AnnCorra : TreeBanks for Indian Languages
Guidelines for Annotating Hindi TreeBank

(version – 2.5) 17/09/2012

Akshar Bharati, Dipti Misra Sharma, Samar Husain, Lakshmi Bai, Rafiya Begam, Rajeev Sangal
Language Technologies Research Center
IIIT, Hyderabad, India
{dipti, samar, lakshmi, sangal}@iiit.ac.in, rafiya@students.iiit.ac.in

Content

1. Background

A major bottleneck in developing various natural language applications for Indian languages is the unavailability of appropriate language resources. For any NLP application, certain linguistic knowledge is required. This knowledge can be prepared in the form of dictionaries, grammars, wordformation rules etc. An alternative approach is to annotate linguistic knowledge in electronic texts. The annotated texts can be used for machine learning, developing these resources by extracting the knowledge etc. Penn Treebank for English (Marcus et al., 1993), Prague Dependency Tree bank for Czech (Hajicova, 1998) etc. are some of the efforts in this direction.

The idea of developing such a resource for Indian languages was first decided to be taken up at the ”Workshop on Lexical Resources for Natural Language Processing”, 58 Jan 2001, held at IIIT Hyderabad. The task was named as AnnCorra, shortened for ”Annotated Corpora”.

For achieving this, certain standards had to be drawn in terms of selecting a grammatical model and developing tagging schemes for the three levels of sentential analysis, POS tagging, chunking and syntactic parsing. Since Indian languages are morphologically richer, they allow the order of the words to be more flexible. This also implies that the information at the morphological level can be crucial for sentence analysis. Hence, coming up with standards for morph feature representations for various Indian languages also becomes critical. The standards for POS tagging, Chunking and Morph feature representation were initially arrived at in the project ILILMT System’. In this project nine language pairs were taken for developing bidirectional MT systems. The project is being carried out in a consortium mode and is funded by DIT, Government of India. For defining the standards for the above, several workshops were conducted with participation from major NLP groups working on the nine languages undertaken in the project.

The natural next step after POS tagging, chunking and morph analysis is sentence level parsing. Thus, it was decided to work out a scheme for annotating tree bank for Hindi. Hindi was chosen as an example language. The theoretical model that has been adopted for the sentence analysis is Panini's grammatical model which provides a level of syntactico-semantic analysis.

This document, a guidelines on dependency annotation of Hindi has two Parts. Part-1 contains a description of the grammatical model and the details of the tagging scheme. Part-2 contains examples of certain typical constructions of Hindi and their analysis in Paninian dependency model.

2. The Task

The task is to develop a dependency Treebank for Hindi. As part of the task, it is decided to annotate the corpora for the following linguistic information,

a). Relevant morph features for the token in the context (lexical level)
b). POS tag (lexical level)
c). Chunk (phrasal level (without distorting the internal dependencies))
d). Dependencies (sentential level – syntactico-semantic)
e). Shared and missing arguments
f). Sentence type
g). Voice type
h). Conference in specific cases

The task can be better explained with the help of an illustration. Given below is a sentence from Hindi:

Ex1	*Hin-wx:*	*rAma*	*ne*	*mohana*	*ko*	*nIlI*	*kiwAba*	*xI*
	Hin-Roman:	Ram	ne	Mohan	ko	niilii	kitaaba	dii
	Gloss :	ram	erg	Mohan	acc	blue	book	gave
	Eng :	'Ram gave a blue book to Mohan.'						

The above example would have the following dependency analysis:

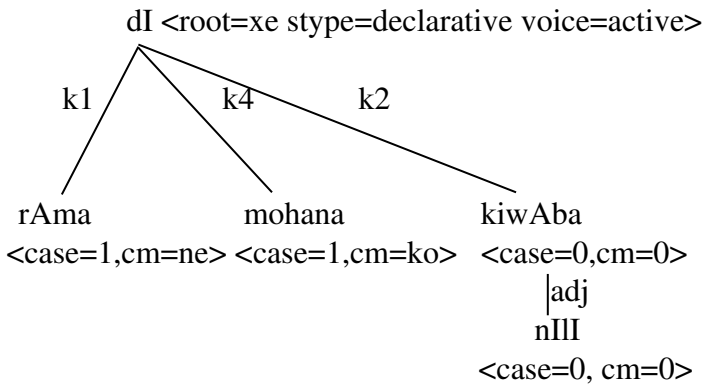

Figure 1

The dependency representation (Figure 1) of the example (1) represents that *Ram* is the 'kartaa' (doer marked as k1) of the action denoted by the verb *dI* 'gave', *Mohan* is the 'sampradana' (recipient marked as k4) and *nIlI kitAba* 'blue book' is the 'karma' (locus of result of the action denoted by the verb marked as k2) of the verb. The root node of a dependency tree is normally a verb. In the Hindi treebank, each node is annotated for the morphological information (not fully represented here). Apart from the morphological information annotated for the main verb (the root node) in a sentence, two additional features (sentence type and the voice type) are also annotated.

The main task, however, is to explicitly mark the relations (arc labels) between various elements (words) of a sentence. This obviously requires a grammatical model basing which the dependency relations can be annotated.

3. PART 1-A

This section of the document has a description of the grammatical model used in designing the tagging scheme and the details of the tagging scheme. Some details about the corpora and where it has been taken from are also provided.

3.1 Grammatical Model

Paninian grammatical model has been chosen for annotating the dependency relations in the Hindi-Urdu Treebanks. Since the analysis is in Paninian framework, the tag names also reflect that. As mentioned in the previous section, the model offers a syntactico-semantic level of linguistic knowledge. Preference for this model is based on:

a) The model, not only offers a mechanism for SYNTACTIC analysis, but also incorporates the SEMANTIC information (dependency analysis).
b) Indian languages have a relatively free word order, hence a dependency grammar based approach would be better suited for sentence analysis.

The Paninian grammatical model treats a sentence as a series of modifier – modified elements starting from a primary modified (generally a finite verb) . The objective of the grammarian, according to this framework, is to extract meaning from a sentence as spoken by a lay person. It works with the assumption that language is used for communication. The meaning in a sentence is encoded, not only in words (the lexical items), but also in the relations between words. Thus, every word in a sentence has a twofold role towards composing the larger meaning; (i) the concept it represents and (ii) the participatory role it plays in the sentence in relation to the other words. The latter is most often expressed through some explicit markers such as nominal inflections, verbal inflections etc. This implies that certain linguistic cues are explicitly available in a sentence using which one can extract the meaning from a sentence. Morphologically rich languages such as Sanskrit (a classical Indian language), Telugu, Tamil etc (some of the modern Indian languages) have the grammatical information in the words themselves (through affixes). However, for languages such as Hindi, one has to go beyond lexical items and use postpositions (for case marking) and auxiliaries (for tense, aspect, modalities) for this purpose. A step of local word grouping (LWG - Bharati et al, 1995) helps in computing the grammatical information easily. Thus, the Paninian Grammatical model (let us refer to it as Computational Paninian Grammatical (CPG) model) can easily be designed to meet the parsing requirements and also help in extracting meaning from a sentence.

The grammatical relations which have been considered here are of two types; (1) karaka, and (2) Relations other than karakas.

A number of direct participants are needed for an action to be completed successfully. The 'doer' of an action, time when the action is carried out, receipient of an action which requires transfer of some sort, source of an action which denotes a point of departure etc are some examples of the direct participants (karakas) of an action. There could also be other players when an action is being

carried out. These players may not have any direct role in the action though. *Reason* and *purpose* are two examples of such players. 'karakas' are the roles of various direct participants in an action. An action in a sentence is normally denoted by a verb. Hence, a verb becomes the primary modified (root node of a dependency tree) in a sentence. Panini has spelled out six karakas (Bharati et al., 1995). The sentence may contain a number of relations between words which are not ’karaka’ relations. The scheme adopted for annotating dependency relations in the Hindi treebank refers to these relations as ’other than karaka’ relations. As mentioned earlier, purpose, reason, genitive etc. would fall under the second type of relations in CPG.

The six kaarakas given by Panini are *kartaa* (doer of an actions), *karma* (locus of the result of the action), *karana* (instrument), *sampradaana* (receipient/beneficiary), *apaadaana* (source) and *adhikarana* (location).

kartaa is defined as the ’most independent’ of all the *karakas* (participants). *kartaa* is the one who carries out the action. It is conceptually different from the agent theta role as it does not always have volitionality. It is the locus of the activity implied by the verb root. In other words, the activity resides in or springs forth from the ’kartaa’ (Bharati et al., 1995). For example:
Ex2. Ram made the basket.

Ram is ’kartaa’ here as he is performing the action of making the basket. In Paninian grammar, every action is a bundle of sub-actions and all the participants (karakas) in an action have a sub-action located in them. Thus every karaka is the ’kartaa’ (doer) of its own action. Therefore, if we take Ex3.a,

Ex3.a *Ram opened the lock with a key*

’Ram’(’kartaa’), ’lock’(karma) and ’key’(instrument) are the three karakas (participants) in the action of ’opening’. The larger action of 'opening the lock' involves following sub-actions (i) action of Ram, (ii) action of the lock and (iii) action of the key. (i) involves Ram's action of inserting the key in the lock and also turning it. (ii) is the action of key of unlocking the lever and (iii) involves lock's action of opening. Therefore, all the three 'Ram', ’lock’ and ’key’ are the ’kartaa’ of the sub-actions carried out by each of them. Each of these actions can be brought into focus by structuring a sentence with a changed 'kartaa'. (Ex3.b) and (Ex3.c) exemplify this.

Ex3.b *The lock opened*

Here, the action is of the opening of the lock. If a lock is rusted, then even if the key turns the lever, the lock would not open as the lock’s action is not carried out.

Thus, in (Ex3.b) the focus is on the ’lock’s action’. This is expressed by making 'lock' as the 'kartaa'.

Ex3.c *This key opened the lock*

Similarly, in (Ex3.c) the key's sub-action is brought into focus by making it the 'kartaa'. A wrong key cannot open a lock.

3.2 The Scheme

The tagging scheme here includes tagsets at various levels of annotation, the representation format, the naming conventions etc.

3.2.1 A Little History

The first step in the direction of coming up with a tagging scheme for annotating dependencies at the sentential level for Indian languages was conceived and worked out in 2000 itself. At the time it was decided to break the dependency annotation into two parts. Local dependencies and the dependencies of postpositions and auxiliaries to their respective nouns or verbs etc would be done separately. Since it is easy to mark such dependencies automatically with fairly high degree of accuracy, it was decided to leave these out of the manual task of annotation. Thus, the dependency annotation would be manually marked only between the heads of the chunks, i.e., at the inter-chunk level. A chunk is taken to be a basic unit for marking the syntactico-semantic relations with the assumption that the intra-chunk dependencies could be obtained automatically by using a rule based system. The verb chunk is more or less a grouping of the verb base form and its tense, aspect and modality (TAM) auxiliaries. The practical aspect of this decision was that it allowed saving the effort in manual annotation. Once inter-chunk annotation is over, the intr-chunk dependencies could be automatically obtained using a relatively highly accurate rule based tool. Thus, the dependency annotation guidelines do not include a description of intra-chunk relations.

The task of treebanking could not be immediately carried forward at the time as other tasks such as POS tagging and chunking etc for Indian languages needed prior attention. Substantial amount of work was then done in the direction of developing standards for POS tagging and chunking for Indian languages and a tagging scheme for the same (Bharati et al. 2006). It was decided to revisit the AnnCorra Tagset for inter-chunk dependency relations in Jan 2005. Each of the tag was discussed and a revised list was arrived at. The tagset contained around 26 tags.

Based on the tagset developed in 2005, a small set of sentences (about 2000) from Hindi were annotated. During this process it was noted that there were constructions which could not be satisfactorily captured in the existing tagset. Subsequently, the tagset was revisited and the tagset given in these guidelines was evolved. The intra-chunk dependency labels (see Appendix 10.4) were also spelled out subsequently.

3.2.2 Corpora

The corpora for the treebank has been acquired from ISI, Calcutta. The Hindi corpus is mainly newspaper texts from Dailies. The domains chosen for the annotation are general news articles (350k), tourism and conversational texts (50k).

3.2.3 Treebank Representation (SSF)

The annotated data is stored in SSF format (Bharati et al., 2007). The SSF is a four column format in which the first column is for address, the second column is for the token, the third column is for the category of the node and the fourth column has other features. Any required linguistic or other information can be annotated in the fourth column using an attribute value pair. Thus, POS and chunk category of the tokens would be in the third column and the morph, dependency and any other information pertaining to a node would appear in the fourth column. For more details on SSF read (Appendix10.2)

3.2.4 Naming conventions

The naming conventions adopted in the treebank are described in the following sub-sections.

A. Naming tokens

Every lexical item and chunk would have a name. The attribute for naming is 'name'. Values for **lexical nodes** would be the concerned lexical item. In case there are more than one occurrences of the same word the value for the name attribute would be the lexical item followed by a numerical. For example, if the token is ’phala’ (fruit), it would be represented as name=’phala’. In case ’Pala’ occurs twice in a sentence, the first time its naming feature would be name=’phala’ and the second time it will be named as name=’phala2’. Some more examples are :

Hari <name='Hari'>
said <name='said'>
Ram <name='Ram'>
Ram <name='Ram2'>
! <Name='!'>

B. Naming convention for Chunks

The chunks are named as their respective phrase tags(NP/VP/JJP). As in the case of lexical items, the subsequent occurrences of the chunks are also named by appending an iterated number (starting with 2) to the phrase tag. For example,

((Hari/NNP)) NP <name='NP'>
((gave/VBD)) VP <name='VP'>
((Ram/NNP)) NP <name='NP2'>
((a/DET book/NN)) NP <name='NP3'>

C. Naming NULL nodes

In case a NULL node is inserted, the NULL node would be assigned a appropriate POS tag. The naming of a NULL node would also be similar to the naming of tokens. That is the node would be named name=’NULL’ and the

subsequent NULL nodes within the same sentence would be assigned names NULL2, NULL3 etc. Similarly, at the chunk level, a chunk containing a NULL node would have the chunk category of the type NULL__NP, NULL__VGF, NULL__JJP etc depending on the POS category of the NULL node within a chunk. The naming on these chunks would be similar to the other chunks, i.e. a NULL__NP chunk would be named as 'NULL__NP' etc.

The above are the naming conventions adopted in the Treebank.

D. Naming the examples in this manual

For ease of access, the examples for various labels and constructions have also been given ids in this document. In PART-1B, the convention is that every example starts with Relation-DS-. Thereafter, the id has the relation label for which the example stands for, followed by a numerical. For example, examples for *kartaa karaka* would have the following ids – Relation-DS-k1-1, Relation-DS-k1-2 and so on. Similarly, for *karma karaka* examples the ids would be Relation-DS-k2-1, Relation-DS-k2-2 and so on. This allows us a flexibility of adding more examples for each type of relation at a later stage.

In PART-2, the examples are named as [Construction type-DS-examplenumber]. Thus, examples for causative constructions would read as follows : Causative-DS-1, Causative-DS-2 and so on.

3.2.5 Relations and Tag labels

(A) The POS and Chunk Tags

The tagging scheme for POS and Chunk annotation has been developed through conducting various workshops in which scholars representing several major languages of India participated. The scheme aimed at coming up with a tagset which would be comprehensive to the extent possible covering issues from all Indian languages and should be simple for the annotators.

Annotation guidelines based on the above scheme are also prepared (Appendix 10.3). The task of annotating POS and chunk in several Indian languages is already going on under the ILMT project funded by Department of Information Technology (DIT), Ministry of Communication and Information Technology (MCIT), Government of India.

(B) Dependency labels

The scheme contains about 68 tags for the inter-chunk dependency relations (these include certain fine grained distinctions as well) which are arrived at considering various types of sentence constructions in Hindi. These labels contain relations (a) *karaka* and *non-karaka* dependency relations (b) some underspecified

tags of the type vmod, nmod etc and (c) some tags which indicate relations which are not exactly dependency relations but are required for representing certain nodes in the tree (more details are given below).

As mentioned earlier, the grammatical model captures certain syntactico semantic relations. The tag labels represent various karaka and other than *karaka* relations. All *karaka* relation labels start with a 'k-' followed by a numerical. Although the basic number of *karakas* is six, there are a number of relations which are subtypes (destination(at a finer a level of granularity) of *karakas*. Some of these are k2g (secondary *karma*), k2p (destination, a subtype of *karma*), k7t (time), k7p (place) etc.

There are some relation labels which begin with a 'k-' but are not really *karaka* labels. These relations, instead, in some or the other way are related to a *karaka*. Examples of some such relations are k1s (noun complement of *karta*), k2s (noun complement of *karma*), k1u (comparative of a *karta*), k2u (comparative of a *karma*) etc. More details about each of these relation types are described below.

The labels for dependency relations other than *karaka* relations start with an 'r-'. For example, r6 (genitive), rt (purpose), rh (reason) etc.

There are certain relations which do not fall under 'dependency relation' directly but are required for showing the dependencies indirectly. For example, the labels 'ccof ' and 'pof ' in the tagging scheme appear to represent 'co-ordination' and 'complex predicates' respectively. Both of these are not really dependency relations.

Figure 2 gives the type hierarchy of the dependency relations. The figure shows the relations from coarser to finer on a modifier modified paradigm. The classification shown in Figure 2 allows underspecification of certain relations in cases where a finer analysis is not very significant for this level of annotation and is also more difficult for decision making for the annotators. Therefore, the labels such as k1, k2 etc

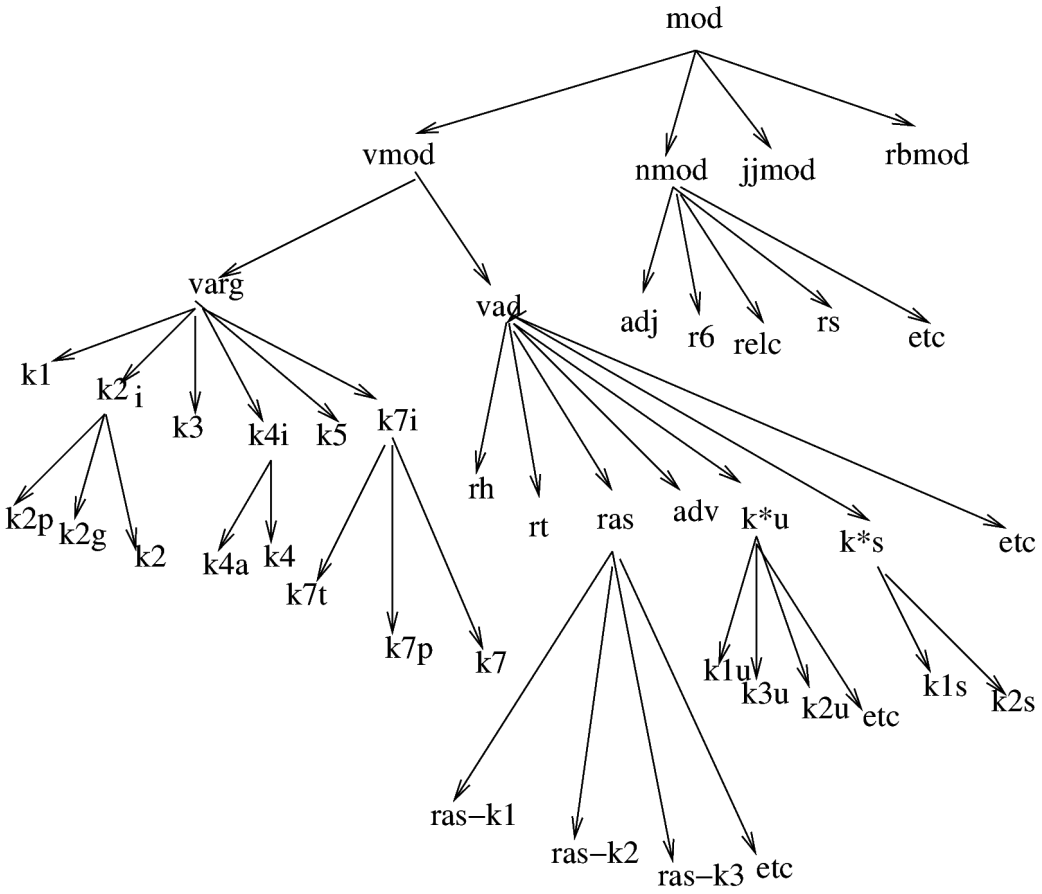

Figure 2 : Inter-chunk Dependency Relation Types

The classification shown in Figure 2 allows underspecification of certain relations in cases where a finer analysis is not very significant for this level of annotation and is also more difficult for decision making for the annotators. Therefore, the labels such as k1, k2 etc represent a finer level depicted deeper in the tree, whereas, labels such as 'vmod', 'nmod' show an underspecified representation of the relation. More details for this are given under respective labels in Section 4.1 of this document.

The semantics of a verb plays a major role in deciding the *karaka* relations of various elements in a sentence. However, there are syntactic cues which help too in these decisions. Normally, *karta* and *karma* agree with the verb. The *karta* takes a *zero* vibhakti (nominative case) when it agrees with the verb. Similarly, if the *karma* agrees with the verb, it occurs in its nominative form. In case the *karta* does not agree with the verb, it takes the following vibhaktis (it is followed by the postpositions): *ne, ko, se, xvArA*. In all these cases the verb is inflected by different tense, aspect and moods. Therefore, a mapping between vibhakti (noun case markers) and TAM (tense, aspect and modality) can be quite useful for identifying relations such as *karta* and *karma*.

A default for annotating *karakas* in sentences with more than one verb is that all *karaka*s attach to the nearest verb on the right. *k1* has a special default rule for shared *karta* relationship between two or more verbs where there is one finite verb and the rest of the verbs are non-finite. In this case it attaches to the finite verb.

4. PART- 1B

The issues related to actual annotation task such as how to mark various relations, how to handle shared arguments, what to do in case of missing arguments are described in this part of the document. All the relations and the labels to be used for them are also listed here. As mentioned above, the framework provides two kinds of dependency relations - kaaraka relations and other relations. Detailed description for each of the labels and the syntactic cues for marking them are also provided.

NOTE : Gloss has been provided for the examples given in this document. But often the gloss provides only the relevant lexical information and not all the information which might be there in a Hindi word. For example, most often the gender and number information is missing.

4.1 The Dependency Relations and How to mark them

We will now describe all the dependency relations and the tag labels for each of them one by one . A detailed description of every relation and its tag is provided below. The objective of this section is to help the annotators with the actual annotation of various relations in a sentence. All the *karaka* relations which have labels starting with 'k-' are listed first followed by *non-karaka* relation labels which begin with 'r-.'

4.1.1 karaka Relations

DRel-1. k1 (karta 'doer/agent/subject')

In a sentence, *kartaa* is the one who carries out the action denoted by a verb. Different cases of a *kartaa* in a sentence are listed below:

The grammar talks of two types of 'kartaa', (a) primary and (b) secondary. Primary 'kartaa' has volitionality whereas the secondary 'kartaa' does not. Therefore, 'kartaa' in Ex3.b and Ex3.c given under section 3.1 above do not have volitionality.

In A.B.C. and D. below various conditions under which a 'kartaa' occurs in Hindi are explained with the help of some examples.

A. If the verb denotes an action, then the k1 is the doer of the action. In examples (Relation-DS-k1-1 to 2 and 3 to 7), 'rAma' is the doer of the action, thus 'rAma' is the *kartaa*.

Relation-DS-k1-1 : ***rAma*** *bETA hE*
Ram sit-perf is
'Ram is sitting'

Syntactic Cues : Most general or **default syntactic cues** for identifying *karta* in a Hindi sentence are:

(a) *Karta* is normally in nominative case which is realized as 0 in Hindi.
(b) By default verb in active voice (list of TAMs attached) agrees with the *karta* in number, gender and person.

IMPORTANT NOTE on syntactic cues: It is important to note that *karta* is not the only *karaka* which may appear with a 0 *vibhakti.* Some other relations may also appear without an explicit case marker. The conditions under which various *karakas* etc occur with a particular *'vibhakti'* may not always be syntactic. Therefore, one may have to use various cues such as the context, the semantic properties of the word under consideration, semantic properties of the words to which the given word is related etc. In short, the cues provided here are only to help take a decision but are not to be followed fully mechanically.

Some more examples of *karta* with the above syntactic cues are :

Relation-DS-k1-2 : ***rAma*** *KIra KAwA hE*
Ram rice-pudding eat-hab-sg-m is
'Ram eats rice-pudding'

Relation-DS-k1-3 : ***sIwA*** *KIra KAwI hE*
Sita rice-pudding eat-hab-sg-f is
'Sita eats rice-pudding'

B. However, *karta* in Hindi can also occur with case markers other than nominative case (0 vibhakti).

NOTE : The terms case marker, *vibhakti* or postposition are used interchangeably in this document.

Relation-DS-k1-4 : ***rAma ne*** *KIra KAyI*
Ram erg rice-pudding ate
'Ram ate rice-pudding.'

Relation-DS-k1-5 : ***rAma ko*** *Kira KAnI padZI*
Ram dative rice-pudding eat+inf+fem had+fem
'Ram had to eat the rice-pudding'

Relation-DS-k1-6 : ***rAma ko*** *KIra KAnA cAhiye*
Ram Dat rice-pudding eat+inf should
'Ram should eat the rice-pudding'

Syntactic cues for identifying a *'karta'* in the above constructions are : If a noun occurs with the postpositions belonging to the list given below and the verb has the corresponding TAM in the list below then the noun would always be a *karta* in Hindi.

Postposition (Vibhakti)	TAM	
(i) *ne*	*yA*	(past)
(ii) *ko*	*nA_padZA*	(compulsive, past)
(iii) *ko*	*nA_cAhiye*	(prescriptive)

C. In **passive constructions,** normally a karta would be absent. However, if it occurs , it will appear either with 'xvArA' or 'se' as its vibhakti.

Relation-DS-k1-7 : ***rAma xvArA*** *KIra* *KAyI gayI*
ram by rice-pudding ate Passv
'Rice-pudding was eaten by Ram.'

Syntactic cues: (a) A noun followed by the postposition *'xvArA' or 'se'* and (b) the verb having a passive TAM (tense, aspect and modality) would be a 'karta'. A list of passive TAMs in Hindi is provided in Appendix for reference.

D. Karta with a genitive marker : Karta in Hindi can also occur with a genitive marker. Following are some examples of the same.

Relation-DS-k1-8 : ***rAma kA*** *mAnanA hE ki kala bAriSa hogI*
Ram of belief is that tomorrow rain will-happen
'Ram believes that it will rain tomorrow.'

The *karta* with a genitive postposition (kA) occurs only with a few verbs such as *'kaha', 'soca', 'mAna'* etc. The verb in these cases would have the TAM '-nA' (gerundive)

E. Some more examples of '*karta'* in Hindi sentences

Relation-DS-k1-9 ***:*** ***rAma*** *acCA hE*
ram good is
'Ram is good.'

Relation-DS-k1-10 : *muJako* ***cAzxa*** *xiKA*
I-Dat moon appeared
'I saw the moon.'

In the stative verbs, the state of a person or a thing is mentioned. The person or thing whose state is mentioned will be the *karta.* In example (Relation-DS-k1-8), state of *rAma* is mentioned so *rAma* becomes the *karta*.

Similarly, the subject of an unaccusative verb would also be marked as *karta*. In example (Relation-DS-k1-9), ***cAzxa*** 'moon' is the *karta* as *'xiKanA' (to be seen)* is an unaccusative verb in Hindi. Following the definition of a *karta* as the doer of the

activity denoted by the verb, the doer of the activity of 'xeKanA' (to see) is different from the activity of 'xiKanA' (to be seen). Therefore, the element (rAma in Relation-DSfrom where this activity springs forth would be *karta*.

F. Clausal *karta* : A clause can also be *karta.* For example,

Relation-DS-k1-11 : ***rAma kA yaha mAnanA*** *sahI nahIM hE*
Ram of this belief true not is
'This belief of Ram is not true.'

In the above example the non-finite clause, **'***rAma kA yaha mAnanA'* is the *karta* of the verb *'hE'.* The k1 tag in such cases would be annotated on the verb of the clausal *karta*. Therefore , (annotatedexample is represented in SSF)

```
((      NP      <drel=r6:VGNN>
rAma    NNP
kA      PSP
))
((      NP      <drel=k2:VGNN name=VGNN>
yaha    PRP
))
((      VGNN            <drel=k1:VGF>
mAnanA          VM
))
((      JJP     <drel=k1s:VGF>
sahI    JJ
))
((      VGF     <name=VGF>
nahIM   NEG
hE      VM
))
```

Figure 3: *SSF-1*

Robust cues for identifying karta:

1. A noun chunk with '*ne*' case marker is always k1. For example,

   rAma ne *KAnA KAyA.*
   Ram ERG food ate.
   'Ram ate food.'

2. For a sentence in active voice, the verb generally agrees with the *karta*. For example,

   rAma *KIra KA rahA hE.*
   Ram rice-pudding eat cont is
   'Ram is eating rice-pudding.'

3. There is always at most one k1 for a verb. For example,
   rAma *skUla jAkara Gara A gayA.*
   Ram school gone home came went
   'Having gone to school, Ram came home.'

4. All first and second person personal pronouns in nominative case are k1. For example,
   mEM *KAnA KA rahA hUz .*
   I food eat is am
   'I am eating food.'

DRel-2. pk1, jk1, mk1 (causer, causee, mediator-causer)

Causatives in Hindi are realized through a morphological process. An intransitive or a transitive verb changes to a causative verb when affixed by either an '–A' or a '-vA' suffix. In our scheme, both 'causer' and 'causee' are marked. In addition to the causer and causee, there can also be a mediator who is both causee and causer.

A. pk1 (*prayojaka karta* 'causer')

Relation-DS-pk1-1 : ***mAz ne*** *bacce ko KAnA* ***KilAyA***
mother erg child acc food caused to eat
'The mother fed the child.'

Relation-DS-pk1-2 : ***sIwA ne*** *AyA se bacce ko KAnA Kil****avAyA***
Sita erg came by child to food caused to eat
'Sita made the maid to feed the child.'

Relation-DS-pk1-3 : ***rAma ne*** *mohana se BiKArI ko xAna x****ilavAyA***
Ram erg Mohan by beggar acc food caused to give
'Ram made Mohan give the alms to the beggar'

Syntactic cues : Syntactically, 'pk1' will behave like *'karta'*. Therefore, all the syntactic cues which are used for *'karta'* would apply in the case of a *'prayojak karta' (pk1-causer)* as well. The difference between a *'karta'* and a '*prayojaka karta'* is to be noted from the verb form. '-vA' suffix in the verb is a clear indicator of it being a causative.

B. jk1 (*prayojya karta* 'causee')

The causee in a causative construction is annotated as **jk1**. All the tags capture the information of agentive participation in various nouns.

Relation-DS-jk1-1 : *mAz ne AyA se* ***bacce ko*** *KAnA KilavAyA*
mother erg ayah by child acc food caused to feed
'Mother made the ayah to feed the child'

Relation-DS-jk1-2 : *rAma ne* ***mohana xvArA/se*** *tikata KArixavAye*
Ram erg Mohan by ticket caused to buy
'Ram made Mohan to buy tickets for Raja.'

Relation-DS-jk1-3 : *rAma ne* ***mohna xvArA/se*** *rAjA ko tikata xilavAye*
Ram erg Mohan by Raja Dat ticket caused to give
'Ram made Mohan to buy tickets for Raja.'

Syntactic cues : Syntactically, a causee would have either a 'ko' vibhakti or a 'se' vibhakti. The choice of 'ko' or 'se' would depend on the type of verb. Therefore, there is no definite syntactic cue. In this case also, it is the verb form and its semantics which are the determining factors for identifying this relation.

C. mk1 (*madhyastha karta* 'mediator causer')

Causative constructions have at least one causer and one causee. However, more than one causers can also occur in a sentence. The second causer (a mediator) in such cases is a causee-causer. The mediator (causee-causer) is marked as **mk1.** It is possible that more than one causee-causers can occur in a sentence. In case there are more than one mediators in a causative construction they are all marked as **mk1**. See the examples below :

Relation-DS-mk1-1 : *mAz ne* ***AyA se*** *bacce ko KAnA KilavAyA*
mother erg Ayah by child acc food made to eat
'The mother made the Ayah to make the child eat the meal'

Relation-DS-mk1-2 : *rAma ne* ***SyAma xvArA*** *mohana se BiKArI ko xAna xilavAyA*
Ram erg Shyam by Mohan by beggar Dat food caused to give
'**Ram** made Shyam to make Mohan give the alms to the beggar'

Relation-DS-mk1-3 : *sIwA ne* ***mIrA xvArA AyA se*** *bacce ko KAnA KilavAyA*
Sita erg mira by maid by child acc food caused to feed
'Sita made Mira to make the maid feed the child.

Syntactic cues : The vibhakti for a 'mk1' would either be *xvArA* or *se*. In case more than one mk1 occurs in a sentence, then the first one would have 'xvArA' vibhakti and the second one would have 'se' vibhakti.

However, the causer – causee relation is derived more from the verb morphology rather than other clear syntactic cues.

Robust cues:

1. Causatives can be identified by the presence of the TAMs –A or –vA. For example,
mAz ne *bacce ko KAnA* ***KilAyA***
mother erg child acc food caused to eat
'The mother fed the child.'

Possible cases of confusion:

1. Sometimes transitive verbs also end with -A TAM. Also, sentences with passive voice construction be confused as causatives.

DRel-3. k1s (*vidheya karta - karta samanadhikarana* 'noun complement of karta')

Noun complements of *karta* are marked as 'k1s'. The term *samanadhikarana* indicates 'having the same locus'. Therefore, *karta samanadhikarana* indicates having the same locus as *karta*.

Relation-DS-k1s-1 : *rAma* ***buxXimAna*** *hE*
Ram intelligent is
'Ram is intelligent.'

Relation-DS-k1s-2 : *xaniyA iwanI* ***vyavahArakuSala*** *na WI*
Dhaniya so-much diplomatic not was
'Dhaniya was not that diplomatic.'

Robust cues:

1. **k1s** can only be there when a **k1** is marked for a verb.

DRel-4. k2 (*karma* 'object/patient')

The element which is the object/patient of the verb is marked as *karma*. *Karma* is the locus of the result implied by the verb root.

A. karma in active voice sentences:

Given below are some examples of the occurrence of *karma* in active voice sentences :

Relation-DS-k2-1 : *rAma rojZa* ***eka seba*** *KAwA hE*
Ram everyday one apple eat-hab pres
'Ram eats an apple everyday'

Relation-DS-k2-2 : *rAma ne* ***KIra*** *KAyI*
ram erg rice-pudding ate
'Ram ate rice-pudding.'

Relation-DS-k2-3 : *rAma ne bAjZAra meM* ***ravi ko*** *xeKA*
Ram erg market in Ravi acc saw
'Ram saw Ravi in the market'

Syntactic cues : *Karma* occurs either with a zero vibhakti (postposition) or a 'ko' vibhakti (postposition). Often, in Hindi, both *karta* and *karma* occur without a postposition/vibhakti (zero vibhakti). In case both *karta* and *karma* occur with a zero

vibhakti in a sentence and the two nouns are of different gender then the noun which **does not** agree with the verb would be *karma* (see example Relation-DS-k2-1 above).

If the *karta* is followed by a postposition in a sentence, then the noun which agrees with the verb would be *karma* (Relation-DS-k2-2). *Karma* can also occur with a 'ko' postposition. *Karma* would be marked by a 'ko' vibhakti when it is a human noun (Relation-DS-k2-3) . Sometimes, *karma* is marked by a 'ko' vibhakti to indicate definiteness as well.

B. In **passive constructions**, the noun which agrees with the verb is the *karma.*

Relation-DS-k2-4 : *xivAlI para KUba **miTAI** KAyI gayI*
Diwali on lots of sweets eat-Passv
'Lots of sweets were eaten on Diwali'

Relation-DS-k2-5 : x*ivAlI para KUba **pataKe** CodZe gaye*
Diwali on lots of crackers leave go-Passv
'Lots of crackers were burst on Diwali'

Syntactic cues : If the verb in a sentence occurs with a passive TAM then the noun which agrees with the verb is the *karma*

C. Vakya-karma (Sentential object – 'complement clauses')

Finite clauses occur as sentential object the verb of the subordinate clause is attached to the verb of the main clause and the arc is tagged as 'k2'. For example,

Relation-DS-k2-6 : *rAma ne bawAyA **ki bAhara pAnI barasa rahA hE***
Ram erg told that outside water raining prog pres
'Ram told that it was raining outside'

Relation-DS-k2-7 : *usane kahA **ki rAma kala nahIM AyegA***
he-erg told that ram tomorrow not will-come
'He told that Ram will not come tomorrow.'

DRel-5. k2p (Goal, Destination)

The destination or goal is also taken as a *karma* in this framework. However, it is marked as *k2p* in the treebank. *k2p* is a subtype of *karma* (k2). The goal or destination where the action of motion ends is a k2p. These are mostly the objects of motion verbs. They also occur with other types of verbs. The syntactic behavior of k2p is slightly different from other k2. That is why a separate tag has been kept for them. Unlike other *karma*, the goal/destination *karma* do not agree with the verb under similar syntactic context (see example Relation-DS-k2p-2 below).

Relation-DS-k2p-1 : *rAma* ***Gara*** *gayA*
Ram home went
'Ram went home.'

Relation-DS-k2p-2 : *vaha saba ko* ***apane Gara*** *bulAwA hE*
he all acc his home invite be-Pres
'He invites everybody to his home.'

Relation-DS-k2p-3 : rAma ko xillI jAnA padZA
Ram acc Delhi go lie
'He had to go to Delhi'

Relation-DS-k2p-3 b : * *rAma ko* ***xillI*** *jA****nI*** *pa****dZI***

'xillI' is a feminine noun in Hindi. However, an agreement between 'xillI' and the verb 'jAnA padZA' in example Relation-DS-k2p-3 b above is ungrammatical. This is why, though a destination is also a *karma*, it is treated as a special case.

In general, verbs such as *jAnA* (to go), *AnA* (to come), *pahucanA* (to reach), etc. will take k2p.

DRel-6. k2g (secondary *karma*)

It is possible to have more than one 'karma' of the same verb in a sentence. For example:

Relation-DS-k2g-1 : *ve loga gAMXIjI ko* ***bApU*** *BI kahawe hEM*
those people Gandhi+hon acc Bapu also say+hab be-Pres
'They also call Gandhiji as Bapu.'

Verbs such as *kahanA* (to say/to call) can have two *karma.* In sentence Relation-DS-k2g-1 above, 'kahate hEM' (say/call) has two *karma*s - *gAMXIji* and *bApU*.

DRel-7. k2s (*karma samanadhikarana* 'object complement')

The object complement is called as *karma samanadhikarana* and the tag used for it is 'k2s'.

Relation-DS-k2s-1 : ve *gAMXIjI ko* ***bApU*** *BI mAnawe hEM*
they Gandhiji acc father also believe+hab be-Pres
' They consider Gandhiji as a father .'

Relation-DS-k2s-2 : *rAma mohana ko* ***buxXimAna*** *samaJawA hE*
ram mohan acc intelligent consider-Impf be-Pres
'Ram considers Mohan to be intelligent.'

Notice that both *kahanA* 'to say' and *mAnanA* 'to believe' seem to have two

karmas, but only *kahanA* can be treated as taking two ‘karma’. This is because in (Relation-DS-k2g-1), *bApU* ‘bapu’ is a word or substance, whereas in (Relation-DS-k2s-1), *bApU* ‘bapu’ is a property that resides in *gAMXIjI*. That is why in Relation-DS-k2s-1 *bApU* is the object of a ditransitive verb and in Relation-DS-k2s-1 *bApU* is the complement of *gAMXIjI* and thus would be marked as k2s.

Robust cues:

1. k2s can only be there if there is a k2 in a sentence.

Possible case of confusion:

There may be some inconsistency in marking the additional argument in the form of either 'rs' or 'k2s' in the case of perception and communication verbs like, *xeKa* (to see), *soca* (to think) , *suna* (to hear/listen), *pUCa* (to ask), *bola* (to speak), etc. The additional argument should consistently be marked as 'k2s' and be directly attached to the main verb.

DRel-8. k3 (*karana* 'instrument')

karana karaka denotes the instrument of an action expressed by a verb root. The activity of *karana* helps in achieving the activity of the main action. The *karana karaka* is annotated as **k3**. Some examples of sentences having *karana karaka* are given below.

Relation-DS-k3-1 : *rAma ne **cAkU** **se** seba kAtA*
Ram erg knife inst apple cut
'Ram cut the apple with a knife.'

The element ‘with a knife' in the above sentence is *karana* as with the help of the knife, the result, i.e. the ‘pieces of the apple’, is achieved. Some more examples of sentences having *karana karaka* are given below.

Relation-DS-k3-2 : *rAma ne **cammaca se** KIra KAyI*
Ram erg spoon with rice-pudding ate
‘Ram ate the rice-pudding with a spoon.’

Relation-DS-k3-3 : *sIwA ne **pAnI se** GadZe ko BarA*
Sita erg water with clay-pot acc filled
‘Sita filled the clay-pot with water.’

Any element/noun which is instrumental in achieving the result would be marked as 'k3' for *karana.* The noun need not necessarily denote a physical object which is an instrument. For example, the noun 'pAnI' (water) in the sentence Relation-DS-k3-3, is instrumental in achieving the action of 'BaranA' (to fill). Thus, 'pAnI' (water) would be marked as 'k3' (*karana).*

Syntactic cues : *karana* karaka always takes a *se* vibhakti (postposition) in Hindi.

Possible cases of confusion:

1. Many other non-k3 karakas can also take *se* vibhakti. We saw this in the case of *karta* karaka above. *se* vibhkati can also be taken up by k4 (cf. section 4.1.9). It can also appear with rh (cf. section 4.1.23), and k5.

2. *se* is quite an ambiguous vibhakti. The following examples list out some varied cases. You will notice that one cannot solely depend on the vabhakti to decide the relations and that the semantics of the verb is an equally important factor

- koI **[kisi se]/k2** milawA hE
- koi **[kisi se]/k2** samparka banAwA hE
- koI **[kisi se]/k4** kehawA hE
- koI **[kisi se]/k4** pUcawA hE
- koI **[kisi se]/ras-k1** bAwa karwA hE
- koI **[kisi se]/ras-k1** milwA hE
- … kisi ke **[havAle se]/k3** …
- koI **[kisi se]/k5** ubawA hE
- koI **[kisi se]/mk1** kuCa karvAwA hE

DRel-9. k4 (*sampradana* 'recipient')

Sampradana karaka is the recipient/beneficiary of an action. In other words, the person/object for whom the *karma* is intended for is *sapradana*.

Relation-DS-k4-1 : *rAma ne* ***mohana ko*** *KIra* *xI*
Ram erg Mohan dat rice-pudding gave
'Ram gave rice-pudding to Mohan.'

Relation-DS-k4-2 : *rAma ne* ***mohana ko*** *kahAnI* *sunAyI*
Ram erg Mohan dat story told
'Ram narrated a story to Mohan.'

The final destination of the action *xI* 'gave' in Relation-DS-k4-1 above is *mohana* 'Mohan' which is marked with *ko*. Similarly the final destination of the action *sunAyI* 'told' in Relation-DS-k4-2 is *mohana* 'Mohan' which is again marked with *ko*.

Syntactic cue : *sampradana* karaka normally takes a *ko* vibhakti in Hindi.

B. Certain cases where sampradana does not take a 'ko' postposition

Verbs such as 'kahanA' take a 'se' vibhakti for K4.

Relation-DS-k4-3 : *rAma ne* ***hari se*** *yaha kahA*
Ram erg Hari to this said
'Ram said this to Hari.'

It appears that some communication verbs take 'se' vibhakti for k4 but not all. Therefore, k4 of verbs such as 'bawAnA', 'sunAnA' does not take a 'se' vibhakti. It takes a 'ko' vibhakti in these cases also.

Relation-DS-k4-4 : *rAma ne* ***hari ko*** *yaha bAwa bawAyI*
Ram erg Hari to this matter told
'Ram told this (matter) to Hari.'

Robust cues:
1. For verbs like bola, kaha, puCa, etc. noun with 'se' vibhakti is k4. For example,
rAma ne mohana se kuCa bolA.
Ram erg Mohan ABL something said
'Ram said something to Mohan.'

DRel-10. k4a (*anubhava karta* 'Experiencer')

Perception verbs such as *seems, appear* etc have a perceiver/experiencer participant. In the Hindi example Relation-DS-k4a-1 below, *rAma* is *k1*, *buxXimAna* is *k1s* and *muJako* 'I-Dat' is *k4a* (perceiver). Here *muJako* 'I-Dat' is a passive agent i.e. experiencer who is not making any effort but just receiving or perceiving the activity carried out by another agent is identified as *anubhava karta* and is marked as *k4a*. The term *anubhava karta* does not occur in Sanskrit grammatical literature. This has been introduced here for Hindi based on the observations of Hindi syntax. Also, since the passive participation of perceiving is that of a receipient, it has been placed under *sampradana* here. The *anubhava karta* can be equated with a dative subject.

Relation-DS-k4a-1 : ***muJako*** *rAma buxXimAna lagawA hE*
I-Dat ram intelligent seems be-Pres
'Ram seems intelligent to me.'

Syntactic cues : *anubhava karta* always takes a *ko* vibhakti. Argument of unaccusative verbs having a 'ko' vibhakti would also be marked as *anubhava karta* (Example Relation-DS-k4a-2 below).Verbs such as *laganA* 'to seem' and *xiKanA* 'to appear' take passive agents and would be marked as 'k4a'. On the other hand, verbs such as *mAnanA* 'to believe' and *xeKAnA* 'to see' take active agents and would be marked as 'k1'. See the following examples:

Relation-DS-k1-10 : ***vaha*** *mAnawA hE ki rAma buxXimAna hE*
he believe+hab be-Pres that Ram intelligent be-Pres
'He believes that Ram is intelligent.'

Relation-DS-k4a-2 : ***muJako*** *cAzxa xiKA*
I-Dat moon appeared
'I saw the moon.'

Relation-DS-k1-11 : ***mEne*** *cAzxa xeKA*
I-Erg moon saw
'I saw the moon.'

In examples (Relation-DS-k1-10 and 11), *vaha* 'he' and *mEne* 'I-erg' respectively are *k1* as they are active agents. On the other hand, in examples Relation-DS-k4a-1 and 2 , *muJako* 'I-Dat' is *k4a* as in both the examples it appears as a passive agent (experiencer). Some more examples of *anubhava karta* are:

Relation-DS-k4a-3 : ***rAma ko*** *kiwAba milI*
Ram Dat book got
'Ram found a book.'

Relation-DS-k4a-4 : ***rAma ko*** *BUka lagI*
Ram Dat hungry felt
'Ram felt hungry.'

Relation-DS-k4a-5 : ***rAma ko*** *xuKA* ***hE***
Ram Dat unhappiness is
'Ram is unhappy.'

Relation-DS-k4a-6 : ***muJe/muJako*** *acCA lagA*
I-Dat good felt
'I felt good.'

Relation-DS-k4a-7 : ***muJe/muJako*** *laddU acCe lagawe hEM*
I-Dat sweet good feel-hab be-Pres
'I like sweets.'

DRel-11. k5 *(apadana* 'source'*)*

apadana karaka indicates the source of the activity, i.e. the point of departure. A noun denoting the point of separation for a verb expressing an activity which involves movement 'away from' is *apadana*. In other words, the participant which remains stationary when the separation takes place is marked *k5*.

Relation-DS-k5-1 : *rAma ne cammaca se* ***katorI se*** *KIra KAyI*
Ram erg spoon with bowl from rice-pudding ate
'Ram ate the rice-pudding from a bowl with a spoon.'

Relation-DS-k5-2 : *cora* ***pulisa se*** *BAgawA* *hE*
thief police from run-away-hab pres
'The thief runs away from the police.'

Syntactic cues : *apadana karaka* always takes a *se* vibhakti in Hindi. However, since 'se' postposition in Hindi is functionally overloaded, it is not a very reliable cue for identifying a karaka. Therefore, one has to look for additional cues in cases where 'se' is a vibhakti. The other cue in case of *apadana karaka* would be the verb semantics. If the verb denotes some motion, then the point of departure would be marked with 'se' and that would be *apadana karaka*.

B. Emotional verbs such as *gussA honA '*to be angry', *KuSa honA* 'to be happy' also take an *apadana karaka.* The entity which triggers these emotions is annotated as k5

Relation-DS-k5-3 : *rAma* ***mohana se*** *gussA hE*
Ram Mohan from angry is
'Ram is angry with Mohan'

The example Relation-DS-k5-3 shows a case where there is no explicit point of separation from the noun 'mohana' (Mohan). However, it will still be marked as 'k5' since it expresses the source of anger. At an abstract level, the anger is triggered from Mohan. Thus, 'mohana' (Mohan) would be the point of departure for the emotion of anger triggered in 'rAma' (Ram) and will be marked as 'k5'.

C. Verbs such as *pUCanA '*to ask' also take a k5. The entity from which the information has to be elicited is marked as k5 as it functions as the source.

Relation-DS-k5-4 : *mEMne* ***usase*** *eka praSna pUCA*
I-erg him-abl one question asked
'I asked him a question.'

DRel-12. k5prk (*prakruti apadana* 'source material' in verbs denoting change of state)

Examples such as the following pose an interesting problem for appropriate *karaka* assignment.

Relation-DS-k5prk-1 : *jUwe* ***camade se*** *banawe hEM*
shoes leather from make-hab be-pres-pl
'The shoes are made of leather.'

The issue here is whether 'camade' (leather) in the above example is *karana karaka* or *apadana.* Both these *karakas* in Hindi take a *'se'* postposition. Therefore, how do we decide what role 'camade' (leather) is playing in the action of 'banate' (make). An instrument participates in an action as a mediator for accomplishing the result of the action and is not itself affected by it, i.e., it does not undergo a change. However,

‘camade’ as a participant in the action of 'banate' (make) undergoes a change and also has a relation with the finished product. Change of state verbs such as 'make' require at least two participants 'a raw material' ('leather' in this case) with the aid of which a finished product ('shoes' in this case) is made. Hence, it is a relation which involves a kind of separation – separation from the larger raw material from which a product is made. The karaka relation will then be a special case of apadaan i.e k5. This is because there is a conceptual separation point from the original raw material ‘camade’ (leather) to the finished product ‘jUte’ (shoes). The two states in this change of state action are referred to as *prakriti* 'natural' and *vikruti* 'change'. Therefore the tag for this type of *apadana* is named as 'k5prk'.

NOTE: Currently, this distinction of k5 is not being annotated in the treebank.

DRel-13. k7t (*kAlAdhikarana* 'location in time')

Adhikaran karaka is the locus of karta or karma. It is what supports, in space or time, the karta or the karma. The participant denoting the time of action is marked as 'k7t'. For example,

Relation-DS-k7t-1 : *rAma cAra baje AegA*
Ram 4’o clock come
'Ram will come at 4’o clock.'

In the example above, ‘*cAra baje*’ is k7t. *adhikarana* can be of time or space. It is not mandatory of *adhikarana* to always take a vibhakti. Therefore, even k7t may occur with or without a vibhakti. For instance, in example Relation-DS-k7t-2 and 3 there are no vibhaktis, whereas Relation-DS-k7t-4 and 5 take a *meM*.

Relation-DS-k7t-2 : ***kala*** *pAnI barasA WA*
yesterday water rained be-Past
‘It rained yesterday.’

Relation-DS-k7t-3 : *rAma* ***pahale*** *AyA*
Ram first came
‘Ram came first.'

Relation-DS-k7t-4 : *usa* ***jZamAne meM*** *mahazgAI kama WI*
that period in expensive-ness less be-Past
‘The cost of living was less those days'

Relation-DS-k7t-5 : ***bacapana meM*** *vaha bahuwa SEwAna WA*
childhood in he very naughty be-Past
‘He was very naughty in his childhood.’

Syntactic cue : As mentioned above, 'k7t' is often marked by a 'meM' vibhakti. Some time expressions (such as 'subaha' – morning, 'pahale' – before/first, 'kala' – yesterday/today, 'mahIne' – month etc) when participating in an *adhikarana* role do not take any vibhakti. However, there are some specific cases where 'k7t' has other

vibhaktis as well. For example,

Relation-DS-k7t-6 : w*uma mere Gara* ***SAma ko*** *AnA*
you my home evening acc come
'You come to my place in the evening.'

Relation-DS-k7t-7 : *rAma apanA kAma* ***samaya para*** *karawA hE*
Ram own work time on do-hab be-pres
'Ram does his work on time'

DRel-14. k7p (*deshadhikarana* 'location in space')

The participant denoting the location of *karta* or *karma* at the time of action is called as *deshadhikarana.* It will be marked as 'k7p'. Some examples of 'k7p' are given below.

Relation-DS-k7p-1 : ***mejZa para*** *kiwAba hE*
table on book *is*
'The book is on the table.'

Relation-DS-k7p-2 : ***havA meM*** *TaMdaka hE*
air in chill is
'The air is very chill.'

Relation-DS-k7p-3 : *rAma* ***vahAz*** *KadZA hE* ***jahA***z *SyAma KadZA hE*
Ram there standing is where Shyam standing is
'Ram is standing there where Shyam is standing.'

Syntactic cues : Like location of time(k7t), some locations of place carry explicit vibhaktis (case markers) and some don't. When a location of place does take an explicit vibhakti then most of the postposition would be *meM* 'in' or *para* 'on'. In example Relation-DS-k7p-3 'k7p' has no vibhakti. The tag k7p refers to a location of place which is an **actual physical place** and **not a metaphorical or abstract place**.

DRel-15. k7 (*vishayadhikarana* 'location elsewhere')

Another kind of *adhikarana* is *vishayadhikarana* which can be roughly translated as 'location in a topic'. For example

Relation-DS-k7-1 : *ve rAjanIwi para carcA kara rahe We*
they poilitics on discussion do prog be-past
'They were discussing politics.'

However, the term 'topic' can be misleading as it is not restricted to the 'topic' of discourse alone. It is in fact a location other than time and place. Some more

examples of *vishayadhikarana* are :

Relation-DS-k7-2 : *harI ne* ***svawanwrawA saMgrAma meM*** *hissA liyA*
Hari erg independence movement in part took
'Hari took part in the independence movement.'

Relation-DS-k7-3 : *unhoMne apane SiSya ko ASrama kI sevAoM se*
he-erg own student acc ashram of services from
mukwa karane meM *saMkoca nahIM kiyA.*
Free doing in hesIwAtion not did
'He didn't hesitate in freeing his student from the services of the ashram.'

Relation-DS-k7-4 : *mere mana meM gussA hE*
my mind in anger is
'I am angry'

Relation-DS-k7-5 : *merA mana amarIkA meM hE*
my mind America in is
'I am mentally in America.'

In the example (4) above *'mana'* is not a concrete physical place, therefore, it will be marked as k7. In the example (5), *'amerikA'* is an actual physical place, but this will also be NOT marked as k7p. Instead, it will be marked as k7. The reason for marking it as k7 is that though America is an actual physical place, but the entity (*mana* in this case) which is in America is not. So, for a participant to be marked as k7p there has to be an actual physical contact, i.e., the located and the location have to be concrete objects. If they are not, then the location would be marked as k7.

Syntactic cue : Like other types of *adhikarana, vishayAdhikarana* also takes 'meM' and 'para' postpositions as its case markers.

DRel-16. k7a (according to)

For noun chunks with vibhaktis, *ke_muwAbika/ke_anusAra/ke_wahawa* should be marked as k7a. For example,

Relation-DS-k7a-1: *rAma ke muwAbika sIwA Gara para nahIM hE.*
Ram gen according Sita home loc not is
'According to Ram, Sita is not at home.'

DRel-17. k*u (*sAdrishya* 'similarity/comparison')

The tag to mark similarity is 'k*u'. This can be used for annotating both similarity and comparison. The tag is marked on the 'comparand' in a comparative construction. Since the compared entity can compare with any *karaka*, the tag

includes a star. '*' in the tag label is a variable for whichever *karaka* is the comparee of the comparand. Therefore, while marking the comparand (the compared entity), the * would be replaced by the appropriate *karaka* label. For example,

Relation-DS-k*u-1 : *rAXA **mIrA jEsI** sunxara hE*
Radha Mira like beautiful is
'Radha is beautiful like Mira.'

In the above example, 'rAXA' is the karta of the verb 'hE'. 'mIrA' is the comparand (entity with which 'rAXA', the karta, is being compared) and 'rAXA' is the comparee (entity which is being compared). Therefore, 'mIrA' in the above example will be annotated as 'k1u'. Some more examples are given below
:

Relation-DS-k*u-2 : *sIwA ko mIrA **rAXA jEsI** sunxara lagI*
Sita Dat Mira rAXA like beautiful appeared
' To Sita Mira appeared as beautiful as Radha.'

Relation-DS-k*u-3 : *sIwA mIrA ko rAXA jEsI sunxara mAnatI hE*
Sita Mira acc Radha like beautiful consider pres
'Sita considers Mira as beautiful as Radha.'

Relation-DS-k*u-4 : *rAXA **mIrA kI tulanA** **meM** adhika sunxara hE*
Radha Mira of comparison in more beautiful is
'Radha is more beautiful in comparison to Mira.'

Similarly, in the example Relation-DS-k*u-2, 'mIrA' is the comparee and 'rAXA' comparand. Therefore, 'rAXA' would be marked as 'k1u'. However, in example Relation-DS-k*u-3, 'Mira', the comparee is 'k2', thus 'rAXA', the comparand will be annotated as 'k2u'.

Syntactic cue : In the comparative constructions the comparand will take either 'jEsA' or 'se' postposition.

DRel-18. r6 (shashthi 'genitive/possessive')

The genitive/possessive relation which holds between two nouns has to be marked as 'r6'. For example,

Relation-DS-r6-1 : ***sammAna kA** BAva*
respect of feeling
'Feeling of respect.'

Relation-DS-r6-2 : ***puswaka kI** kImawa*
book of price
'Price of the book.'

Relation-DS-r6-3 : ***pATaka kI*** *krayaSakwi*
reader of purchasing-power
'Purchasing power of the reader.'

Syntactic cues : This is one of an easy to identify relation. It has a relatively reliable syntactic cue. It mostly occurs with a 'kA' postposition. A reliable cue for its identification is that the postposition 'kA' agrees with the noun it modifies in number and gender. Thus, in example Relation-r6-1 above 'kA' has masculine gender and singular number which agrees with the following noun (its modified) ". In Relation-r6-2 and 3, the postposition 'kA' agrees with 'kImawa' and 'krayaSakwi', both feminine nouns in Hindi.

Possible case of confusion:

The 'kA/ke/kI' vibhakti can occur with relations other than r6. We see this in section DRel-18, DRel-19. Sometimes, this vibhakti can also be taken up by a k1 (cf. example Relation-DS-k1-8)

DRel-19. r6-k1, r6-k2 (*karta* or *karma* of a conjunct verb (complex predicate))

Indian languages have extensive use of conjunct verbs. A conjunct verb is composed of a noun or an adjective followed by a verbalizer. Some times the argument *(karta or karma)* occur in a genitive case. Whenever the argument of a conjunct verb is in genitive case it will have a dependency relation with the noun of the conjunct verb. This is because the argument in the genitive case agrees with the noun of the conjunct verb and not with the verb. The noun of the conjunct verb agrees with the verb. In the exmple Relation-DS-r6-k1-1 below, *maMxira kA* 'temple of' will be marked as *r6-k1*with *uxGAtana* 'inauguration'. *maMxira* has *r6* relation with the noun of conjunct verb and in the sentence, *maMxira* has karaka relation *k1 of the conjunct verb 'uxGAtana karanA'.* In example Relation-DS-r6-k2-1, *maMxira kA* 'temple of' will be marked as *r6-k2* with *uxGAtana* 'inauguration'. *maMxira* has *r6* relation with the *uxGAtana* 'inauguration' which is the noun of conjunct verb and in the sentence, *maMxira* has karaka relation of *k2.*

Relation-DS-r6-k1-1 : *kala* ***manxira kA*** *uxGAtana huA*
yesterday temple of inauguration happened
'Yesterday, the temple got inaugurated.'

Relation-DS-r6-k2-1 : *manwrIjI ne kala* ***manxira kA*** *uxGAtana kiyA*
minister erg yesterday temple of inauguration did
'The minister inaugurated the temple yesterday.'

Remarks:

1. A genitive noun attached to the nominal part of the complex predicate should be r6-k*.
2. Presence of r6-k* indicates that the verb is complex.

3. A genitive k1/k2 attached to a complex verb must be r6-k1/r6-k2 respectively. Also, its attachment should be with the nominal part of the complex verb. Thus, the example Relation-DS-r6-k2-1, would be annotated as follows,

*((manwrIjI ne))*__NP <drel='k1:VGF' name='NP'>
*((kala))*__NP <drel='k7t:VGF' name='NP2'>
*((manxira kA))*__NP <drel='r6-k2:NP4' name='NP3'>
*((uxGAtana))*__NP <name='NP4' drel='pof:VGF'>
*((kiyA))*__VGF <name='VGF'>

Possible case of confusion:

1. r6-k* and pof should not have the same parent.

DRel-20. r6v ('kA' relation between a noun and a verb)

There are instances where a noun with 'kA' is attached to the verb but does not have any *karaka* relation. Instead, it does indicate a sense of possessesion. For example,

Relation-DS-r6v-1 : ***rAma ke*** *eka betI* ***hE***
Ram of one daughter is
'Ram has a daughter.'

The above example has a possessive relation between the noun *rAma ke* 'Ram's' and the verb *hE* 'is'. The relation between this noun and the verb is marked as *r6v*.

Syntactic cue: In a r6v relation, the 'kA' vibhakti normally does not agree with the noun after it.

DRel-21. adv (*kriyAvisheSaNa* 'adverbs - ONLY 'manner adverbs' have to be taken here').

Adverbs of manner are marked as 'adv'. Note that the adverbs such as place, time, etc. are not marked as 'adv' under this scheme. Place adverbs are assigned 'k7p' tag and time adverbs are marked as 'k7t'.

Relation-DS-adv-1 : *vaha* ***jalxI jalxI*** *liKA rahA WA*
He fast fast write prog be-past
'He was writing fast'

Relation-DS-adv-2 : *vaha bahuwa* ***wejZa*** *bolawA* *hE*
he very fast speak-hab be-pres
'He speaks very fast'

Remarks:

1. Sometimes an adv can occur with a *se* vibhakti such as in *kaWiwa rUpa se*

DRel-22. sent-adv (Sentential Adverbs)

Some adverbial expressions have the entire sentence in their scope. For example,

Relation-DS-sent-adv-1 : ***isake alAvA,*** *BakaPA (mAovAxI) ke rAmabacana yAxava*
this-of apart, BKP (maoist) of Rambacana Yadav
ko giraPZawAra kara liyA gayA
ACC arrest do reflx-perf go-perf
'Apart from this, Rambacana Yadav of BKP (Maoist) was arrested.'

In the above example, phrase *'isake alAvA'* is a connective which is modifying the verb but has the entire clause in its scope. Such expressions would be attached to the verb of the sentence they are modifying and the attachment would be labeled as 'sent-adv'.

Remarks:

1. A list of possible lexical items that act as *sent-adv* is given in Appendix 10.3

Possible case of confusion:

1. A conjunct cannot be sent-adv of a verb.

DRel-23. rd (relation *prati* 'direction')

The participant indicating 'direction' of the activity has to be marked as 'rd'. The label 'rd' stands for 'relation direction'.

Relation-DS-rd-1 : *sIwA* ***gAzva kI ora*** *jA rahI WI*
Sita village of direction go prog be-past
'Sita was going towards her village.'

Relation-DS-rd-2 : ***pedZa ke Upara*** *pakR udZa rahA hE*
tree of above bird fly prog be-pres
'The bird is flying over the tree.'

Relation-DS-rd-3 ***: rAma ke prawi*** *mohana ko SraxXA hE*
Ram of direction Mohan dat respect be-Pres
'Mohan has respect for Shyam.'

Syntactic cues : An element having postpositions such as 'kI_ora' or 'ke_prati' is to be marked as 'rd'.

DRel-24. rh (*hetu* 'reason')

The reason or cause of an activity is to be marked as 'rh'.

Relation-DS-rh-1 : *mEne* ***mohana kI vajaha se*** *kiwAba KArIxI*
I-erg Mohan of because book bought
'I bought the book because of Mohan.'

Relation-DS-rh-2 : *mohana* ***vyavasAyika lakSya se*** *kAma karawA hE*
mohan professional goal because of work do-Impf be-Pres
'Mohan works for professional goals.'

Syntactic cues : Complex postpositions such as 'ke_karana', 'kI_vajaha_se' etc are indicators of 'rh' relation. An 'rh' relation might also occur with a 'se' postposition. However, since 'se' postposition in Hindi is highly overloaded, its presence alone can not be a deciding factor.

Robust cues:

1. Noun chunk with vibhakti *ke_kAraNa/ki_vajaha_se* should be rh.
2. Conjunct '*kyoMki*' should be rh and should have a parent and a child.

DRel-25. rt (*tadarthya* 'purpose')

The purpose of an action is called as *tadarthya* which is marked as *rt*.

Relation-DS-rt-1: *mEne* ***mohana ke liye*** *kiwAba KArIxI*
I-erg mohan for book bought
'I bought the book for Mohan.'

Relation-DS-rt-2: *mEne* ***jAne ke liye*** *tiketa KArIxA*
I-erg going for ticket bought
'I bought the ticket for going.'

Relation-DS-rt-3: *mohana* ***padZane ke liye*** *skUla jAwA hE*
Mohan studying for school go-hab be-Pres
'Mohan goes to school for studying.'

Notice that in the second and third examples above, have verbs which are purpose of the action. For example in the example Relation-rt-2 *jAne ke liye* 'for going' is the purpose of the action *KArIdI* 'bought'.

Syntactic cue : Most often '*ke_liye*' postposition in Hindi indicates a 'rh' relation.

DRel-26. ras-k* (*upapada_ sahakArakatwa* 'associative')

In sentences where two participants perform the same action but syntactically one is expressed as primary and the other as its associate, the associate participant is marked as 'ras-k*'. *k** can be any karaka of which it is an associative. In this tag 'r' stands for relation and 'as' stands for 'associative'. The associative, like comparative can be for any relation, *karaka or non-karaka*. The * stands for the label whose associative it is.

Relation-DS-ras-k1-1 : *rAma apane **pIwAji ke sAWa** bAjZAra gayA*
Ram own father of with market went
'Ram went to the market **with his father**'

Relation-DS-ras-k1-2 : ***rAma ke sAWa** mohana ne bhI **xUXa ke sAWa** kele KAye*
Ram of with Mohan erg also milk of with banana ate
'Along with Ram, Mohan also ate bananas with milk'

In the first example *rAma* is 'k1' of the action 'gaya' (went) and since *pIwAjI* 'father' is associative of *rAma* so it will be marked as 'ras-k1'. The second example (Relation-ras-2) has two instances of *associative karakas.* 'rAma' is associative of 'mohana', thus will be marked as 'ras-k1' and 'k1' respectively. Also, xUXa 'milk' is associative of *kele* 'bananas' which is *k2* so xUXa will be marked as 'ras-k2'.

Similarly, we can have associatives for other tags as well. Given below are examples for 'ras-k4', and 'ras-k7'

Relation-DS-ras-k4-1 : *praXAna manwrI ne anya pawrakAroM kI waraha hI*
Prime minister erg other reporters like-that emph
taruNa vijaya ko milane kA samaya xiyA WA
Tarun Vijay Acc. meeting of time gave was
'The Prime Minister had given Tarun time for meeting like he had given to the other reporters.'

In the above example, *anya pawrakAroM* 'other reporters' is associative of *taruNa vijaya* 'Tarun Vijay' which is *k4* so *anya pawrakAroM* 'other reporters' will be marked as 'ras-k4'.

Relation-DS-ras-k7-1 : *unhoMne rAjnIwika viSayoM ke sAWa sAWa anya viSayoM*
He-erg political topics of with with other topics
par BI kiwAbeM liKI
on emph books wrote
'He has written books on other topics including political issues.'

rAjnIwika viSayoM 'political topics' is associative of *anya viSayoM* 'other topics'

which is *k7* so *rAjnIwika viSayoM* ‘political topics’ will be marked as ‘ras-k7’.

Syntactic cues : *Postposition 'ke_sAWa', ‘ke sAWa sAWa’, and ‘kI waraha’ normally marks an associative relation.*

DRel-27. ras-neg (Negation in Associatives)

In sentences where a karaka and its associative participate in an action but the associative does not perform the action, the associative is participant is marked as 'ras-NEG'.

Relation-DS-ras-NEG-1 *: rAma* ***pIwAjI ke binA*** *gayA*
Ram father without went
'Ram went without his father'

rAma is k1 and *pIwAjI ke binA* ‘without his father' has an associative relationship with *rAma*. The relation is denoted by ras-NEG.

Syntactic cues : Postposition *ke binA* ‘without’ indicates the sense of negation of associative.

DRel-28. rs (relation *samanadhikaran* 'noun elaboration')

Elements (normally clauses) which elaborate on a noun/pronoun are annotated as 'rs'.

Relation-DS-rs-1 : bAwa *yaha* hE ***ki vo kal nahIM AyegA***
fact this is that he tomorrow not will-come
'The fact is that he will not come tomorrow'

bAwa 'fact' is 'k1' (karta) in the above example and ***yaha*** 'this' is its 'k1s' (k1 samanadhikaran). The relations k1 and k1s will be attached to the verb whereas the clause ***ki vo kal nahI AyegA*** 'that he will not come tomorrow' will have a dependency relation with ***yaha*** 'this'. The relation is denoted by 'rs' (relation samanadhikaran). The main verb will take one samanadhikaran as its argument. If there are two samanadhikarans then the second samandhikaran is related with one of *karakas* with which it is associated.

Relation-DS-rs-2 : *usane yaha kahA* ***ki rAma kala nahIM AyegA***
he-erg this told that ram tomorrow not will-come
‘He told that Ram will not come tomorrow.’

In Relation-DS-rs-2 above the complement clause is the complement of the karma pronoun *yaha* ‘this’. Therefore, it will be attached to the pronoun ‘yaha’ and would also be labeled as ‘rs’. While annotating the sentence, the conjunct ‘ki’ will be

annotated as ‘rs’ will be attached to the ‘yaha’ which is the k2 of the verb of the main clause ('kahA' in this case). The finite verb of the complement clause ('nahIM AyegA' in the above example) will be attached to the conjunct ‘ki’ (that) and would be labeled as ‘ccof’.

Remarks:
Possible case of confusion:

1. There may be some inconsistency in marking the additional argument in the form of either 'rs' or 'k2s' in the case of perception and communication verbs like, xeKa, soca, suna, pUCa, bola, etc. The additional argument should consistently be marked as 'k2s' and be directly attached to the main verb.

DRel-29. rsp (relation for duratives)

The durative expressions have two points – a point of starting and an end point. The expression as a whole may express time, place or manner etc. The tag 'rsp' shows the relation between the starting point and the end point of a durative expression. For example,

Relation-DS-rsp-1 : ***1990 se lekara 2000 waka*** *BArawa kI pragawi wejZa rahI*
1990 from taking 2000 till India of development fast was
'India was fast developing from 1990 till 2000'

The entire expression *kala se lekara Aja waka* ‘from yesterday till today’ is a time expression. There are two parts in this time expression, one is starting point(*kala*) and the other is the ending point(*Aja*). The vibhaktis *se* ‘from’ and *waka* ‘till’ give us the information of starting point and ending point in time. As the entire expression *kala se lekara Aja waka* is a time expression it will have a k7t (time relation) relation with the verb. Now internally the two parts of the time expressions are related to each other. So the relation of *kala se lekara* ‘from yesterday’ with *Aja waka* ‘till today’ will be rsp (relation source of a durative).

Syntactic cues : Duratives will have 'se lekara - - - waka' construction.

DRel-30. rad (address terms)

Terms such as S**rImAnajI, paMdiwajI** etc. are the address terms. Such terms are annotated as 'rad'.

Relation-DS-rad-1 : ***mAz,*** *muJe kala xillI jAnA hE'*
mother, I-Dat tomorrow Delhi to go be-pres
'Mother, I have to go to Delhi tomorrow'

Relation-DS-rad-2 : ***mAstara sAhaba****, kyA kala skUla KulA hE*
master hon what tomorrow school open be-Pres
'Teacher, is the school open tomorrow?'

DRel-31. nmod__relc, jjmod__relc, rbmod__relc (relative clauses, jo-vo constructions)

A relative clause construction in Hindi has a 'jo' pronoun. Typically, the modified element has a pronoun 'vaha' in it. Such relative clauses where there is a corresponding 'vaha' pronoun in the main clause are called relative-correlative *(jo-vo)* constructions. The *jo-vo* constructions in Hindi are highly productive. These occur not only as noun modifiers but also as modifiers of adjectives and manner adverbs.

Relative_clause-DS-1 : ***merI bahana [jo*** *xillI meM rahawI hE] kala A rahI hE*
my sister who Delhi in live-hab pres tomorrow come prog pres
'My sister who lives in Delhi is coming tomorrow'

The above example does not have a 'vaha' pronoun in the modified NP. Relative clauses without a 'vaha' pronoun in the modified NP normally are elaborative in nature. These are also not so frequent.

A relative clause can be either prenominal or postnominal.

(a) Prenominal: The relative clause occurs to the left of the head noun and it carries a relative pronoun 'jo' as a demonstrative along with the noun. For example,

Relative_clause-DS-2: ***[jo ladZakA*** *vahAz KadZA hE] [**vaha** merA BAI hE]*
who boy there standing pres he my brother is
'The boy who is standing there is my brother'

Relative clause in the above example is modifying 'vaha' of the main clause. However, 'vaha' itself refers to *'ladZakA'* which occurs in the subordinate relative clause along with the relative pronoun 'jo'. Thus, the relative clause has *'jo ladZakA'* as the relativizing element. The pronoun *vaha* 'he' in the main clause has ***' jo*** *ladZakA'* as its referent. The prenominal relative clauses in Hindi moslty have this structure.

(b) Postnominal: The relative clause occurs to the right of the head noun and the relative pronoun in such cases behaves like a full-fledged pronoun and is not a demonstrative any more.

Relative_clause-DS-3 : ***vaha ladZaka [jo*** *vahAz KadZA hE] merA BAI hE*
that boy who there standing pres my brother is
'The boy who is standing there is my brother'

A relative clause can also occur to the right of the main verb as in the following example:

Relative_clause-DS-4 : ***vaha ladZakA*** *merA BAI* ***hE [jo*** *vahAz KadZA hE]*
that boy my brother is who there standing pres
'That boy is my brother who is standing there.'

A relative clause can modify any element in the main clause whatever its participatory role it might have. Thus a relative clause can modify a *karta* (subject/agent), *karma* (direct object), *samradana* (indirect object), *karana, adhikarana* (oblique object) etc.

(i) *karta* (subject) modification :-

Relative_clause-DS-5 : ***jo*** *ladZakA vahAz KadZA hE* ***vaha*** *merA bhAI hE*
who boy there standing pres he my brother pres
'The boy who is standing there is my brother.'

(ii) *karma* (object) modification) :-

Relative_clause-DS-6 : *rAma ne* ***vaha seba*** *KAyA* ***jo*** *KArAba ho gayA WA*
Ram erg that apple ate which rotten happen go-perf be-past
'Ram ate an apple which was rotten.'

(iii) *sampradana* (Indirect object) modification :-

Relative_clause-DS-7 : *rAma ne* ***usa ladZake ko*** *kiwAba xI* ***jo*** *vahAz KadZA WA*
Ram erg that boy acc book gave who there standing be-past
'Ram gave the book to that boy who was standing there.'

(iv) *karana* (Oblique object) modification :-

Relative_clause-DS-8 : *rAma ne* ***usa cAkU*** *se seba kAta* ***jo*** *wejza WA*
Ram erg that knife by apple cut which sharp was
'Ram cut an apple with the knife which was very sharp.'

Given below are the examples and corresponding tags for the 'jo-vo' constructions of Hindi :

a. nmod__relc (relative clause constructions modifying a noun)

Relation-DS-nmod__relc-1 : ***jo ladZakA vahAz bETA hE, vaha*** *merA BAI hE*
who boy there sat is he my brother be-pres
'The boy who is sitting there is my brother.'

Since it is an entire clause which modifies an element in the main clause, the convention which is followed in the current annotation scheme is to attach the verb of the subordinate clause to the element it modifies. The relation between 'jo' and 'vo' is

marked by showing a co-referential tag (coref). Therefore, a tree representation for the above example would be as follows:

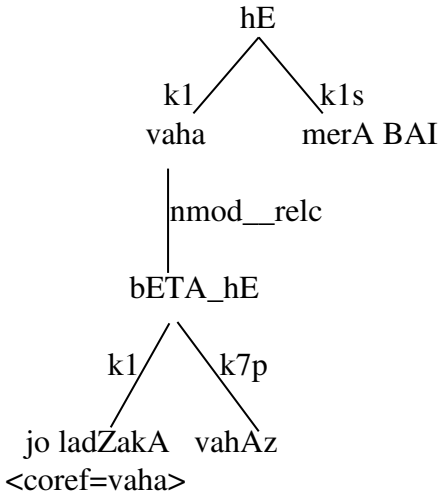

figure 3

b. rbmod__relc ('jo' construction modifying an adverb)

A relative-corelative construction can occur for an adverbial expression as well. Such 'jo' clauses would be attached under the adverb they modify with a tag 'rbmod__relc'.

Relation-DS-rbmod__relc-1 : rAma ne ***jEsA*** ***kiyA,*** *mEMne BI* ***vaisA*** *hI kiyA*
Ram erg like-what did, I-erg also like-that emph did
'I did exactly what Ram did.'

c. jjmod_relc ('jo' construction modifying an adjective)

A 'jo' clause can also modify an adjective. It will be annotated as jjmod__relc

Relation-DS-jjmod__relc-1 : *makAna* ***vEsA*** *hI suMxara banAo,* ***jEsA*** *kahA gayA hE*
house like-that part.beautiful build like-what told go-perf pres
'Build a house as beautiful as has been told'

(Here the clause containing ***jEsA*** is modifying the adjective ***vEsA*** *sunxara*)

DRel-32. nmod (participles etc modifying nouns)

nmod is an underspecified relation label employed to show general noun modification without going into a finer type. Since the dependency relations are being marked at the chunk level, simple adjective modifiers do not normally occur at this level. An adjective - noun sequence is already chunked and their dependency relations are marked only when the chunks are expanded into dependency sub-trees. A tag 'adj' is used for marking simple adjective – noun modification. This tag is not discussed in this document. The nominal modification by adjectival participles falls within the purview of this document. However, an underspecified tag 'nmod' is used to show these dependencies.

Relation-DS-nmod-1 : ***pedZa para bETI*** *cidZiyA gAnA gA rahI WI*
tree on sitting bird song sing prog be-past
'The bird sitting on the tree was singing a song.'

In the above example, the participle clause 'pedZa para bEThI' is modifying the noun 'cidZiyA'. Following a tree representation of the above sentence:

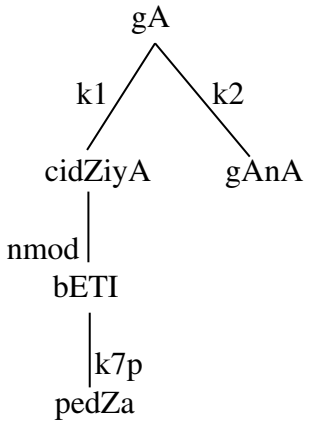

figure 4

Syntactic cues : The non-finite verb form of such participial modifiers agree in gender and number with the noun it modifies. The gender and number of the verb 'bEThI' in the above example agrees with the gender and number of the noun 'cidZiyA'.

Remarks:

Robust cues:

1. An 'nmod' should be attached to a noun chunk.

DRel-33 vmod (verb modifier)

'vmod' is another underspecified tag. For some relations getting into finer subtypes is not yet possible. Such relations are annotated with slightly underspecified tag, a tag high on the dependency tag type tree given in *figure 2* under section 3.2.3. 'vmod' is one such tag. A verb (especially non-finite) that modifies another verb is thus marked as 'vmod'. There can be two types of verb modifiers:

(a) Simultaneous : where the actions denoted by the two verbs modifier and modified happen simultaneously**.**

Relation-DS-vmod-1 : vaha ***KAwe hue*** gayA
he eat-Impf-prtpl went
'He left while eating'

(b) Sequential : where one action happens after the completion of the another action.

Relation-DS-vmod-2 : vaha ***KAnA KAkara*** gayA
he food having-eaten went
'He left after eating the meal'

Relation-DS-vmod-3 : *usako vahAM gaye hue kaI xina bIwa gaye hEM*
he-Dat there go-perf prtpl several days pass go-perf be-pres
'A number of days have passed since he went there.'

(c) '-kara' participles in Hindi: Most Indian languages have a high frequency of participials usages. So does Hindi. Of various participles in Hindi, 'kara' is one of the most frequent one. It also serves several semantic functions. One of them is showing sequentiality of events (example Relation-vmod-2 above). Other than *sequential, kara* participle has other senses also. They are:

(i) Consequential : In case of a 'kara' participle modifying another verb, the 'kara' participle expresses the causality of the other action.

Consequential_kara-DS-1 : *rAma sAzpa ko* ***xeKakara*** *dara gayA*.
Ram snake acc having seen fear go-past
'Having seen the snake Ram got frightened.'

(ii) Manner : The 'kara' participle in certain cases expresses the manner of the verb it modifies.

Manner_kara-DS-1 : *rAma* ***BAgakara*** *AyA*.
Ram running came
'Ram came running.'

(iii) Instrument : 'kara' participle also acts as an instrument of the verb it modifies.

Instrument_kara-DS-1 : *rAma mehanawa* ***karake*** *pEse kamAwA hE.*
ram hard-work having done money earn be-Pres
'Ram earns money by working hard.'

All the above constructions with *kara* and *wA huA* are vmods. Finer analysis for the above is done. However, it has been decided to mark all of the above as 'vmod' only.

Remarks:

Robust cues:

1. A vmod should be attached to a verb chunk.
2. A noun/non-finite verb chunk with vibhakti *ke_viruxXa/ke_KilAPZa* should be vmod.

DRel-34. jjmod (modifiers of the adjectives)

The tag for modifiers of the adjective is also an underspecified tag. In this case finer relations have not been worked out as yet since the need for finer relation tag for adjective modifiers is not felt for syntactic annotation. Therefore, the tag for marking adjective modifiers is 'jjmod'.

Relation-DS-jjmod-1 : ***halkI*** *nIlI kiwAba*
light blue book
'Light blue book'

(The word ***halkI*** 'light' in the above example is modifying the adjective ***nIlI*** 'blue' and not the noun ***kiwAba*** 'book')

Remarks:

1. A jjmod should be attached to an adjectival chunk.

DRel-35. pof (part of units such as conjunct verbs)

A conjunct verb is a verb that is formed by combining a noun or an adjective with a verb. Therefore, the internal structure of a conjunct verb would be [noun/adj + verbalizer]. Conjunct verbs are highly productive in Hindi. 'karana, honA' are the most commonly occurring verbalizers in Hindi. Some of the other verbalizers are 'lenA, denA'. Identifying a conjunct verb is a difficult process in Hindi as the syntactics diagnostic tests work only upto a point and not beyond. Literature on the definite syntactic behaviour of conjunct verbs does suggest a number of diagnostics though (Mohanan 1994; Butt, 2004; Chakrabarty et. al, 2007; Bhatt, 2008).

In the current scheme a special tag 'pof' has been introduced to mark the conjunct verbs. 'pof' does not exactly denote a dependency. It rather represents that the two elements related by this tag are part of a multi word expression (MWE). Therefore, the relation between the two elements of the conjunct verb **snAna + karana '**bath + do' would be shown as follows :

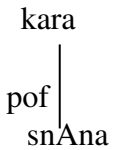

figure 5

Some examples of conjunct verb constructions are given below :

Relation-DS-pof-1 : *rAma ravi kI* ***prawIkSA kara rahA WA***.
Ram Ravi of wait do prog be-past
'Ram was waiting for Ravi'

Relation-DS-pof-2 : *rAma ne eka* ***praSna kiyA***
Ram erg one question did
'Ram asked a question'

In Relation-DS-pof-1, ***prawIkSA kara*** 'to wait' is a conjunct verb. The relationship between *prawIkSA* and the verb *kara* 'do' will be marked as *pof*. In the second example above ***praSna kiyA*** 'questioned' is a conjunct verb. But *praSna* 'question' has a modifier *eka* 'one'. The issue here is – semantically 'praSna karana' is one unit. Therefore, it is logical to group them together within a verb chunk. However, since the noun of a conjunct verb retains its nominal property and can be modified by an adjective (example Relation-DS-pof-2 above),we should be able to represent it in the dependency tree. Grouping them together within a verb chunk would fail to address the problem of an element modifying the noun element of a conjunct verb . *eka* 'one' in the above example is a modifier of *praSna*. *praSna* itself is a part of the conjunct verb *praSna kiyA*. Since *praSna kiyA* is already grouped as one chunk, it is not possible to establish relation between *eka* and *praSna*. Therefore, the noun *'praSna'* would be chunked separately from the verb *'kiyA'* (Bharati et al., 2006). However, the fact of 'praSna' and 'kiyA' being parts of a single unit, a conjunct verb, needs to be captured.

To overcome this problem it was decided that we tag the noun of the conjunct verb as NN at the POS level. Thereafter, the noun is grouped with its preceding adjectival modifiers (if any) as an NP chunk. The only problem in this approach is that the information of a noun verb sequence being a 'conjunct verb' is not captured at the chunk level and the noun of the 'conjunct verb' is separated from its verbalizer. Thus, we show the 'parts-of' relation between the noun and the verbalizer of a conjunct verb, using '*pof*' tag.

The advantage of this solution is that:

1) It allows us to show the modifier-modified relation between an adjective such as *eka* 'one' in the above example with its modified noun *praSna* 'question'.

2) Since the information of a noun verb sequence being a 'conjunct verb' is crucial at the syntactic level, it is captured at this level by marking the relation between the 'noun' and its verbalizer by an appropriate tag.

As mentioned above there are problems in identifying conjunct verbs in a sentence in Hindi. The available syntactic tests (Mohanan 1994; Chakrabarty et. al, 2007; Bhatt, 2008) are not very satisfactory. This appears to be an issue for syntax – semantic interface. There are several cases where a native speaker is quite convinced that a noun verb sequence is a case of conjunct verbs. However, syntactically the noun behaves more like an argument of the verb. In the absence of satisfactory tests for

identifying a conjunct verb, several noun verb sequences pose a major problem for the annotators on whether to treat them as conjunct verbs or otherwise.

Therefore, as of now, the decision has been left to the annotators with a full understanding that this may lead to some inconsistency in the data. The final decision of when a noun verb sequence is a conjunct verb and when not has been left to the senior linguists who would do some checks on the annotated data. Given below are a number of examples of Hindi conjunct verbs :

Conjunct_verb-DS-1 : *usane apanA Ora piSAca kA* ***vriwwAMwa varNana kiyA***
he-Erg own and devil of narration description did
'He described his own story and the story of the ghost.'

Here *varNana* 'description' and *karana* 'to do' have become one verb, and this verb has its *karma karaka 'apanA aur pishAca kA vruttAMta'* in the accusative case. Another possible construction of the same conjunct verb *'varNana karana*' is with the *karma* of the verb occurring with a genitive case. For example,

Relation-DS-pof-3 : *usane apane Ora piSAca ke* ***vriwwAMwa kA varNana kiyA***
he-Erg own and devil of narration description did
'He described his own story and the story of the ghost.'
Relation-DS-pof-4 : *bhAiyoM ne maharSi kI AjfyA* ***svIkAra kI***
brothers Erg saint of command accept did
'The brothers accepted the command of the saint.'

Relation-DS-pof-5 : *isa granWa ko* ***svIkAra kareM***
this book acc accept do-Imper-hon.
'Please accept this book.'

Relation-DS-pof-6 : *sadZaka* ***cOdZI huI***
road wide happened
'The road became wide.'

Some more conjunct verbs which have this alternation are *wyAga karanA* 'to forsake', *AramBa karanA* 'to commence', *pAlana karanA* 'to nurture'.

Another feature of Hindi conjunct verbs is that in some cases the verbalizer agrees with the noun which is a part of the conjunct verb. For example, *grihaNa karanA* 'to receive' or 'accept', *vixA karanA* 'to bid farewell' or 'to dismiss', *kSamA karanA* 'to forgive'

Verbs such as *xayA karanA* 'to display mercy', *rakRA karanA* 'to protect', *pUjA karanA* 'to worship', *sahAyawA karanA* 'to render help' are some more conjunct verbs which are not fully compounded.

B. Since 'pof' indicates a 'part of' relation between two words of a single lexeme, it is generalized to indicate relation between different elements of other MWEs as well.

Hence in the following example, 'PulA nahIM samAyA' is an idiom and 'pof' will be used to mark the relation between 'PulA' and 'nahIM samAyA'.

Relation-DS-pof-7 : *rAma KuSI se PUlA nahIM samAyA*
Ram happiness because of bloated not contained
'Ram was bursting with happiness.'

Label 'pof' has three subtypes :

(1) pof (conjunct verb)
(2) pof-idiom (idiom)
(3) pof-compound (compound noun)

Example (Relation-DS-pos-7) has an idiom *PulA nahIM samAyA'*was bursting with happiness', the parts of this idiom would be connected by the label 'pof'.

Remarks:

1. A genitive noun attached to the nominal part of the complex predicate should be r6-k*.
2. Presence of r6-k* indicates that the verb is complex.
3. A genitive k1/k2 attached to a complex verb must be r6-k1/r6-k2 respectively. Also, its attachment should be with the nominal part of the complex verb.

Possible cases of confusion:
1. r6-k* and pof should not have the same parent.

DRel-36. ccof (co-ordination and sub-ordination)

Another special tag which does not exactly reflects a dependency relation is 'ccof'. This is used for coordinating as well as subordinating conjunctions. The dependency trees will show the conjuncts as heads. In case of coordinating conjuncts, the conjunct is the head and takes the coordinating elements as its children. Likewise, a subordinating conjunct would take the clause to which it is syntactically attached (the subordinate clause) as its child.

(a) co-ordinating conjunct :

Relation-DS-ccof-1 : *rAma seba KAwA hE* ***Ora*** *sIwA xUXa pIwI hE*
Ram apple eat-hab be-pres and Sita milk drink-Imp
'Ram eats apple and Sita drinks milk.'

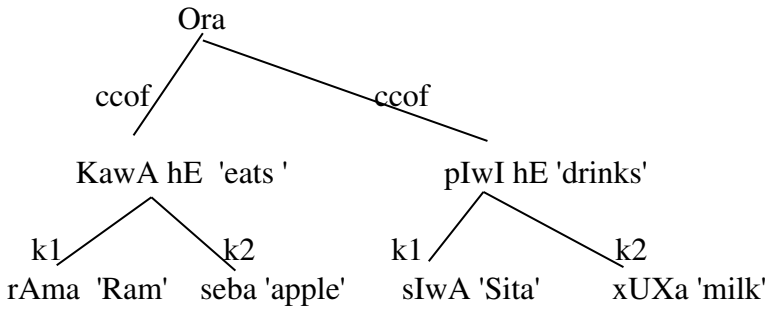

figure 6

The above example is an example of co-ordination of two clauses. However, the tag 'ccof' would be used for any co-ordination. Therefore, co-ordination of nouns, adjectives or adverbs will all be tagged with a 'ccof' tag. Following is an example of noun co-ordination :

Relation-DS-ccof-2 : *rAma **Ora** SyAma skUla jAwe hEM*
Ram and Shyam school go-hab be-pres
'Ram and Shyam go to school.'

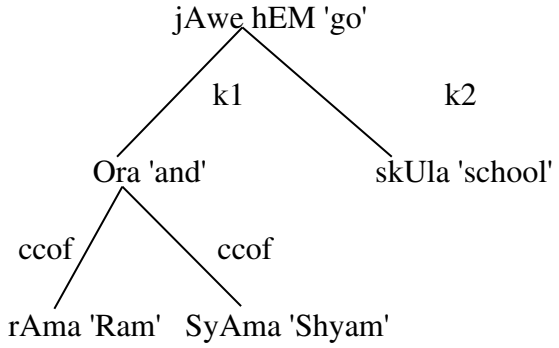

figure 7

(b) sub-ordinating conjunct :

Relation-DS-ccof-2 : *rAma ne SyAma se kahA **ki** vaha kala nahIM AyegA*
Ram erg Shyam to told that he tomorrow not will-come
'Ram told Shyam that he will not come tomorrow.'

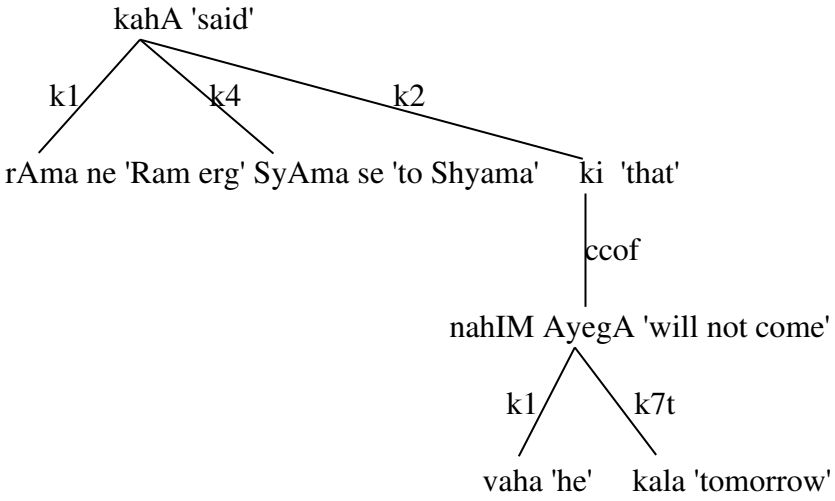

figure 8

A cordinating conjunct would have two or more branches which would be labeled as 'ccof' and a subordinating conjunct would have only one branch.

Remarks:

1. A ccof chunk should be attached to a conjunct.
2. A conjunct chunk should have children of the same type. For example,

rAma <drel=ccof:CCP name=NP> Ora <name=CCP> SyAma <drel=ccof:CCP name=NP>
'Ram and Shyam went to the market.'
Possible case of confusion:

1. A conjunct should not be sent-adv for a verb.

DRel-37 fragof (Fragment of)

'fragof' is a tag which has been included to handle some very special cases.

A. There are examples in the Hindi corpus where a postposition, a negative particle or an auxiliary are separated from the NP or VP of which normally they are a part of. Thus, they do not occur as part of the chunk where they belong. For example,

Relation-DS-fragof-1 : BakaPA (mAovAxI) ke rAmabacana yAxava ko
BKP (maoist) of Rambacana Yadav ACC
giraPZawAra kara liyA gayA
arrest do reflx-perf go-perf
'Apart from this, Rambacana Yadav of BKP (Maoist) was arrested.'

In the above example, the NP chunk *'BAkapA ke'* has been broken through the insertion of additional information *(mAovAxI)* about *'BakapA'*. The noun *'(mAovAxI)'*

itself forms a separate NP chunk. Therefore, the expression ***BAkaPA*** *(mAovAxI)* ***ke*** would appear as follows in chunks :

```
((      NP
BAkapA      NNP
))
((      NP
(       SYM
mAovAxi     NN
)       SYM
))
((      FRAGP
ke      PSP
))
```

SSF-2

The expression '*BAkapA ke'* is broken into two chunks. The postposition *'ke'* which is separated from its noun *'BAkapA'* is chunked as 'FRAGP'. To represent that the post position 'ke' is part of the noun chunk 'BakapA', the postposition chunk would be annotated with the value 'fragof' for the attribute 'drel'.

This is a tag which is an exception in the normal scheme as it marks the relation of two members of the same chunk. Also, this chunk would normally contain a function word which is a part of some other chunk. After annotating the value 'fragof' for the attribute 'drel', the FRAGP chunk would appear as follows :

```
((      FRAGP       <drel=fragof:NP>
ke      PSP
))
```

SSF-3

The occurrence of such cases could be due to some intervening material or some time the main part of the chunk is dropped.

B. There are also instances where the main part of the chunk is missing. It normally happens in cases of gapping particularly with negative particles.

Relation-DS-fragof-2 : *bihAra ke rAjyapAla ko notisa BejA jA sakawA hE ki* ***nahIM***
Bihar of governor acc notice send go can is or not
'Can the notice be sent to the Bihar Governor or not ?'

In the above example, the second occurrence of the verb 'BejA' has been ommitted. Consequently, only the negative particle 'nahIM' is left. To represent the dependencies of the second clause, it is important to insert a verb node. Since, in the current scheme, the negative particles are chunked with the verb, this intra-chunk relation would then be represented by marking the negative particle with 'fragof'.

Therefore, the verb chunk and the negation chunk would appear as follows after annotation :

```
((      NEGP <drel=fragof:NULL__VGF>
        nahIM
        ))
        ((      NULL__VGF <name=NULL__VGF>
        NULL VG
        ))
```

SSF-4

DRel-38. enm (enumerator)

The tag 'enm' is another special tag. This tag also does not represent a dependency in the strict sense. Although, this again is a value for the attribute 'drel'. of the word. This tag is used to mark the enumerators such as 1, 2, 3 or a, b, c, etc in a text. These enumerators occur in the beginning of a sentence and they need to be attached to the root node. In the treebank, the root node normally, is either a verb or a conjunct. Therefore, it has been decided to attach the enumerators to the verb with a label 'enm'. For example,

Relation-DS-enm-1 : ***1.*** *Apa apanA kara samaya se xe sakawe hEM*
1. you your tax time on give can be-pres
'1.You can pay your taxes on time.'

In the above example, numeral '1.' has occurred as an enumerator. This will be chunked separately with a chunkd label 'BLK'. At the dependency level, this chunk will be attached to the verb '*xe sakawe hEM'*. Therefore, the annotated example would be :

```
((      BLK     <drel=enm:VGF>
1       QC
.       SYM
))
```

SSF-5

DRel-39. rsym (tag for a symbol)

'rsym' is a label that marks the arc attaching a sentence end marker (Hindi ' ।) to the verb.

Relation-DS-rsym-1 : *rAma Gara gayA ।*
Ram home went
'Ram went home.'

Here the relation *rsym* exists between *gayA* 'went' and the fullstiop of Hindi ' ।.

DRel-40. psp__cl

'psp__cl' is the relation marked between a clause and the postposition following that clause..

Relation-DS-rsym-1 : "*xillI CodZo" ne halcala macA xiyA*
Delhi leave erg chaos break-out did
'"Quit Delhi" caused chaos.'

Here the relation *psp__cl* is marked between *ne* postposition and the verb of the clause preceding it, i.e., *CodZo* 'leave' and the whole clause ' "*xillI CodZo" ne'* will be marked as *karta* of *'macA xiyA'*.

4.2. How to Mark Elided Elements ?

An issue that came up before us while working on the scheme was whether to mark elided elements in a sentence or not. After due deliberations, it was decided to mark a missing element in the tree for the following cases :

(a) In case of a missing verb since a verb forms the root node of a tree/subtree (see section on Gapping (4.2.1) for more details)
(b) In case of a missing co-ordinating conjunct since it also forms the root of a co-ordinating tree under the current scheme.
(c) In case of any other node which may be a root node for a tree or a sub-tree. For example, ' ulleKanIya hE ki ....,
(d) In case of missing arguments of a verb. Amongst the missing arguments, it was decided to mark only k1 and k2. However, The missing arguments will be inserted only in the following cases:

(i) Shared arguments
(ii) Gapping
(iii) Also in finite subordinate clauses

For making the above missing elements explicit it was decided to introduce a NULL node in the tree. The node would be chunked and the relevant features would be annotated at the chunk level depending on the type of the node inserted. The details of the features to be annotated for various types have been provided under the cases discussed below.

In the following sub-sections each of the above, except 'shared arguments', is discussed in more details. The shared arguments have been discussed in more details under Section 4.3. below.

Remarks:

1. A NULL chunk should not have a ‘drel’ attribute. Instead, it should have a ‘dmrel’ attribute.

4.2.1 Gapping

Gapping is a type of ellipses where a verb is omitted in its repeat occurrences. Some times the arguments of the verb may also be omitted along with the verb. Ross (1967) introduced the term. An example of gapping in Hindi is given below :

Gapping-DS-1 : *rAma xillI gayA Ora SyAma AgarA*
Ram Delhi went and Shyama Agra
'Ram went to Delhi and Shyama to Agra.'

In the above example the occurrence of the verb 'gayA' (went) in the second clause of the co-ordinating construction has been elided. To complete the dependencies of the second clause, it is essential to explicitly show the verb which would be the root node of the tree. The missing verb can be retrieved from the previous clause. Thus, the gapped element would be marked as follows :

(i) First a new node would be created :
NULL VM

No other information about this node would be provided.

(ii) Next, the above node would be chunked. The chunk would be annotated for the following features :

<name=" troot=" mtype=">

Of the three attributes given above, 'name' is an attribute which is annotated on all chunk nodes. The attribute 'troot' is to be added for a gapped verb as it is retrievable from the context. The attribute 'mtype' is to mark every missing element for whether it is a case of 'gap' or 'not'. Therefore, this attribute would have only two values (1) gap and (2) non-gap.

In case the gapped verb is also a dependent of a higher node, an additional attribute of 'dmrel' would be annotated as well. The attribute 'dmrel' is same as 'drel'. The attribute 'drel' is for the words in a sentence and the attribute 'dmrel' would be on elements which are not present in the sentence explicitly. Thus, the chunk annotated for the gapped element in the above example would look as follows:

((NULL__VGF <name='NULL__CCP' troot='jA' mtype='gap'>
NULL VM
))

The example below is another case of gapping.

Gapping-DS-2 : *rAma ne sIwA ko kiwAba xI Ora AwiPZa ne tInA ko*
Ram Erg Sita acc. book gave and Atifa Erg Tina acc.
'Ram gave a book to Sita and Atif to Tina.

However, in the above example, an argument is also dropped in the second clause. This argument and the verb can be retrieved from the previous clause. To build a complete dependency tree for the above example, the following items will be inserted in the the tree, (a) the missing verb and (b) the missing argument. We are, however, are not inserting missing arguments unless they are required as a root node for a sub tree.

The following chunks for (a) and (b) will be created respectively :

```
((      NULL_VGF  <troot='xe'  name='NULL__VGF' mtype='gap'>
NULL VM
))
((  NULL__NP      <dmrel=k2:xe reftype=cotype:kwAba name='NULL__CCP
mtype='gap'>
NULL NN
))
```

Ssf-7

4.2.2 Missing co-ordinating conjunct

Some times the co-ordinating conjunct is implicit and does not occur in the sentence explicitly. For example,

Elided-conjunct-DS-1 : *bacce badZe Ho gaye hEM kisI kI bAwa nahIM mAnawe*
children big happen go-perf be-pres no-one's of talk not listen to
'The children have grown big and do not listen to anyone.'

In the above example, the co-ordinator 'Ora' is missing. Since co-ordinating conjunct forms the root node, a NULL node will be inserted to represent it. Thus, the example after the insertion of NULL would appear as:

Elided-conjunct-DS-1: *bacce badZe Ho gaye hEM* ***NULL*** *kisI kI bAwa nahIM mAnawe*

The feature structure for the NULL node would be :

```
((      NULL__CCP <name=NULL__CCP>
        NULL CC
        ))
```

SSF-8

4.2.3 Missing root node

A commonly occurring construction in Hindi is :

Missing-yaha-DS-1: ***ulleKanIya hE ki*** *unhoMne yaha bAwa mAna lI*
noteworthy is that they this suggestion accept reflx-past
'It is noteworthy that they accepted this proposal.'

In the above example, the sentence begins with an adjective and has a complement clause in the predicative position. The highlighted words show the adjective, verb be and the complement 'ki'. The complement clause in such sentences is actually an NP complement of the subject, which is missing. To represent this a NULL node is to be inserted and the clause is can then be attached to it as its modifier. The inserted NULL node in this case would look like :

((NULL__NP <name=NULL__NP troot=yaha mtype=non-gap>
NULL NN
))

SSF-9

4.2.4 Missing arguments in a co-ordinating construction :

The example Gapping-DS-2 above shows a case of an elided argument along with the gapped verb. In case of gapping, the verb is same in both the clauses and consequently its repeat occurrence is omitted. It is also possible that the two clauses in a co-ordinate structure may have two different verbs. In such a situation both the verbs are realized explicitly. However, the repeated arguments in a co-ordinated construction are dropped even if the verb is different and is realized on surface. For example,

Elided-arg-DS-1 : ***mohana ne kiwAba padZi Ora so gayA***
Mohan Erg book read and sleep go-Past
'Mohan read the book and slept.'

In the above case both the verbs *'padZI'* (read) and *'so gayA'* (slept) have Mohan as their *karta* (k1). However, the second occurrence of Mohan is omitted. In such cases also, the missing argument would be inserted and would be represented as follows:

((NULL__NP <name=NULL__NP mtype='gap' dmrel='k1:VGF2' reftype=corefn:mohana>
NULL NN
))

SSF-10

However, as mentioned above, such missing arguments are not posited at the dependency level of annotation.

4.3 How to mark shared arguments ?

Since Hindi allows omitting of mandatory arguments, there are a number of sentences with missing arguments. Missing arguments in a sentences could be due to being shared between two or more verbs or due to ellipsis. The difference between sharing and omitting is that in sharing the argument occurs once which is shared by two verbs ie. main verb which would be finite and the participle clause which would have a non-finite verb. In sharing the second argument can not be realized syntactically. The other case of missing argument is when the argument can (in principle) occur twice but it has been dropped in the second clause (as in case of gapping).

Since k1 and k2 are otherwise mandatory arguments for several verbs and these two arguments also play a crucial role in several linguistic decisions, it was decided to make them explicit in case they were missing in a sentence. For making the missing k1 and k2 explicit the following procedure has to be followed.

a) Insert a NULL node in the tree for a missing argument.

b) Assign it appropriate POS tag, normally a NN.

c) Chunk the NULL node and assign it appropriate chunk label. However, it has to be prefixed with NULL__ . As shown above (in 4.1), the label for missing verb chunk would be 'NULL__VGF'. For a missing nominal argument, it would be 'NULL__NP'.

d) As mentioned earlier, a new dependency attribute is introduced in the scheme to mark the dependency relations of the inserted nodes. The attribute is 'dmrel'. 'dmrel' stands for 'dependency relation for a missing element'.

e) Missing argument could either be co-referential with another element in the tree or could be of the same type but not exactly co-referential. Thus, to mark this distinction an attribute 'reftype' has been introduced. The values for the 'reftype' would be 'corefn:X' or 'cotype:X'. The value has three parts to it. The first part (corefn, cotype) indicates the 'type' of reference, the second part (:) indicates 'of' and the third part 'X' stands for 'what'. Please see example under section on shared argument for more clarity.

Therefore, the following information is annotated in an inserted node for a missing argument :

```
((      NULL__NP   <name='NULL__NP' dmrel='' reftype='' mtype=''>
      NULL NN
))
```

NOTE : The attribute 'troot' is not annotated for a missing argument as it is captured by the 'reftype'. In principle, the morph features (root, number, gender, person) of the corresponding element in the sentence can be copied to the inserted node and need not be manually annotated.

Coming back to the sharing of arguments, the sharing of arguments can be of two types :

4.3.1 Sharing in non-adjectival participles:

In non-adjectival partiples, an argument of a verb(main) is shared with another verb(participle). The argument occurs only once in the sentence but is semantically related to both the verbs. The shared argument syntactically always attaches with the main verb. For the other verb this argument is semantically realized but not syntactically. Arguments of *-kara* constructions and *ke_bAxa* constructions in Hindi would fall under this type. Note the following sentence :

Non-adjectival-Shared-arg-DS-1 : *rAma ne KAnA KAkara pAnI piyA*
Ram Erg food having eaten water drank
'Ram drank water after eating the food.'

It may be noted that linguistically *rAma ne* is explicit *karta* of only *piyA* 'drank' and not of *KAkara* 'having eaten', even though, semantically it is the agent for both *KAkara* and *piyA*. Since agreement and its vibhakti are controlled by the main verb *'piyA'* (drank) it will be attached to it. However, its semantic presence of being an argument of *'Kakara'* will be annotated by following the steps given above. After the annotation the inserted node would look as follows :

((NULL__NP <name='NULL__NP' dmrel='k1:'VGNF' reftype='corefn:NP' mtype='non-gap'>
NULL NN
))

SSF-13

'VGNF' and 'NP' in the values of attributes dmrel and reftype respectively are the names of the chunks to which this chunk would attach (VGNF) and would refer to (NP). Some more examples of this type of sharing are given below :

Non-adjectival-Shared-arg-DS-2 : *rAma KAnA KAne ke bAxa pAnI pIwA hE*
Ram food eating after water drinks be-Prs.Sg
'Ram drinks water after eating food.'

Noun 'Ram' in the above example is shared by '*KAne*' (eating) and '*piwA_hE'* (drinks) The inserted chunk for 'rAma' in the above example would be :

((NULL__NP <name='NULL__NP' dmrel='k1:'VGNN' reftype='corefn:NP' mtype='non-gap'>
NULL NN
))

SSF-14

Non-adjectival-Shared-arg-DS-3 : *rAma xillI jAnA cAhawA hE*
Ram delhi to-go want-hab be-Pres
'Ram wants to go to Delhi to Delhi.

4.3.2 Sharing in adjectival participles (*wA_huA* constructions, *KAye_gaye* constructions)

In another kind of sharing of arguments, a participle clause modifies the noun. and the modified noun, apart from being an argument of a higher verb, is also an argument of the verb in the participle clause. Therefore, the noun is shared by the main verb and its modifier verb. The adjectival participle, obviously, does not have the modified noun as its explicit argument. Again, although the argument in this case also is semantically realized but cannot occur syntactically. For example,

Adjectival-Shared-arg-DS-1 : *bEnca para bETA huA ladZakA seba KA rahA hE*
bench on sit-perf be-ptpl boy apple eat prog pres
'The boy sitting on the bench is eating an apple.'

Adjectival-Shared-arg-DS-2 : *mere xvArA Kaye gaye Pala acCe We*
My-obl by eat-perf go-Perf fruits good past
'The fruits eaten by me were good.'

In example (Adjectival-Shared-arg-DS-1) above, *bETA huA* 'sit-perf be-ptpl' is modifying the noun *ladZakA* 'boy'. Noun *ladzakA* 'boy' is an argument of the higher verb *KA rahA he* 'eat prog pres'. *ladZakA* 'boy' is also an argument of the non-finite verb *bETA huA* 'sit-perf be-ptpl'. Similarly, in example (Adjectival-Shared-arg-DS-2) the noun *Pala* 'fruits' is an argument of both, the finite verb *We* 'were' and the non-finite verb *Kaye* 'eaten'.

As in the case of shared arguments of the non-adjectival participles, the arguments of this type will also be annotated. However, for such shared arguments, a new node will not be created. Instead, it will be captured by the label on the arc between the modifying clause and the modified noun. For example, the *karaka* relation of *ladZakA* 'boy' with *KAwA huA* 'eat.Impf.Ptpl' (in Adjectival-Shared-arg-DS-1) is *k1 (karta* karaka relation), it will be represented as *nmod__k1inv.* Similarly, in example (Adjectival-Shared-arg-DS-2), *KAye gaye* 'ate go-Prf.' is the participle which modifies the noun *Pala* 'fruit', the noun *Pala* 'fruit' is *k2* (*karma* karaka relation) of the verb *Kaye hue* 'eaten'. The relation between *Pala* 'fruits' and *KAye hue* 'eaten' will be represented as *nmod__k2inv.*

Therefore, we have one more tag *'nmod__k*inv,* which means *nmod* of the type *k*inv,* where *k** stands for the type of *karaka* relation i.e. *k1 or k2* etc. and *inv* stands for inverse. Along with the *karaka* relation we also specify *inv* which denotes that, here the relation arc is going from child to the parent instead of parent to the child. In this type of sharing a new node is not created, the label *nmod__k*inv* is sufficient.

Adjectival-Shared-arg-DS-3 : *dAliyoM para Kile Pula mahaka rahe We*
branches on blossomed flowers smell prog past
'The flowers flowering on the branches were spreading a scent'

In the above example, *Pula* 'flowers' is the shared argument. Verb *Kile* 'blossomed' is modifying *PUla* 'flowers'. The feature structure of *Kile* 'blossomed' would be as follows :

((VGNF <name='VGNF' drel='nmod__k1inv'>
Kile VM
))

SSF-15

Since in this case, a new node is NOT inserted, none of the attributes which are annotated in an inserted node will be annotated here.

We also have *'nmod__pofinv'*. Its example is given below:

Adjectival-Shared-arg-DS-4 : *rAma ke sIwA se kiye gaye vAxe JUTe WeM*
Ram of Sita with did go-Prf promises false were
'The promises done by Ram to Sita were false'

In the example (Adjectival-Shared-arg-DS-4), *kiye gaye* 'ate go-Prf.' is the participle which modifies the noun *vAxe* 'promises', the noun *vAxe* 'promises' is *pof* of the verb *kiye gaye* 'ate go-Prf.'. The relation between *vAxe* 'promises', and *kiye gaye* 'ate go-Prf.' will be represented as *nmod__pofinv*.

5. Some Additional Features

During the discussion on what all information would be useful for various applications, it was decided to add two more features on every finite verb clause. The two features are :

5.1 stype (Sentence type)

The attribute 'stype' is to be annotated on every finite verb chunk. The values for this are : declarative, imperative, interrogative etc. A complete list of the sentence type is provided separately. For example,

Sentence-type-DS-1 : *Apa xAna rASi para Cuta kA xAvA kara leM*
you donation amount on exception of claim do imp
'You claim (tax) exception on the donated amount'

The attribute 'stype' will be marked on the verb chunk. Thus, the annotated verb chunk with the 'stype' attribute would be as follows :

```
((  VGF  <stype=imperative>
kara  VM
leM  VAUX
))
```

SSF-16

5.2 voicetype (Voice type)

The other feature to be annotated on every finite verb chunk is 'voicetype'. The values for this are only two (1) active and (2) passive. For example,

Voice-type-DS-1 : *borda kA gaTana kiyA gayA*
board of formation do-perf go-perf
'The board was formed'

The voice type feature would be annotated on the verb as follows :

```
((  VGF  <voicetype=passive>
kiyA  VM
gayA  VAUX
))
```

SSF-17

Voice-type-DS-2 : *Apa xAna rASi para Cuta kA xAvA kara leM*
you donation amount on exception of claim do imp
'You claim (tax) exception on the donated amount'

```
((  VGF  <voicetype=active>
kara  VM
leM  VAUX
))
```

SSF-18

5.3 coref (Coreference)

As mentioned in the section DRel-28, relative clauses are attached to the noun they modify with a label 'nmod__relc'. The attachment is between the main verb of the relative clause and the noun it modifies. Thus, an important information about the relative pronoun playing a crucial role in this relation is missed out. To capture this information, it has been decided to annotate the relative pronoun of the relative clause with an additional attribute of 'coref'. The value for the attribute 'coref' would be the referent noun in the main clause, i.e. the noun modified by the relative clause. An example of the same is :

Relative_clause-DS-1 : ***merI bahana*** *[* ***jo*** *xillI meM rahawI hE] kala A rahI hE*
my sister who Delhi in live-hab pres tomorrow come prog pres
'My sister who lives in Delhi is coming tomorrow'

In the above example, the relative pronoun will, in addition to other features will also be marked with the attribute coref. Thus,

((NP <name=NP>
merI
bahana
))
((NP <coref=NP
jo
))

6. PART – 2 : Hindi Example Constructions

This section of the document contains some example constructions of Hindi and their relevant dependency analyses. The constructions given here are based on criteria normally considered for identifying construction types. Broadly these are :
(a) For simple sentences, realization of a syntactic structure based on the verb type such as transitive, unergative, unaccusative etc.
(b) For complex sentences, the type of subordination a clause may have. For example, relative clause, complement clause etc
(c) Constructions which result due to certain linguistic operations such as ellipsis, sharing of arguments etc.

(Most examples in this PART are taken from PS Guidelines)

6.1 Simple Transitives

Simple transitives in Hindi have mostly both karta and karma taking nominative case (0 vibhakti).

a. Nominative

Transitive-Verbs-DS-1 : *AwiPZa kiwAba paDZegA*
Atif.M book.f read-Fut.3MSg
`Atif will read (a/the) book.'

DS analysis (only the relevant dependency features are shown) ;

AwiPZa <drel=k1:VGF> kiwAba <drel=k2:VGF> paDZegA <name=VGF>

b. Dative

Transitive-Verbs-DS-2 : *AwiPZa ko kiwAba paDZanI hE*
Atif-Dat book.f read-Inf.f be.Prs.Sg
'Atif has to read (a/the) book.'

DS analysis ;
AwiPZa ko <drel=k1:VGF> kiwAba <drel=k2:VGF> paDZanI hE <name=VGF>

The dependency analysis considers the postposition of the noun and the TAM markers of the verb to ascertain the *karaka* relations (refer Section 3.1 on Grammatical model)

c. Ergative

An ergative construction in Hindi occurs when the verb is transitive and its TAM is past perfective.

Transitive-Verbs-DS-3 : *AwiPZa ne kiwAba paDZI*
Atif-Erg book.f read-Pfv.F
'Atif read (a/the) book.'

DS analysis ;
AwiPZa ne <drel=k1:VGF> kiwAba <drel=k2:VGF> paDZI <name=VGF>

6.2 Unergatives

a. Nominative

Unergatives-DS-1 : *AwiPZa bAxa meM nahAegA*
Atif.M later bathe-Fut.3MSg
'Atif will bathe later.'

DS Analysis ;
AwiPZa <drel=k1:VGF> bAxa meM <drel=k7t:VGF>

nahAegA<name=VGF>

b. Dative

Unergatives-DS-2 : *AwiPZa ko nahAnA hE*
Atif-Dat bathe-Inf be.Prs
'Atif has to bathe.'

DS Analysis ;
AwiPZa ko <drel=k1:VGF> nahAnA hE <name=VGF>

The analysis of the dative construction within Paninian dependency framework would remain same for both transitives and unergatives as within Paninian framework what is considered as a syntactic cue for identifying the k1 of a verb is its TAM and the postpositions of the participating nouns. Therefore, the TAM *nA_hE* in active voice assigns a *'ko'* vibhakti to the *karta* of a verb (refer to Transformation rules in Appendix) irrespective of the verb type. In other words, it is purely a syntactic operation in Hindi which applies to any verb.

c. Ergative

Unergatives-DS-3 : *AwiPZa ne nahA liyA*
Atif-Erg bathe TAKE.Pfv
'Atif has bathed.'

This is a sentence which can be contested by many native speakers of Hindi as bad. This also does not go well with the rule given under ergative above. However, it is found in the speech of some Hindi speakers so included here.

6.3 Unaccusatives

a. Nominative

Unacusatives-DS-1: *xaravAjZA Kula rahA hE*
door.M open Prog.MSg be.Prs.Sg
'The door is opening.'

b. Dative

Unacusatives-DS-2: *xaravAjZe ko bAraha baje KulanA hE*
door-Dat 12 o'clock open-Inf be.Prs
'The door has to open at noon.'

DS Analysis;

xaravAjZe ko <k1:VGF> bAraha baje <k7t:VGF> KulanA hE<name=VGF>

6.4 Dative Subject Constructions

The dative subject constructions of PS analysis correspond to the k4a constructions in DS analysis. For cross reference please see section DRel-10 of PART -1B.

6.5 Ditransitives

Ditransitive-DS-1 : *AtiPZa ne kala monA ko sabake sAmane*
Atif Erg yesterday Mona Dat all-Gen.Obl.of in.front

wohaPZA xiyA
present give.Pfv.MSg
'Atif gave a present to Mona yesterday in front of everyone.'

DS Analysis ;
AtiPZa ne <k1:VGF> kala <k7t:VGF> monA ko <k4:VGF>
sabake sAmane <k7:VGF> wohaPZA <k2:VGF> xiyA <name=VGF>

6.6 Existentials

a. Existential

Existential-DS-2 : usa kamare meM cUhe hEM
that.Obl room in rats be.Prs.Pl
'There are rats in that room.'

DS Analysis;
usa kamare meM <k7p:VGF> cUhe <k1:VGF> hEM <name=VGF>

b. Predicate Locative:

Predicative-locative-DS-1 : *mInA kamare meM hE*
Mina room in is
'Mina is in the room.'

DS Analysis;
mInA <k1:VGF> kamare meM <k7p:VGF> hE<name=VGF>

As can be observed in the above examples, the dependency analysis of the predicative locative and simple existential would remain same.

6.7 Copular constructions

Copular-DS-1 : *rAma dAktara hE*
Ram doctor be.Prs.Sg
'Ram is a doctor.'
DS Analysis;
rAma <k1:VGF> dAktara <k1s:VGF> hE <name=VGF>

6.8 Causatives

Causative-DS-1 : *AwiPZa ne kala mInA ko kiwAba xilavAyI*
Atif.obl erg yesterday Mina.obl acc book.Sg give.Caus.Pfv.F.Sg
'Atif caused Mina to buy a book yesterday.'
DS Analysis ;
AwiPZa ne <pk1:VGF> kala <k7t:VGF> mInA ko <jk1:VGF>
kiwAba <k2:VGF> xilavAyI <name=VGF>

Causative-DS-2 : *AwiPZa ne kala Arif se mInA ko kiwAba xilavAyI*
Atif.obl erg yesterday Arif.Obl instr Mina.obl acc book.Sg give.Caus.Pfv.F.Sg
'Atif caused Arif to make Mina buy a book yesterday.'

DS Analysis;
AwiPZa ne <pk1:VGF> kala <k7t:VGF> Arif se <mk1:VGF> mInA ko <jk1:VGF>
kiwAba <k2:VGF> xilavAyI <name=VGF>

6.9 Relative clauses (to be included)
6.10 Participles (to be included)
6.11 Complement clauses (to be included)

7. Conclusion

The tagging scheme presented above has been designed to annotate syntactic analysis within a dependency framework. The task of annotation for Hindi is underway. The basic scheme developed initially has been improved and revised. It is planned to conduct some experimental annotation on other languages and test if it can be applied to other Indian languages as well.

8. Acknowledgments

The scheme presented in the document has been developed through intense discussions with several Sanskrit scholars. However, Professor Ramakrishmacharyulu of Rashtriya Sanskrit Vidyapeetha (Tirupati) has been the main resource person who not only explained the theoretical aspects of various Hindi constructions but also helped us in deciding how deeper into analysis we need to go for various Hindi constructions. The scheme would not have taken a shape without his constant support. We are thankful to him for being there for us whenever we are lost (which we often are).

9. References:

R. Begum, S. Husain, A. Dhwaj, D. M. Sharma, L. Bai, and R. Sangal. 2008. Dependency annotation scheme for Indian languages. In Proceedings of IJCNLP-2008.

A. Bharati, V. Chaitanya and R. Sangal. 1995. *Natural Language Processing: A Paninian Perspective*, Prentice-Hall of India, New Delhi, pp. 65-106.

A. Bharati, D. M. Sharma, L. Bai and R. Sangal. 2006. AnnCorra : Annotating Corpora Guidelines For POS And Chunk Annotation For Indian Languages. *LTRC Technical Report-31*

A. Bharati, R. Sangal and D. M. Sharma. 2007. SSF: Shakti Standard Format Guide. *LTRC Technical Report-33*

Rajesh Bhatt. 2008. A Lecture at EFLU, Hyderabad. http://people.umass.edu/bhatt/papers/eflu-aug18.pdf

M. Butt. 2004. The Light Verb Jungle. In G. Aygen, C. Bowern & C. Quinn eds. *Papers from the GSAS/Dudley House Workshop on Light Verbs.* Cambridge, Harvard Working Papers in Linguistics, p. 1-50.

D. Chakrabarty, V. Sarma and P. Bhattacharyya. 2007. Complex Predicates in Indian Language Wordnets, *Lexical Resources and Evaluation Journal, 40 (3-4), 2007*.

E. Hajicova. 1998. Prague Dependency Treebank: From Analytic to Tectogrammatical Annotation. In *Proc. TSD'98*.

M. Marcus, B. Santorini, and M.A. Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank, *Computational Linguistics 1993*.

T. Mohanan, 1994. *Arguments in Hindi*. CSLI Publications.

J. R. Ross. 1967. *Constraints on variables in syntax,* doctoral dissertation, MIT (published as 'Infinite syntax!' Ablex, Norwood (1986)).