Hindi PropBank Annotation Guidelines

Archna Bhatia, Ashwini Vaidya, Bhuvana Narasimhan, Martha Palmer

16 November 2013

Contents

Chapter 1

Propbank Annotation Goals

PropBank is a large annotated corpus consisting of information regarding the argument structure of predicates. PropBanking involves creating a semantic layer of annotation that adds predicate argument structure to syntactic representations[5]. The Hindi PropBank annotations are done on top of the Hindi dependency treebank [2] [4]. For each verb, PropBank represents the information about the arguments that appear with the verb in its corresponding **framefile**. The arguments of the verbs are labeled using a small set of numbered arguments, e.g. Arg0, Arg1, Arg2, etc. The following table shows the framefile for the verb *pii* drink which has two arguments: Arg0 and Arg1.

'pI'	'to drink', Transitive
rAma ne SarAba pI	
'Ram drank liquor.'	
Arg0	Drinker: rAma
Arg1	Liquid: SarAba

Figure 1.1: A framefile for the verb "pI" in

In the framefile for each verb, the numbered argument labels are associated with fine-grained verb-specific descriptions. For instance, in the case of the *pI* 'drink', Arg0 is the 'agent' who performs the action of drinking (the 'drinker') and Arg1 is the entity that is affected by the action of the agent (the 'liquid'). Thus typically, when the verb *pI* appears in a sentence, it will have two arguments which have the semantic roles indicated in its framefile.

Additionally, the sentences in a corpus occur with modifiers that are not part of the semantic specifications of the verb. E.g. the verb *pI* 'drink' implies a 'drinker' and a liquid that is drunk. But the verb itself does not specify *when* the drinking happened or *where* or *how*, so this kind of information is not provided in the framefile for the verb. But if one comes across a sentence such as 'Ram drank liquor in a bar yesterday', the expressions 'in a bar' and 'yesterday' that provide additional temporal, spatial (or manner) information about the

situation, have to be annotated. There is a special set of labels to indicate the kind of information provided by these modifiers. Typically, these modifiers are annotated using function tags such as ArgM-LOC, ArgM-TMP, ArgM-MNR. This information is not provided in the list of core arguments in the framefile, but the examples used in the framefiles may contain these additional arguments and are annotated using the appropriate function tags. The annotators can make use of the examples to see how these ArgM labels are used. However it is not possible to include all the possible modifiers in the examples for each verbframe. Hence the annotators can make use of these guidelines to look for the descriptions for each of these function tags and their corresponding examples in section 4 below.

Hindi-Urdu is a language that allows the speaker to freely omit arguments of the verb in discourse-pragmatically licensed contexts. To take the example above, one can say 'Ram drank liquor', but if 'Ram' has been talked about before, or is otherwise salient in the context, one could say 'drank liquor' without overtly mentioning the subject, 'Ram'. In a corpus, one comes across many such sentences where the arguments of the verb are missing although they can be retrieved from the context. Although PropBank annotation does not typically involve adding empty arguments to syntactic trees, in the case of Hindi-Urdu we have taken a somewhat different approach. We insert the core empty arguments of the verb (subject, object or indirect object) and then go ahead and assign them semantic role labels just as we do for the overt arguments. The information contained in the verb framefiles can act as a valuable resource in allowing for recovery of the different kinds of empty arguments before we annotate them using the semantic role labels.

PropBank annotation is carried out on data that has already been parsed syntactically, or treebanked. In English, this has been done on the Penn Treebank. In the case of Hindi, it will be carried out on a Hindi dependency treebank [1]. Below we mention three goals we consider important for the Hindi PropBank.

(i) As mentioned above an important goal is to provide semantic role labels to the arguments. These are in the form of numbered arguments, e.g Arg0, Arg1, Arg2 etc. They are numbered in order to be more generic and theory neutral [6]. The same numbered argument should be consistent in terms of its semantic role across different syntactic realizations of the same argument of the verb. For example, *darvAzA* 'door' receives Arg1 in both the sentences (1a) where it is the grammatical object of the verb and (1b) where it is the grammatical subject.[1] The reason it receives the same numbered argument label is because the argument *darvAzA* bears the same semantic role in both sentences it is the argument that undergoes motion.

[1] All Hindi examples will be transliterated using the wx format. For a mapping between Devnagari script and the WX characters, please see /data/home/verbs/shared/cleardata/hindi/wx-chart.jpg.

(1) a. *[ARG0 rAma ne]* ***[ARG1 darvAzA]*** *KolA*
Ram Erg door opened
'Raam opened the door.'

b. ***[ARG1 darvAzA]*** *KulA*
door opened
'The door opened.'

The consistency in semantic role labeling is also helpful in the training of machine learning systems such as automatic semantic role labellers. However consistency in semantic role labeling for the arguments of the same verb does not mean that only one set of semantic role (SR) labels is available for a specific verb. PropBank also takes into account the different senses of the same verb while annotating the semantic roles. It is possible that each of the two senses of the same verb has a different set of SR labels. For example, the verb *KilanA* takes an Arg1 only in (2a) with sense1 'to bloom', but it takes an Arg1 and Arg2_LOC in (2b) with sense2 'to look good'.

(2) a. ***[ARG1 kaliyAM]*** *KilIM*
buds bloomed
'The buds bloomed'

b. ***[ARG1 poSAk]*** ***[ARG2_LOC Ap par]*** *Kil rahI hE*
dress you on bloom PROG are.
'The dress is looking good on you.'

(ii) The second goal of the PropBank annotation involves assigning function tags to all modifiers of the verb (or of the verb's event structure- these modifiers correspond with syntactic adverbs). Some examples are manner (ArgM_MNR), locative (ArgM_LOC), temporal (ArgM_TMP). For a complete list of these function tags, please refer to section 4 below.

(3) ***[ARG0 buS]*** ***[ARG1 us se]*** ***[ARGM_TMP guruvAr ko]***
Bush him with Thursday on

[ARGM_LOC White House mEM] ***[ARGM_MNR akele mEM]*** *mile*
White House in alone in met

'Bush met with him privately in the White House on Thursday.'

(iii) Finally, Hindi PropBank annotation involves annotation of null arguments in the context of a particular verb sense, e.g. it is possible to say *calA gayA* '(he) left' in Hindi without overtly specifying the core argument Arg0 of the verb *cala* 'leave' if the argument is recoverable from the discourse-pragmatic

context. We also identify arguments of the verb that are obligatorily null: (a) the null subject of complement clauses of verbs such as *cAha* 'want' (*mohana* NULL *Gara jAnA cAhatA hE* 'Mohan wants [NULL to go home]'); (b) the null subject of adjunct clauses (*Gara jAkar mohana KAnA KAegA* '[(After) NULL going home], Mohan will eat food'); (c) the gapped argument in participial modifiers (Kile PUla [NULL] blossomed flowers); (d) the omitted arguments in coordinate constructions (*mohana Gara jAegA aur KAnA KAegA* 'Mohan will go home and [NULL] eat food').

An example where we insert a null element in a sentence with a complement clause (so-called 'PRO' in generative grammar) is shown below (4a – 4b). Note, in (4b), the subject of the verb *jAnA* 'to go' in this example is inserted as an empty category PRO at the PropBank stage and eventually it is annotated like any other overt argument.

(4) a. *[ARG0 **mEMne**] [[ARGM-GOL **Gara**] jAnA] cAhA*
I-Erg NULL home to-go wanted
'I wanted to go home.'

b. *[ARG0 **mEMne**] [[ARG0 **PRO**] **[ARGM-GOL Gara]** jAnA] cAhA*

These three tasks of PropBank annotation: argument labeling, annotation of modifiers, and insertion and annotation of empty categories are discussed in detail below. Besides these three tasks, Hindi PropBank annotation is also concerned with the annotation of complex predicates, (discussed in section 7) and with distinguishing unaccusative vs. unergative verbs (discussed in section 6). In section 4, we will discuss various issues and constructions which would aid the process of empty category insertion and PropBank annotation of arguments (overt elements or empty categories), modifiers and complex predicates. But first in section 2, we describe how various tools, such as Cornerstone and Jubilee, are used for these tasks.

Chapter 2

Using the tools

2.1 Framefiles

Framefiles are an important source of information for annotators (In PropBank, the annotation is carried out verb-by-verb, i.e. a verb is selected, then all the sentences in which that verb appears are identified in the treebank, the annotators then look at the framefile for that verb and annotate the arguments for that verb in all these sentences based on the information provided in the verb's framefile. This process is repeated for each verb.). The framefile for a verb provides information such as what arguments a verb takes, what semantic roles the verb assigns to these arguments, what different combinations of arguments can appear with a specific verb form, what different forms a verb appears in (alternations based on transitivity etc). Hindi examples are also provided in the framefiles as ilustrations for the annotation of Hindi sentences.

On framefile naming convention: Note that in Hindi, a verb can show up in various forms depending on the transitivity alternation, causativization etc. All these related verb forms are included in the same framefile. If an intransitive form of the verb is possible, then the framefile is named after this intransitive form. For example, for the intransitive verb form *diKa* 'be seen', we have the transitive counterpart *diKA* 'show', and the causative form *diKavA* 'make someone show something/someone'. Thus the framefile is named after the verb form *diKa* (the intransitive form), hence named as "diKa-v.xml". However, for some verbs, such as *KarIda* 'buy', there is no intransitive counterpart. In such cases, the transitive verb form is used in naming the framefile for the verb, hence the name "KarIda-v.xml".

Before we discuss the framefiles in detail, let's talk briefly about two terms which we will be using in the discussions of framefiles: rolesets and predicate frames. The term roleset refers to a block in the framefile of a verb containing the information about the set/ list of arguments a verb takes in that specific form (intransitive form or transitive form etc). The term predicate frame refers to a block in the framefile of a verb which consists of one or more rolesets and

corresponds with each possible verb form (i.e. one predicate frame associated with an intransitive verb form, another predicate frame associated with a transitive verb form, another one associated with a causative verb form etc). These terms will become more clear as we see them used in the text below.

Let us now talk in detail what information framefiles contain. The framefiles for each verb provide fine-grained information specific to that verb, e.g. it contains information about the arguments 'sender' and 'sendee' etc for the verb 'send'. Thus the framefile provides a description of verb-specific semantic roles as is shown for the verb *Beja* 'send' in figure 2.1 below. Note the set of semantic roles Arg0, Arg2_Gol and Arg1 as one of the possible rolesets[1] "Beja.02" for the verb *Beja*. Note that a description is provided for each semantic role mentioned in this roleset. For example, Arg0 is used for "the one who sends something", i.e. the sender etc.

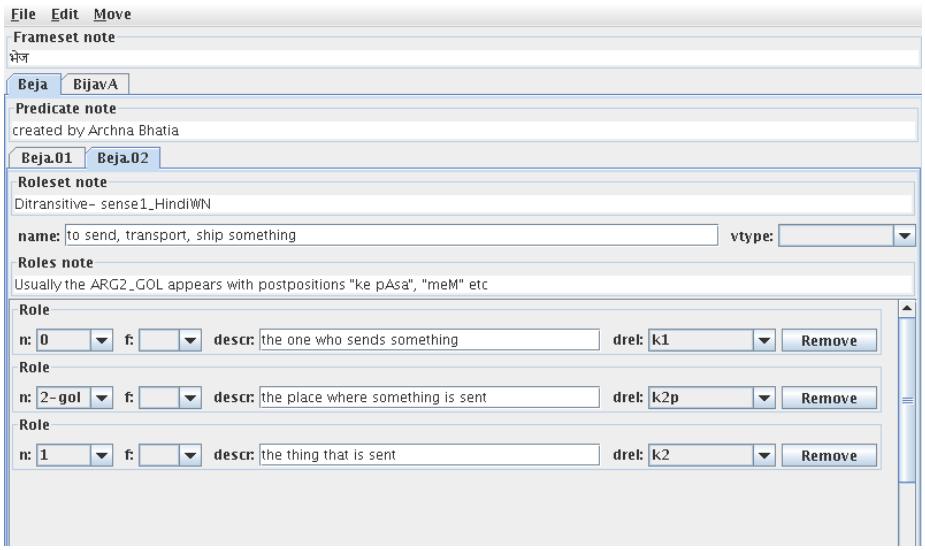

Figure 2.1: A screenshot for a roleset "Beja.02" for the verb *Beja* 'send'

Besides a description of these roles, the framefile also provides examples which illustrate the use of these semantic roles. This is shown through figure 2.2 below. Note while the roleset only provides a list of core arguments for the verb (as in figure 2.1 above), the examples may contain modifiers as well, e.g. ArgM_MNR in the example in figure 2.2 below.

[1] A roleset for a verb is a pattern in which a set of semantic roles for that specific verb are realized in a sentence (please note that the order of phrases representing these roles is flexible in the actual sentences). The same verb can have multiple rolesets, i.e. it can have different patterns in which the semantic roles are realized for that verb. For example, for the verb Beja, two rolesets "Beja.01" and "Beja.02" have been identified in the framefile. The roleset "Beja.01" consists of semantic roles Arg0, Arg2 and Arg1, whereas the roleset "Beja.02" consists of semantic roles Arg0, Arg2$_G$ol and Arg1. See figure 2.1 in the text above for reference.

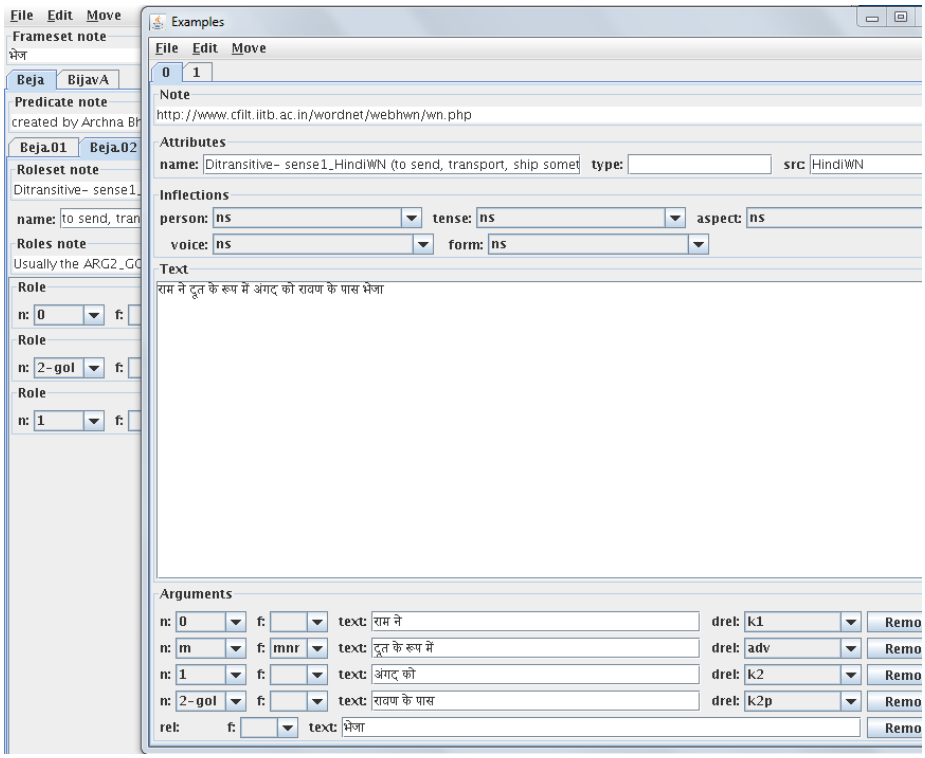

Figure 2.2: An example for the roleset "Beja.02" in the framefile for the verb *Beja* 'send'

If a verb has more than one sense but the argument structure of the verb is the same corresponding to those senses (i.e. same set of semantic roles is used), then this may also be represented in the framefile, with corresponding examples for these senses within the same roleset. This is shown through the following two figures (figures 2.3 & 2.4) which represent an example each for two senses of the same verb with the same argument structure (i.e. belonging to the same roleset "Jalaka.04" here). Note in figure 2.3 as well as in figure 2.4, the core arguments used are Arg0 and Arg1. The figure 2.3 represents sense 'to make something shine', while figure 2.4 represents the sense 'to make something be visible'.

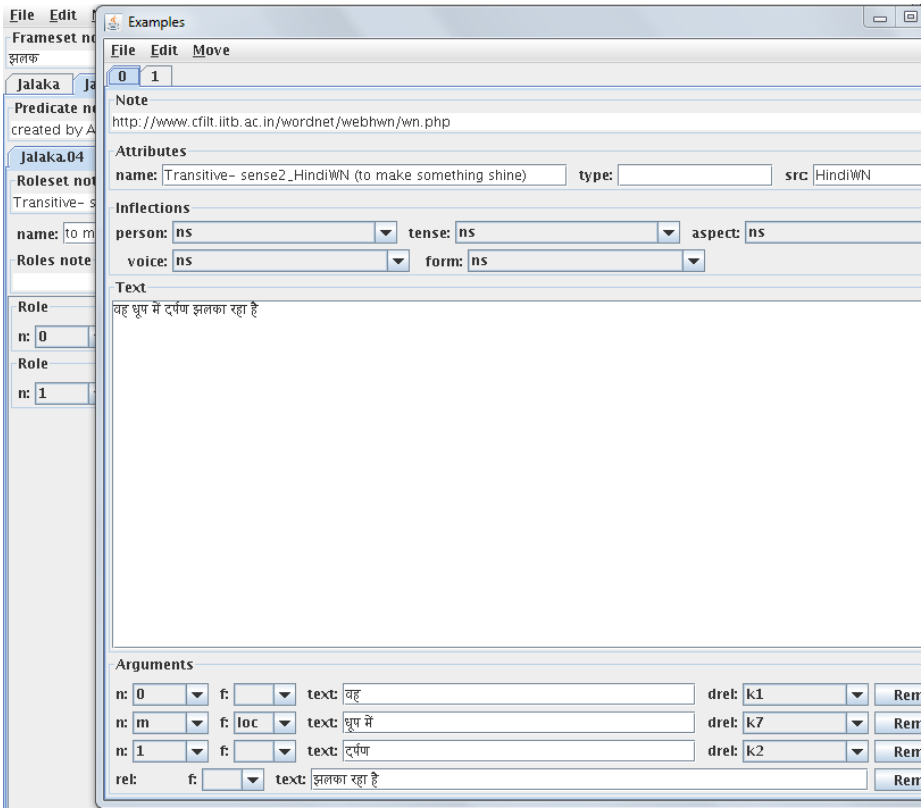

Figure 2.3: An example for one sense for the roleset "Jalaka.04" in the framefile for the verb *Jalaka*

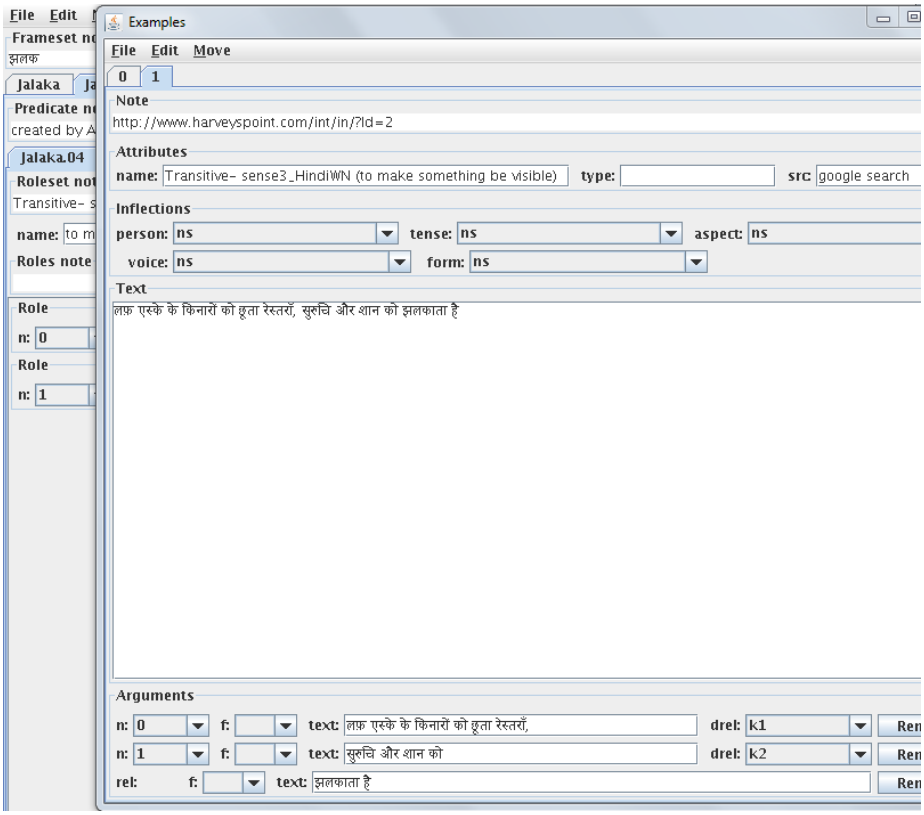

Figure 2.4: An example for another sense for the same roleset "Jalaka.04" in the framefile for the verb *Jalaka*

If, however, a verb has more than one sense and the argument structure is also different corresponding to these senses (i.e. a different set of semantic roles is used for each sense), then this is represented in the framefile using a separate roleset for each of the two senses of the verb. This is shown through the following two figures (figures 2.5 & 2.6). Note figure 2.5 represents the sense 'to stay/lodge/be located at' and the verb takes arguments Arg0 and Arg2_Loc, whereas in the figure 2.6 with sense 'to last/endure', the verb takes a different set of arguments, viz. Arg1 and Arg2_Loc. Hence we see that the same verb can have two different argument structures for two different senses. Such cases are represented in the framefile using two separate rolesets for the verb, here "Tahar.01" and "Tahara.04" respectively.

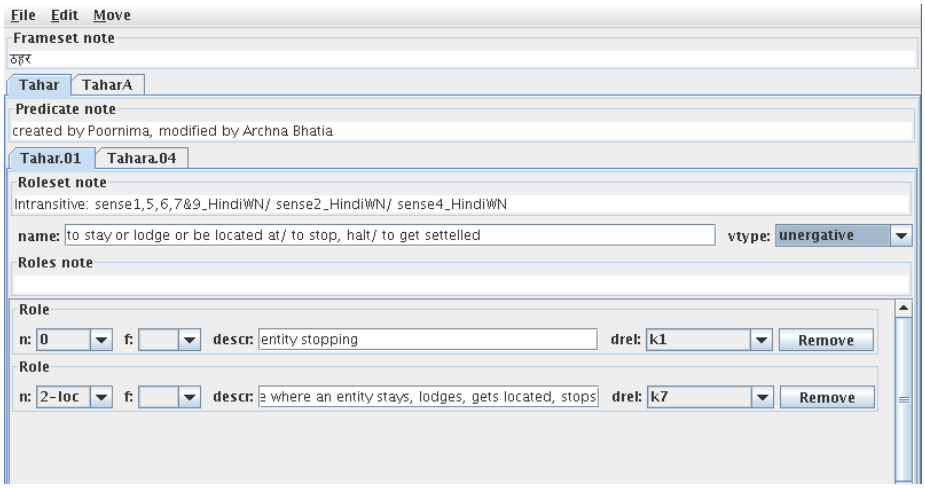

Figure 2.5: One roleset "Tahar.01" for one sense of the verb *Tahara* with one set of semantic roles, namely, Arg0 and Arg2_Loc.

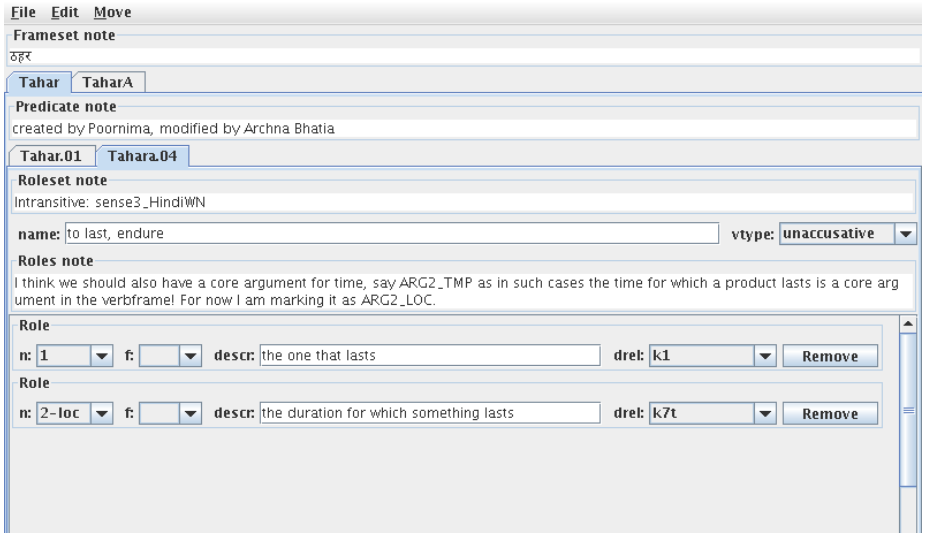

Figure 2.6: Another roleset "Tahara.04" for another sense of the verb *Tahara* as it appears with another set of semantic roles, namely, Arg1 and Arg2_Loc.

Thus we see that the framefile for a verb can have different rolesets, e.g. "Tahar.01" and "Tahara.04" in the figure 2.6 above. These rolesets belong to the same predicate frame, namely "Tahar" as can be seen in the figure 2.6 above. However the framefile for a verb can also have multiple predicate frames (and usually framefiles do have multiple predicate frames). Notice the figure 2.6 above has two predicate frames, "Tahar" and "TaharA". The separate predicate frames are used to refer to various morphological forms a verb can have. In Hindi, we usually see that a verb can have more than one morphological realization depending on its transitivity alternations or causativization. For example, the intransitive verb *cala* 'walk' *cala* 'walk' has a transitive counterpart *calA* 'make someone/something walk', and a causative counterpart *calvA* 'make someone make someone else/something walk'. These different forms are represented using three separate predicate frames in the framefile for the verb *cala*. Besides the above mentioned possibilities of predicate frames in a framefile for a verb (intransitive verb, transitive verb, causative verb etc), the complex predicates also constitute as separate predicate frames in the framefile. Hence when a verb is combined with a noun or adjective to form a complex predicate (also known as support verb or light verb construction), e.g *PEslA kar* 'make a decision' (lit. decision do), this also appears as a predicate frame in the framefile for the verb *kara* 'do'. See section 7 for more details on the complex predicates.

2.2 Framefiles creation and use of Cornerstone

The procedure used in creating Hindi verb framefiles is described in this section. As mentioned above, the issues that arise in this context are also discussed in 4 below (also discussed in the literature on syntax-semantics mappings in South Asian languages, e.g. Mohanan 1994, Butt 1995).

2.2.1 Verb selection

We begin creating framefiles, starting with the most frequent verbs in the corpus and then going on to the lower-frequency verbs. As was mentioned in section 2.1 above, when framefiles for verbs are created, the primary senses of the verbs (with different sets of numbered arguments) have corresponding rolesets and examples to illustrate the semantic role labelling on each argument of the verb for each of these rolesets. These examples are either drawn from the corpus or , internet or they may be self created by the framefile creators (who generally are native Hindi speakers).

2.2.2 Annotation using Cornerstone

The screenshot below (as well as in the figures 2.1-2.6 above) shows the tool Cornerstone which is used to create a framefile. Cornerstone may be accessed at /home/verbs/shared/propbank/cornerstone. Make sure that you use version 1.35.

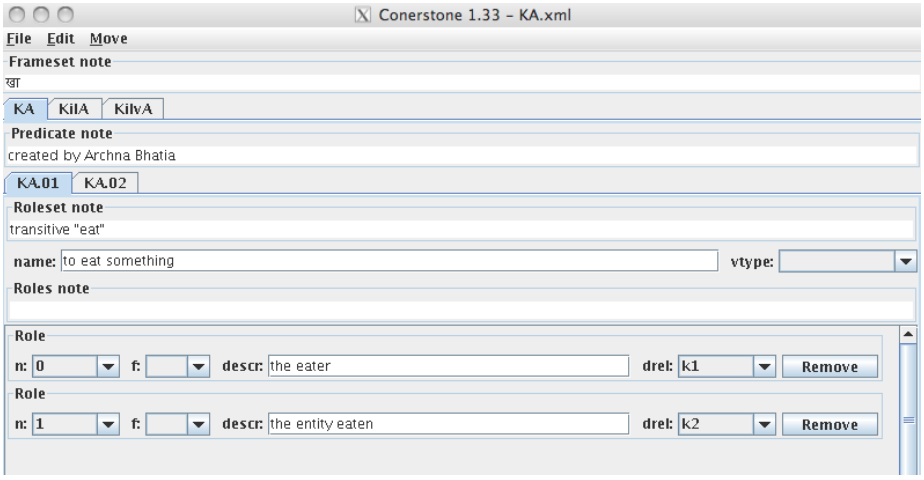

Figure 2.7: A screenshot of the tool Cornerstone

Below, we outline the format for entering values in each of the fields seen in Cornerstone above:

Frameset note

This field is used to enter the verb LEMMA in Devnagari script (the script that is used to write Hindi). The lemma is the citation form of the verb (in Hindi, this is the bare form of the verb, e.g. *KA* 'eat', and not its inflected forms, e.g. *KayA* 'ate').

Frameset tabs

Here we enter different forms of the verb's lemma (in Romanized script) based on its transitivity, causativization and its involvement in the complex predicates. In the example above, we can see that *KA* 'eat' becomes *KilA* 'make someone eat' (i.e. 'feed') and *KilavA* 'cause someone to feed someone else'. Each of these will get a separate predicate frame as mentioned in section 2.1. above. If the verb appears as a support verb (alternatively known as a light verb) in a complex predicate, e.g. the verb *kara* in the complex predicate *snaan kar* 'bath do' then we will enter "*kara_LV*" as one of the predicate frames (see section 7 for more details).

Predicate note

This field has the information about who created the predicate frame (for future reference in case we need to contact the creator to inquire about some of the decisions).

Roleset tabs with Roleset Ids

Roleset note

This field carries some identification information about the roleset, e.g. what kind of a verb it is, whether it is an intransitive verb or a transitive verb or a causative verb. If it is an intransitive verb, we further specify whether it is unaccusative or unergative. The unaccusativity or unergativity of an intransitive verb is determined based on a set of diagnostics that are applied to all the intransitive verbs. The diagnostic criteria that the verb passes for being considered as an unaccusative or as an unergative verb are also mentioned in this note tab. 1 Moreover, if there are exceptional cases among unaccusatives like the motion verbs which may display variable behavior with respect to the unaccusative diagnostics, then this information should also be specified in the roleset note.

Name

This field provides the meaning or sense of the verb in English. This is the best English gloss the framefile creators could provide for the verb Many a times, this is one of the senses mentioned in the Hindi wordnet.

vtype

This field is used to mention the type of intransitive Verb, i.e. whether it is an unaccusative verb (Unacc) or an unergative verb (Unerg). The unaccusativity/ unergativity of the intransitive verb is determined by applying a set of diagnostics to these verbs, see section 6 below for more details.

Roles note

This field is used to enter any information to either disambiguate the roles or to provide any additional syntactic/ morphological information relevant to the roles. For example, in the roleset "Jada.03" in figure 2.8 below, syntactic/ morphological information about possible case markings on the Arg1 argument is mentioned.This information is provided as it usually aids the annotators further in identifying the argument or clear some confusion about the argument.

Figure 2.8: An example of additional syntactic/ morphological information provided in the Roles note in a Roleset

Role

This field is used to add any numbered tags or function tags used to represent the semantic role of the core arguments for the verb in that roleset. For example, n: 0 for an Arg0 argument; or n: 2, f: GOL for an Arg2_GOL argument. Note in the figure 2.8 above, Role field suggests that the verb Jada takes two core arguments Arg0 and Arg1 in the Roleset "Jada.02".

descr

This field is used to provide a description about that argument, e.g. in the figure 2.8 above, Arg0 gets the description "the one who scolds somebody", the Arg1 gets the description "the one who is scolded". Thus note that these descriptions are verb specific and are not selected from a predetermined list, such as Agent, Patient etc.

drel

This field is used to enter karaka label, e.g. k1/k2, mapping for the PropBank role. Note in the figure 2.8 above, Arg0 argument gets the k1 drel label, and the Arg1 argument gets the k2 drel label.

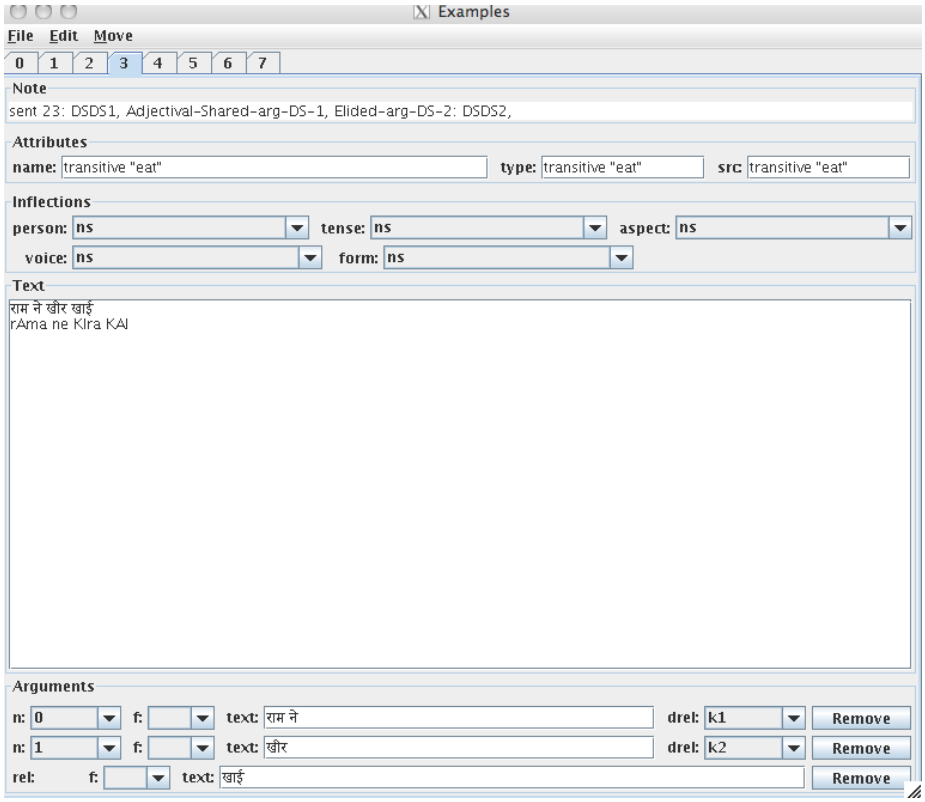

Figure 2.9: A screenshot of the example for a roleset in the framefile for the verb *KA* 'eat'

The above is a screenshot of the example screen. As mentioned above in section 2.1, in the framefiles, we provide an example for each of the rolesets that is included in the framefile. Following is a brief description of the fields that are used in the examples.

Note

This field provides the information about the source of the example sentence, e.g. whether the example was drawn from our Hindi DS-treebanked corpus, Hindi wordnet, web or if it was self created by the framefile creators. The above example shows the actual sentence number in the corpus from which it was taken. However, it is not necessary to include the number, a note about the source should be enough.

Name

This field is a brief identifier of the example/meaning. For example, in the above screenshot, it says that the verb used in this example is transitive and has the

meaning 'eat'.

src

This field is optional, it provides the title of the corpus that was the source.

Text

This field is used to provide the actual example in Devnagari script. For example, in the screenshot above, it gives the following sentence. We also provide the romanized script, the glosses and the translation for the example to aid readers in understanding how the arguments get PB roles which are mentioned further below.

(5) ***[rAma ne]*** ***[Kira]*** *KAI*
Ram-Erg *pudding* *ate*
'Ram ate the pudding.'

The example may also include indices where relevant.

Besides, the example may also use bracketing if required. This is used in case of complex predicates. Notice the use of the square brackets in the devanagari part in the example 6 below, the main verb *KAI* and the light verb *gal* form a unit together which is shown using the square brackets in the examples on cornerstone.

(6) ***[kala]*** ***[yahAM]*** ***[bahuwa Kira]*** *KAI* *gaI*
yesterday *here* *a lot pudding* *eaten* *go.Perf*
'Yesterday a lot of pudding was eaten here.'

Arguments

This field is used to add any numbered tags or function tags used to represent the semantic role of the core arguments for the verb in that roleset. For example, we may see, n: 0, or n: 1, etc. , and f: TMP, f: LOC, etc . Note the use of numbered tags Arg0 (n: 0) and Arg1 (n:1) in the example in figure 2.9 above, Below we present another example screen in figure 2.10, copied from figure 2.3 above. This figure shows the use of numbered tags Arg0 (n: 0) and Arg1 (n: 1) and of function tag ArgM-Loc (n: m, f: loc).

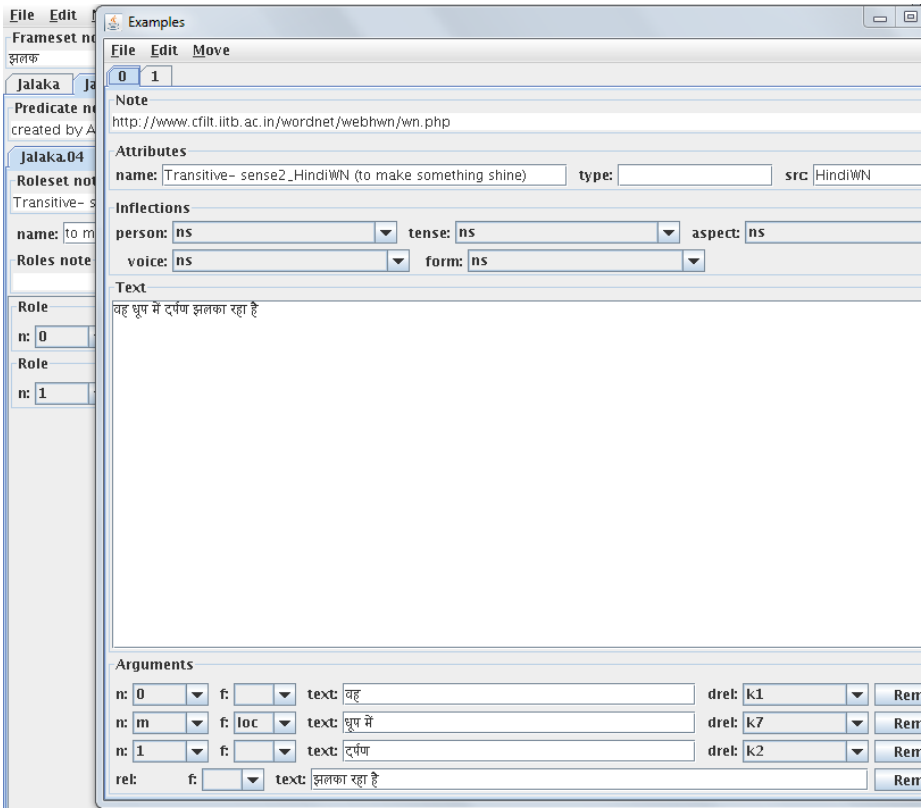

Figure 2.10: A screenshot of the example for a roleset in the framefile for the verb *JalakA* 'make something shine'

text

Relevant argument chunk from the example is inserted in this field. Notice the text in Devanagari script in the text fields in figures 2.9 and 2.10 above.

drel

This field is used to enter karaka labels such as k1, k2 etc that maps to the PB roles. Note the drel fields in figure 2.9 (use of k1 and k2) and figure 2.10 (use of k1, k7, and k2).

Sometimes more than one example has been used for a specific roleset if it is necessary to illustrate a point. This is usually done to represent each of the senses of the verbs mentioned in a specific roleset, or to illustrate the use of function tags which are not necessarily present in the other examples. Notice the two example tabs labelled 0 and 1 in figure 2.10 above, which show that for this roleset of the verb JalakA, two examples (example 0 and example 1) have been provided.

2.3 Using the Framefiles for PropBank annotation

As is mentioned and also shown through the figures in sections 2.1 and 2.2 above, the framefiles contain information about the verb senses, the set of arguments the verbs have corresponding to these senses, examples showing the use of PB labels etc. For an example, let's look at the information contained in the framefile for the verb *kara* for one sense of the verb, viz. 'do', represented by the roleset numbered "kara.01".

kara.01 ; sense 'do'
Arg0: doer
Arg1: thing done

(7) *[ARG0 **rAma ne**] [ARG1 **kAma**] kiyA*
Ram Erg work did
"Ram worked."

However, as was mentioned above too, even though the verbs can be quite polysemous, it would be time consuming and not so useful to provide a set of roles for all the different senses of the verb if the set of roles is the same. Instead, we only differentiate between two senses of the verb when they take different sets of arguments, hence these senses are represented using different rolesets. This is shown through the example below where we provide some information for two senses of the verb *baca* ("baca.01" and "baca.02") from its framefile.

baca.01 'remain'
Arg1: thing remaining
Arg2: benefactive, entity getting the remaining thing

(8) *[ARG02 **mere pAsa**] [ARG01 **10 rupaye**] bace hEM*
me-obl near/possession 10 rupees remained be
'I have 10 rupees left with me.'

baca.02 'avoid'
Arg0: avoider
Arg1: thing avoided

(9) *[ARG0 **yaha**] [ARG0 **SarAba se**] bacawA hE*
He liquor from avoid be
'He avoids (drinking) liquor.' i.e. 'He runs from the liquor.'

In this case, we have two rolesets indicating the difference in the senses for

the verb *baca* which also have different sets of argument roles. When annotating, it is necessary to look at all the different senses specified in the framefile before annotating. The annotation tool- Jubilee will load the framefile in the PropBank annotation pane. An example of this is shown in figure 2.11 below.:

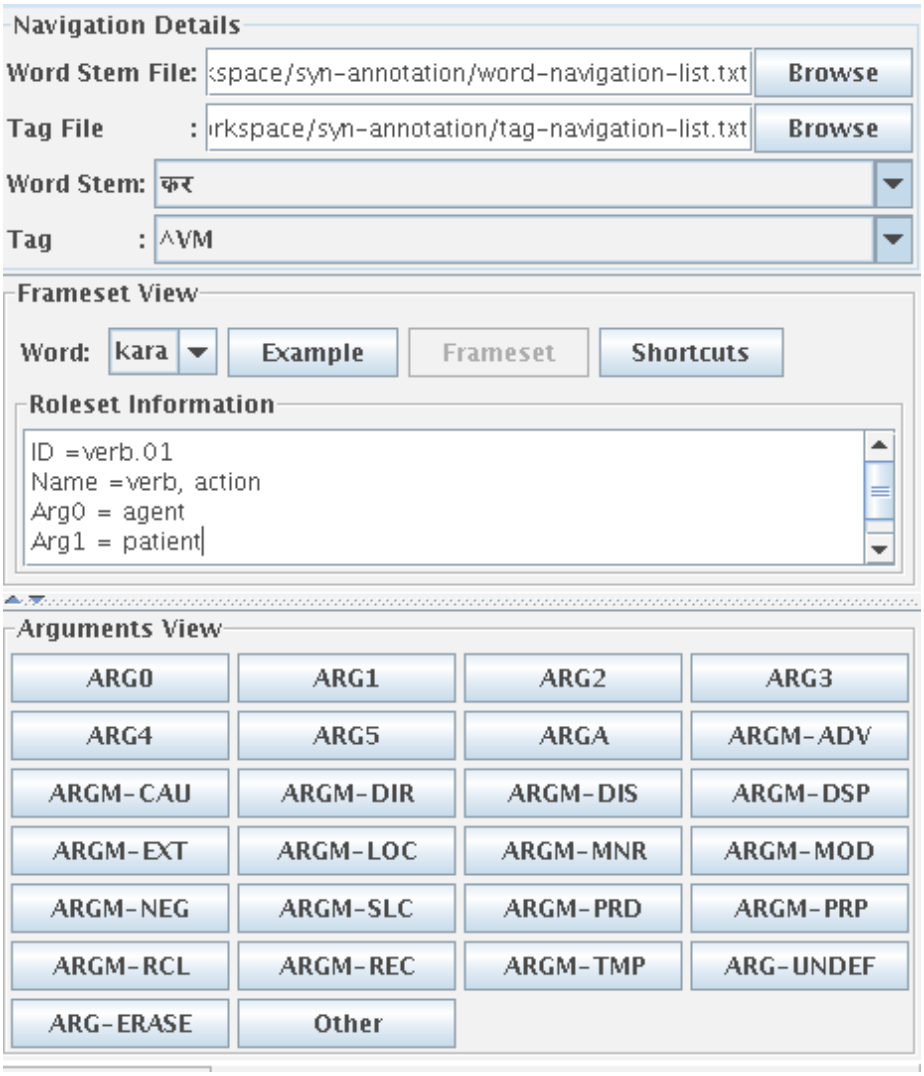

Figure 2.11: A screenshot of the tool Jubilee used by annotators for PB annotations of the Hindi treebank

While the annotators are annotating a sentence, the framefile for the relevant verb opens in Jubilee.Thus the annotators can look at the different rolesets the verb has. Depending on the arguments present in each roleset, the annotators determine which roleset the sentence represents. They then select each argument

present in the sentence one-by-one and for that argument the relevant PB tag is selected from the list of arguments in the "Argument view" pane as shown in figure 2.11 above. In the following two sections, we provide the descriptions for each of these argument tags which were shown in the "Arguments view" pane above.

Chapter 3

Annotation of numbered arguments

3.1 Description of the numbered arguments

The core arguments of a verb are annotated using numbered arguments, except for ArgA and ArgA_MNS which are not numbered as such, however they are also included in the list of numbered arguments (and will be referred to as the numbered arguments henceforth) since they are also used to annotate the core arguments in the causative constructions just like the other numbered arguments are used to annotate the core arguments in a construction/ a sentence. We use 12 numbered arguments. However, some of these also carry a function tag besides being numbered to further differentiate between the different types of arguments, for example Arg2_GOL, Arg2_LOC, etc. The complete list of numbered arguments used in the Hindi PropBank annotation is as follows:

Numbered Arguments	Description
1. Arg0	Prototypical agent, actor, experiencer
2. Arg0_GOL	The doer of an action which is also the recipient or beneficiary of the caused action (only used in causative constructions)
3. Arg0_MNS	The doer of an action which is caused by a causer to perform the action and it is not the obvious recipient or beneficiary of the action and when the action is performed on an Arg1 argument (only used in causative constructions)
4. ArgA	The only causer present or the external-most causer in case of multiple causers
5. ArgA_MNS	Intermediate Causer
6. Arg1	Prototypical patient, theme
7. Arg2	Beneficiary, receiver
8. Arg2_GOL	Goal, destination
9. Arg2_SOU	Source
10. Arg2_ATTR	Attribute
11. Arg2_LOC	Location
12. Arg3	Instrument

Below we provide the descriptions for each of these numbered argument labels. But before we look at these labels individually, a general observation about these lables should be noted. The argument labels for the core arguments are not repeated within the same sentence in PropBank. This observation represents the fact that each possible core argument manifests itself only once in a sentence, i.e. multiple arguments cannot be treated as the same core argument for the same verb in the same sentence. However, the label ArgA_MNS and Arg3 are exceptions to this rule. The label ArgA_MNS, despite being considered as a numbered argument label, can be used multiply within the same sentence, used to annotate each intermediate causer present in the sentence. This is possible since intermediate causers are such core arguments that can repeat, i.e. more than one intermediate causers can be present within the same sentence. Similarly the label Arg3 can be used repeatedly since multiple instruments are possible within the same sentence. See sections 3.6 and 3.13 for examples. Let us now look at each of these numbered argument labels in detail.

3.2 ARGO

In general, the numbered argument Arg0 corresponds to the prototypical agent of a verb. The verb itself can be intransitive, or transitive (i.e. monotransitive as well as ditransitive). Below we provide an example for an Arg0 argument

with each an intransitive verb, a monotransitive verb and a ditransitive verb, In example 10a with an intransitive verb *ro* 'cry', the entity performing the act of crying rAma is considered the agent and hence, it is annotated with the PB tag Arg0. In example 10b with a monotransitive verb *KA* 'eat', the entity performing the act of eating is considered the agent and hence, it is annotated with the PB tag Arg0. Similarly, in example 10c with a ditransitive verb *xe* 'give', the entity performing the act of giving is considered the agent and hence, it is annotated with the PB tag Arg0.

(10) a. *[ARG0 **rAma]** rowA hE*
Ram cry-Hab be.Pres
'Ram cries'

b. *[ARG0 **rAma]** [KIra] KAwA hE*
Ram ricepudding eat-Hab be.Pres
'Ram eats rice pudding'

c. *[ARG0 **rAma]** [sIwA ko] [kiwAba] xewA hE*
Ram [Sita to] book give-Hab be.Pres
'Ram gives Sita a book'.

Among the intransitive verbs, the unergative type verbs get an Arg0 argument. As was mentioned above and is also discussed in detail in section 6 below, there are a number of diagnostics that are applied to the intransitive verbs to determine whether a verb is unergative type or unaccusative type. However, just as a rough clue, we typically find Arg0 in those intransitive verbs that take an animate subject (i.e. a subject that has voluntary control over the action expressed by the verb). For example, the verbs *nAca* 'dance' and *xORa* 'run' are examples of intransitive verbs that require an agentive subject, typically animate subjects, because the actions expressed by these verbs involve a participant that has voluntary control over the action described by the verb. Thus, for example, in the following sentence, the agentive participant *sIwA* gets the label Arg0.

(11) *[ARG0 **sIwA]** nAca rahI hE*
Sita dance Prog be.Pres
'Sita is dancing.'

Inanimate subjects of intransitive verbs can also be Arg0 when they display strong agentivity. For instance, natural forces such as *AMdhi* 'storm' or *wUfAna* 'typhoon' are labeled with Arg0 in sentences such as the following

(12) *[ARG0 **tUfAna ne]** [KhidzkiyoM ko] woda dAlA*
typhoon Erg windows Acc break put.Pst
'(The) typhoon broke the windows.'

Metaphorical extensions also can involve an Arg0 if the verb typically takes an agentive subject. For example, in the following sentence, the metaphorical agent *bAriSa* ne 'rain Erg' also gets the argument label Arg0.

(13) *[ARG0 **bAriSa ne]** [PuloM ko] [nayA jIvana] xiyA*
rain Erg flowers Dat new life give.Pst
'(The) rain gave the flowers new life.'

We also assign the Arg0 label for possessor subjects as in the following example. The framefile indicates that such arguments lack the degree of agentivity associated with 'prototypical' transitive verbs such as *toRa* 'break' or *mAra* 'hit':

(14) *[ARG0 **rAma ke]** [ek beTI] hE*
Ram Gen one daughter is
'Ram has one daughter.'

For passivized subjects, although syntactically, the subject has been demoted, it is still the doer of the action and hence, it gets the label Arg0. See 15 below. Note, however, that the agentive argument is annotated with the PB label Arg0 only if it is explicitly realized in the sentence. That is, PB does not insert a null argument (and annotate it as Arg0) for the agents of the passive verbs if they are not already explicitly present in the sentence.

(15) *[ARG0 **rAma xvArA]** [Kira] KAyI gayI*
Ram by ricepudding eaten was
'(The) rice pudding was eaten by Ram.'

In Hindi, the morpheme *–ne* is often indicative of an agent, and as Arg0 is mostly associated with agentivity, this can sometimes provide a clue about the identity of Arg0. However, note that an Arg0 argument does not necessarily get ne-marking. The *ne*-marking is a helpful clue only when the verb is transitive and in the perfective aspect.[1] For example, in 16a below, the transitive verb is in the perfective aspect and the agentive subject gets the *ne*-marking, but

[1]Even though case marking can provide a clue to the use of a PB label for an argument in some tenses and aspect combinations etc, PropBanking does not entirely depend on it, For PropBanking, we analyze the verbs event and its participants irrespective of the aspect and transitivity of the main verb or the use of a light verb that changes the case on the noun.

in 16b, the verb is in non-perfective aspect and the agentive subject does not get the *ne*-marking. Similarly in 17a, since the verb is intransitive, the agentive subject does not get *ne*-marking. Also certain light verbs do not allow the use of *ne*-marking and hence the agentive subject appears without it. For example, in 17b below, the light verb *jA* 'go' is used which cannot cooccur with *ne*-marking on the subject, hence the subject *rAma* appears without *ne*-marking.

(16) a. *[ARG0 **rAma ne]** [Kira] KayI*
Ram Erg ricepudding eat.Pst
'Ram ate the rice pudding.'

b. *[ARG0 **rAma]** [Kira] KawA hE*
Ram ricepudding eat-Hab be.Pres
'Ram eats rice pudding.'

(17) a. *[ARG0 **rAma]** xOdA*
Ram run.Pst
'Ram ran.'

b. *[ARG0 **rAma]** [sArI Kira] Ka gayA*
Ram all ricepudding eat go.Pst
'Ram ate up all the rice pudding.'

3.3 ARG0_GOL

The Arg0_GOL label is used in two types of constructions in the Hindi PropBank: the causative constructions, and the experiencer subject constructions.

In causative constructions, it is used for a participant that is a causee who is the agent of the action and also the receiver/beneficiary of the caused action, e.g. the argument *bacce ko* 'child to' in the following sentence gets the Arg0_GOL label. Notice that in the example, the kid is the agent of the action denoted by the innermost verb 'eat' (hence "Arg0" part of the label is appropriate), but he/she is also the recepient of the caused action 'feed' (hence the "GOL" part of the PB label is appropriate). Also notice that even though the child is performing the act of eating, he/she does not initiate the eating by him/herself but is a causee, he/she is made to eat by a causer. Thus we notice that the PB label Arg0_GOL is used only in presence of another label ArgA used to represent the causer in the sentence. For further details, refer to 5 below.

(18) *[ARGA sIwA ne] [AyA se] [ARG0_GOL **bacce ko**] [KAnA] KilvAyA*
Sita Erg maid by child to food feed.CAUS.Pst
'Sita had the maid feed the child food'

The label Arg0_GOL is also used in cases where the sentence has an experiencer subject. In Hindi, the experiencer subject can be found in sentences like 19 below. Notice that the subject always appears with the Dative case marker *-ko*. In these constructions, the subject does not have agent like properties, because it is not controlling the action, but nevertheless the act expressed by the verb is an internally caused reaction to some external/ internal factor. In such cases, the subject experiences or feels some physical state such as a fever, hunger, cold etc or an emotional state such as anger, embarrassment etc, hence the term experiencer subject construction. For these cases, the framefile indicates the 'non-prototypicality' of the agents by specifying that the Arg0_GOL is an 'experiencer' in the description of the role in the framefile.

(19) a. *[ARG0_GOL **muJ-ko**] [TaMDa] lagI*
I-Dat cold feel.Pst
'I felt cold (it felt cold to me).'

b. *[ARG0_GOL **muJ-ko**] [buKAra] huA*
I-Dat hunger happen.Pst
'I got fever (the fever happened to me).'

c. *[ARG0_GOL **muJ-ko**] [gussA] AyA*
I-Dat anger come.Pst
'I got angry (the anger came to me).'

d. *[ARG0_GOL **muJ-ko**] [Sarma] AyI*
I-Dat shyness come.Pst
'I felt embarrassed (teh embarrassment occured to me).'

Besides the physical or emotional states, the experiencer subject is also found with certain verbs like *diKa* glimpse, *mila* find, *laga* feel/seem, *sUJa* be struck (with an idea).

e. *[ARG0_GOL **muJ-ko**] [cAMda] dikhA*
I-Dat moon be.seen.Pst
'I glimpsed the moon.'

3.4 ARG0_MNS

The label **Arg0_MNS** (20a-20b) is also used to annotate the doer of an action but when the doer is caused by a causer to perform the action (similar to Arg0_GOL above) but it is not the obvious recipient or beneficiary of the action (different from the Arg0_GOL above). Note that this label is applied only when the the action is performed on an Arg1 argument, thus basically this label is used only when the event for the action is represented by a causative form of a base unaccusative verb or a base transitive verb. The Arg0_MNS may appear with postpositions "se" or "xvArA" in some cases. These postpositions are used with an argument that acts as a means to get some action done. Note Arg0_MNS is used as a means by its causer to achieve some result, i.e. to get these arguments to perform some action. (this is observed with verbs such as *giravA* 'cause (someone) to make (something) fall', *KulavA* 'cause (someone) to make (something) open', *tuRavA* 'cause (someone) to break (something)', *bikavA* 'cause to sell' and so on.)

Note that when the base verb is an unergative verb instead of an unaccusative verb, even when the causers are present, the doer of the action gets Arg0 label, not Arg0_MNS label. This is so because for an argument to be an Arg0_MNS, the caused argument has to perform an action on an Arg1 argument, but for base unergative verbs, that condition is not met. For further details, refer to 5 below.

(20) a. *[ARGA mohana ne]* ***[Arg0_MNS rAma se]*** *[Arg1 peRa]* *kat-vAyA*
Mohan Erg Ram Inst tree cut-CAUS.Pst
'Mohan made/had Ram cut the tree.'

b. *[ARGA mohana ne]* ***[Arg0_MNS rAma se]*** *[Arg1 tikata]* *KarIxavAyA*
Mohan Erg Ram Inst ticket buy-CAUS.Pst
'Mohan made/had Ram buy the ticket.'

3.5 ARGA

This label is used for the externalmost causer or the only causer present in a causative construction. In Hindi, it is possible to add the causative morpheme *A* to a transitive verb such as *KA* 'eat', *sIKa* 'learn' to derive the ditransitive *KilA* 'feed', *siKA* 'teach' respectively, or to intransitive verbs such as *ro* 'cry', *gira* 'fall' to derive the transitive *rulA* 'make someone cry', *girA* 'make someone fall' respectively. Similarly the causative morpheme *vA* is added to a transitive or a ditransitive verb in order to get the meaning: cause someone to do X (for further details see 5 on causatives). For example, *beca* (sell) becomes *bikavA* (cause someone to sell something). In such constructions, the causer has a special status in the sentence as it denotes the person/entity who is causing the

agent to actually perform the action or who is causing the action on an entity be performed. The label ArgA is assigned to such an argument. For example, in the sentence below, *rAma* is annotated as ArgA:

(21) a. *[ARGA **rAma ne]** [sIwA se] [bacce ko] [KAnA] KilvAyA*
Ram Erg Sita Instr child to food feed.CAUS.Pst
'Ram made Sita feed the child food.'

b. *[ARGA **rAma ne]** [ARG0_MNS sIwA se] [ARG1 mohana ko] giravAyA*
Ram Erg Sita Instr Mohan Acc fall.CAUS.Pst
'Ram caused Sita to make Mohan fall.'

c. *[ARGA **rAma ne]** [sIwA se] [mohana ko] rulavAyA*
Ram Erg Sita Instr Mohan Acc cry.CAUS.Pst
'Ram caused Sita to make Mohan cry.'

3.6 ARGA_MNS

The label ArgA_MNS is used for the intermediate causers present in a causative construction. The causative morpheme *vA* allows one or more intermediate causers besides the externalmost causer in the sentence, For example, in the sentence 22 below, besides the externalmost causer *rAma ne*, an intermediate causer *sIwA se* is also present for the causative verb *KilvAyA* (for further details see 5 on causatives). This intermediate causer *sIwA se* gets the label ArgA_MNS.

(22) *[ARGA rAma ne] **[ARGA_MNS sIwA se]** [bacce ko] [KAnA] KilvAyA*
Ram Erg Sita Instr child to food feed.CAUS.Pst
'Ram made Sita feed the child food.'

The following example shows that it is also possible to have more than one intermediate causer within the same sentence. Note the use of label ArgA_MNS for the argument *sIwA se* as well as for the argument *rIwA xvArA* within the same sentence. As mentioned above, the argument labels for the core arguments are not repeated within the same sentence in PropBank, the label ArgA_MNS being the only exception. The label ArgA_MNS, despite being considered as a numbered argument label, can be used multiply within the same sentence, as there can be more than one intermediate causer present in the sentence.

(23) *[ARGA rAma ne] [ARGA_MNS **sIwA se**] [ARGA_MNS **rIwA xvArA**]*
Ram Erg Sita Inst Rita by
[bacce ko] [KAnA] KilvAyA
child to food feed-CAUS.Pst
'Ram made Sita, who via Rita made the child eat food.'

3.7 ARG1

In contrast to Arg0 which is the prototypical agent, Arg1 is the label that is assigned to arguments that are acted upon by another participant and are affected by the action described in the verb. That is, the label Arg1 is used for a prototypical patient or a theme. Syntactically, in case of a transitive verb construction, Arg1 is typically applied to the object in the sentence. Hence, in example 24 below, *rotI* 'bread', which is the object of the transitive verb, gets the label Arg1. Note it is the affected entity that is acted upon by another participant *rAma*, the agent of the action here.

(24) *[ARG0 rAma] [ARG1 **rotI**] KAwA hE*
Ram bread eat-Hab be.Pres
'Ram eats bread.'

The notion 'affected by the action described by the verb' is however quite broadly construed. For example, in sentence 25 below, *newA kI mAMga* '(the) leader's appeal' gets the Arg1 label even though 'an appeal' is not an entity that is acted upon by another participant and affected by the action expressed by the verb *suna* 'listen' in any concrete way.

(25) *[ARG0 praXAna maMwrI ne] [ARG1 **newA kI mAMga**] sunI*
prime minister Erg leader of demand hear-Pst
'(The) prime minister heard the leader's appeal.'

The Arg1 label typically appears with the 'affected' entity of verbs that are transitive or ditransitive. But it also appears with some intransitive verbs. Among the intransitive verbs, there is a special class of intransitives called the **unaccusative** verbs. These have a syntactic subject which does not have any agent like properties (contrast it with the case of an intransitive verb like *nAca* 'dance', which needs an agentive subject). Verbs such as *Kula* in 26 below do not require an agentive subject and hence the syntactic (non-agentive) subject *xarvAzA* 'door' gets the PB label Arg1. The Arg1 label here is thus assigned to the 'theme' argument, typically defined as the entity that is at rest or undergoing motion or change of state.

(26) *[ARG1 **xarvAzA**] KulA*
door open-Pst
'The door opened.'

Similarly, in the following sentence, we have the verb *jala* 'to burn', which is unaccusative, the syntactic subject *CAwroM ke hAWa* 'students' hands' is the affected entity by the event of the verb and hence, it gets the PB label Arg1.

(27) *[ARG1 **CAwroM ke hAWa**] jala gaye*
students of hands burn LV.Pst
'The students' hands got burnt.'

Arg1 is also found with verbs like *ho* 'be/become' where the subject is a theme argument with some attribute. The verb *ho* is also described as a linking verb [3], which establishes a relationship between the subject (the theme argument) and a complement. The complement could be an attribute like *acCA* 'good' or a location *Gara meM* 'at home' or an identity, e.g. 'doctor'.

In the following example, the subject rAma gets the PB label Arg1:

(28) a. *[ARG1 **rAma**] [acCA] hE*
Ram good be.Pres
'Ram is good.'

b. *[ARG1 **saDaka**] [cODI] huI*
road wide become-Pst
'(The) road became wide.'

3.8 ARG2

The Arg2 label is used to denote the beneficiary/ recepient of the action expressed by the verb. For example, in the sentence below, the argument *SyAma ko* is the beneficiary/ recepient of the verb *xe* 'give', hence it gets the PB label ARG2.

(29) *[ARG0 rAma ne] [ARG2 **SyAma ko**] [ARG1 kiwAba]* xI
Ram Erg Shyam to book give.Pst
'Ram gave Shyam a book.'

An Arg2 labelled argument is found in verbs that need a recipient argument e.g. *parosa* 'serve', *lA* 'bring'. *bawA* 'tell', etc.

Notice an Arg2 argument is different from an Arg0_GOL argument (men-

tioned in section 3.2 above) in that even though both these arguments are recepients/ beneficiaries of the event expressed by the verb, Arg2 is not an agent whereas Arg0_GOL is also the agent of the base verb besides being a recepient of the caused action.

3.9 ARG2_GOL

The Arg2_GOL is the destination or goal argument for the verb. It is useful to think of Arg2_GOL being used for an argument that represents the end point to a motion . Thus, it is used to annotate the destination arguments. For example, a verb like *pahuMca* 'reach' takes a destination argument, hence the label ARG2_GOL is used to annotate that argument.

(30) *[ARG0 cAcA] [ARG2_GOL **dillI]** pahuMca rahe hEM*
uncle Delhi reach Prog be.Pres
'Uncle is going to reach Delhi.'

lOta 'return', *Beja* 'send', *A* 'come', *jA* 'go', *cala* 'walk/go', *Gusa* 'push one's way into/ enter', *Coda* 'drop/ leave something/someone' are some of the verbs that take the Arg2_GOL label.

(31) *[ARG0 rAma] **[ARG2_GOL Gara]** gayA*
Ram home went
'Ram went home.'

The Arg2_GOL label is also used for directions that do not terminate in a goal (directed paths). See 32, where 'backwards' specifies the direction of motion although no endpoint/ goal is specified.

(32) *[ARG0 vo] [ARG2_GOL **pICe]** KisakA*
he backwards scooted.
'He scooted backwards.'

3.10 ARG2_SOU

This label is applied to the arguments that could be conceived as a source or the starting point of the event of the verb, these usually appear with the postposition *se* 'from'. See the following examples. In example 33 below, the starting point for the event is *Cawa se*, hence it is marked as Arg2_SOU. In example 34, again the argument marked as Arg2_SOU is the point where the comparison originates (i.e. the source of competition). In example 35, the Arg2_SOU is the source

from which the blessings are requested.

(33) *[ARG0 rAma]* ***[ARG2_SOU Cawa se]*** *[nIce] girA*
Ram roof from down fall.Pst
'Ram fell down from the roof.'

(34) *[ARG0 sIwArAma]* ***[ARG2_SOU dUsare prawiniXI se]*** *[Age] nikal*
Sitaram other candidate Abl ahead go.out
gayA hE
LV.Perf be.Pres
'Sitaram has moved ahead of the other candidate.'

(35) *[ARG0 mEMne]* ***[ARG2_SOU apne BagavAna se]*** *[ARG1 ASirvAxa]*
I-Erg my god Abl blessings
mAMgA
ask.Pst
'I asked for blessings from my god.'

This label can also be used as a starting point for any action/ motion. For example:

(36) *[ARG0 kiSorI]* ***[ARG2_SOU harixvAra se]*** *[ARG2_GOL xillI] AyI WI*
Kishori Haridvar from Delhi come.Pst be.Pst
'Kishori had come to Delhi from Haridvar.'

This label is also applied to those arguments for the verbs of transfer from which the transfer starts, e.g *Karixa* 'buy', *mAMga* 'demand'. For example:

(37) *[ARG0 KariSma ne]* ***[ARG2_SOU saMjay se]*** *[ARG1 xo lAKa rupaye]*
Karishma Erg Sanjay source two lakh rupees
mAMge We
demand.Pst be.Pst
'Karishma had demanded two lakh rupees from Sanjay.'

3.11 ARG2_ATTR

This label is applied to those arguments that describe some property of another argument, which is often (but not always) the subject. The most common example is the predicative element that occurs with *ho* 'be'. For example:

(38) *[ARG1 rAma]* ***[ARG2-ATTR buximAna]*** *hE*
Ram intelligent be.Pres
'Ram is intelligent.'

Arg2_ATTR can also be thought of as an identity marking argument with Arg1 in the case of a verb like *bana* 'become';

(39) *[ARG1 rAma]* ***[ARG2-ATTR xukAna kA mAlika]*** *ban gayA*
Ram shop Gen owner become go.Pst
'Ram became the owner of the shop.'

For verbs like *mAna* 'believe/consider' we give the following analysis:

(40) *[ARG0 ve loga]* *[ARG1 gAMXIjI ko]* ***[ARG2-ATTR bApU]*** *mAnawe hEM*
those people Gandhi-Hon Acc bapu consider-Hab be.Pres
'Those people consider Gandhiji as Bapu.'

In both the above cases, the highlighted argument is the ARG2-ATTR.

3.12 ARG2_LOC

The Arg2_LOC label is given to locations that do not involve a direction like the Arg2-SOU and Arg2-GOL arguments involve but these locations also seem to be essential to the understanding of the verb event. For example, see 41 below.

(41) a. *[ARG0 rAma]* ***[ARG2-LOC mere Gara mEM]*** *rahtA hE*
Ram my house in stay-Hab be.Pres
'Ram stays at my house.'

b. *[ARG1 billI]* ***[ARG2-LOC pedza mEM]*** *Pasa gayI hE*
cat tree Loc stuck go.Perf be.Pres
'The cat is stuck in the tree.'

Some examples of the verbs that require Arg2_LOC are *Guma* 'to roam', *Kisaka* 'to slip', *guzara* 'to pass by', *dUba* 'to drown' etc.

This label can also be used for abstract/metaphorical locations. For example, it can be used for time for the verb (*samaya*) *kAta* 'spend time' , see 42a below. It can be used for an abstract/metaphorical location, such as *xIvAra se* 'with the wall', *SikSA kSewra se* 'with education field', as in 42b and 42c below.

(42) a. *[ARG0 loga] [yahAM] [sadzakoM par] [ARG2_LOC **rAta**] kAta rahe hEM*
People here streets on night spend prog be.Pres
'People are spending the night on the street here.'

b. *[ARG1 yaha kursI] [ARG2_LOC **xIvAra se**] judI hE*
this chair wall with attach.Perf be.Pst
'This chair is attached to the wall.'

c. *[ARG2_LOC **SikSA kSewra se**] [ARG1 anek DarmasaMsWAeM] judI hEM*
education field with many religious.institutions connect.Perf be.Pres
'Many religious institutions are connected to the field of education.'

We also use Arg2_LOC for arguments that express accompaniment, notice the company can be taken as an abstract location. See 43 below.

(43) *[ARG0 rAma] [ARG2_LOC **SyAma ke sAWa**] [ARG2_GOL bAjAra] gayA*
Ram Shyam with market go.Pst
'Ram went to the bazar with Shyam'.

3.13 ARG3

The label Arg3 is used to annotate arguments that act as instruments in the sentence, i.e. they enable the action to take place. Usually, they are artifacts rather than persons. We can use the test *X kA upayoga* (use of X) to check whether an argument can be assigned Arg3. In the following sentence, *cAkU se* 'with a knife' gets Arg3:

(44) *[ARG0 rAma ne] [ARG3 **cAkU se**] [ARG1 Ama] kAtA*
Ram Erg knife Instr mango cut.Pst
'Ram cut the mango with a knife'

It is possible to rewrite the above sentence as in (41) below. Since the construction in (41) is possible, we assign *cAkU* the label Arg3. The verb *kAta* 'to cut' is an example of a verb that needs an instrument, and hence the Arg3 label.

(45) *[ARG0 rAma ne] [* ***cAkU kA upayoga karke]*** *[ARG1 Ama] kAtA*
Ram Erg knife of use having.done mango cut.Pst
'Ram, having used a knife, cut the mango.'

However the *X kA upayoga* test is quite stringent and includes only cases of prototypical instruments. There are other cases where the means whereby an action is performed can be labeled using the Arg3 label. An interesting case is the verb *Bara* 'fill' where the argument *pAnI* 'water' gets *se* case-marking (typically used with instruments) and invites a construal whereby the pot is filled BY MEANS of using water:

(46) *[ARG0 sIwA ne] [ARG3* ***pAnI se]*** *[ARG1 Gade ko] BarA*
Sita Erg water Instr pot Acc fill.Pst
'Sita filled the pot with water.'

However, one might argue that *pAnI* 'water' is a theme argument (it describes an entity in motion) and should get an Arg1 label. However, notice that Arg1 cannot be applied to multiple arguments within the same sentence, and since *Gade ko* is a theme argument (an entity affected by the event of the verb, it gets filled, notice the focus is on filling something up in this construction, hence *Gade ko* is assigned the Arg1 label), we cannot assign *pAnI se* also the same label. Hence for the -*se* marked argument, if it can be construed as an instrument, we use the label Arg3.[2] We treat an argument, that possibly could also act as an instrument, as Arg1 only in cases where -*se* case marking is not used, such as in 47 below. Notice, in contrast to 46 above, here the focus is on the entity *pAnI* that is moved and put in a container and hence can be taken as a theme and thus marked as Arg1.

[2]A note about some other -*se* marked arguments:

The -*se* marked argument for a verb such as *SAxI kara* 'marry' or *SAxI ho* 'marriage happen' get Arg1 label and the argument that would syntactically be treated as a subject for these verbs gets teh Arg0 label, following the English PropBank. Hence in a sentence such as 1 below, Ram gets Arg0 and Sita, Arg1.

(1) *[Arg0* ***rAma ne]*** *[Arg1* ***sIwA se]*** *SAxI kI*
Ram Erg Sita with marriage do.Pst
'Ram married Sita.'

Further, for verbs such as *gussA ho* 'be angry' as in 2 below, 'Sita se' will be treated as an adjunct rather than an argument.

(2) [Arg0 rAma] [ArgM siWa se] gussA hE
Ram Sita with angry be.Pres
'Ram is angry with Sita.'

(47) *[ARG0 sIwA ne]* ***[ARG1 pAnI]*** *[ARG2-LOC Gade meM]* *BarA*
Sita Erg water Instr pot Acc fill.Pst
'Sita filled water in the pot.'

As was mentioned above, we can have multiple instrument core arguments within the same sentence, hence we can use Arg3 repeatedly in the sentence. This is shown through the example in 48 below.

(48) *[ARG0 sIwA ne]* ***[ARG3 cammaca se]*** *[ARG1 ghade ko]* ***[ARG3 pAnI se]***
Sita Erg spoon Instr pot Acc water Instr
BarA
fill.Pst
'Sita filled the pot with water with a spoon.'

Chapter 4

Annotating of modifiers

The modifiers of a verb are annotated using the semantic role tags beginning with ArgM. The following types of modifiers are being used in PropBank:

Modifiers	**Directionals**
1. DIR	Directionals
2. LOC	Locatives
3. MNR	Manner
4. MNS	Means, path
5. GOL	Goal
6. EXT	Extent
7. TMP	Temporal
8. REC	Reciprocals
9. PRD	Secondary Predication
10. PRP	Purpose
11. CAU	Cause
12. DIS	Discourse
13. ADV	Adverbials
14. DSP	Direct speech
15. MOD	Modals
16. LV	Light verbs
17. NEG	Negation

4.1 ARGM_DIR

Directional modifiers show motion along some path. Both "source" and "goal" are grouped under "direction." On the other hand, if there is no clear path being followed a "location" marker should be used instead. Thus, *Gara kI* or *BAga* 'run towards home' involves a directional, but *KetoM meM Pira* 'roam in the fields' involves a location.

(49) *mohana* ***uDara xORA***.
Mohan there ran.
'Mohan ran there'

4.2 ARGM_LOC

Locative modifiers indicate where some action takes place. The notion of a locative is not restricted to physical locations, but abstract locations are being marked as LOC as well:

(50) *rAma ne* ***bAzAra mEM*** *ravI ko xeKA.*
Ram erg market in ravi dat saw
'Raam saw Ravi in the market'

(51) ***apne BASana mEM***, *mohana ne newa kI KUba wArIfa kI.*
self lecture in Mohan erg leader gen much praise did.
'In his lecture, Mohan praised the leader highly'

4.3 ARGM_MNR

Manner adverbs specify how an action is performed. For example, *duusroN ke saath kaam acchaa kartaa hae* 'he works well with others' is a manner. Manner tags should be used when an adverb is an answer to a question starting with *kaise* 'how'?

(52) *vaha bahuwa* ***wejZa*** *bolawA hE*
he very fast talk.Imp be
'He talks very fast'

ArgM-MNR is also used

4.4 ARGM_MNS

This is like an ARGM equivalent to the instrumental ARG3. It functions to accommodate cases such as the following:

(53) *yanA ko wuraMwa [ARGM-MEANS ek kAra se] aspawAla le jAyA*
Yanaa dat immediately one car with hospital take go
gayA
AUX
'Yanaa was immediately taken to the hospital by car'

(54) *rAja [ARGM-MEANS pArtI xvArA] BJP se sambaMWa woda lene ke*
Raj party by BJP with connection break LV gen
pakSa mEM hE
side in is
'Raj is in favour of breaking connection with the BJP via the party.

4.5 ARGM_GOL

This tag is for the goal of the action of the verb. This includes beneciaries and the nal destination of some motion verbs. 'Goal' would be used for modiers that indicate that the action of the verb was done for someone or something, or on their behalf:

(55) *mohana ne ravI ke liye cAy banAI.*
Mohan erg Ravi for tea made.
'Mohan made (some) tea for Ravi'

4.6 ARGM_EXT

ArgM-EXT indicates the amount of change occurring from an action, and are used mostly for (a) numerical adjuncts like *aaluu kii kiimat **10 prawiAaw** baRha gaI hE* 'the price of potatoes has increased **by 10%**' (b) quantifiers such as *bahuwa* 'a lot' and (c) comparatives such as '(he worked) **more than she did**':

(56) 1998 *se* 2008 *wak usne mumbai mEM kAma kiyA*
1998 from 2008 until he-erg Mumbai in work did
'He worked in Mumbai from 1998 to 2008'

In comparative constructions such as the following:

(57) *mohana ne [ArgM-Ext **sIwA se**] weza BAgA.*
Mohan erg Sita abl fast ran.
'Mohan ran faster than Sita did'

(58) *rAWA **[mIrA kI]** wulanA mEM [Argm-mnr **aXika**] suMdar hE*
Radha Mira gen comparison in more beautiful is
'Radha is more beautiful than Radha'

4.7 ARGM_TMP

Temporal ArgMs show when an action took place, such as 1987 *mein* 'in 1987', *pichle hafte* 'last week', *turant* 'immediately'. Also included in this category are adverbs of frequency (eg. *aksar* 'often', *hameshaa* 'always', *kabhii-kabhii* 'sometimes' (with the exception of *kabhii nahii* 'never', see NEG below), adverbs of duration (*ek saal keliye/ek saal mein* 'for a year/in an year'), order (e.g. *pehlaa* 'first'), and repetition (eg. *baar-baar*, *phir se* 'again'):

(59) ***kala*** *paanii barasaa thaa*
yesterday water rained past
'it rained yesterday'

4.8 ARGM_REC

These include reflexives and reciprocals such as *khud*, *apne aap* 'him/her self', *saath(-saath)/ek saath* 'together', *ek duusre* 'each other', *dono* 'both', which refer back to one of the other arguments. Often, these arguments serve as the Arg 1 of the relation. In these cases, the argument should be annotated as the numbered argument as opposed to the reciprocal modier.

(60) *Mohan aur siitaa ne ek saath kaam kiyaa.*
Mohan and Siitaa erg together work did
'Mohan and siitaa worked together'

(61) *mohan ne apne aap kaam kiyaa.*
Mohan erg by-himelf work did
'Mohan worked by himself'

Note that the reciprocal is **not** an ARGM_REC in the following sentence, since they function as an argument.

(62) *Mohan aur raam ne ek duusre ko dekhaa.*
Mohan and Raam erg one another Dat looked.
'Mohan and Ram looked at each other'

4.9 RGM_PRD: markers of secondary predication (PRD)

These are used to show that an adjunct of a predicate is in itself capable of carrying some predicate structure. In Hindi, secondary predication does not appear to be as productive as it is in English. A typical example might include:

(63) *bacce ne seb ko* ***choTe-choTe TukRon mein*** *kaaTaa*
child erg apple dat small-small pieces in cut.
'The child cut the apple **in little pieces**'

4.10 ARGM_PRP

Purpose clauses are used to show the motivation for some action. Clauses beginning with "in order to" are canonical purpose clauses.

(64) ***maine dillii jaane ke liye Tikata khariida.***
I-erg Delhi go-inf for ticket bought
'I bought tickets in order to go to Delhi.'

4.11 ARGM_CAU

Similar to "Purpose clauses", these indicate the reason for an action. Clauses beginning with "because" or "as a result of" are canonical cause clauses. Also questions starting with 'why':

(65) ***[muKya gavAha ke is bayAna se***$_{\text{ARGM-CAU}}$***]*** *sarabjIwa kI*
primary witness gen this testimony instr Sarabjit gen
rihAI kI ASA badha gayI hE
acquittal gen hope increase go be
'By the primary witness' testimony, the hope for Sarabjit's acquittal has increased'

(66) *maine* ***mohana kii bhuul kii vajaha se*** *kitaab kho dii.*
I-erg Mohan gen mistake gen because inst book lose LV.
'I lost the book because of Mohan's mistake'

4.12 ARGM_DIS

These are markers which connect a sentence to a preceding sentence. Examples of discourse markers are: *lekin/parantu* 'but', *aur* 'and', *iske alaavaa* 'instead', *yaa* 'or', *yaanii* 'namely', etc. Note that conjunctions corresponding to 'but', 'or', 'and' in Hindi are only marked in the beginning of the sentence. Do not mark 'and', 'or', 'but', when they connect two clauses in the same sentence.

(67) ***lekin*** *kal hamaare ghar kaun aane waalaa hae?*
But tomorrow our house who come-inf one is?
'But who is going to come to our house tomorrow?'

(68) ***isake alaava,*** *maovaadi ke raambacana yaadav ko giraftaar*
this-gen moreover, Maoist gen Rambachan Yadav dat arrest
kara liyaa gayaa
do LV AUX.
'In addition to this/Moreover, Maoists' Rambachan Yadav was arrested'

Another type of discourse markers includes vocatives and interjections:

(69) ***maa,*** *mujhe kal dilli jaana hai*
mother, I-dat tomorrow Delhi go-inf be
'Mother, I want to go to Delhi tomorrow'

(70) ***he bhagwaan,*** *mujhe maafii do.*
Oh God, I-dat forgiveness give.Imp.
'Oh God, forgive me'

4.13 ARGM_ADV (Adverbials)

These are used for syntactic elements which clearly modify the event structure of the verb in question, but which do not fall under any of the headings above.

Temporally related (modifiers of events): ***mohan kii pratiiskhaa mein*** *siitaa wahiin khaRii rahii* 'Sita remained standing there, **in anticipation of Mohan**'

Intensional (modifiers of propositions): *shaayad* 'probably, possibly' Focus-sensitive: *sirf/keval* 'only', *bhii* 'also, even'

Sentential (evaluative, attitudinal, viewpoint, performatives): *soubhaagya se* 'fortunately', *asal mein* 'in actuality', *kaanoon ke anusaar* 'legally', *ke baavjuud* 'despite', *agar* 'if', etc.

Conditional clauses like *agar-to*, where the dependent clause gets the Adv label

As opposed to ArgM-MNR, which modify the verb, ARGM-ADVs usually modify the entire sentence.

(71) ***shaayad*** *siitaa khaanaa khaayegii*
maybe Sita food will eat
'Maybe Sita will eat food'

(72) ***keval*** *do laRkO ne saaraa kaam kiyaa.*
Only two boys erg all work did
'Only two boys did all the work'

4.14 ARGM_MOD (modals)

Modal constructions in Hindi convey notions such as ability, desire, obligation, permission, etc. In Pbank, we will annotate the following cases using the ARGM-Mod label.

(73) *mohan kaam kar* ***sakegaa.***
Mohan work do able.fut
'Mohan will be able to do the work'

(74) *mohan kaam kar* ***paaegaa.***
Mohan work do able.fut
'Mohan will be able to do the work'

(75) *mohan ko ghar jaanaa* ***caahiye.***
Mohan dat home go-inf ought
'Mohan ought to go home'

(76) *mohan ko kaam karnaa* ***padZaa.***
Mohan dat work do-inf had
'Mohan had to work'

(77) *mohan ko kulfii khaanii* ***hai.***
Mohan dat icecream eat-inf is
'Mohan has/wants to eat icecream'

4.15 ARGM_VLV (light verbs)

Light verbs are semantically bleached verbs that combine with the bare form of the verb (e.g. *kar* 'do', *ro* 'cry', *so* 'sleep') to convey different meanings. Typically these meanings are aspectual, but may also involve suddenness, inception, surprise, etc. In DS, all these bare forms get the label VAUX. We can automatically insert ARGM_VLV because these verbs form a closed class of 15-20 verb roots.

(78) *mohan ro* ***paRaa***
Mohan cry lie
'Mohan burst out crying'

(79) *siitaa saaraa khaanaa khaa* ***gayii***
siitaa all food eat went
'Siitaa ate up all the food'.

(80) *siitaa saaraa khaanaa khaa* ***cukii*** *hae*
siitaa all food eat complete.prf is
'Siitaa has eaten all the food'.

(81) *saritaa ne saRii khariid* ***lii.***
Sarita erg sarii buy took
'Saritaa has bought the sarii'

(82) *bacce ne mAA ko gend de* ***diyaa.***
Boy erg mother dat ball give gave.
'(The) boy has given his mother the ball'

Typically, the light verb is not negated independently of the main verb, nor can they be scrambled or have an adverbial inserted between the main verb and the light verb. Hence they will not be annotated as independent verbs.

The light verb 'de' can combine with other verbs in the infinitival form i.e ending with *–ne* e.g. *sone* 'to sleep' or *rone* 'to cry'. In these cases, we will adopt a slightly different approach. For example, if we have the case of 'sone diyaa' ; to let (someone) sleep, we will annotate 'diyaa' as the rel and not 'sone'. Instead of annotating the infinitival verb which has the tag VM, the PropBank 'rel' label will point to 'diyaa', which has the label VAUX in DS.

(83) *raam ne mohan ko duudh piine* ***diyaa.***
Raam erg Mohan dat milk drink-inf gave.
'Ram allowed Mohan to drink millk'

Also, the change will be reflected in the framefile for 'denaa'. It will now have a special sense for the VLV light verb sense. The main reason for separating out the case of 'denaa' is because it licenses an agentive argument 'raam'.

Note, that this applies **only** to those cases where the light verb 'de' appears with the infinitival (or 'ne') form of the preceding verb. For instance cases like 'dikhaai denaa' 'to be seen/ appear' will get the VLV automatic label as the other cases above.

Also discuss what we do with *vaalaa*, e.g. in the sentence *mohan kal kaam karne vaalaa hae* 'mohan is going to work tomorrow']

4.16 ARGM_NEG

This tag is used for elements such as *nahii* 'not', *kabhii nahii* 'never, *naa* 'not' and other markers of negative sentences. Negation is an important notion for Propbank annotation; therefore, all markers which indicate negation should be marked as NEG. For example, when annotating adverbials like *kabhii nahii* 'never,, which could be marked as either TMP or NEG, the NEG tag should be used.

(84) *laRkii ghar* ***nahII*** *gayii*
girl home not went
'(The) girl did not go home'

(85) *laRkii kaam* ***kabhii nahII*** *karegii.*
Girl work never do.fut
'(The) girl will never do work'.

Be careful to distinguish these from conjunctions such as *naa hii* 'not only,' which does not actually indicate that the verb is negative and should not be annotated because it is a conjunction.

Chapter 5

Causatives

Hindi it is possible to add the causative morpheme –A to an unaccusative or an unergative verb in order to get the meaning: to cause someone to do X.

It is also possible to add another causative morpheme –vA to a transitive or a ditransitive verb in order to get the meaning or indirect causation: to cause A to cause B to do X.

The treatment of causatives in Hindi PropBank is quite uniform in terms of the assignment of the core semantic roles to the arguments of verbs with the –A or –vA. The core semantic roles that we see in the causative constructions are typically one or more of the following: ArgA, ArgA_MNS, Arg0, Arg0_MNS, Arg0_GOL, Arg1 and Arg2. Let us talk briefly about these semantic roles and then we will see the causativization process in Hindi for the unaccusative verbs, unergative verbs, transitive verbs and the ditransitive verbs respectively where we show the use of these semantic roles.

There are two types of causer arguments ArgA and ArgA_MNS in our analysis:

(a) The externalmost causer argument in case of multiple causers or the only causer argument present in the sentence is annotated with the **ArgA** label (example 86 It can be the direct causer in case only one causer argument is present or it can be the indirect causer (the externalmost causer) in case multiple causer arguments are present in the sentence. It may appear with case marker "ne" or zero case marking, although it should be noted that each argument that appears with ne or zero-case marking is not necessarily an ArgA, the argument has to be a causer for it to be an ArgA.

(b) Any intermediate causers are annotated with the **ArgA_MNS** label example 86. Note that ArgA_MNS can only be used in presence of ArgA (i.e. the leftmost/ externalmost causer argument will always be ArgA) . Hence we use the label ArgA_MNS for all the causers starting from the direct causer, moving leftwards in the sentence until we reach the externalmost causer (which is ArgA). Thus we see that ARGA_MNS is one of the core arguments that can be repeated (i.e. multiple arguments can get it within the same sentence). In case of multiple ARGA_MNS arguments, the innermost causer (the rightmost)

is the direct causer, the others are indirect causers. The logic behind employing the function tag MNS (means) is that the externalmost causer is using these intermediate causers as a means to get some action performed. These arguments are typically marked with either the postposition *–se* or a postpositional phrase *–(ke) xvArA*.

(86) *Sita ne* $_{ARGA}$*] [mohan se*$_{ArgA\_MNS}$*] [ram ko*$_{Arg0}$*] rul-vAyA*
Sita erg mohan instr ram acc make-cry-CAUS
'Sita had/made Mohan make Ram cry'

For the remaining arguments, we use numbered argument labels. Besides the normal labels Arg0, Arg1 and Arg2 used elsewhere too, we also use more specified labels, namely Arg0_MNS and Arg0_GOL for core numbered arguments in causative constructions. These are described below.

The agent (doer) of an action, such as the agent of a verb ronA 'to cry' is marked **Arg0** as is expected. The label **Arg0_MNS** (87a-87b) is also used to annotate the doer of an action but when the doer is caused by a causer to perform the action and it is not the obvious recipient or beneficiary of the action. Please also note that this label is applied only when the the action is performed on an Arg1 argument, thus basically this label is used only when the event for the action is represented by a base unaccusative verb or a base transitive verb. The Arg0_MNS can also get postposition "se" or "xvArA" in some cases just like an ArgA_MNS can get these postpositions. Hence, note the postpositions "se" or "xvArA" are not limited to the argument marked as ArgA_MNS. In fact, there is a shared property between these two arguments (Arg0_MNS and ArgA_MNS) that these postpositions express, these postpositions are used with an argument that acts as a means to get some action done. Note Arg0_MNS is used as a means by its causer to achieve some result just as ArgA_MNS is used as a means by its causer to get some results, viz to get these arguments to perform some action. (this is observed with verbs such as *giravA* 'cause (someone) to make (something) fall', *KulavA* 'cause (someone) to make (something) open', *tuRavA* 'cause (someone) to break (something)', *bikavA* 'cause to sell' and so on.) When the base verb is an unergative verb, in that case, even when the causers are present, the doer of the action gets Arg0 label, not Arg0_MNS label. This is so because for an argument to be an Arg0_MNS, the caused argument has to perform an action on an Arg1 argument, but for base unergative verbs, that condition is not true.

(87) a. *[Mohan ne$_{ARGA}$] [raam se$_{Arg0_MNS}$] [peRa$_{Arg1}$] kat-vAyA*
Mohan erg raam inst tree cut-CAUS
'Mohan made/had Raam cut the tree'

b. *[John ne$_{ARGA}$] [sItA se$_{Arg0_MNS}$] [ticket$_{Arg1}$] KarIdvAyA*
Mohan erg raam inst tree cut-CAUS
'John made/had Sita buy the ticket.'

If there is a doer of an action (represented by the noncaused-inner verb, e.g. KA in example 88 below) which also happens to be the recipient or beneficiary of the caused action (KilavA in example 88 below), **Arg0_GOL** label is assigned to it rather than Arg0 or Arg0_MNS. When the beneficiary of an action is not the doer of the action too, then it is simply marked as **ARG2**.

The 'affected' or theme argument gets the Arg1 label as is expected (example 88).

(88) *[rAma ne$_{ARGA}$] [sIwA se$_{ArgA_MNS}$] [bacce ko$_{Arg0_GOL}$] [KanA$_{Arg1}$] Kil-vAyA*
Raam erg Sita instr child dat food feed-CAUS

Ram had/made Sita feed the child food'

Let's go through the causativization process in Hindi now to see the use of these argument labels. Let us look at how the transitivization and causativization of the base unaccusative verbs adds the arguments in the sentence.

Besides, the label Arg0 is also used for the agentive participants of those monotranistive verbs or ditransitive verbs which are themselves derived from other intransitive or monotransitive verbs respectively, using the transitivizing *–A* morpheme in Hindi. For example, *-A* morpheme can be added to the intransitive verb *ro* 'cry' to derive the monotransitive verb *rulA* 'make someone cry'. Note when the *-A* morpheme is added, certain other internal changes may also take place in the root morpheme (e.g. here the change in vowel o to u, and insertion of the consonant l). Similarly, *-A* morpheme can also be added to the monotransitive verb *KA* 'eat' to derive the ditransitive verb *KilA* 'make someone eat/ feed someone', another example is derivation of the ditransitive verb *siKA* 'make someone learn/ teach someone' from the monotransitive verb *sIKa* 'learn'. Again notice the internal changes in the root morphemes besides the addition of *-A* here. The agentive participant of such derived transitive verbs (monotransitive or ditransitive verbs) also receive the Arg0 label. Some examples are shown below.

(89) a. *[ARG0 **mohana ne]** [rAma ko] rulAyA*
Mohan erg rama Dat cry.made
'Mohan made Ram cry.'

b. *[ARG0 **mohana ne]** [rAma ko] [KIra] KilAyI*
Mohan Erg Ram Dat ricepudding fed
'Mohan fed Ram the rice pudding.'

c. *[ARG0 **mohana ne]** bacce ko siKAyA*
Mohan erg child dat taught
'Mohan taught the child.'

5.1 Unnacusative

(90) *[ARG1 mom] pighal-ii*
wax melted
The wax melted.

5.1.1 Unaccusative → Transitive

(91) *ARG0 siita-ne ARG1 mom pighaal-ii*
Sita wax melted
Sita melted the wax.

Here the Arg1 argument maintains its semantic role in the transitive sentence. An agent argument Arg0 is added.

5.1.2 Unaccusative → Causative

(92) *ARG_A John-ne ARG0_MNS siita-se ARG1 mom pighal-vaayii*
John Sita-from/by wax made to melt
John made Sita melt the wax.

Since only one causer argument is present, it gets the label ArgA. Since the doer of the action is caused by ArgA to perform the action, it gets the label Arg0_MNS.

5.1.3 Unaccusative → Causative

(93) ARG_A *Bill-ne* ARG_A_MNS *John-dwaaraa* ARG0_MNS *siita-se* ARG1 *mom*
Bill John-by Sita-from/by wax
pighal-vaayii
made to melt
Bill made John make Sita melt the wax.

Since the intermediate causers get labels ArgA_MNS, the argument John-dwaaraa gets ArgA_MNS label, while the externalmost causer argument Bill-ne gets the label ArgA.

Now let us look at the transitivization and causativization of an unergative verb.

5.2 Unergative

(94) ARG0 *raam ro-yaa*
Ram cried
Ram cried.

5.2.1 Unergative → Transitive

(95) ARG_A *siita-ne* ARG0 *raam-ko rul-aayaa*
Sita Ram made-cry
Sita made Ram cry.

Here the doer of the unergative verb gets the Arg0 label. The only causer present gets the ArgA label.

5.2.2 Unergative → Causative

(96) ARG_A *John-ne* ARG_A_MNS *siita-se* ARG0 *raam-ko rul-vaayaa*
John Sita-from/by Ram made-cry
John made Sita make Ram cry.

The intermediate causer gets the ArgA_MNS label, while the externalmost causer gets the label ArgA.

Now let us look at the causativization of a transitive verb.

5.3 Transitive

(97) ARG0 *bacce-ne* ARG1 *khaanaa khaa-yaa*
child food ate
The kid ate the food.

5.3.1 Transitive → Ditransitive

(98) ARG_A *siitaa-ne* ARG0_GOL *bacce-ko* ARG1 *khaanaa khil-aa-yaa*
Sita child-to food made-eat
Sita made the kid eat food.

The only causer present gets the label ArgA. The doer of the action also happens to be the receiver of the caused action, hence it gets the label Arg0_GOL.

5.3.2 Transitive → Causative

(99) ARG_A *John-ne* ARG_A_MNS *siita-se* ARG0_GOL *bacce-ko* ARG1 *khaanaa*
John Sita-from/by child-to food
khil-vaa-yaa
made-eat
John made Sita make the kid eat food.

The externalmost causer argument gets the label ArgA, the intermediate causer argument gets the label ArgA_MNS.

Another type of transitive verb alternation: not the transitive- ditransitive alternation, but just the transitive- causative alternation

5.4 Transitive

(100) ARG0 *siitaa-ne* ARG1 *ticket khariid-aa*
Sita ticket bought
Sita bought the ticket.

5.4.1 Transitive → Causative

(101) ARG_A *John-ne* ARG0_MNS *siitaa-se* ARG1 *ticket khariid-vaayaa*
John Sita-from/by ticket make-buy
John made Sita buy the ticket.

Here also the only causer argument present gets the label ArgA. The doer of the action gets the role Arg0_MNS and not Arg0_GOL as it is not the receiver of the caused action. Also it is not Arg0 but Arg0_MNS since the action affects an Arg1 argument.

5.4.2 Transitive → Causative

(102) ARG_A *Bill-ne* ARG_A_MNS*J ohn-dwaaraa* ARG0_MNS *siitaa-se* ARG1 *ticket*
Bill John-by Sita-from/by ticket
khariid-vaayaa
make-buy
Bill made John make Sita buy the ticket.

The externalmost causer argument gets the label ArgA while the intermediate causer argument gets the label ArgA_MNS.

Another type of transitive verb alternation: the transitive- ditransitive; transitive- causative; transitive- causative ditransitive alternation

5.5 Transitive

(103) ARG0 *siitaa-ne* ARG1 *ticket khariid-aa*
Sita ticket bought
Sita bought the ticket.

5.5.1 Transitive → Ditransitive

(104) ARG0 *siitaa-ne* ARG2 *Bill-ke liye* ARG1 *ticket khariid-aa*
Sita Bill-for ticket buy
Sita bought the ticket for Bill.

5.5.2 Transitive → Causative

(105) ARG_A *John-ne* ARG_A_MNS *Pat-dwaaraa* ARG0_MNS *Siita-se* ARG1 *ticket*
John Pat-by Sita-from/by ticket
khariid-vaayaa
make-buy
John made Pat make Sita buy the ticket.

Again the ArgA is assigned to the externalmost causer, ArgA_MNS to the intermediate causer, and Arg0_MNS to the caused doer who is not the recepient

of the caused action.

5.5.3 Transitive → Ditransitive Causative

(106) ARG_A *John-ne* ARG_A_MNS *Pat-dwaaraa* ARG0_MNS *Siita-se*
John Pat-by Sita-from/by
ARG2 *Bill-ke liye/ Bill-ko* ARG1 *ticket khariid-vaayaa*
For Bill/ to Bill ticket make-buy
John made Pat make Sita buy the ticket for Bill.

Here also the ArgA is assigned to the externalmost causer, ArgA_MNS to the intermediate causer, and Arg0_MNS to the caused doer who is not the recepient of the caused action.

5.6 Ditransitive

(107) ARG0 *Mohan-ne* ARG2 *John-ko* ARG1 *kitaab dii*
Mohan John-to book gave
Mohan gave John the book.

5.6.1 Ditransitive → Causative

(108) ARG_A *raam-ne* ARG0_MNS *Mohan-dwaaraa* ARG2 *John-ko* ARG1 *kitaab*
Ram Mohan-by John-to book
dil-vaa-yii
make-give
Ram made Mohan give John the book.

Here again Again the ArgA is assigned to the only causer, and Arg0_MNS to the caused doer who is not the recepient of the caused action.

5.6.2 Ditransitive → Causative

(109) ARG_A *bill-ne* ARG_A_MNS *raam-dwaaraa* ARG0_MNS *Mohan-se* ARG2 *John-ko*
Bill Ram-by Mohan-from/by John-to
ARG1 *kitaab dil-vaa-yii*
book make-give
Bill made Ram make Mohan give John the book.

Here also the ArgA is assigned to the externalmost causer, ArgA_MNS to the intermediate causer, and Arg0_MNS to the caused doer who is not the recipient

of the caused action.

Chapter 6

Unaccusatives and unergatives

6.1 Decision process

Expected answers:

(a) UNACC: NO

(b) UNERG: YES

(c) cognate object: NO

(110) *us-ne laRaaii laRii*
he-Erg fight_N fight_V-Perf
'He fought the fight.'

(d) ergative case: NO

(111) *us-ne chiiNkaa*
he-Erg sneeze-Perf
'He sneezed.'

(e) transitivity alternation: if YES, apply F.
khul-naa vs *khol-naa*
'to be opened' vs 'to open'

(f) the ones undergoing alternation entail animacy of single argument of the verb: NO

khulnaa does not entail animate argument(subject): so UNACC. *haNsanaa* entails animate argument: so not UNACC

(g) impersonal passives (only works for intrans verbs with human animate subjects): NO

(112) *kuch naac li-yaa jaae*
some dance_V take-Perf Pass
'Should we dance a little?'

(h) past participial relative possible with passive syntax- NO (UNACC allows use of active syntax "huaa")

(113) *gir-ii hu-ii/*ga-ii kitaab*
fall-Perf be_Active-Perf/Pass-Perf book
'The fallen book (not good in English but is good in Hindi)'

(i) inanimate nonspecific subject cannot appear without overt Genitive: NO (UNACC allows it to appear without overt Genitive)
[akhbaar (kaa) vakt par] aanaa zaruurii hai

1 round with intuitions. 2nd round- search on google
A description of the diagnostics:

6.2 Ergative Subjects

Unergatives may sometimes allow ergative subjects esp. when paired with the right adverbials and compound verbs.

(114) *laajvanti-ne man bhar naac liyaa thaa*

*Ram-ne aa-yaa/pit-aa

6.3 Cognate objects

Unergatives may also sometimes allow cognate objects.

(115) *purush kataaar ghumma ghumaa-kar laraai-ke naac naac-aa*
men knife twist twist-CP fighting-Gen dance dance-Pfv
kar-te the
do-Hab.MPI be.Pst.MPI
'Twirling their knives, the men used to dance fighting dances.'

*Ram aa-naa aa-yaa/pit-naa pit-aa

6.4 Impersonal passive

Unergatives allow for the impersonal passive (in intransitives where the implicit agent is [+human]/animate). So this test is inapplicable to the vast majority of unaccusatives whose sole argument is inanimate.

(116) *kuch jhuum liya jaa-e, kuch naac liyaa jaa-e*
some 'move' TAKE.Pfv go-Sbjv some dance TAKE.Pfv go-Sbjv

*kuch aa-yaa jaa-e/*kuch pit-aa jaa-e

6.5 Past participial relatives

Unergatives lack an internal argument and so they should be ungrammatical with past participial relatives.

(117) a. *Dilli-se kal aa-ye (hue/*gaye)*
Delhi-From yesterday come-Pfv.MPI be.PfV.MPI/*GO.Pfv.MPI
chaatra)
student

b. **kal duar-aa/naac-aa*
yesterday run-Pfv.MSg/dance-Pfv.MSg

c. *(huaa/*gayaa) chaatra*
be.PfV.MSg/GO.Pfv.MSG student

Confound: it is possible that this test targets the achievement/accomplishment vs. activity distinction and that ergatives typically pattern with activities and unaccusatives with achievement/accomplishment. In some cases, it might be possible to create an achievement/ accomplishment out of an unergative by adding suitable adverbs. For the test to go through therefore, we should test the relatives without adding adverbs that would lead to the creation of an achievement/accomplishment.

6.6 Inabilitatives

Unaccusatives enter the inabilitative with active syntax; unergatives cannot enter the inabilitative with active syntax:

(118) *Ravi-se darwaazaa nahi: khul-aa*

*Ravi-se Ramma nahi: has-ii

The tests in this section apply to the class of verbs that undergo the transitivity alternation. They do not apply to the motion verb subclass of unaccusatives; the motion verbs do not participate in the transitivity alternation e.g. **aa, jaa, pah~uc** 'come, go, arrive'.

6.7 Use of an instrumental phrase to introduce an agent (who acts inadvertently)

The tests in this section also apply to the class of verbs that undergo the transitivity alternation.

(119) a. *Rina-se gamlaa tuut gayaa*

b. *Rina-se gamlaa kaha: tuut-aa*

*Rina-se Ramesh naac gayaa
*Rina-se Ramesh kaha: naac-aa

6.8 Compound verb selection

The unaccusative compound verb jaa 'go' appears most naturally with unaccusatives; unergatives seem unhappy with jaa 'go'.

(120) *gamlaa gir gayaa*
Ravi (apne-aap) nahaa le-gaa/???jaa-egaa

Sometimes unergatives can combine with **jaa** 'go' as a light verb but this leads to a change in meaning, i.e. the verb is not interpreted agentively any more:

(121) *doosraa din khuun-se mahaa gayaa*
aasmaan roshnii-me nahaa gayaa

6.9 Unmarked subjects of non-finite clauses

Non-finite clauses in Hindi-Urdu typically do not permit overt unmarked subjects.

(122) *[Atif-kaa/*Atif mehnat kar-naa] zaruurii hai*

But inanimate (and non-specific) subjects of unaccusative predicates can appear without an overt genitive.

(123) *[akhbaar(-kaa) waqt-pe aa-naa] zaruurii hai*

The inanimacy restriction means that we can't really use this test freely with unergatives as most of them only have animate subjects.

Chapter 7

Complex predicates

The annotation of complex predicates will be carried out separately from simple verbs. In the dependency Treebank, the nominal hosts of complex predicates are markd with a special label 'pof'. These cases are automatically annotated with the label ARGM-PRX. During the annotation of simple verbs, we will not look at those cases of the verb where ARGM-PRX exists.

We will then have to create frames for the nominals that take the POF label.

All the complex predicates will then be annotated using nominal verb frames, but keeping in view the common argument structure (in accordance with English and Arabic).

In order to create nominal verb frames, we will follow a slightly different procedure, where each roleset will represent a particular combination of a noun with a light verb. We know that the argument structure of the sentence changes with a change in the light verb. For example, *Ram ne paese chorii kiye* 'Ram stole the money' differs from *paese chorii hue* 'The money got stolen'. Hence, if *chorii* 'theft' occurs with two light verbs *kar* 'do' and *ho* 'be', the frameset for *chorii* will consist of two rolsets: *chorii kar* 'steal' and *chorii ho* 'get stolen'.

As complex predicates have already been identified in the Treebank with the 'pof' label, we will annotate directly using this label for reference and the noun frame files as a guideline for annotation. However, at this stage, we will also explicitly mark those 'pof' marked cases that do not appear to be complex predicates as errors. They will later be annotated as simple verbs.

7.1 Complex Predicates Diagnostics

Does not allow for accusative marker
**us-ne mere kaam me us rucii-ko li-yaa jo*

Does not allow movement of the noun
**bhaag us-ne pratiyogita meN li-aa*

Constituent response test
raam-ne kyaa ki-yaa? Response should be including the N and V

co-ordination test
**us-ne bacce-ko [[kshamaa] aur vidaa]] k-ii*

Adding modifier to NP
us-ne pratiyogita meN bhaag li-aa

Wh-words and pronouns don't occur with second reading**
anil kyaa bec-egaa? OR anil yah bec-egaa.

Gapping not possible with CP meaning
anil ghodei khariid-taa hai aur raam —- bec-taa hai

N+V has a generic meaning for N
anil kitaabeM beca-taa hai

salient or name-worthy predicate
ghaas kaat-na VS ghaas dekh-naa

Is Agentive -vaalaa possible in conjunction with overt ko etc (with second reading**)
ghode-ko bec-ne vaalaa

Relativization:Incorporated nominals cannot be relativized
**vah bharosaa [jo raam-ne mohan par ki-yaa] jhhotha thaa*

Ellipses: In answer to yes-no questions in hindi, the predicate of a clause can stand for the entire clause.
*raam-ne mohan par bharosaa ki-yaa? *haa, ki-yaa. haa, bharosa ki-yaa*

ApnA -ApnA test
**apnii apnii corii ho ga-ii but sab-ne apnaa apnaa snaan ki-yaa*

Chapter 8

Empty categories

Hindi-Urdu is a language that allows the speaker to freely omit arguments of the verb in discourse-pragmatically licensed contexts. For instance, one can say 'Ram drank liquor', but if 'Ram' has been talked about before, or is otherwise salient in the context, one could say 'drank liquor' without overtly mentioning the subject, 'Ram'. In a corpus, one comes across many such sentences where the arguments of the verb are missing although they can be retrieved from the context.

Although PropBank annotation does not typically involve adding empty arguments to syntactic trees, in the case of Hindi-Urdu we have taken a somewhat different approach.

- We insert empty categories corresponding to the core arguments of the verb including **subjects, direct and indirect objects**. We mark 4 empty categories in all: pro, PRO, GAP and RELPRO.
- During the annotation process, we mark semantic roles and carry out empty argument insertion simultaneously. In this respect, we make use of the verb frame file to determine the number and semantic roles of the arguments that are not overtly realized (and that must be inserted).

For instance, in the following example, the subject argument (Arg0) of the transitive verb *paRh* 'read' can be elided, e.g. when it is recoverable from the prior discourse or situational context. In the second sentence, the object argument (Arg1) is missing.

(124) **pro** *kitaab* *paRh-egii*
kitAb *paDzegI*
NULL book read-fut
'(She) will read the book'.

(125) *kis ne xarvAjA KolA mohan ne *pro* KolA*
*kis ne darwaazaa khol-aa? mohan ne *pro* khol-aa*
who erg door open-perf? Mohan erg *pro* khol-aa
'Who opened the door? Mohan opened (it)'.

In addition, three other kinds of obligatorily non-overt categories are inserted in PropBank:

- empty subject arguments occurring in nonfinite complement and adjunct clauses (big PRO marked as *PRO*);
- empty arguments in relative clauses (labeled as *RELPRO*)
- empty arguments in coordination and gapping constructions (labeled as *GAP-pro*)

For instance, in the following example, the empty subject of the nonfinite complement of the verb *chaah cAh* 'want' is labeled with *PRO*. Note that the empty subject is controlled by the subject of the matrix clause (*mohan ne* 'mohan erg')

(126) *mohan nei [[PRO]$_i$ kitaab paRh-nii] chaah-ii*
mohan ne kiwAb paDznI cAhI
Mohan erg NULL book read-Inf want-perf
'Mohan wanted to read the book.'

The category RELPRO in the following example represents gaps in participial relative clauses that are used as pronominal modifiers of noun phrases:

(127) *zyaadaatar [RELPRO] kal khul-e] darvaaze*
zyAxAwar [RELPRO] kal Kule xarvAje
most-of-the NULL yesterday open-perf doors
'Most of the doors that opened yesterday'

The category GAP in the following examples represents gaps in coordination constructions (where clauses are linked together using conjunctions such as *aur* 'and', *lekin* 'but') and gapping constructions (where the verb is missing along with one or more of its arguments):

(128) *mohan-ne kitaab$_i$ paRh-ii aur [GAP]$_i$ so ga-yaa*
mohan-ne kiwAb paDzI Or [GAP]$_i$ so gayA
M.-Erg book read-Perf and NULL sleep go-Perf
'Mohan read the book and slept.'

(129) *John-ne seb$_i$ khaa-yaa aur Mary-ne bhii [GAP]$_i$ (khaayaa)*
jOn-ne seb KayA Or mErI-ne BI [GAP]$_i$ KAyA
J.-Erg apple eat-Perf and M.Erg also NULL
'John ate an apple and Mary too.'

8.1 Insertion and annotation

The process of identification of null arguments will take place concurrently with addition of PropBank labels. Since the environments in which **PRO** and **REL-PRO** occur can be identified deterministically, these labels will be inserted automatically during a preprocessing step. The null elements **GAP** and **pro** are inserted manually.

The contexts in which each of the empty arguments are inserted will depend mainly on **valency** considerations. This means that in an underlying fashion, verbs will have the capability of licensing arguments, but in the surface syntax they are unable to do so.

For the annotation of the null categories, we use a special drop-down menu in Jubilee. This shows the null argument label and the PropBank label together. The annotator has to choose the appropriate label for a particular instance of the annotation. Figure 8.1 below shows the drop-down menu:

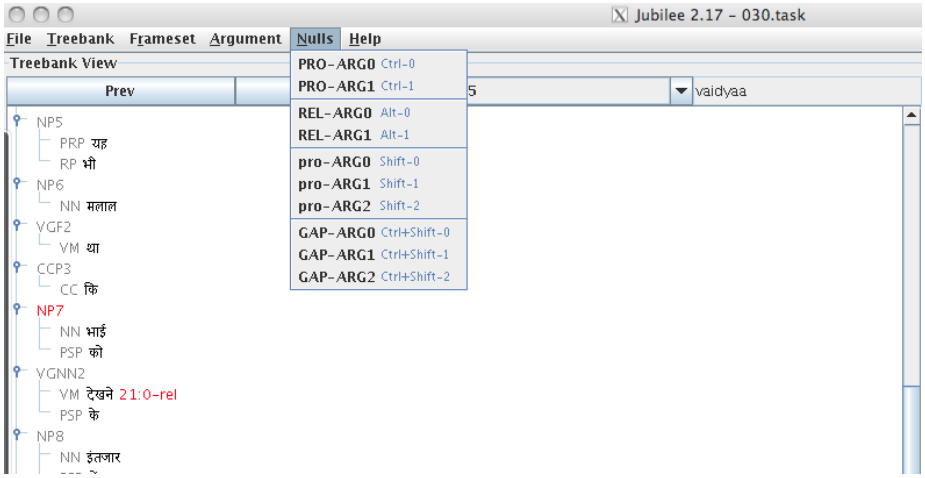

Figure 8.1: The null annotation drop down menu in Jubilee.

This method of annotation will identify the null argument type through the labels PRO, RELPRO, pro and GAP. Simulatenously, it will allow for the annotation the correct PropBank label Arg0, Arg1 or Arg2 with each null type. Note that null arguments are always numbered (i.e. obligatory arguments).

After having identified the correct type of null argument, each of these needs to be annotated using a different strategy. The actual annotation process for nulls will involve two types of actions:

- annotation on a shared argument: GAP, RELPRO, PRO
- annotation on the verb: pro

For any missing arguments that are coreferential with another argument in the same sentence (labeled **PRO**) or in one of the conjuncts of a conjoined sentence (labeled **GAP-PRO**, the antecedent argument will be double-annotated with the argument roles of each of the predicates with which it is associated. A similar strategy is used for **RELPRO**, although these cases dont have a shared argument, the label is simply placed on the modified noun.

For any missing arguments that are not co-referential with another argument that is 'local' (but which may be found in prior discourse), the verb will be annotated to indicate the missing argument(s) (labeled **pro** or the special cases of PRO without antecedents).

8.2 Annotation on shared argument

The correct null label should be chosen from the drop-down menu and then annotated, depending on whether the null type calls for shared annotation or annotation on the verb. The examples for shared annotation i.e. the GAP-pro and PRO cases are shown in Figure 8.2 and Figure 8.3

Figure 8.2: Example of shared annotation for GAP-pro

(130) *[udhar, John Carey ne apni haar swiikar kar lii]* **aur** *[*gap-pro*$_{Arg0}$ Bush ko$_{Arg2}$ phone par$_{ArgM\text{-}MNS}$ badhaai dii]*
[uXar, jOn kErI ne apni hAr svIkAr kar lI] Or *[*gap-pro*$_{Arg0}$ buS ko$_{Arg2}$ Pon par$_{ArgM\text{-}MNS}$ baXAI xI]*

The verb being annotated is 'dii hai' but the GAP-ARG0 label is applied onto the shared argument in the previous clause, in this case its John Carey. Note that John Carey will not be highlighted in red like the other arguments of the sentence. This is because it is a shared argument and belongs to the corresponding conjoined sentence with the verb 'svIkAr kar'.

The same strategy is applied to the cases where PRO is inserted and it has a shared argument. Figure 8.3 shows the annotation.

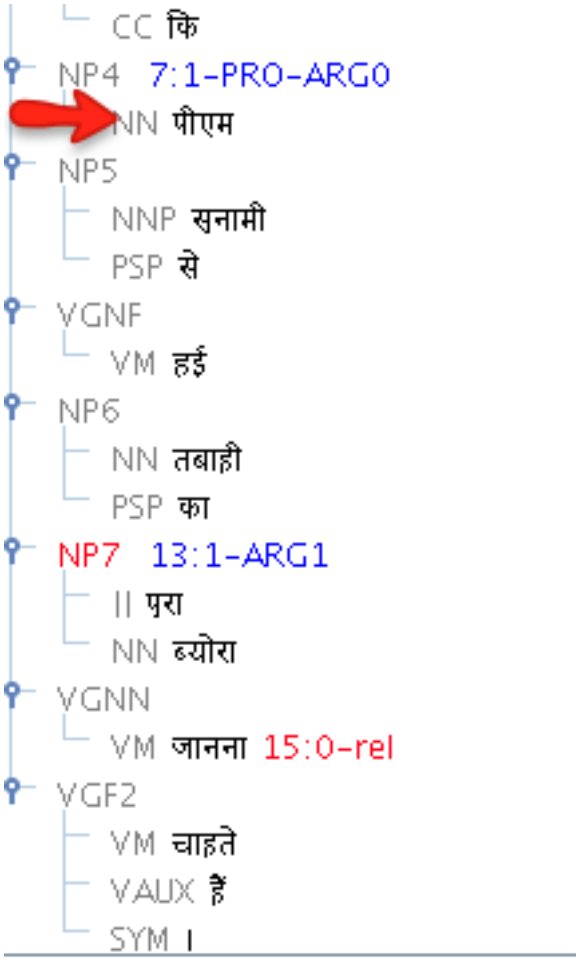

Figure 8.3: Null annotation for PRO 'shared' cases

(131) *PM [PRO$_{Arg0}$ tabaahi kaa puura byoraa$_{ARG1}$* ***jaananaa****] caahte hai*
PM [PRO$_{Arg0}$ wabAhI kA pUrA byorA$_{ARG1}$ ***jAnanA****] cAhwe hEM*

The argument 'PM' is shared across both verbs and hence PM gets the label PRO-ARG0. Similar to GAP-pro, it is the argument from another clause and wont be highlighted in the tree.

In the case of RELPRO, a strategy of double annotation is carried out, but the arguments are not shared in the sense of GAP-pro and PRO. The noun that is modified acts as a placeholder for the empty category.

(132) *hamaare dvaara$_{Arg0}$ RELPRO-$_{Arg1}$ abhi tak dekhe gaye$_{rel}$] grahon mein yaha sabse yuvaa hai*
[hamAre XvArA$_{Arg0}$ RELPRO-$_{Arg1}$ aBI wak xeKe gaye$_{rel}$] grahoM meM yahAM sabse yuvA hE

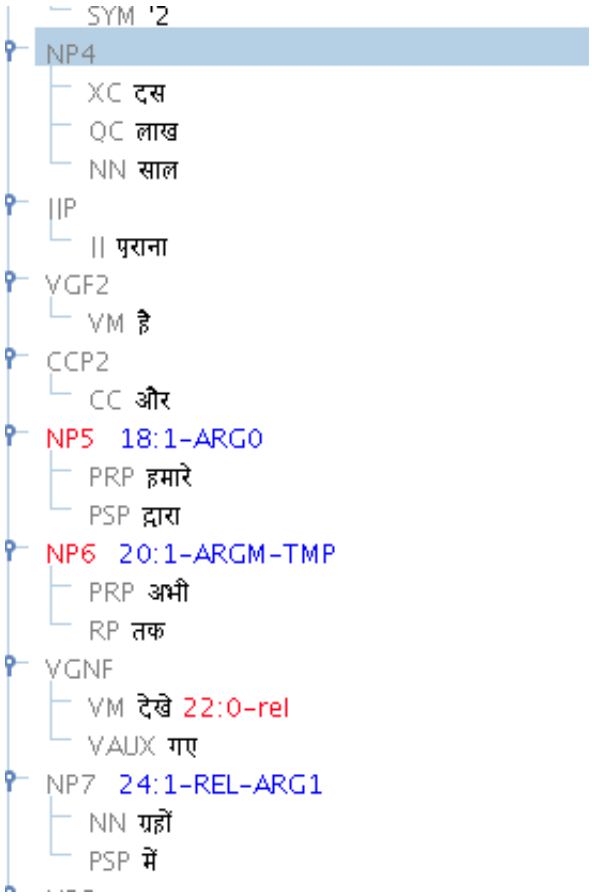

Figure 8.4: RELPRO annotation on the noun

In this case, 'grahoM meM' is the modified noun and gets the label REL-ARG1.

8.3 Annotation on the verb

Single annotation is carried out for those cases where there is no shared argument, and hence there is no actual placeholder for the null argument. This takes place for the pro type of null argument and also for the special cases of PRO that do not have antecedents. Figure 8.5 shows the single annotation strategy for pro, where the verbs node gets the pro-ARG0 label.

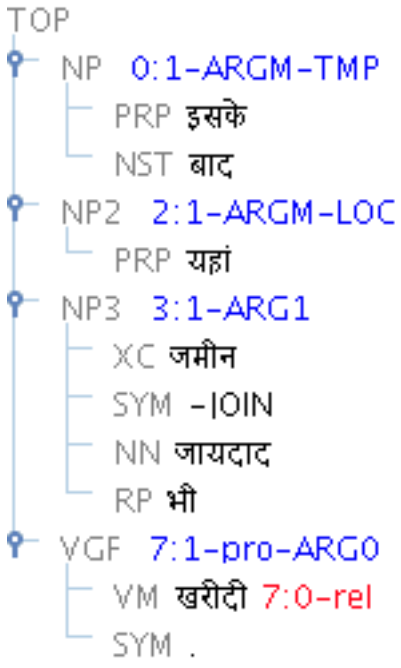

Figure 8.5: Single annotation of pro on the verb

The figure above shows the annotation *on the verb* for the null argument pro.

(133) *pro*$_{ARG0}$ *iske baad*$_{ARGM\text{-}TMP}$ *yahaan*$_{ARGM\text{-}LOC}$ *zameen-jaaydad bhi*$_{ARG1}$ ***kharidi***.
pro$_{ARG0}$ *iske bAx* $_{ARGM\text{-}TMP}$ *yahAM*$_{ARGM\text{-}LOC}$ *jamIn-jAyxax BI*$_{ARG1}$ ***kharIxI***.

The subsections below summarize the kinds of null elements that will be inserted and annotated by PropBank.

8.3.1 Empty relative pronoun: *RELPRO*

(134) *jyAxAwar [*RELPRO*] kal Kul-e] xarwAje*
most-of-the NULL yesterday open-Perf doors
'Most of the yesterday opened (by themselves) doors'

Examples of other cases:

(135) RELPRO *xillI jAne vAlA ladzkA*
'RELPRO Delhi going one boy'

(136) RELPRO *pIne kA pAnI*
'RELPRO drinking of water'

(137) [RELPRO_i t_i KAe gae] Pal
'RELPRO gotten eaten fruit'

(138) *[RELPRO rotI KAne vAlA] ladzkA*
'RELPRO bread eating one boy'

(139) *[RELPRO pro KAne vAlA] ladzkA*
RELPRO pro eating one boy'

(140) *[PRO RELPRO KAne vAlI] rotI*
PRO RELPRO eating one bread'

8.3.2 Empty arguments: pro*

(141) **pro* kitaab paRh-egii*
**pro* kiwAb paDz-egI*
NULL book read-fut
'(She) will read'

(142) *john-ko kitaab dii aur mary-ne *pro* magazine di*
*jOn ko kiwAb xI Or mErI ne *pro* mEgajIn xI*
john-acc book give-prf.f and mary-erg *pro* magazine give-prf.f

8.3.3 Empty arguments in coordination constructions: *GAP-pro*

(143) *mohan-ne kitaab$_i$ paRh-ii aur [*GAP-pro*]$_i$ so ga-yaa*
*mohan ne kiwAb paDzI Or [*GAP-pro*]$_i$ so gayA*
M.-Erg book read-Perf and NULL sleep go-Perf
'Mohan read the book and slept.'

(144) *kitaab$_i$ mohan-ne likh-ii aur [*GAP-pro*]$_i$ raam-ne paRh-ii.*
*kiwAb mohan ne liKI Or [*GAP-pro*]$_I$ rAm ne paDzI*
Book Mohan-erg write-Perf and NULL raam-erg read-Perf.
The book Mohan wrote and Ram read.

8.3.4 Empty k1 Argument (control): *PRO*

(145) *mohan-ne$_i$ [*PRO*]$_i$ kitaab paRh-nii] chaah-ii*
*mohan ne$_i$ [*PRO*]$_i$ kiwAb paDznI cAhI*
M.-Erg NULL book read-Inf want-Perf
'Mohan wanted to read the book.'

Null elements corresponding to 'pro' are automatically inserted at the beginning of the VGF chunk in the order Arg0, Arg2/Arg2 Arg1.

8.4 Special cases for Empty categories

The following sentence has a depictive clause, explaining the state in which the money is found. The phrase 'giraa huaa' is explaining the state of the money, but as it is infinitival, we give it a PRO.

(146) *Do sau rupaya skuul ke shauchalay ke paas [PRO giraa huaa] milaa*
Two hundred rupees school gen toilet gen near PRO fallen find toilet
'Two hundred rupees fallen (on the ground) were found near the school toilet'

8.5 Coreference

We will co-index two types of empty categories: PRO and GAP-pro. In the case of PRO, the coindexation type is 'pbref' and the PRO is linked with its antecedent in the matrix clause. The actual representation of co-reference is carried out in a post-processing step, using the argument linking positions marked during annotation.

(147) *mohan-ne$_i$ [[*PRO*]$_i$ kitaab paRh-nii] chaah-ii*
M.-Erg NULL book read-Inf want-Perf
'Mohan wanted to read the book.'

We do not co-index all cases of PRO. In those sentences containing the so-called arbitrary PRO, where the antecedent is not to be found in the same clause, no coindexation is provided for the PRO. For instance;

(148) *yahAA [[**PRO**] dhuumrapaan karnaa] manaa hae.*
Here NULL spitting do forbidden is.
'It is forbidden to spit here'

(149) *mohan-ne$_i$ kitaab paRh-ii aur [**GAP**]$_i$ so ga-yaa*
M.-Erg book read-Perf and NULL sleep go-Perf
'Mohan read the book and slept.'

(150) *kitaab$_i$ mohan-ne likh-ii aur [**GAP**]$_i$ raam-ne paRh-ii.*
Book Mohan-erg write-Perf and NULL raam-erg read-Perf.
'The book Mohan wrote and Ram read'

Chapter 9

Relatives and Correlatives

Chapter 10

Passives

Sentences can be either active (The executive committee approved the new policy) or passive (The new policy was approved by the executive committee). In active sentences, the subject is the agent or a do-er of the action, marked as Arg0 in Propbank. In passive sentences, the subject of the sentence is acted upon by some other agent or by something unnamed, and is being marked as Arg1 in Propbank.

In Hindi Propbank, the demoted subject of passives is labeled as Arg0 if it is overtly realized.

Chapter 11

Questions

Wh-phrases in questions are in-situ in Hindi, and are simply annotated with the argument role that would be assigned to it by the verb if it had been a regular argument. E.g. in the following example *mohan* would be annotated with Arg0.

(151) *Mohan* ***kyaa*** *khaataa hae?*
Mohan what eat.Imp is
'What does Mohan eat'

Chapter 12

Special cases of topicalization

These cases involve instances where a constituent is topicalized and is then repeated, often in the form of a pronoun, and the two constituents are not already indexed in the tree. For example, ***gandhiijii, unho-ne*** *hamaare desh kii sevaa mE sabkuch balidaan kiyaa* 'Gandhiijii, he sacrificed everything in the service of our nation.'

If the rel to be annotated is *balidaan kar* 'sacrifice do' the annotator would rst have to annotate the pronoun *unho-ne* 'he-erg' as ARG-1, then provide a coreference link to the pronoun's referent, *gandhiijii*. Select and annotate the pronoun in the argument position, then select the topicalized node and click Argument on the Jubilee menu bar, followed by clicking Functions. From the options therein, select '*' (shortcut: Ctrl+Shift-8). The linked annotation should appear in the TreeBank view and in the annotation view at the top of the screen.

Chapter 13

Span of annotation

13.1 Boundaries of annotation

For the purposes of PropBank annotation, annotators should only assign arguments within a certain syntactic span surrounding the rel. The structure of the tree reects which constituents in an utterance are truly arguments of a particular predicate; thus, even when annotators feel that a constituent outside of this span has some semantic bearing on the rel, it should not be annotated. Rather, the syntactic span of annotation should be respected: everything within that span should be encompassed by an argument label (with exceptions described below), and nothing outside of that span should be annotated (with exception of linking annotation, such as that of relative clauses).

Do not tag noun modifiers (labeled ? in DS) or conjunctions (labeled ?? in DS), unless these begin the sentence and are being used in a discourse function. Do not tag auxiliary verbs or modals or light verbs. These verbs will come up for annotation and at that point the appropriate will be selected without further annotation.

Bibliography

[1] Rafiya Begum, Samar Husain, Arun Dhwaj, Dipti Misra Sharma, Lakshmi Bai, and Rajeev Sangal. Dependency Annotation Scheme for Indian Languages. In *Proceedings of The Third International Joint Conference on Natural Language Processing (IJCNLP). Hyderabad, India*, 2008.

[2] Rajesh Bhatt, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Sharma, and Fei Xia. A Multi-Representational and Multi-Layered Treebank for Hindi/Urdu. In *In the Proceedings of the Third Linguistic Annotation Workshop held in conjunction with ACL-IJCNLP 2009*, 2009.

[3] Yamuna Kachru. *Hindi*. John Benjamins, 2006.

[4] Martha Palmer, Rajesh Bhatt, Bhuvana Narasimhan, Owen Rambow, Dipti Misra Sharma, and Fei Xia. Hindi Syntax: Annotating Dependency, Lexical Predicate-Argument Structure, and Phrase Structure. In *Proceedings of ICON-2009: 7th International Conference on Natural Language Processing*, Hyderabad, 2009.

[5] Martha Palmer, Daniel Gildea, and Paul Kingsbury. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106, 2005.

[6] Martha Palmer, Daniel Gildea, and Nianwen Xue. Semantic role labeling. In Graeme Hirst, editor, *Synthesis Lectures on Human Language Technologies*. Morgan and Claypool, 2010.