Hindi PropBank Annotation Guidelines
December 2012

Contents
Propbank Annotation Goals
Using the tools
Annotation of numbered arguments
Annotating of modifiers
Causatives
Unaccusatives and unergatives
Complex predicates
Empty categories
Passives
Questions
Special cases of topicalization
Span of annotation
Hindi PropBank Guidelines

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 (Palmer et al, 2005). The Hindi PropBank annotations are done on top of the Hindi dependency treebank (Bhatt et al, 2009, Palmer et al 2009). 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.

<table>
<thead>
<tr>
<th>'pii'</th>
<th>'to drink’, Transitive</th>
</tr>
</thead>
<tbody>
<tr>
<td>rAma ne SarAba pii</td>
<td></td>
</tr>
<tr>
<td>'Ram drank liquor.'</td>
<td></td>
</tr>
<tr>
<td>Arg0</td>
<td>Drinker: rAma</td>
</tr>
<tr>
<td>Arg1</td>
<td>Liquid: SarAba</td>
</tr>
</tbody>
</table>

Table 1: A framefile for the verb “pii” in Hindi

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 pii ‘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 pii appears in a sentence, it will have two arguments that 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 pii ‘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 expressions 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 (Begum et al. 2008). 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 relatively general and theory neutral (Palmer et. al, 2010). The number of the argument should be consistent if it maintains its semantic role across different syntactic realizations of the same 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
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'.

(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) modifiers. For a complete list of these function tags, please refer to section 4 below.

(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* *KAAnA* *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* *jAgA* *aur* *KAAnA* *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-b). 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 NULL *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.

2. Using the tools

2.1 Framefiles
Framefiles are an important source of information for annotators. 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 illustrations for the annotation of Hindi sentences. In PropBank, the annotation is carried out verb-by-verb. First, a verb is selected, then all the sentences in which that verb appears are identified in the Treebank. Second, the annotators consult the framefile for that verb, and annotate the arguments for that verb in all the sentences in which the verb occurs, based on the information provided in the verb's framefile. This process is repeated for each verb.

2.2 Framefile naming conventions

In Hindi, a verb can show up in various forms owing to transitivity alternations or causativization. All these related verb forms are included in the same framefile. If an intransitive form of the verb exists, then the framefile is named after this intransitive form. For example, the intransitive verb form diKa 'be seen' has the transitive counterpart diKA 'show', and the causative form diKavA 'make someone show something/someone'. The framefile is named after the intransitive verb form diKa: "diKa-v.xml". However, some verbs such as KarIda 'buy' do not have an 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 that we will be using in the discussions of framefiles: 'rolesets' and 'predicate frames'. The term 'roleset' refers to a part of 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). The term 'predicate frame' refers to a part of the framefile of a verb that 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 clearer as we see them used in the sections below.

2.3 Information in framefiles
Let us now talk in detail about the information contained in framefiles. The framefile for each verb provides fine-grained information specific to that verb. For instance, the verb-specific roles for the arguments of the verb ‘send’ include ‘sender’, ‘sendee’, and ‘thing sent’ as shown in the figure below for the Hindi verb *Beja* ‘send’. Note the set of semantic roles Arg0, Arg2_Gol and Arg1 is only one of the possible rolesets (“Beja.02”) for the verb *Beja.*

**Figure 1. Framefile for the verb Beja ‘send’**

Besides a description of these roles, the framefile also provides examples that illustrate the use of these semantic roles. This is shown in figure 2 below. While the roleset only provides a list of core arguments for the verb (as in figure 1 above), the examples may contain modifiers as well, e.g. ArgM_MNR in the example in figure 2 below.

**Figure 2. Example sentence in the framefile for Beja ‘send’**

If a verb has more than one sense but the argument structure of the verb is the same for both senses (i.e. the same set of semantic roles is used), then this information may also be represented in the framefile, with corresponding examples for the additional senses listed within the same roleset. Figures 3a and 3b illustrate two senses of the same verb with the same argument structure (belonging to the same roleset "Jalaka.04"). Note that in figure 3a as well as in figure 3b, the core arguments used are...
Figure 3a: An example for one sense for the 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 representing different senses for the verb Tahar ‘endure’/’stay’ (figures 4a & 4b). Note figure 4a represents the sense 'to stay/lodge/be located at' and the verb takes arguments Arg0 and Arg2_Loc, whereas in the figure 4b with the 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 corresponding to 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 3b: An example for another sense for the same roleset “Jalaka.04” in the framefile for the verb Jalaka
Figure 4a: One roleset “Tahar.01” for one sense of the verb Tahara with one set of semantic roles, namely, Arg0 and Arg2_Loc.

Figure 4b: 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 4b above. These rolesets belong to the same predicate frame, namely "Tahar" as can be seen in the figure 4b above.

The framefile for a verb can also have multiple predicate frames (and usually framefiles do have multiple predicate frames). The separate predicate frames are used to refer to various morphological forms a verb can have. In Hindi, a verb can have more than one morphological realization depending on its transitivity alternations or causativization. For example, the intransitive verb *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), complex predicates are also included 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.

2.4 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 section 4 below (also discussed in the literature on syntax-semantics mappings in South Asian languages, e.g. Mohanan 1994, Butt 1995).

2.4.1 Verb selection
We begin creating framefiles, starting with the most frequent verbs in the corpus and then proceeding 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, the internet, or they may be created by the framefile creators (who generally are native Hindi speakers).
2.4.2 Annotation using Cornerstone

The screenshot below (as well as in the figures 1-4 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.

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. "खा " KA ‘eat’, and not its inflected forms, e.g. "खाया " 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 complex predication. 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. Moreover, if there are exceptional cases among unaccusatives, e.g., 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. Often, 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 information needed to disambiguate the roles of the arguments, or to provide additional syntactic/morphological information relevant to the roles. For example, in the roleset "Jada.03" in figure 6 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.
Role: This field is used to add 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 that in figure 6 above, the '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 figure 6 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 the karaka label (e.g. k1/k2) mapping for the PropBank role. Note in the figure 6 above, Arg0 argument gets the k1 drel label, and the Arg1 argument gets the k2 drel label.
The above figure is a screenshot of the example screen. As mentioned 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 information about the source of the example sentence, e.g. whether the example was drawn from our Hindi DS-treebanked corpus, Hindi Wordnet, the internet, or if it was created by the framefile creators. The example in Figure 7 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 suffice.

**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, the following sentence is provided (example 5). We also provide the romanized script, the English glosses and the translation for the example to
aid readers in understanding how the arguments get PB roles which are mentioned further below.

(5) राम ने किरा खायी
[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 the case of complex predicates. Notice the use of the square brackets in the devanagari part in the example (6) below, the main verb "खायी" and the light verb "गयी" form a unit together which is shown using the square brackets in the examples on cornerstone.

(6) कल यहाँ बहुत खायी [खायी गयी]
[kala] [yahAM] [bahuwa Kira] KAI gal
yesterday here a lot pudding eaten go.Perf
'Yesterday a lot of pudding was eaten here.'

**Arguments**: This field is used to add 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 7 above, Below we present another example screen in figure 8, copied from figure 3a above. This figure shows the use of numbered tags Arg0 (n: 0) and Arg1 (n: 1) as well as function tag ArgM-Loc (n: m, f: loc).
Figure 8: 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 7 and 8 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 7 (use of k1 and k2) and figure 8 (use of k1, k7, and k2).

Sometimes more than one example is 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.

2.5 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 that the verbs have corresponding to these senses, examples showing the use of PB labels etc. As an example, let us 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

Example: \[ARG0 r\textit{Ama} ne] [ARG1 k\textit{Ama}] kiyA
Ram Erg work did
“Ram worked.”

However, as was mentioned above too, even though 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 \textit{baca} ("baca.01" and
"baca.02") from its framefile.

\textbf{baca.01 ‘remain’}
Arg1: thing remaining
Arg2: benefactive, entity getting the remaining thing

Example: \[ARG02 \textit{mere} p\textit{Asa}] \[ARG01 \textit{10 rupay}e\] bace hEM
me obl near/possession 10 rupees remained be
‘I have 10 rupees left with me.’

\textbf{baca.02 ‘avoid’}
Arg0: avoider
Arg1: thing avoided

Example: \[ARG0 yaha] [ARG0 Sar\textit{Aba} 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
\textit{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 9 below.:
Figure 9: 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 of the verb. Depending on the arguments present in each roleset, the annotators determine which roleset the verb in 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 9 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.

3. 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. We use 12 numbered arguments. However, some of these also carry a function tag in addition to a number in order 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:
Below we provide descriptions for each of the numbered argument labels. But before we look at these labels individually, a general observation about these labels 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 to label more than one argument within the same sentence, i.e. it is used to annotate intermediate causers in the sentence of which there may be more than one. Similarly the label Arg3 can be used repeatedly since multiple instruments are possible within the same sentence. See sections 3.5 and 3.12 for examples. Let us now look at each of these numbered argument labels in detail.
3.1. ARG0
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 an intransitive verb, a monotransitive verb and a ditransitive verb. In example (8a) 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 (8b) 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 (8c) 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.

8.a. [*Arg0 rAma*] rowA hE
   Ram cry-Hab be.Pres
   'Ram cries.'

b. [*Arg0 rAma*] [Klra] KAwA hE
   Ram ricepudding eat-Hab be.Pres
   'Ram eats rice pudding'

c. [*Arg0 rAma*] [slwa 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 or unaccusative. However, just as a general rule-of-thumb, 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 slwagets the label Arg0.

(9) a. [*Arg0 slwa*] nAca rahl 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.

**b.** [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.

**c.** [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’:

**10** [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 (11) 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.

**11** [Arg0 rAma xvArA] [Kira] KAyl gayI
   Ram by rice pudding 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 all Arg0 arguments do not necessarily get *ne*-marking. The *ne-*
marking is a helpful clue only when the verb is transitive and in the perfective aspect. For example, in (12a) below, the transitive verb is in the perfective aspect and the agentive subject gets the ne-marking, but in (12b), the verb is in non-perfective aspect and the agentive subject does not get the ne-marking. Similarly in (13a), 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 (13b) 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.

(12) a. [ARGO rAma ne] [Kira] KayI
    Ram   Erg  rice pudding   eat.Pst
    'Ram ate the rice pudding.'

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

(13) a. [ARGO rAma] xOdA
    Ram run.Pst
    'Ram ran.'

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

3.2. 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 child 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 recipient of the caused

---

3 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 verb’s 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.
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 section 5 below.

(14) \[\text{ArgA sIwA ne} \ [\text{AyA se} \ [\text{Arg0_GOL bacce ko} \ [\text{KiivAyA}]

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 (15) 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.

(15) a. \[\text{Arg0_GOL muJ-ko} \ [\text{TaMDa} \ lagI]

I-Dat cold feel.Pst
‘I felt cold (it felt cold to me).’

b. \[\text{Arg0_GOL muJ-ko} \ [\text{buKara} \ huA]

I-Dat hunger happen.Pst
‘I got fever (the fever happened to me).’

c. \[\text{Arg0_GOL muJ-ko} \ [\text{gussA} \ AyA]

I-Dat anger come.Pst
‘I got angry (the anger came to me).’

d. \[\text{Arg0_GOL muJ-ko} \ [\text{Sarma} \ AyI]

I-Dat shyness come.Pst
‘I felt embarrassed (the embarrassment came to me).’
Besides physical or emotional states, the experiencer subject is also found with certain verbs like *diKa* ‘glimpse’, *mila* ‘find’, *lagā* ‘feel/seem’, *sUJa* ‘be struck (with an idea)’.

(15) e. \[ARG0_GOL mj-ko] \[cAMda] dikhA  
I-Dat moon be.seen.Pst  
‘I glimpsed the moon.’

3.3. ARG0_MNS

The label ARG0_MNS (16a-b) is also used to annotate the doer of an action when the doer is caused by a causer to perform the action (similar to Arg0_GOL in example 14 above), but it is not the obvious recipient or beneficiary of the action (different from the Arg0_GOL in example 14). Note that this label is applied only when the action is performed on an Arg1 argument. Thus 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 the participants labeled by these arguments to perform some action. This can be 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 the 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 section 5 below.

(16) 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] KarlxavAyA  
Mohan Erg Ram Inst ticket buy-CAUS.Pst  
‘Mohan made/had Ram buy the ticket.’

3.4. ARGA
This label is used for the most external 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' and siKA 'teach' respectively. It can also be added to intransitive verbs such as ro 'cry' or gira 'fall' to derive the transitive rulA 'make someone cry' and 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 section 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:

(17) a. [ArgA rAma ne] [slwA 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 slwA se] [arg1 mohana ko] girvAyA
   Ram Erg Sita Instr Mohan Acc fall.CAUS.Pst
   ‘Ram caused Sita to make Mohan fall.’

c. [ArgA rAma ne] [slwA se] [mohana ko] rulvAyA
   Ram Erg Sita Instr Mohan Acc cry.CAUS.Pst
   ‘Ram caused Sita to make Mohan cry.’

3.5. 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 most external causer in the sentence, For example, in sentence (18) below, besides the most external causer rAma ne, an intermediate causer slwA se is also present for the causative verb KilvAyA (for further details see section 5 on causatives). This intermediate causer slwA se gets the label ArgA_MNS.

(18) [ArgA rAma ne] [argA_mns slwA 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 slwA se as well as for the argument rlwA 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 multiple times within the same sentence, as there can be more than one intermediate causer present in the sentence.

(19) [ArgA rAma ne] [ArgA_MNS slwA se] [ArgA_MNS rlwA xvArA] [bacce ko]
  Ram Erg Sita Inst Rita by child to
  [KAnA] KilvAyA food feed-CAUS.Pst
  ‘Ram made Sita, who via Rita made the child eat food.’

3.6. 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 (20) 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.

(20) [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 (21) below, newA kl 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.

(21) [Arg0 praXAna maMwrI ne] [Arg1 newA kl 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 that 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 ‘open’ in (22) 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.

(22) \[\text{ARG1} \ xarvAzA \] Kula
\hspace{1cm} door \quad \text{open-Pst}
\hspace{1cm} ‘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.

(23) \[\text{ARG1} \ CAwroM \ ke \ hAWa \] jala \ gaye
\hspace{1cm} students of hands \quad \text{burn} \quad \text{LV.Pst}
\hspace{1cm} ‘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 (Kachru, 2006), 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:

(24) a. \[\text{ARG1} \ rAma \] [acCA] \ hE
\hspace{1cm} Ram \quad \text{good} \quad \text{be.Pres}
\hspace{1cm} ‘Ram is good.’

b. \[\text{ARG1} \ saDaka \] [cODI] \ hul
\hspace{1cm} road \quad \text{wide} \quad \text{become-Pst}
\hspace{1cm} ‘(The) road became wide.’
3.7. ARG2
The Arg2 label is used to denote the beneficiary/recipient of the action expressed by the verb. For example, in the sentence below, the argument SyAma ko is the beneficiary/recipient of the verb xe 'give', hence it gets the PB label ARG2.

(25) [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 that an Arg2 argument is different from an Arg0_GOL argument (mentioned in section 3.2 above) in that even though both these arguments are recipients/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 recipient of the caused action.

3.8. ARG2_GOL
The Arg2_GOL is the destination or goal argument for the verb. It is useful to think of Arg2_GOL as 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.

(26) [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.

(27) [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 (28), where 'backwards' specifies the direction of motion although no endpoint/ goal is specified.

(28) [ARG0 vo] [ARG2_GOL piCe] KisakA  
   he backwards scooted.  
   'He scooted backwards.'

3.9. ARG2_SOU
This label is applied to the arguments that could be conceived of as a source or the starting point of the event of the verb. These usually appear with the postposition se 'from'. In example (29) below, the starting point for the event is Cawa se, hence it is marked as Arg2_SOU. In example (30), again the argument marked as Arg2_SOU is the point where the comparison originates (i.e. the source of competition). In example (31), the Arg2_SOU is the source from which the blessings are requested.

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

(30) [ARG0 slwArAma] [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.'

(31) [ARG0 mEMne] [ARG2_SOU apne BagavAna se] [ARG1 ASirvAxa] mAMgA  
   I-Erg my god Abl blessings 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:

(32) [ARG0 kiSorI] [ARG2_SOU harixvAra se] [ARG2_GOL xillI] Ayl 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:
Karishma had demanded two lakh rupees from Sanjay.

3.10. 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:

(34) [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’;

(35) [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:

(36) [ARG0 ve  loga] [ARG1 gAMX[i]j] ko] [ARG2_ATTR bApU]
    those people  Gandhi-Hon  Acc  bapu
    mAnawe  hEM
    consider-Hab  be.Pres
    ‘Those people consider Gandhiji as Bapu.’

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

3.11. 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 (37) below.
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 (38a) below). It can be used for an abstract/metaphorical location, such as xIvAra se ‘with the wall’ or SikSA kSewra se ‘with education field’ as in (38b) and (38c) below.

We also use Arg2_LOC for arguments that express accompaniment, notice the company can be taken as an abstract location. See (39) below.
3.12. 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:

\[(40) [ARG0 rAma ne] [ARG3 cAkU se] [ARG1 Ama] kAtA\]

\[
\begin{array}{llllll}
\text{Ram} & \text{Erg} & \text{knife} & \text{Instr} & \text{mango} & \text{cut.Pst} \\
\end{array}
\]

‘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.

\[(41) [ARG0 rAma ne] [cAkU kA upayoga karke] [ARG1 Ama] kAtA\]

\[
\begin{array}{llllllll}
\text{Ram} & \text{Erg} & \text{knife} & \text{Gen} & \text{use} & \text{having.done} & \text{mango} & \text{cut.Pst} \\
\end{array}
\]

‘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:

\[(42) [ARG0 slwA ne] [ARG3 pAnI se] [ARG1 Gade ko] BarA\]

\[
\begin{array}{llllllll}
\text{Sita} & \text{Erg} & \text{water} & \text{Instr} & \text{pot} & \text{Acc} & \text{fill.Pst} \\
\end{array}
\]

‘Sita filled the pot with water.’

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.\(^4\)

\(^4\) A note about some other -se marked arguments:

The -se marked argument for a verb such as SÀxl kara ‘marry’ or SÀxl ho ‘marriage happen’ get Arg1 label and the argument that would syntactically be treated as a subject for these verbs gets the Arg0
possibly could also act as an instrument as Arg1 only in cases where -se case marking is not used, such as in (43) below. Notice, in contrast to (42) 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.

(43) $[ARG0 \text{ sIwA ne}] [ARG1 \text{ pAnI}] [ARG2 \text{ Loc Gade meM}] \text{ BarA}$

\[\text{Sita} \ \text{Erg} \ \text{water} \ \text{pot} \ \text{Loc} \ \text{fill.Pst}\]

‘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 (44) below.

(44) $[ARG0 \text{ sIwA ne}] [ARG3 \text{ cammaca se}] [ARG1 \text{ ghade ko}] [ARG3 \text{ pAnI se}]$

\[\text{Sita} \ \text{Erg} \ \text{spoon} \ \text{Instr} \ \text{pot} \ \text{Acc} \ \text{water} \ \text{Instr} \ \text{BarA}\]

\[\text{fill.Pst}\]

‘Sita filled the pot with water with a spoon.’

4. Annotating 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:

<table>
<thead>
<tr>
<th>Modifiers</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. DIR</td>
<td>Directionals</td>
</tr>
<tr>
<td>2. LOC</td>
<td>Locatives</td>
</tr>
</tbody>
</table>

label, following the English PropBank. Hence in a sentence such as (i) below, Ram gets Arg0 and Sita, Arg1.

(i) $[ARG0 \text{ rAma ne}] [ARG1 \text{ sIwA se}]$ $\text{SAxl kl}$

\[\text{Ram} \ \text{Erg} \ \text{Sita with marriage} \ \text{do.Pst}\]

‘Ram married Sita.’

Further, for verbs such as $\text{gussA ho ‘be angry’}$ as in (ii) below, ‘Sita se’ will be treated as an adjunct rather than an argument.

(ii) $[ARG0 \text{ rAma}] [ARGM \text{ siWa se}] \text{ gussA hE}$

\[\text{Ram} \ \text{Sita with angry} \ \text{be.Pres}\]

‘Ram is angry with Sita.’
### 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.

35.  *mohana [ARGM-DIR 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:

36.  *rAma ne [ARGM-LOC bAzAra mEM] ravI ko xeKA.*
    Ram erg       market in       ravi dat saw
    ‘Raam saw Ravi in the market’
37. \[ \text{ARGM-LOC apne BASana mEM}, \text{mohana ne newa kl} \text{ KUba wArIfa kl.} \]
   self lecture in Mohan erg leader gen much praise did.
   ‘In his lecture, Mohan praised the leader highly’

4.3. \text{ARGM_MNR}
Manner adverbs specify how an action is performed. For example, \text{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 \text{kaise ‘how’}?

38. \text{vaha bahuwa [ARGM-MNR wejZa] bolawA hE}
   he very fast talk.Imp be
   ‘He talks very fast’

4.4. \text{ARGM_MNS}
This is like an ARGM equivalent to the instrumental ARG3. It functions to accommodate cases such as the following:

39. \text{yanA ko wuraMwa [ARGM-MEANS ek kAra se] aspawAla le jAyA gayA}
   Yanaa dat immediately one car with hospital take go AUX
   ‘Yanaa was immediately taken to the hospital by car’

40. \text{rAja [ARGM-MEANS pArtI xvArA] BJP se sambaMWa woda lene}
   Raj party by BJP with connection break Light Verb
   ke pakSa mEM hE
   gen side in is
   ‘Raj is in favour of breaking connection with the BJP via the party.’

4.5. \text{ARGM_GOL}
This tag is for the goal of the action of the verb. This includes beneficiaries and the final destination of some motion verbs. ‘Goal’ would be used for modifiers that indicate that the action of the verb was done for someone or something, or on their behalf:

41. \text{mohana ne [ARGM-GOL ravi ke liye] cAy banAl.}
   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 mat* 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’:

42. [ARGM-EXT 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:

43. mohana ne [ARGM-EXT slwA se] weza BAgA.
    Mohan erg Sita abl fast ran.
    ‘Mohan ran faster than Sita did’

44. rAWA [ARGM-EXT [mlrA kl wulanA mEM 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 (eg. *pehlaa* ‘first’), and repetition (eg. *baar-baar, phir se* ‘again’):

45. [ARGM-TMP 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 modifier.

46. **Mohan aur siitaa ne**  [ARGM-REC ek saath] *kaam kiyaa.*
   Mohan and Siitaa erg together work did
   ‘Mohan and siitaa worked together’

47. **mohan ne**  [ARGM-REC 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.

48. **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. **ARGM_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:

49. **bacce ne seb ko**  [ARGM-PRD 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.

50. **maine**  [ARGM-PRP 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':

51. \textit{ARGM-CAU} mu\textit{Kya gavAha ke is bay\textit{Ana se} sarab\textit{jlwa kl rih\textit{Al kl}} primary witness gen this testimony instr Sarabjit gen acquittal gen ASA badha gayl hE hope increase go be 'By the primary witness' testimony, the hope for Sarabjit’s acquittal has increased'

52. \textit{ARGM-CAU} 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: \textit{lekin/parantu} ‘but’, \textit{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.

53. \textit{ARGM-DIS 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?'

54. \textit{ARGM-DIS isake alaava}, maovaadi ke raambacana yaadav ko gir\textit{ftaar} 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 marker includes vocatives and interjections:

55. \textit{ARGM-DIS maa}, mujhe kal dilli jaana hai mother, I-dat tomorrow Delhi go-inf be 'Mother, I want to go to Delhi tomorrow'
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.

i. Temporally related (modifiers of events): \textit{mohan kii pratiiskhaa mein siitaa wahiin khaRii rahii} ‘Sita remained standing there, in anticipation of Mohan’

ii. Intensional (modifiers of propositions): \textit{shaayad} ‘probably, possibly’

iii. Focus-sensitive: \textit{sirf/keval ‘only’, bhii ‘also, even’}

iv. Sentential (evaluative, attitudinal, viewpoint, performatives): \textit{soubhaagya se ‘fortunately’,}


vi. Conditional clauses like \textit{agar-to}, where the dependent clause gets the Adv label

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

57. \textit{[ARGM-ADV shaayad] siitaa khaanaa khaayegii} maybe Sita food will eat
   ‘Maybe Sita will eat food’

58. \textit{[ARGM-DIS 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 ARG-M-Mod label.

59. \textit{mohan kaam kar [ARGM-MOD sakegaa].} Mohan work do able.fut
   ‘Mohan will be able to do the work’
60. *mohan kaam kar [ARGM-MOD paaegaa].*  
Mohan work do able.fut  
‘Mohan will be able to do the work’

61. *mohan ko ghar jaanaa [ARGM-MOD caahiye].*  
Mohan dat home go-inf ought  
‘Mohan ought to go home’

62. *mohan ko kaam karnaa [ARGM-MOD padZaa].*  
Mohan dat work do-inf had  
‘Mohan had to work’

63. *mohan ko kulfii khaanii [ARGM-MOD hai].*  
Mohan dat icecream eat-inf is  
‘Mohan has/wants to eat icecream’

4.16. **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.

64. *mohan ro [ARGM-VLV paRaa]*  
Mohan cry lie  
‘Mohan burst out crying’

65. *siitaa saaraa khaanaa khaa [ARGM-VLV gayii]*  
siitaa all food eat went  
‘Siitaa ate up all the food’.

66. *siitaa saaraa khaanaa khaa cukii [ARGM-VLV hae]*  
siitaa all food eat complete.prf is  
‘Siitaa has eaten all the food’.

67. *saritaa ne saRii khariid [ARGM-VLV lii].*  
Sarita erg sarii buy took
‘Saritaa has bought the sarii’

68.  
| bacce ne  mAA  ko gend de  [ARGM-VLV 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.

69.  
| raam ne  mohan ko duudh piine diyaa. |
| Raam erg Mohan dat milk drink-inf gave. |
| ‘Ram allowed Mohan to drink milk’ |

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.

4.17. 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.

70.  
| laRkii ghar  [ARGM-NEG nahii] gayii |
| girl  home  not  went |
| ‘(The) girl did not go home’ |
5. Causatives

In 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 72). 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 72). 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.

72. \[Sita \text{ ne}_{\text{ARGA}} \text{ mohan se}_{\text{ArgA,MNS}} \text{ ram ko}_{\text{Arg0}} \text{ rul-vAyA} \]
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 (73a-b) 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.
73a. [Mohan ne ] [raam se ] [peRa ] kat-vAyA
     Mohan erg raam inst tree cut-CAUS
     ‘Mohan made/had Raam cut the tree’

b. [John ne ] [sltA se ] [ticket ] KarldvAyA
    John erg siita 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 74 below) which also happens to be the recipient or beneficiary of the caused action (KilavA in example 72 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 74).

74. [rAma ne ] [slwA se ] [bacce ko ] [KanA ] 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 monotranstive or ditransitive verbs which are themselves derived from other intransitive or monotranstive 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 monotranstive 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 monotranstive 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 monotranstive verb slka ‘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.

75. a. \[ ARG0 \text{mohana ne} \] \[ rAma ko \] \text{ru}AyA
   Mohan \text{erg} \hspace{5mm} \text{rama Dat cry.made}
   ‘Mohan made Ram cry.’

b. \[ ARG0 \text{mohana ne} \] \[ rAma ko \] \[ Kira \] \text{KilAyI}
   Mohan \text{Erg} \hspace{5mm} \text{Ram Dat rice pudding fed}
   ‘Mohan fed Ram the rice pudding.’

c. \[ ARG0 \text{mohana ne} \] \text{bacce ko siKAyA}
   Mohan \text{erg} \hspace{5mm} \text{child dat taught}
   ‘Mohan taught the child.’

**Unaccusative**

<table>
<thead>
<tr>
<th>Sentence 1</th>
<th></th>
<th>mom</th>
<th>pighal-ii</th>
</tr>
</thead>
<tbody>
<tr>
<td>translation</td>
<td></td>
<td>wax</td>
<td>melted</td>
</tr>
<tr>
<td>PB labels</td>
<td></td>
<td>ARG1</td>
<td></td>
</tr>
<tr>
<td>Meaning</td>
<td></td>
<td>The wax melted.</td>
<td></td>
</tr>
</tbody>
</table>

**Unaccusative --> Transitive**

<table>
<thead>
<tr>
<th>Sentence 2</th>
<th></th>
<th>siita-ne</th>
<th>mom</th>
<th>pighaal-ii</th>
</tr>
</thead>
<tbody>
<tr>
<td>translation</td>
<td></td>
<td>Sita</td>
<td>wax</td>
<td>melted</td>
</tr>
<tr>
<td>PB labels</td>
<td></td>
<td>ARG0</td>
<td>ARG1</td>
<td></td>
</tr>
<tr>
<td>Meaning</td>
<td></td>
<td>Sita melted the wax.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

**Unaccusative --> Causative**

<table>
<thead>
<tr>
<th>Sentence 3</th>
<th></th>
<th>John-ne</th>
<th>siita-se</th>
<th>mom</th>
<th>pighal-vaayii</th>
</tr>
</thead>
<tbody>
<tr>
<td>translation</td>
<td></td>
<td>John</td>
<td>Sita-from/by</td>
<td>wax</td>
<td>made to melt</td>
</tr>
</tbody>
</table>
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.

**Unaccusative --> Causative**

<table>
<thead>
<tr>
<th>PB labels</th>
<th>ARG_A</th>
<th>ARG0_MNS</th>
<th>ARG1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meaning</td>
<td>John made Sita melt the wax.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.

**Unergative**

<table>
<thead>
<tr>
<th>Sentence 5</th>
<th>raam</th>
<th>ro-yaa</th>
</tr>
</thead>
<tbody>
<tr>
<td>translation</td>
<td>Ram</td>
<td>cried</td>
</tr>
<tr>
<td>PB labels</td>
<td>ARG0</td>
<td></td>
</tr>
<tr>
<td>Meaning</td>
<td>Ram cried.</td>
<td></td>
</tr>
</tbody>
</table>

**Unergative --> Transitive**

<table>
<thead>
<tr>
<th>Sentence 6</th>
<th>siita-ne</th>
<th>raam-ko</th>
<th>rul-aayaa</th>
</tr>
</thead>
<tbody>
<tr>
<td>translation</td>
<td>Sita</td>
<td>Ram</td>
<td>made-cry</td>
</tr>
<tr>
<td>PB labels</td>
<td>ARG_A</td>
<td>ARG0</td>
<td></td>
</tr>
<tr>
<td>Meaning</td>
<td>Sita made Ram cry.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Here the doer of the unergative verb gets the Arg0 label. The only causer present gets the ArgA label.

### Unergative --> Causative

<table>
<thead>
<tr>
<th>Sentence 7</th>
<th>John-ne</th>
<th>siita-se</th>
<th>raam-ko</th>
<th>rul-vaayaa</th>
</tr>
</thead>
<tbody>
<tr>
<td>translation</td>
<td>John</td>
<td>Sita-from/by</td>
<td>Ram</td>
<td>made-cry</td>
</tr>
<tr>
<td>PB labels</td>
<td>ARG_A</td>
<td>ARG_A_MNS</td>
<td>ARG0</td>
<td></td>
</tr>
<tr>
<td>Meaning</td>
<td>John made Sita make Ram cry.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.

### Transitive

<table>
<thead>
<tr>
<th>Sentence 9</th>
<th>bacce-ne</th>
<th>khaanaa</th>
<th>khaa-yaa</th>
</tr>
</thead>
<tbody>
<tr>
<td>translation</td>
<td>child</td>
<td>food</td>
<td>ate</td>
</tr>
<tr>
<td>PB labels</td>
<td>ARG0</td>
<td>ARG1</td>
<td></td>
</tr>
<tr>
<td>Meaning</td>
<td>The kid ate the food.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Transitive --> Ditransitive

<table>
<thead>
<tr>
<th>Sentence 10</th>
<th>siitaa-ne</th>
<th>bacce-ko</th>
<th>khaanaa</th>
<th>khil-aa-yaa</th>
</tr>
</thead>
<tbody>
<tr>
<td>translation</td>
<td>Sita</td>
<td>child-to</td>
<td>food</td>
<td>made-eat</td>
</tr>
<tr>
<td>PB labels</td>
<td>ARG_A</td>
<td>ARG0_GOL</td>
<td>ARG1</td>
<td></td>
</tr>
<tr>
<td>Meaning</td>
<td>Sita made the kid eat food.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.
## Transitive --> Causative

<table>
<thead>
<tr>
<th>Sentence</th>
<th>John-ne</th>
<th>siita-se</th>
<th>bacce-ko</th>
<th>khaanaa</th>
<th>khil-vaa-yaa</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>translation</th>
<th>John</th>
<th>Sita-from/by</th>
<th>child-to</th>
<th>food</th>
<th>made-eat</th>
</tr>
</thead>
<tbody>
<tr>
<td>PB labels</td>
<td>ARG_A</td>
<td>ARG_A_MNS</td>
<td>ARG0_GOL</td>
<td>ARG1</td>
<td></td>
</tr>
<tr>
<td>Meaning</td>
<td>John made Sita make the kid eat food.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

## Transitive

<table>
<thead>
<tr>
<th>Sentence</th>
<th>John-ne</th>
<th>siita-ne</th>
<th>ticket</th>
<th>kharid-aa</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>translation</th>
<th>Sita</th>
<th>ticket</th>
<th>bought</th>
</tr>
</thead>
<tbody>
<tr>
<td>PB labels</td>
<td>ARG0</td>
<td>ARG1</td>
<td></td>
</tr>
<tr>
<td>Meaning</td>
<td>Sita bought the ticket.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Transitive --> Causative

<table>
<thead>
<tr>
<th>Sentence</th>
<th>John-ne</th>
<th>siita-se</th>
<th>ticket</th>
<th>kharid-vaayaa</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>translation</th>
<th>John</th>
<th>Sita-from/by</th>
<th>ticket</th>
<th>make-buy</th>
</tr>
</thead>
<tbody>
<tr>
<td>PB labels</td>
<td>ARG_A</td>
<td>ARG0_MNS</td>
<td>ARG1</td>
<td></td>
</tr>
<tr>
<td>Meaning</td>
<td>John made Sita buy the ticket.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.
### Transitive --> Causative

<table>
<thead>
<tr>
<th>Sentence 15</th>
<th>Bill-ne</th>
<th>John-dwaaraa</th>
<th>siitaa-se</th>
<th>ticket</th>
<th>khariid-vaayaa</th>
</tr>
</thead>
<tbody>
<tr>
<td>translation</td>
<td>Bill</td>
<td>John-from/by</td>
<td>Sita</td>
<td>ticket</td>
<td>make-buy</td>
</tr>
<tr>
<td>PB labels</td>
<td>ARG_A</td>
<td>ARG0-MNS</td>
<td>ARG1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Meaning</td>
<td>Bill made John make Sita buy the ticket.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

#### Transitive

<table>
<thead>
<tr>
<th>Sentence 16</th>
<th>siitaa-ne</th>
<th>ticket</th>
<th>khariid-aa</th>
</tr>
</thead>
<tbody>
<tr>
<td>translation</td>
<td>Sita</td>
<td>ticket</td>
<td>bought</td>
</tr>
<tr>
<td>PB labels</td>
<td>ARG0</td>
<td>ARG1</td>
<td></td>
</tr>
<tr>
<td>Meaning</td>
<td>Sita bought the ticket.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Transitive --> Ditransitive

<table>
<thead>
<tr>
<th>Sentence 17</th>
<th>siitaa-ne</th>
<th>Bill-ke liye</th>
<th>ticket</th>
<th>khariid-aa</th>
</tr>
</thead>
<tbody>
<tr>
<td>translation</td>
<td>Sita</td>
<td>Bill-for</td>
<td>ticket</td>
<td>buy</td>
</tr>
<tr>
<td>PB labels</td>
<td>ARG0</td>
<td>ARG2</td>
<td>ARG1</td>
<td></td>
</tr>
<tr>
<td>Meaning</td>
<td>Sita bought the ticket for Bill.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Transitive --> Causative

<table>
<thead>
<tr>
<th>Sentence 18</th>
<th>John-ne</th>
<th>Pat-dwaaraa</th>
<th>Siita-se</th>
<th>ticket</th>
<th>khariid-vaayaa</th>
</tr>
</thead>
<tbody>
<tr>
<td>translation</td>
<td>John</td>
<td>Pat-from/by</td>
<td>Sita</td>
<td>ticket</td>
<td>make-buy</td>
</tr>
<tr>
<td>PB labels</td>
<td>ARG_A</td>
<td>ARG0-MNS</td>
<td>ARG1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Meaning</td>
<td>John made Pat make Sita buy the ticket.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.

**Transitive --> Ditransitive Causative**

<table>
<thead>
<tr>
<th>Sentence</th>
<th>John-ne</th>
<th>Pat-dwaaraa</th>
<th>Sita-se</th>
<th>Bill-ke liye/ Bill-ko</th>
<th>ticket</th>
<th>kharid-vaayaa</th>
</tr>
</thead>
<tbody>
<tr>
<td>translation</td>
<td>John</td>
<td>Pat-by</td>
<td>Sita-from/by</td>
<td>For Bill/to Bill</td>
<td>ticket</td>
<td>make-buy</td>
</tr>
<tr>
<td>PB labels</td>
<td>ARG_A</td>
<td>ARG_A_MNS</td>
<td>ARG0_MNS</td>
<td>ARG2</td>
<td>ARG1</td>
<td></td>
</tr>
<tr>
<td>Meaning</td>
<td>John made Pat make Sita buy the ticket for Bill.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.

**Ditransitive**

<table>
<thead>
<tr>
<th>Sentence</th>
<th>Mohan-ne</th>
<th>John-ko</th>
<th>kitaab</th>
<th>dii</th>
</tr>
</thead>
<tbody>
<tr>
<td>translation</td>
<td>Mohan</td>
<td>John-to</td>
<td>book</td>
<td>gave</td>
</tr>
<tr>
<td>PB labels</td>
<td>ARG0</td>
<td>ARG2</td>
<td>ARG1</td>
<td></td>
</tr>
<tr>
<td>Meaning</td>
<td>Mohan gave John the book.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Ditransitive --> Causative**

<table>
<thead>
<tr>
<th>Sentence</th>
<th>raam-ne</th>
<th>Mohan-dwaaraa</th>
<th>John-ko</th>
<th>kitaab</th>
<th>dil-vaa-yii</th>
</tr>
</thead>
<tbody>
<tr>
<td>translation</td>
<td>Ram</td>
<td>Mohan-by</td>
<td>John-to</td>
<td>book</td>
<td>make-give</td>
</tr>
<tr>
<td>PB labels</td>
<td>ARG_A</td>
<td>ARG0_MNS</td>
<td>ARG2</td>
<td>ARG1</td>
<td></td>
</tr>
</tbody>
</table>
Meaning: Ram made Mohan give John the book.

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

**Ditransitive --> Causative**

<table>
<thead>
<tr>
<th>Sentence</th>
<th>bill-ne</th>
<th>raam-dwaaraa</th>
<th>Mohan-se</th>
<th>John-ko</th>
<th>kitaab</th>
<th>dil-vaa-yii</th>
</tr>
</thead>
<tbody>
<tr>
<td>translation</td>
<td>Bill</td>
<td>Ram-by</td>
<td>Mohan-from/by</td>
<td>John-to</td>
<td>book</td>
<td>make-give</td>
</tr>
<tr>
<td>PB labels</td>
<td>ARG_A</td>
<td>ARG_A_MNS</td>
<td>ARG0_MNS</td>
<td>ARG2</td>
<td>ARG1</td>
<td></td>
</tr>
<tr>
<td>Meaning</td>
<td>Bill made Ram make Mohan give John the book.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.

**6. Unaccusatives**

As discussed in earlier sections, the single argument of intransitive verbs is labeled as Arg0 or Arg1 based on a variety of factors including the agentivity and/or volitionality of the argument. This distinction corresponds to differences in the syntactic behavior of intransitive verbs. When creating framefiles for intransitive verbs, a notation is inserted in the file identifying the verb as unaccusative or unergative based on its behavior with respect to a number of diagnostics. A description of the diagnostics follows.

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

2. Cognate objects: Unergatives may also sometimes allow cognate objects.

3. Impersonal passives: 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.

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

5. Inabilitatives: Unaccusatives enter the inabilitative with active syntax; unergatives cannot enter the inabilitative with active syntax:
6. Use of an instrumental phrase to introduce an agent (who acts inadvertently)

7. Compound verb selection: The unaccusative compound verb jaa `go' appears most naturally with unaccusatives; unergatives seem unhappy with jaa `go'. 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:

8. Unmarked subjects of non-finite clauses: Non-finite clauses in Hindi-Urdu typically do not permit overt unmarked subjects. But inanimate (and non-specific) subjects of unaccusative predicates can appear without an overt genitive. The inanimacy restriction means that we can't really use this test freely with unergatives as most of them only have animate subjects.

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 marked with a special label `pof'. These cases are automatically annotated with the label ARGMPRX. During the annotation of simple verbs, we will not look at those cases of the verb where ARGMPRX 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 conventions developed for English and Arabic PropBank).

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.
8. Empty Categories

Null arguments will be annotated in the following way. 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. For any missing arguments that are not coreferential 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 PRO-arb). For arguments of reduced relatives (participial modifiers), a preprocessing step will be created to introduce these nulls automatically (labeled ‘RELPRO’).

8.1 Insertion and annotation

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 (so-called “little pro, marked as *pro*”), using the context to determine which sense of the verb is relevant. We also use the verb framefile 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.

76. *pro* kitaab paRh-egii
   NULL book read-fut
   ‘(She) will read the book’.

77. 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:

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

Note that we will not be actually distinguishing between GAP-pro and pro when we insert null categories. Rather they will be uniformly labeled as pro when they are inserted. However, we will be distinguishing between the two types of empty categories in a postprocessing step. The ‘pro’ that we insert in coordination and gapping constructions will be coindexed to its antecedent using a special coindexation label (e.g. ‘gapref’). The uses of ‘pro’ elsewhere will not be coindexed (since the antecedent may be found only by looking outside the sentence, in prior discourse), except for any ‘pro’ instances found in relatives or correlatives. See the section on coreference for further details.

In all these cases, the empty argument is controlled by an antecedent within the same sentence. (Please note that we distinguish between “big” PRO represented in caps as *PRO*, and “little” pro represented as *pro*).

Since the environments in which *PRO* and *RELPRO* occur can be identified deterministically, these labels will be inserted automatically during a preprocessing step. The null elements *GAP-pro* and *pro* are inserted manually.

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

78. mohan ne, [[*PRO*]i kitaab paRh-nii] chaah-ii
    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:
79. **zyaadaatar [*RELPRO*] kal khul-e darvaaze**

   most-of-the NULL yesterday open-perf doors

   ‘Most of the doors that opened yesterday’

The category *GAP-pro* 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):

80. **mohan-ne kitaab paRh-II aur [*GAP-pro*] so ga-yaa**

   Mohan-Erg book read-Perf and NULL sleep go-Perf

   ‘Mohan read the book and slept.’

81. **John-ne sebI khaa-yaa aur Mary-ne bhii [*GAP-pro*] (khaayaa)**

   John-Erg apple eat-Perf and M.Erg also NULL

   ‘John ate an apple and Mary too.’

Below is a table that summarizes the kinds of null elements that will be inserted and annotated by PropBank.
<table>
<thead>
<tr>
<th>Description</th>
<th>Examples</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empty relative pronoun</td>
<td>*jaadaatar [&quot;RELPRO&quot;] kal khul-e] darwaaze most-of-the NULL yesterday open-Perf doors'*most of the yesterday opened (by themselves) doors'</td>
<td><em>RELPRO</em></td>
</tr>
<tr>
<td></td>
<td>Examples of other cases:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><em>RELPRO dilli jaane waala laRkaa</em></td>
<td></td>
</tr>
<tr>
<td></td>
<td>‘RELPRO Delhi going one boy’</td>
<td></td>
</tr>
<tr>
<td></td>
<td><em>RELPRO piine kaa paanii</em></td>
<td></td>
</tr>
<tr>
<td></td>
<td>‘RELPRO drinking of water’</td>
<td></td>
</tr>
<tr>
<td></td>
<td><em>RELPRO_i t_i khaaye gaye] phal</em></td>
<td></td>
</tr>
<tr>
<td></td>
<td>‘RELPRO gotten eaten fruit’</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[RELPRO roTii khaane waalaa] laRkaa</td>
<td></td>
</tr>
<tr>
<td></td>
<td>‘RELPRO bread eating one boy’</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[RELPRO pro khaane waalaa] laRkaa</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RELPRO pro eating one boy’</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[PRO RELPRO khaane waali] roTii</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PRO RELPRO eating one bread’</td>
<td></td>
</tr>
<tr>
<td>Empty arguments of the verb (regular pro-drop)</td>
<td><em>pro</em> kitaab paRh-egii NULL book read-fut <em>(She) will read the book</em>.</td>
<td><em>pro</em></td>
</tr>
<tr>
<td></td>
<td>siitaa-ne john-ko kitaab dii aur mary-ne <em>pro</em> magazine dii</td>
<td></td>
</tr>
<tr>
<td></td>
<td>S.-Erg J.-Dat book give.Perf and M.Erg NULL magazine give.Perf</td>
<td></td>
</tr>
<tr>
<td></td>
<td>‘Sita gave John a book and Mary a magazine.’</td>
<td></td>
</tr>
</tbody>
</table>
Null elements corresponding to ‘pro’ are automatically inserted at the beginning of the VGF chunk in the order Arg0, Arg2/Arg2 Arg1.

### 8.2 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.

82. **Do sau rupaya skuul ke shauchalay ke paas [PRO giraa huaa] milaa**

Two hundred rupees school gen toilet gen near PRO fallen find

‘Two hundred rupees fallen (on the ground) were found near the school toilet’

### 8.3. Coreference

We will co-index two types of empty categories: PRO and GAP-pro. In the case of PRO, the coindexation type is ‘coref’ and the PRO is linked with its antecedent in the matrix clause.
83. mohan-ne, ["PRO*"] kitaab paRh-nii chaah-ii
   Mohan-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
“PROarb”, where the antecedent is not to be found in the same clause, no coindexation
is provided for the PRO. For instance;

84. yahAA ["PRO*"] dhuumrapaan karna] manaa hae.
   Here NULL spitting do forbidden is.
   'It is forbidden to spit here'

We also insert indices to indicate the link between the empty category and its
antecedent in coordination constructions and gapping constructions. The type of
coindexation here is different from that provided for the PRO cases because here we
are simply “copying” lexical material from the initial conjunct to the second conjunct.
Therefore we will use a different coindexation convention here, annotating such cases
as “gapref”.

85. mohan-ne kitaab,paRh-ii aur ["GAP-pro*"] so ga-yaa
   M.-Erg book read-Perf and NULL sleep go-Perf
   'Mohan read the book and slept.'

86. kitaab, mohan-ne likh-ii aur ["GAP-pro*"] raam-ne paRh-ii.
   Book Mohan-erg write-Perf and NULL raam-erg read-Perf.
   'The book Mohan wrote and Ram read'

9. 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.
10. 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.

87.  Mohan kyaa khaataa hae?
Mohan what eat.Imp  is
‘what does Mohan eat’

11. 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 first 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.

12. 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 reflects 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.
References


Resources used in creating the Hindi framesets

i. Unified Verb Index (http://verbs.colorado.edu/verb-index/)
ii. Hindi WordNet (http://www.cfilt.iitb.ac.in/wordnet/webhwn/)
iii. DS annotation guidelines (http://ltrc.iit.ac.in/MachineTrans/research/tb/final_guidelines-ver2.pdf)
iv. Hindi Treebank (IIIT-Hyderabad database)