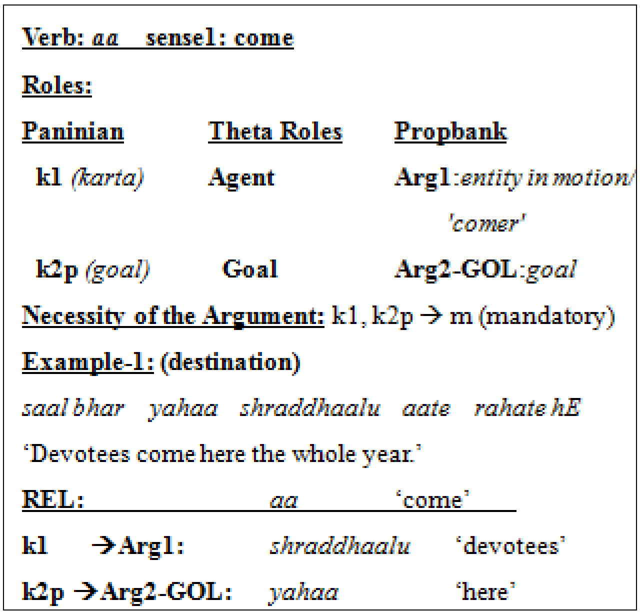
Example 19 has two outputs because *raam* 'Ram' and *seba* 'apple' have 0 (zero) *vibhakti.* In such a situation, two outputs will be generated as both *k1* (karta) and *k2* (karma) take 0 (zero) *vibhakti*. In output-I (figure 9), *raam* is *k2* (karma) and *seba* is *k1* (karta), and in output-II (figure 10), *raam* is *k1* (karta) and *seba* is *k2* (karma).

The *verb frames* developed have been used in Constraint based parser. Constraint based parsing using integer programming [36,18,37] has been successfully tested for Indian languages [38,39]. 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 [3] 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. The performance is not directly comparable as the dependency tagset used for the constraint-based parser is coarse-grained, while the data-driven uses fine-grained. On average, however, the data-driven parser will do better than the constraint-based parser. The small lexicon (linguistic demands of various heads) has a negative impact on parser performance. The efficiency will automatically increase with the increase in the coverage of this lexicon [36,18,37].

7. Comparison of Dependency Annotation with Propbank Annotation

In this work, the Propbank annotation and dependency annotation done on the Hindi data is compared to get a mapping between them. This mapping will help in analysing the relation between dependency annotation and propbank annotation.

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. 110 verb chunks (main verb+auxiliary verb) were extracted. The main verb tokens were taken from the verb chunks. There were 58 different verbs and then framesets were created for theses Hindi verbs. 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. *Arg0*→agent, causer, experience*;* *Arg1*→ patient, theme; *Arg2*→ beneficiary; *Arg3*→ instrument); (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; (**8**)If the verb had more than one sense, each sense was separately represented with all the information mentioned above in (1) to (7). A sample verb entry is provided below in the figure-11:



Verb: *aa* sense1: come

Roles:

Paninian	**Theta Roles**	**Propbank**
k1 *(karta)*	**Agent**	**Arg1**:*entity in motion/ 'comer'*
k2p *(goal)*	**Goal**	**Arg2-GOL**:*goal*

Necessity of the Argument: k1, k2p → m (mandatory)

Example-1: **(destination)**

saal bhar yahaa shraddhaalu aate rahate hE

'Devotees come here the whole year.'

REL:	*aa*	'come'
k1 →Arg1:	*shraddhaalu*	'devotees'
k2p →Arg2-GOL:	*yahaa*	'here'

Figure 11. Frameset for Hindi verb *aa* "come"

In the above figure, frameset for 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 - time location)* in *karaka* relations; *Agent, Goal,* and *Time* in Theta roles and; *Arg1:entity in motion/*'comer', *Arg2-GOL:goal* and *ArgM-TMP:temporal* in Propbank respectively. *k1* (*shraddhaalu* "devotees") and *k2p* (*yahaa* "here") are mandatory; *k7t* (*saal bhar* "year long") is optional.

In the comparative study of Hindi verbs in both dependency annotation and Propbank annotation, it is found that *karta* maps with both *Arg0* and *Arg1* whereas *karma* maps with *Arg1.* In PropBank, Arg0 and Arg1 are understood as framework-independent labels. They are closely linked with Dowty's Proto-roles [40]. Arg0 correlates with the agent, causer, or experiencer, even if it is realized as the subject of an active construction or as the object of an adjunct (by phrase) of the corresponding passive. In this case, Arg0 and Arg1 are very much like k1 and k2 in Hyderabad Dependency Treebank (HyDT) [36]. *k1* and *k2* are annotated based on their semantic roles and not their grammatical relation. HyDT treats the sentences given below in a similar manner, whereas PropBank does not:

Ex- (20) ***The boy opened the door.***
Ex- (21) ***The door opened.***

In HyDT, *the boy* (Ex-20) and *the door* (Ex-21) are annotated as *k1*, whereas in PropBank, *the boy* (Ex-20) is annotated as Arg0 and *the door* (Ex-21) as Arg1. In example *21, the door* is not a primary causer as the verb is unaccusative for Propbank. In HyDT, the concept of unaccusativity is not taken into consideration. This is a significant differentiation that has to be considered when doing the mapping. Hence, k1 is ambiguous, i.e., it maps to Arg0 as well as ARG1. In HyDT, Experiencer subjects are annotated as k4a (anubhavkarta) whereas in PropBank, experience subjects don't have a separate label and are marked as ARG0. Therefore, k4a maps to ARG0 [41].

In Hindi, necessary arguments are dropped frequently and it can be retrieved from the previous discourse. In HyDT, all the dropped arguments are not represented but some of the empty categories like ellipsis, gapping, and empty conjunctions, etc. [42]. Hence, to provide total representations of predicate argument structures involving dropped arguments, Hindi PropBank does semantic role annotation and also empty argument insertion [43].

While working on creation of *verb frames* for Hindi, a mapping was also done between *karaka* roles, *theta* roles, and *propbank* roles. The table 3 given below shows some of the mappings:

Table 3. Mapping between *karaka* roles, *Theta* roles, and *Propbank* roles

karaka	Propbank	Theta-Role	Verb	Sense
k1 (karta)	**Arg1**:*entity in motion / 'comer'*	Agent	*Aa*	**sense1:** come
	Arg0:*agent, doer*		*Kara*	**sense1:** do
	Arg0:*Taker*		*Le*	**sense1:** take
	Arg0:*buyer*		*Le*	**sense2:** buy
	Arg0:*creator*		*Banaa*	**sense1:** make
	Arg0:*believer*		*Maan*	**sense1:**believe
	Arg0:*reacher*		*pahu.Ncha*	**sense1:** reach
	Arg0:*giver*		*de*	**sense1:**give
	Arg0:*entity leaving*		*Cho.Da*	**sense1:**leave
	Arg0:*entity doing the dropping*		*Cho.Da*	**sense2:**drop
k5 (apadaan)	**Argm:** *direction or location or extent*)	Source	*Baha*	**sense1:** flow
	Arg2-SOU:*entity taken from, orprepositional complement of arg1*		*Le*	**sense1:** take
	Arg2-SOU:*seller*		*Le*	**sense2:** buy
	Arg2-SOU:*thing/place exited from*		*nikala*	**sense2:**emerge
k2 *(karma)*	**Arg1**:*thing done*	Theme	*Kara*	**sense1:** do
	Arg1:*thing taken*		*Le*	**sense1:** take
	Arg1: *thing bought*		*Le*	**sense2:** buy
	Arg1:*belief*		*maan*	**sense1:**believe
	Arg1: *entity respected*		*Maan*	**sense2:** respect
	Arg1:*thing given*		*de*	**sense1:**give
	Arg1:*place left*		*Cho.Da*	**sense1:**leave
	Arg1:*Logical subject, patient, thing falling*		*Cho.Da*	**sense2:**drop
k4 (sampradan)	**Arg2**:*entity given to*	Beneficiary	*de*	**sense1:** give
	Arg2:*Hearer*	Recipient	*bataa* *Kaha*	**sense1:**tell

After the creation of *verb frames* for Hindi, the Hindi verbs were classified based on their argument structure given in the *verb frames*. The classification of Hindi verbs is discussed in the section given below.

8. Hindi Verb Classification

This section presents the approach followed for the classification of Hindi verbs. Hindi verbs have been classified based on similar *verb frames,* i.e., based on same argument structure. In other words, the verbs have been classified based on the dependency relations taken by the verbs. These dependency relations are syntactico-semantic in nature. They include mostly *karaka* relations and also some other dependency relations. *Verb frames* of Hindi verbs are same when they have same set of dependency relations. Other information in the *verb frame* (table in the verb frame file) 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.

In the whole process of creating *verb frames,* 49 unique *verb frames* (unique set of *dependency relations*) were developed and the verbs were classified based on these *verb frames.* The verbs which take these 49 unique verb frames are grouped and this group of verbs is listed under each unique verb frame. Table 4 given below shows all the 49 unique *verb frames* (dependency relations):

Table 4. Unique *Verb Frames* (dependency relations)

Frames	Frames	Frames
k1+ k2p+verb	k1+rh+k7+verb	k4a+k1+k1s+ verb
k4a+k1+verb	k1+k2+k2g+verb	pk1+jk1+k2+ verb
k4a+k7p+verb	k1+k7t+verb	pk1+ jk1+k7p+verb
k1+k2+verb	k1+ras+verb	pk1+ jk1+k3+verb
vmod(sequential)+k1 +k2+verb	k1+rt+verb	pk1+ jk1+verb
k1+k7+verb	k1+rt+k2+verb	pk1+ jk1+k5+verb
k1+r6v+verb	k1+k2+k7+verb	pk1+jk1+k4+k2+ verb
k1+k7p+verb	k1+k5+k7+ verb	pk1+jk1+k2+k5+ verb
k1+k5+verb	k1+k2+k7p+verb	pk1+ jk1+ k2p+verb
k1+vmod(simultaneous)+verb	k1+k2p+k2+verb	k1+k1u+verb
k1+vmod(reason)+ verb	k1+k2p+k7t+verb	k1+k1u+k1s+ verb
k1+verb	k1+k7t+k5+verb	pk1+mk1+jk1+ verb
k7p+k7t+k1+verb	k1+rt+k5+k2+verb	pk1+mk1+jk1+k2+ verb
k1+k2+k5+verb	k7+ k7p+k1+verb	
k1+rd+verb	k1+k5+k7p+k2+ verb	
k1+rd+k2+verb	k1+ k2+k3 +verb	
k1+k1s+verb	k1+k2+k4+verb	
k1+rh+verb	k1+k2+k2s+verb	

Out of the verbs that are classified under each unique verb frame, few of them are synonyms as synonyms also take same *verb frames* but not always. Also, some of the verbs that are classified under each unique *verb frame* share certain semantics. So, within the verb frame classification the verbs are grouped under a semantic class and these classes are closer to Levin's semantic classes. The verbs within the main class were sub-classified based on similar *vibhaktis* and *other* information (*necessity, lextype*) in the *verb frame*. Section given below describes the classification process (In the frames given below, for the ease of understanding, the complete format of sense-id and frame-ids of the verbs are not provided. Only sense number of the verb is mentioned. Those verbs which don't have sense and frame ids are the populated verbs).

8.1. Verb Frame: k1+k2+k4+verb

The below listed verbs take the 'k1+k2+k4+verb' verb frame:

(1)bataa1, (2)bola3, (3)kaha4, (4)bataa4, (5)bataa6, (6)bola4, (7)kaha3, (8)sunaa, (9)bataa3, (10)bataa2, (11)bheja1, (12)bola2, (13)kaha1, (14)pahuMchaa2, (15)Daala2, (16)lauTaa1, (17)vaapasa_kara, (18)parosa1, (19)de1, (20)sauMpa1, (22)Cho.Da, (23)baa.NTa1, (24)likha

Out of the above listed verbs, some verbs are 'Verbs of Communication' and some are 'Give verbs'. The verbs that share the semantics of 'Communication' and 'Give' are listed under these semantic classes:

Verbs of Communication: (1)bataa1, (2)bataa2, (3)bataa3, (4)bataa4, (5)bataa6, (6)bola3, (7)bola4, (8)bola2, (9)kaha4, (10)kaha3, (11)kaha1, (12)sunaa, (13)likha

Give verbs: (1)lauTaa1, (2)vaapasa_kara, (3)parosa1, (4)de1, (5)baa.NTa1

In 'verbs of communication' semantic class there are 5 senses of *bataa* verb, 3 senses of *bola* verb, and 3 senses of *kaha* verb.

Verbs that take '*k1+k2+k4+verb' verb frame*, are sub-divided based on the *vibhaktis* and other information (*necessity, lextype*) given in the *verb frame* (tabular form). Main classification of the verbs is based on dependency relations and the sub-classification is based on *vibhaktis* and other information given in the *verb frame* (tabular form). There are 8 sub-classes of verbs based on this sub-division. Below given are the sub-classes of the '*k1+ k2+k4+verb' verb frame:*

I. Frame for the following verbs is given below: (1)bataa1

arc-label	Necessity	vibhakti	lextype	Posn
k1	M	0	n	l
k4	M	ko	n	l
k2	M	0	n|cl	r

II. Frame for the following verbs is given below: (1)bola3, (2)kaha4

arc-label	Necessity	vibhakti	lextype	Posn
k1	M	0	n	l
k4	M	ko|se	n	l
k2	M	0	cl|n	r

III. Frame for the following verbs is given below: (1)bataa4, (2)bataa6, (3)bola4, (4)kaha3, (5)sunaa, (6)bheja1, (7)Daala2, (9)lauTaa1, (10)vaapasa_kara, (11)parosa1, (12)de1 (de%VDT%S1%FID1), (13)sauMpa1, (14)likha, (15)baa.NTa1

arc-label	Necessity	vibhakti	lextype	Posn
k1	M	0	n	l
k4	M	ko	n	l
k2	M	0	n	l

IV. Frame for the following verbs is given below: (1)bataa3

arc-label	necessity	vibhakti	lextype	Posn
k1	M	0	n	l
k4	M	ko	n	l
k2	M	0	cl	l

V. Frame for the following verbs is given below: (1)bataa2

arc-label	necessity	vibhakti	lextype	Posn
k1	M	0	n	l
k4	M	ko	n	l
k2	M	ke_baare_meM	n	l

VI. Frame for the following verbs is given below: (1)bola2, (2)kaha1

arc-label	Necessity	vibhakti	lextype	Posn
k1	M	0	n	l
k4	M	ko	n	l
k2	M	ko|ke_liye	n	l

VII. Frame for the following verbs is given below: (1)pahuMchaa2

arc-label	necessity	vibhakti	lextype	posn
k1	M	0	n	l
k4	M	0|ko	n	l
k2	M	taka	n	l

In the above *verb frame,* i.e., *k1 + k2 + k4 + verb,* all the verbs that take this verb frame, are listed under it. Certain verbs listed in the verb class have semantic similarities such as *verbs of Communication* and *Give verbs.* Then the verbs were sub-divided based on similar *vibhaktis* and other information given in the *verb frame* (tabular form).

Within the verb frame classification, the verbs that share certain semantics are grouped under a semantic class and these classes are closer to Levin's semantic classes. Semantic similarities are not found among the verbs of most of the verb frames. Table 5 shows some of the verb frames which contain semantic classes and also shows the Levin's semantic classes to which our semantic classes are closer to (refer table 5).

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. The verbs are classified based on their dependency relations (*karaka* relations and other relations) 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) [3]. The verbs were again sub-classified based on the *vibhaktis*, *necessity* and *lex-type* given in the verb frame. The reason behind doing this was that verbs had the same dependency relations but varied in case of *vibhaktis.* The sub-classification was done to capture finer semantics among the verbs. *Necessity* and *lex-type* features were also considered as a base for sub-classification because they also may help in getting finer classes.

In the major classification, it is found that different groups of verbs share certain semantics. For example, the *k1+k2p+verb* frame 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', etc. 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 participate in different sub-classes. The motion verbs mentioned above are not listed under one single sub-class but distributed among 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 there are 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 it is observed 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 *k1+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, certain classes are formed based on some Hindi sentence constructions, such as

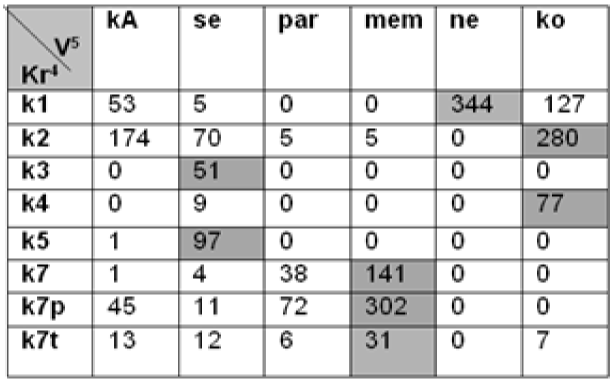
Table 5. Semantic Classes in the Verb Frames

Verb Frame	Semantic Classes and verbs within those classes	Levin’s Semantic Class names
k1+k2+k2s	**Appoint Verb:** *chun3, nirvaachit_kar,samajh2, bataa5* **Characterize Verb:** *maan2, Thaharaa, ghoshit_kar*	**Appoint Verbs,** **Characterize Verb**
pk1+jk1+k2p	**Verbs of Motion:** *pahuzchaa1, Cho.D3*	**Verbs of Motion**
k4a+k7p	**Pain Verbs:** *dukh1, darda_ho, pii.Daa_ho*	**Pain Verbs**
k1+k7t+k5	**Verbs of Motion:** *ChuuT3, chal3, nikal6, ravaanaa_ho*	**Verbs of Motion**
k4a+k1	**Verbs of Perception:** *laga5, dikha1, nazar_aa, dikhaaii_de*	**Verbs of Perception**
k1+k7p	**Lodge Verbs:** *raha1, basa, tika2, ruka2, Tahara1, raha2, uwara1* **Run Verbs:** *ghuuma1, ghuuma-phira, phira1, bhaag1, dau.Da , kuud1, u.Da1, taira1, phudak1, bhaTak1, reMga, uchala*	**Lodge Verbs** **Run Verbs**
pk1+jk1+k2	**Verbs of Ingesting:** *chaTaa, chakhaa, jitaa, khilaa, pilaa* **Verbs of Communication:** *samajhaa1, pa.Dhaa1*	**Verbs of Ingesting** **Verbs of Communication**
k1+k2+k7p	**Verbs of Putting:** *barasaa1, Daal3, biChaa,sajaa2, lagaa, jamaa, lagaa5, jadZa, baiThaa/biThaa, fita_kar, lagaa7, TaaMka, laTakaa1, TaaMga1* **Poke Verbs:** *chubhaa1, ga.Daa*	**Verbs of Putting** **Poke Verbs**
k1+k2	**Verbs of Ingesting:** *chakha1, khaa1, chabaa1, nigal1, gataka, khilaa1, pii1, pii2, chuga* **Cooking Verbs:** *pakaa1, ubaal1, tala1, banaa2* **Destroy Verbs:** *rauMda1, ujaa.Da1, miTaa, ukhaa.Da, kuchala1, piisa , khaa2, barbaad_kar* **Anger Verbs:** *biga.Da2, bha.Dak, gussaa_kar , chillaa3, barasa, DaaMTa1, phaTakaar1* **Social Interaction Verbs:** *caaha1, la.Da1, jhaga.Da , jhaga.Daa_kar, la.Daaii_kar , ulajha , la.Da2, bola6, mila2*	**Verbs of Ingesting** **Cooking Verbs** **Destroy Verbs** **Psych Verbs** **Social Interaction Verbs**
k1+k2p	**Motion Verbs:** *aa1, pahuMcha, padhaar1, jaa1, aa*6 , *lauTa1, vaapas_aa, chal, dau.Da, bhaag, cha.Dha1, chala4* **Raise Verbs (Quantity Increment):** *cha.Dha2, ba.Dha*	**Motion Verbs** **Verbs of Change of State**
k1+k2+k4	**Verbs of Communication:** *bataa1, bataa2, bataa3, bataa4, bataa6, bola3, bola4, bola2, kaha4, kaha3, kaha1, sunaa, likha* **Give verbs:** *lauTaa1, vaapas_kar, parosa1, de1*	**Verbs of Communication** **Give verbs**

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.

Syntactico-Semantic helps better classification and has its own advantages. The dependency relations have both syntactic and semantic properties. Since *karaka* labels express the roles of various participants in an action, they incorporate some degree of semantic information. On the other hand, the labels also interact with the syntactic properties such as agreement and case marking which provide cues for better parsing. It is found that there is a strong co-relation between most *vibhakti-karaka* occurrences (shaded cells in Table 6 given below). k7 ('place') for example, overwhelmingly takes *meM* post-position, k3 (*karana*) takes *se* in all the cases. Of course, there are some competing relations which show preference for the same post-position. In such cases only the post-position information will not be sufficient and it will be required to take into account other syntactic cues as well. These syntactic cues can be TAM (tense, aspect and modality) of the verb, verb class information, etc.

Table 6. *karaka-vibhakti* correlation



V5 Kr4	kA	se	par	mem	ne	ko
k1	53	5	0	0	344	127
k2	174	70	5	5	0	280
k3	0	51	0	0	0	0
k4	0	9	0	0	0	77
k5	1	97	0	0	0	0
k7	1	4	38	141	0	0
k7p	45	11	72	302	0	0
k7t	13	12	6	31	0	7

9. Conclusion and Future Work

This paper discusses two aspects of Hindi verbs: (1) Creation of verb frame files within Paninian Grammatical Framework and (2) Classification of Hindi verbs based on the verb frames. The verb frame file contains the following information: (1) description of the verb; and (2) verb frame for the verb. 49 unique *verb frames* were developed and the verbs were classified based on these unique *verb frames.* There exist certain semantic similarities among the verbs participating in the above classification. So, within the verb frame classification the verbs are grouped under a semantic class and these classes are similar to Levin's semantic classes. The verbs within the main class were also sub-classified based on similar *vibhaktis* and *other* information (*necessity,lextype*) given in the *verb frame*.

The uses of these verb frames are: (i)It becomes a knowledge base for various NLP applications, e.g., parsers, MT, language generation, etc; (ii)It becomes a linguistic resource which gives the classification of Hindi verbs; (iii)It is helpful for the annotators in deciding various dependency relations for a given verb in the corpus; (iv)It forms a basis for linguistic analysis.

Our verb classification is similar to the classification given by Sahay [32] who has classified the Hindi verbs based on their *karaka* requirements. The only difference between our classification and Sahay's classification is that our classification has more verb frames on whose basis the verbs were classified. This work contains 49 unique verb frames whereas Sahay [32] has 21 verb frames. This work also attempted to capture the semantic similarities between the verbs that are classified based on dependency relations.

A mapping is done between Propbank annotation and dependency annotation based on Paninian Grammatical Framework [36,3]. This will help in analysing the relation between dependency annotation and propbank annotation. In the comparative study of Hindi verbs in both dependency annotation and Propbank annotation, it is found that *karta* maps with both *Arg0* and *Arg1* whereas *karma* maps with *Arg1.* Arg0 correlates with the agent, causer, or experiencer, even if it is realized as the subject of an active construction or as the object of an adjunct (by phrase) of the corresponding passive. In this case, Arg0 and Arg1 are very much similar to *k1* and k2 in Hyderabad Dependency Treebank (HyDT) [36]. *k1* and *k2* are annotated based on their semantic roles and not their grammatical relation. In HyDT, the concept of unaccusativity is not taken into consideration. This is an important differentiation that has to be considered when doing the mapping. Hence, k1 is ambiguous, i.e., it maps to Arg0 as well as ARG1. In HyDT, Experiencer subjects are annotated as k4a (anubhavkarta) whereas in PropBank, experience subjects don't have a separate label and are marked as ARG0. Therefore, k4a maps to ARG0 [41].

A comparative study of theta roles with *karaka* relations was also done. This study shows that any theta role such as *agent*, *theme*, or *instrument* can occur as a *subject*. Object is generally *theme/patient* in a transitive verb. Both *subject-agent* and *subject-theme/patient* combinations are *karta. Karma* maps to *object-theme/patient.*

This paper has compared a couple of semantic classes that were created with the Levin's semantic classes and proposes a plan to do an extensive comparison as a future work.

REFERENCES

[1] R. Begum, S. Husain, D. M. Sharma, and L. Bai. Developing Verb Frames for Hindi. In *Proceedings of The Sixth International Conference on Language Resources and Evaluation (LREC)*. Marrakech, Morocco, 2008

[2] I. A. Mel'cuk. *Dependency Syntax: Theory and Practice*, State University, Press of New York, 1988.

[3] A. Bharati, V. Chaitanya, and R. Sangal. *Natural Language Processing: A Paninian Perspective,* Prentice-Hall of India, New Delhi, 1995.

[4] K. P. Mohanan. Grammatical relations in Malayalam, In Joan Bresnan (ed.), The Mental Representation of Grammatical Relations, MIT Press, Cambridge, 1982.

[5] T. Mohanan. *Arguments in Hindi.* CSLI Publications, 1994.

[6] B. Comrie. *Language Universals and Linguistic typology: Syntax and Morphology.* The University of Chicago Press, 1981.

[7] L. Tesnière. *Eléments de Syntaxe Structurale*. Klincksiek, Paris, 1959.

[8] T. Kakkonen. Dependency Treebanks: Methods, Annotation Schemes and Tools. *Proceedings of the 15th Nordic Conference of Computational Linguistics*, Joensuu, Finland, 2005.

[9] P. Kiparsky, J. F. Staal. 'Syntactic and Semantic Relations in Panini', *Foundations of Language*, 1969.

[10] C. Shastri. *Vyakarana Chandrodya (Vol. 1 to 5).* Delhi: Motilal Banarsidass (In Hindi), 1973

[11] Z. Žabokrtský, O. Smrž. Arabic syntactic trees: from constituency to dependency. *In Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics (EACL'03)* – Research Notes, Budapest, Hungary, 2003.

[12] N. Habash, O. Rambow. Extracting a Tree Adjoining Grammar from the Penn Arabic Treebank. In *JEP-TALN 2004, Session Traitement Automatique de l'Arabe*, Fes, Morocco, 2004.

[13] N. Habash, R. Gabbard, O. Rambow, S. Kulick and M. Marcus. Determining Case in Arabic: Learning Complex Linguistic Behavior Requires Complex Linguistic Features. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1084–1092, 2007.

[14] O. Smrz, V. Bielicky, I. Kourilova, J. Kracmar, J. Hajic and P. Zemanek. Prague Arabic Dependency Treebank: A Word on the Million Words. In *Proceedings of the Workshop on Arabic and Local Languages*, LREC, Marrakech, Morocco, 2008. [http://ufal.mff.cuni.cz/~smrz/LREC2008/padt-lrec.pdf]

[15] L. Tounsi, M. Attia and J. V. Genabith. Automatic treebank-based acquisition of Arabic LFG dependency structures. In *Proceedings of the EACL 2009 Workshop on Computational Approaches to Semitic Languages*, pages 45–52, Athens, Greece, 2009.

[16] N. Habash, R. Faraj and R. Roth. Syntactic Annotation in the Columbia Arabic Treebank. In *Proceedings of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR)*, Cairo, Egypt, 2009.

[17] S. Husain. *A Generalized Parsing Framework based on Computational Paninian Grammar*. Ph.D. thesis, IIIT-H, Hyderabad, India, 2011.

[18] A. Bharati, S. Husain, D. M. Sharma, and R. Sangal. Two stage constraint based hybrid approach to free word order language dependency parsing. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT09),* Paris, 2009

[19] J. M. Gawron. Lexical Representations and the Semantics of Complementation. New York: Garland Publisher, pp. 111, 1988

[20] A. Radford. Transformational Grammar: A First Course. Cambridge University Press, pp. 263, 1988.

[21] B. Levin. *English verb classes and alternations, A Preliminary Investigation*. The University of Chicago Press, 1993

[22] B. Levin, M. Rappaport Hovav *Unaccusativity: At the Syntax-Lexical Semantics Interface*, Linguistic Inquiry Monograph 26, MIT Press, Cambridge, MA, 1995

[23] K. Kipper, H. T. Dang and M. Palmer. Class-based construction of a verb lexicon. In *AAAI/IAAI,* pages 691-696, 2000.

[24] K. Kipper. *VerbNet: A broad-coverage, comprehensive verb lexicon.* Ph.D. thesis, Computer and Information Science Dept., University of Pennsylvania, Philadelphia, PA, 2005.

[25] G. A. Miller, R. Beckwith, C. D. Fellbaum, D. Gross and K. Miller. *WordNet*: *An online lexical database*. Int. J. Lexicograph. 3, 4, pp. 235–244., 1990.

[26] C. Fellbaum.. *WordNet: An Electronic Lexical Database.* Cambridge, MA: MIT Press, 1998, ed.

[27] M. Palmer, D. Gildea, and P. Kingsbury. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics, 31:1.*, pp. 71-105., 2005

[28] P. Kingsbury, M. Palmer. From Treebank to PropBank. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation,* Las Palmas, Canary Islands, Spain, 2002.

[29] C. F. Baker, C. J. Fillmore and J. B. Lowe. The Berkeley FrameNet project. In *Proceedings of the COLING-ACL*, Montreal, Canada, 1998.

[30] R. Begum, S. Husain, A. Dhwaj, D. M. Sharma, L. Bai, and R. Sangal. Dependency Annotation Scheme for Indian Languages. In *Proceedings of The Third International Joint Conference on Natural Language Processing (IJCNLP)*, Hyderabad, India, 2008.

[31] S. Jha, D. Narayan, P. Pande and P. Bhattacharyya. A WordNet for Hindi. *International Workshop on Lexical Resources in Natural Language Processing,* Hyderabad, India, January 2001.

[32] C. Sahay. *Hindi ke Muul Vaakya Saanche*, Kumar Prakaashan, Agra, pp. 91-100., 2004

[33] A. R. Tripathi. *Hindi Bhashaanushasan.* Bihar Granth Academy, Patna, 1986.

[34] T. Reinhart. *Causativization and Decausativization*: Unpublished technical report. LSA 2005.

[35] R. Begum, D. M. Sharma. A Preliminary Work on Causative Verbs in Hindi. In *Proceedings of the 8th Workshop on Asian Language Resources* at COLING 2010

[36] A. Bharati, S. Husain, D. M. Sharma, and R. Sangal. A Two-Stage Constraint Based Dependency Parser for Free

Word Order Languages. In *Proceedings of the COLIPS International Conference on Asian Language Processing (IALP)*. Chiang Mai, Thailand, 2008.

[37] A. Bharati, S. Husain, M. Vijay, K. Deepak, D. M. Sharma, and R. Sangal. Constraint Based Hybrid Approach to Parsing Indian Languages. In: *The 23rd Pacific Asia Conference on Language, Information and Computation (PACLIC 23)*. Hong Kong, 2009

[38] A. Bharati, R. Sangal. Parsing Free Word Order Languages in the Paninian Framework, *ACL93: Proc. of Annual Meeting of Association for Computational Linguistics*, 1993

[39] A. Bharati, R. Sangal, and T. P. Reddy. A Constraint Based Parser Using Integer Programming, In *Proc. of ICON-2002: International Conference on Natural Language Processing, 2002.*

[40] M. Palmer, D. Gildea, and N. Xue. Semantic role labeling. In Graeme Hirst, editor, *Synthesis Lectures on Human Language Technologies*. Morgan and Claypool, 2010.

[41] A. Vaidya, J. D. Choi, M. Palmer, B. Narasimhan. Analysis of the Hindi Proposition Bank using Dependency Structure, *Proceedings of ACL workshop on Linguistic Annotation*, 21-29, Portland, Oregon, June, 2011

[42] A. Bhatia, R. Bhatt, B. Narasimhan, M. Palmer, O. Rambow, D. M. Sharma, M. Tepper, A. Vaidya, and F. Xia. Empty Categories in a Hindi Treebank. In *Proceedings of the 7th International Conference on Language Resources and Evaluation,* pages 1863–1870, 2010.

[43] A. Vaidya, J. D. Choi, M. Palmer, and B. Narasimhan. Empty Argument Insertion in the Hindi PropBank**,** In the Proceedings of LREC, Instanbul, Turkey, 2012