PROPBANK ANNOTATION GUIDELINES

Olga Babko-Malaya
September 2005

1. PropBank Annotation Goals

PropBank is an annotation of syntactically parsed, or treebanked, structures with `predicate-argument' structures. An important goal is to provide consistent argument labels across different syntactic realizations of the same verb, as in

> [*ARG0* John] broke [*ARG1* the window]
> [*ARG1* The window] broke

As this example shows, the arguments of the verbs are labeled as numbered arguments: Arg0, Arg1, Arg2 and so on.

The second task of the PropBank annotation involves assigning functional tags to all modifiers of the verb, such as manner (MNR), locative (LOC), temporal (TMP) and others:

> *Mr. Bush met him privately, in the White House, on Thursday.*

> Rel: met
> Arg0: Mr. Bush
> Arg1: him
> ArgM-MNR: privately
> ArgM-LOC: in the White House
> ArgM-TMP: on Thursday

And, finally, PropBank annotation involves finding antecedents for 'empty' arguments of the verbs, as illustrated below:

> *I made a decision [*] to leave.*

The subject of the verb 'leave' in this example is represented as an empty category [*] in Treebank. In Propbank, all empty categories which could be co-referred with a NP within the same sentence are linked in 'co-reference' chains:

> *Rel: leave*
> *Arg0:   [*] -> I*

These three tasks of Propbank annotation: argument labeling, annotation of modifiers, and creating co-reference chains for empty categories are discussed in detail below.

2. Task 1: Argument Labeling

2.1 Frame Files
The argument labels for each verb are specified in the frame files, which are available at http://verbs.colorado.edu/framesets/. Frame files provide verb-specific description of all

possible semantic roles, as well as illustrate these roles by examples.

> *Frame File for the verb 'expect':*
>
> Roles:
> Arg0: expecter
> Arg1: thing expected
>
> Example: Transitive, active:
> *Portfolio managers expect further declines in interest rates.*
>
> Arg0:   Portfolio managers
> REL: expect
> Arg1:   further declines in interest rates

For some verbs, it is impossible to provide one set of semantic roles for all senses of the verb. For example, the two senses of the verb 'leave' in the examples below take different arguments:

> *Mary left the room*
> *Mary left her daughter-in-law her pearls in her will*

In such cases, frame files distinguish two or more verb senses, which are called Framesets, and define argument labels specific to each Frameset:

> *Frameset leave.01 "move away from":*
> Arg0: entity leaving
> Arg1: place left
>
> *Frameset leave.02 "give":*
> Arg0: giver
> Arg1: thing given
> Arg2: beneficiary

When annotating, annotators first select the frameset and then assign the argument labels as specified for this frameset. Please note that the annotation tool allows you to see the semantic roles and one example for the first frameset, but it is absolutely necessary to check the frame files to see if the verb has more than one frameset.

In some cases, frame files define not only several framesets for each verb, but also several predicates. If a verb has a particle (marked as PRT in TreeBank), then it is being considered as a different predicate, and has a different set of semantic roles. For example, the frame file for the verb 'keep' defines three predicates: predicate 'keep' (which has 3 framesets), and predicates 'keep_up' and 'keep_on'. The following example illustrates the definition of the predicate 'keep_up'. Note that the relation (REL) in PB annotation should include both the verb and the particle (which should be selected as one node, if possible, or as a

concatenated constituent (i.e. [keep][up]), if one node is not available in Treebank).

Frameset: keep_up:

keep.05 "keep up: maintain position":
Arg0: maintainer of position
Arg1: relative to what

John can't keep up with Mary's rapid mood swings.
Arg0: John
Argm-MOD: ca
Argm-NEG: n't
REL: keep up
Arg1: with Mary's rapid mood swings

2.2 Choosing Arg0 versus Arg1

In most cases, choosing an argument label is straightforward, given the verb specific definition of this label in the frame files. However, in some cases, the decision needs to be made concerning choosing Arg0 or Arg1 labels.

The Arg0 label is assigned to arguments which are understood as agents, causers, or experiencers. The Arg1 label is usually assigned to the patient argument, i.e. the argument which undergoes the change of state or is being affected by the action.

Arg0 arguments (which correspond to external arguments in GB theory) are the subjects of transitive verbs and a class of intransitive verbs called unergatives.

John (Arg0) sang the song.
John (Arg0) sang.

Semantically, external arguments have what Dowty 1991 called Proto-Agent properties, such as
- volitional involvement in the event or state
- causing an event or change of state in another participant
- movement relative to the position of another participant

Internal arguments (labeled as Arg1) are the objects of transitive verbs and the subjects of intransitive verbs called unaccusatives:

John broke the window (Arg1)
The window (Arg1) broke

These arguments have Proto-Patient properties, which means that these arguments
- undergo change of state

- are causally affected by another participant
- are stationary relative to movement of another participant

Whereas for many verbs, the choice between Arg0 or Arg1 does not present any difficulties, there is a class of intransitive verbs (known as verbs of variable behavior), where the argument can be tagged as either Arg0 or Arg1.

A bullet-Arg1 landed at his feet
He-Arg0 landed

Arguments which are interpreted as agents should always be marked as Arg0, independent of whether they are also the ones which undergo the action.

In general, if an argument satisfies two roles, the highest ranked argument label should be selected, where Arg0 >> Arg1 >> Arg2>>… .

Given this rule, agents are ranked higher than patients. If an argument is both an agent and a patient, then Arg0 label should be selected.

Not all Arg0s are agentive, however. There are many inanimate as well as clausal arguments which are being marked as Arg0s. These arguments are usually the ones which cause an action or a change of state.

A notion which might be useful for selecting Arg0 arguments is the notion of 'internally caused' as opposed to 'externally caused' eventualities, as defined in Levin and Rapapport 1995. In internally-caused eventualities, "some property inherent to the argument of the verb is responsible for bringing about the eventuality… For agentive verbs such as *play*, *speak*, or *work*, the inherent property responsible for the eventuality is the will or volition of the agent who performs the activity. However, an internally caused eventuality need not be agentive. For example, the verbs *blush* and *tremble* are not agentive, but they, nevertheless, can be considered to denote internally caused eventualities, because these eventualities arise from internal properties of the arguments, typically an emotional reaction… In contrast to internally caused verbs, verbs which are externally caused inherently imply the existence of an external cause with an immediate control over bringing about the eventuality denoted by the verb: an agent, and instrument, a natural force, or a circumstance. Thus something breaks because of the existence of some external cause; something does not break because of its own properties" /Levin and Rappaport 1995/. The difference between internal and external causation is important for distinguishing Arg0s and Arg1s: the arguments which are responsible for bringing out the eventuality are Arg0s, whereas those which undergo an externally caused event are Arg1s.

To sum up, Arg0 arguments are the arguments which cause the action denoted by the verb, either agentively or not, as well as those which are traditionally classified as experiencers, i.e. the arguments of stative verbs such as love, hate, fear. Arg1 arguments, on the other hand, are those that change due to external causation, as well as other types of 'patient'-like arguments.

2.3 Annotation of null elements

The inventory of null elements used in Penn Treebank is as follows (see guidelines for English Treebank in www.cis.upenn.edu/~treebank).

[*T*]   (trace of A-movement, including parasitic gaps)
[(NP *)] (arbitrary PRO, controlled PRO, and trace of A-movement)
[0]   (null complementizer, including null wh-operator)
[*U*]   (unit)
[*?*]   (placeholder for ellipsed material)
[*NOT*]   (anti-placeholder in template gapping)
[*RNR*]   (pseudo-attach: right node raising)
[*ICH*]   (pseudo-attach: interpret constituent here)
[*EXP*]   (pseudo-attach: expletive)
[*PPA*]   (pseudo-attach: permanent predictable ambiguity)

This section presents some examples of most commonly used null elements and their Propbank annotation.

2.3.1 Passive sentences

Sentences can be either active (The executive committee <u>approved</u> the new policy) or passive (The new policy <u>was approved</u> 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.

Passive sentences are assumed to be derived from the corresponding active sentences by 'movement' of the object to the subject position. This movement leaves a trace, represented as [*T*] in Treebank.

>   Active: Mary hit John
>   Passive: John was hit [*T*] by Mary.

Since the goal of the Propbank annotation is to mark the 'maximal extent' of the arguments, in the case of passive sentences what is being labeled as Arg1 is the chain [*T*] -> John.

>   *Propbank annotation:*
>   rel: hit
>   Arg1:   [*T*] -> John
>   ArgM-MNR: hard
>   Arg0:   by Mary

In most cases, the link between the trace and the NP is already provided by the Treebank annotation, which is indicated by the number '1' in the trace (NP-3 *-1) and (NP-SBJ-1 he)

below.[1]

Treebank annotation:
(S **(NP-SBJ-1 he)**
   (VP was
      (VP accused
         (NP-3 *-1)
         (PP-CLR of
             (S-NOM (NP-SBJ *-3)
                (VP (VP conducting
                      (NP illegal business))
                   and
                   (VP possessing
                      (NP illegal materials))))))))

Propbank annotation:
Arg1: [NP-3 *-1] -> he
Rel: accused
Arg2: of [*3*] conducting illegal business and possessing illegal materials

2.3.2. Fronted and dislocated arguments:

Other examples of moved constituents are fronted or otherwise dislocated arguments and adjuncts. As in the other cases of movement, fronted elements leave a trace, which should be coindexed with the moved constituent.

In the following example, the Arg2 argument of the verb 'put' is being fronted. In the Propbank annotation, this is indicated by the chain which links the trace [*T*-1] with the adverbial 'There':

Treebank annotation:
(S (**ADVP-PUT-TPC-1 There**)
   ,
   (NP-SBJ I)
   (VP put
      (NP the book)
      (**ADVP-PUT *T*-1**)))

Propbank annotation:
rel: put
Arg0: I
Arg1: the book
Arg2: [*T*-1] -> There

---

[1] When Treebank provides a link, it can be reconstructed in rats by using the command 't'.

Modifiers, or ArgMs, can be fronted as well, as the following example shows:

Treebank annotation:
(S (**SBAR-PRP-TPC-9 Because**
           (S (NP-SBJ I)
            (VP 'm
               (NP-PRD such a bad boy))))
    (NP-SBJ I)
    (VP think
      (SBAR 0
        (S (NP-SBJ I)
          (VP wo n't
            (VP get
              (NP a lollipop)
              (SBAR-PRP *T*-9))))))

Since the 'because' clause modifies the verb 'get' in this example, the trace originates as the modifier of 'get', and should be coindexed with the fronted phrase:

Propbank annotation:
rel: get
Arg1:   a lollipop
Arg0:   I
ArgM-NEG: n't
ArgM-MOD: wo
Argm-CAU: [*T*-9] -> Because I'm such a bad boy

In rare situations, movement does not leave a trace, but rather leaves a pronoun (called a resumptive pronoun). In such cases no chains are created, but rather the pronoun is being labeled as an argument, whereas a dislocated NP is being marked as ArgM-DIS:

Treebank annotation:
  (S (NP-TPC John)
    ,
    (NP-SBJ I)
    (VP like
      (NP him)
      (NP-ADV a lot)))

Propbank annotation:
Rel: like
Arg0: I
Arg1: him
ArgM-MNR: a lot
ArgM-DIS: John

2.3.3. Questions and wh-phrases

Another type of traces is a trace of a wh-phrase in questions.

What do you like?

As in the case of passive sentences, questions are assumed to be derived by movement. In the example below, the Arg1 argument of the verb 'like' is a wh-phrase 'what', which moves from the object position of the verb to the front of the sentence. This movement leaves a trace, as shown below:

*What do you like [*T*]?*

In Propbank, the argument Arg1 is a chain which links the trace and the moved NP:

Propbank annotation:
Rel: like
Arg0: you
Arg1: [*T*] -> What

In Treebank annotations, wh-phrases are marked as WHNP. As in the case of passive sentences, Treebank provides a link between the trace and the moved WHNP, which can be reconstructed in rats by using 't' command:

Treebank annotation:
 (SBARQ (**WHNP-1 what**)
     (SQ do
        (NP-SBJ you)
        (VP like
          (**NP *T*-1**)))

Wh-phrases are not necessarily arguments. Questions can be formed with wh-phrases like when, where, how, in which case they should be tagged as ArgMs.

Treebank annotation:
SBARQ (**WHNP-1 Which day**)
     (SQ did
        (NP-SBJ you)
        (VP get
          (ADVP-DIR there)
          (**NP-TMP *T*-1**)))

Propbank annotation:
Arg1:   you
rel: get
ArgM-TMP: [*] -> which day

Treebank annotation:
```
(SBARQ (WHADVP-42 How)
    (SQ did
       (NP-SBJ you)
       (VP fix
          (NP the car)
          (ADVP-MNR *T*-42)))
    ?)
```

Propbank annotation:
rel: fix
Arg0: you
Arg1: the car
ArgM-MNR: [*T*-42] -> [How]

Questions can also be embedded, as in the example below. Propbank annotation is not different from direct questions in this case:

John didn't know where his parents had met [*].

Arg0:   his parents
rel: met
ArgM-LOC: [*] -> where

2.3.4. Relative clauses

Relative clauses are clauses which modify a N or a NP as in 'answers that we'd like to have'. Relative clauses also include a trace, which is coindexed with the relative pronoun in Treebank (e.g. 'that'/'which'/'who'). Propbank annotation differs from the Treebank annotation in that it also provides a link to the antecedent NP.

For example, in the following Treebank annotation, the Arg1 argument of the verb 'have' is the NP 'answers'. The object position of the verb has a trace (NP *T*-6), which is being coindexed with the relative pronoun (WHNP-6 that/which). Propbank annotation reconstructs this chain, plus adds a link to the NP 'answers':

Treebank annotation:
```
(NP (NP answers)
    (SBAR (WHNP-6 that/which)
       (S (NP-SBJ-3 we)
          (VP 'd
             (VP like
                (S (NP-SBJ *-3)
                   (VP to
                      (VP have
                         (NP *T*-6)))))))))
```

Propbank annotation:
Arg1:   [NP *T*-6] -> which -> answers
rel: have
Arg0:   [NP-SBJ *-3] -> we

Likewise, if a relative clause modifiers a temporal or a locative, the chain should be constructed which links the trace, the relative pronoun 'when' or 'where' and the NP which specifies time or location:

*John found the place where his parents had met *.*
 Arg0:   his parents
 rel: met
ArgM-LOC: [*] -> where -> the place

In some cases, a relative pronoun is missing in a relative clause (as in: 'answers we would like to have'). Such clauses have a null complementizer `0' inside SBAR in Treebank annotations:

 Treebank annotation:
 (NP (**NP answers**)
  (**SBAR (WHNP-3 0)**
   (S (NP-SBJ-4 we)
    (VP 'd
     (VP like
      (S (NP-SBJ *-4)
       (VP to
        (VP have
         (**NP *T*-3**)))))))))

In such cases, the null complementizer should be included in the chain, instead of a relative pronoun:
 Propbank annotation:
Rel: have
Arg1: [*T*-3] -> [WHNP-3 0] -> answers
Arg0: we

A similar analysis applies to infinitival relatives:

 Treebank annotation:
 (NP (**NP a movie**)
  (**SBAR (WHNP-1 0)**
   (S (NP-SBJ *)
    (VP to
     (VP see
      (**NP *T*-1**))))))

Propbank annotation:
Rel: see
Arg1: [*T*-1] -> [WHNP-1 0] -> a movies
Arg0: [SBJ *]

2.3.5. ICH traces (ICH: interpret constituent here)

ICH traces are being used in Treebank to indicate a relationship of constituency between elements separated by intervening material. An example of such 'split constituents' are `heavy shift' constructions, illustrated below:

> *Treebank annotation:*
> (S **(NP-SBJ (NP a young woman)**
>   **(SBAR *ICH*-1**))
>   (VP entered
>   **(SBAR-1 (WHNP-2 whom)**
>   **(S (NP-SBJ she)**
>   **(PP-TMP at**
>   **(ADVP once))**
>   **(VP recognized**
>   **(NP *T*-2)**
>   **(PP-CLR as**
>   **(NP Jemima Broadwood)))))))**

The subject NP in this case is being split into two constituents: the NP 'a young woman' and SBAR: 'whom she at once recognized as Jemima Broadwood'. The ICH trace specifies a link to the SBAR node in this example.

All examples of this type should be annotated as concatenated constituents (using the command 'A' in rats), including the ICH trace:

> *Propbank annotation:*
> Arg0: [a young woman [*ICH*-1]][whom she recognized as Jemima Broadwood]
> Rel: entered

Other typical examples of *ICH* traces are shown below:

> *[Five *ICH*-1] ran, [out of the twenty-five that showed up].*

> Arg0:   [Five *ICH-1*][out of the twenty-five that showed up]
> rel: ran

> *Some people in Paris want to hear more [*ICH*] from me than those fellers over at the conference house do.*

Rel: hear
Arg1: [more [ICH]][than those fellers over at the conference house do]
Arg2: from me

2.3.6. Right Node Raising (RNR) traces

 RNR traces are used when a constituent is interpreted simultaneously in more than one place. An example of a right node raising structure is given below:

>*Treebank annotation:*
>(NP (**NP (ADJP so many) enchained demons)**
>   (VP straining
>   (PP-MNR in
>   (NP anger))
>   (S (NP-SBJ *)
>   (VP to
>   (VP (VP tear
>   (**NP *RNR*-1))**
>   and
>   (VP gnaw
>   (PP-CLR on
>   (**NP *RNR*-1)))**
>   (NP-1 his bones))))))

In this example, the NP 'his bones' is interpreted as both the argument of the verb 'tear' and the verb 'gnaw'. When annotating the verb 'tear', the chain should be constructed which links the trace (NP *RNR*-1) and the NP:

>*Propbank annotation:*
>Rel: tear
>Arg1: [*RNR*] -> his bones
>Arg0: [*] -> so many enchained demons

Likewise, when annotating the verb 'gnaw', the chain should be constructed which links (PP-CLR on (NP *RNR*-1))) (see section 2.3.9 for annotation of prepositional phrases with traces) and the NP:

>*Propbank annotation:*
>Rel: gnaw
>Arg1: on [*RNR*] -> his bones
>Arg0: [*] -> so many enchained demons

A similar annotation applies when the [*RNR*] trace is a clausal argument:

I want *RNR* and like *RNR* [* to eat ice-cream].

Arg0:   I
rel: want
Arg1:   *RNR* -> [* to eat ice-cream]

If the RNR trace is part of the argument of the verb, then use concatenation instead (as always, including the trace):

His dreams had revolved around her so much and for so long that...

Treebank annotation:
```
(S (NP-SBJ His dreams)
   (VP had
      (VP revolved
         (PP-CLR around
             (NP her))
      (UCP-ADV (ADVP (ADVP so much)
              (SBAR *RNR*-1))
            and
            (PP-TMP for
                (NP (NP so long)
                   (SBAR *RNR*-1)))
      (SBAR-1 that...)))))
```

Propbank annotation:
Arg1:   his dreams
rel: revolved
ArgM-LOC: around her
ArgM-MNR: [so much [*RNR]][that …]

The following example illustrates annotation of RNR traces within a small clause (for annotation of small clauses see section 2.4.2 below). There are three concatenated constituents in this case:

But our outlook has been and continues to be defensive

```
(S But
   (NP-SBJ-2 our outlook)
   (VP (VP has
         (VP been
             (ADJP-PRD *RNR*-1)))
      ,
      and
      (VP continues
         (S (NP-SBJ *-2)
           (VP to
              (VP be
                 (ADJP-PRD *RNR*-1)))))
      ,
      (ADJP-PRD-1 defensive)))
```

Propbank annotation:
rel: continue
Arg1: [our outlook][[*-2] to be *RNR-1][defensive]

2.3.7. *EXP* (``EXPletive'')

Expletives like 'it' do not add any meaning to the sentence. In the following example, the syntactic subject of the sentence is an expletive, which includes a trace EXP-1. This trace refers to the logical subject of the sentence, marked as SBAR-1:

Treebank annotation:
```
(S (NP-SBJ (NP It)
        (SBAR *EXP*-1))
   (VP is
      (ADJP-PRD clear)
      (PP to
         (NP me))
      (SBAR-1 that
            (S (NP-SBJ this message)
              (VP is
                 (ADJP-PRD unclear))))))
```

In Propbank annotations, expletives and EXP traces are NOT INCLUDED:

Propbank annotation:
Rel: clear
Arg1: that this message is unclear

Another example:

It required an energy he no longer possessed to be satirical about his father.

Propbank annotation:
Arg0: to be satirical about his father
Arg1: an energy he no longer possessed
Rel: required

In the examples below, the expletives are the objects, rather than the subjects. As in the case of expletive subjects, only the logical argument is being tagged, whereas an expletive and an EXP trace are not part of the Propbank annotation:

Mrs. Yeargin was fired [-1] and prosecuted [*-1] under an unusual South Carolina law that [*T*-79] makes it [*EXP*-2] a crime [*] to breach test security.*

Propbank annotation:
 ARG0:   [*T*-79] -> that -> an unusual South Carolina law
 rel: makes
ARG2: a crime
ARG1: [*] to breach test security

*Any raider would find it [*EXP*-1] hard [*] to crack AG 's battlements.*

Treebank annotation:
(S
    (NP-SBJ (DT Any) (NN raider))
    (VP (MD would)
     (VP (VB find)
      (S
       (NP-SBJ
         (NP (PRP it))
         (S **(-NONE- *EXP*-1)**))
       (ADJP-PRD (JJ hard))
       (S-1
         (NP-SBJ (-NONE- *))
         (VP (TO to)
          (VP (VB crack)
           (NP
             (NP (NNP AG) (POS 's))
             (NNS battlements))))))))
    (. .))

Propbank annotation:
ARG0:   Any raider
ARGM-MOD: would
rel: find
ARG3:   hard
Arg1:   [*] to crack AG 's battlements

Common mistake: Please make sure to distinguish an expletive 'it' from the referring pronoun 'it', where 'it' refers to a previous NP, a clause, or an event. (hint: referring pronouns are not being followed by an EXP trace in Treebank). All referring pronouns, including 'it', should be marked as arguments in Propbank.

It sounds good.
Rel: sound
Arg1:   it
ArgM-MNR: good

Italy 's Foreign Ministry said [0] it is investigating exports to the Soviet Union.
Rel: investigating
Arg0: it
Arg1: exports to the Soviet Union

2.3.8. Other traces

Other types of traces include null complementizer trace, ? trace (used in ellipsis constructions), and *PPA* trace in cases of predictable ambiguous attachments.

Null complementizer trace should be included as part of the clausal argument.

Treebank annotation:
 (S (NP-SBJ I)
    (VP believe
       (SBAR 0
           (S (NP-SBJ you)
             (VP are
                (ADJP-PRD smart))))))

Propbank annotation:
rel: believe
Arg0: I
Arg1: [[O] you are smart]

In cases of ellipsis, ? trace can be reconstructed if the sentence is a conjunction:

```
(S (S (NP-SBJ Robin)
      (VP likes
          (NP ice cream)))
   , and
   (S (NP-SBJ Kim)
     (VP does
         (VP *?*

            ,
           (ADVP too)))))
```

Multiple propositions are being used in this case (see section 3.14.3 for more discussion of multiple propositions). The ? trace is being reconstructed as a relation 'likes', introducing a second proposition:

Propbank annotation (2 propositions):

Rel: likes
Arg0: Robin
Arg1: ice cream

Rel: likes
Arg0: Kim
Arg1: ice cream
ArgM-DIS: too

However, if the antecedent of the ? trace is not in the conjoined clause, then it should not be annotated:

Treebank annotation:
```
(S (S (NP-SBJ She)
      (ADVP-TMP rarely)
      (VP sings))
    ,
    so
   (S (NP-SBJ I)
     (VP do n't
         (VP think
             (SBAR 0
                 (S (NP-SBJ she)
                   (VP will
                       (VP *?*
                          (NP-TMP tonight)))))))))
```

2.3.9. Prepositional phrases with traces

The following sentence illustrates a NP trace within a prepositional phrase (PP). The argument ARG1 is the PP in this sentence. The desired interpretation of this sentence should look like [for [[*-2] -> kid's car]], however the tool does not allow to create a chain within an argument.

> *Propbank annotation:*
> (S (**NP-SBJ-2 (NP kid 's) cars)**
>   (VP are
>   (ADVP-TMP often)
>   (VP paid
>   (**PP-CLR for**
>   (**NP *-2))**
>   (PP by
>   (NP-LGS their parents)))))

The solution which has been adopted (mostly for practical reasons) is to link the PP with a trace and the extracted NP, as shown below:

> Rel: paid
> Arg0:   by their parents
> Arg1:   [for *-2] -> kid's cars
> ArgM-TMP: often

2.4. Special cases

2.4.1. Verbs of Saying

A verb of saying is any verb which has a speaker argument (Arg0) and an utterance (Arg1). If the utterance argument can be selected as one constituent, then Arg1 is a single constituent, or a chain, as shown below:

> *Treebank Annotation*
> (S ``
>   (**S-TPC-1 (NP-SBJ We)**
>   (**VP will**
>   (**VP win**)))
>   ,
>   "
>   (NP-SBJ Mary)
>   (VP said
>   (**S *T*-1))**
>   .))

Propbank Annotation
Rel: said
Arg1: [*T*-1] -> We will win
Arg0: Mary

Unfortunately, in many examples, the utterance does not correspond to one constituent in Treebank. In such cases, all pieces of the utterance should be concatenated, including the trace in the object position of the verb.

Among other things , they said [*T*-1] , Mr. Azoff would develop musical acts for a new record label .

Treebank annotation:
```
(S-1
  (PP (IN Among)
    (NP (JJ other) (NNS things) ))
  (PRN
   (, ,)
   (S
     (NP-SBJ (PRP they) )
     (VP (VBD said)
       (SBAR (-NONE- 0)
         (S (-NONE- *T*-1) ))))
   (, ,) )
  (NP-SBJ (NNP Mr.) (NNP Azoff) )
  (VP (MD would)
    (VP (VB develop)
     (NP
       (NP (JJ musical) (NNS acts) )
       (PP (IN for)
         (NP (DT a) (JJ new) (NN record) (NN label) )))))
  (. .) )
```

Propbank annotation:
ARG1:   [Among other things] [Mr. Azoff] [would develop musical acts for a new record label] [[*T*-1]]
ARG0:   they
rel: said

Punctuation generally should NOT be included, unless it is embedded in the constituent, so that it is impossible to select a constituent without punctuation.

`` By the end of the 1990s , " he said [*T*-4] , `` we want [-2] to be producing roughly two vehicles [*ICH*-3] overseas for every vehicle that we export [*T*-1] from Japan . "*

ARG1:   [By the end of the 1990s] [we] [want [*-2] to be producing roughly two vehicles [*ICH*-3] overseas for every vehicle that we export [*T*-1] from Japan] [*T* -4]
ARG0:   he
rel: said

2.4.2. Small clauses and sentential complements.

This section is concerned with different types of clausal complements and modifiers. For example, verbs like 'consider' are given a small clause analysis in Treebank, which means that the syntactic complement of the verb 'consider' is a clause, marked as S. Propbank frame files do not always follow Treebank in this respect. In the frame files for the verb 'consider', for example, this verb has three semantic arguments, so that the clause S is decomposed into 2 arguments Arg1 and Arg2:

Treebank annotation:
(S (NP-SBJ I)
   (VP consider
      (S (NP-SBJ Kris)
         (NP-PRD a fool))))

Propbank annotation:
rel: consider
Arg0:   I
Arg1:   Kris
Arg2:   a fool

Propbank annotators should follow the frame files, and decompose S clauses into smaller constituents in such cases.

If a verb is passivized, the Arg1 argument is the chain linking the trace in the subject position of the small clause with the subject of the verb 'consider':

Treebank annotation:
 (S **(NP-SBJ-1 Kris)**
    (VP is
       (VP considered
          (S **(NP-SBJ *-1)**
             (NP-PRD a fool))
          (PP by
             (NP-LGS most people)))))

Propbank annotation:
rel: consider
Arg1:   [*1]-> Kris
Arg2:   a fool
Arg0:   by most people

Other examples of verbs with small complements in Treebank, where the small clause is being decomposed in Propbank, are the verbs name, elect, tempt, lead, hold, force, call, deem, schedule, find, ask, make and some others.

```
        (S (NP-SBJ-28 (NP Arthur A. Hatch)
                ,
                (NP 59)
                ,)
           (VP was
              (VP named
                 (S (NP-SBJ *-28)
                    (NP-PRD (NP executive vice president)
                        (PP of
                           (NP the company)))))))
```

Rel: named
Arg1:   [*] -> Arthur A. Hatch, 59,
Arg2:   executive vice president of the company

Small clauses (as well as other sentential complements) should only be decomposed if the frame files specify that the two decomposed constituents are different semantic arguments.

In all other cases, the small clause analysis should be preserved. For example, in the following sentence, the clause S-CLR has a trace in the subject position of 'asleep', which is coindexed with the subject of the verb 'fell' 'I'.

I fell asleep on the floor.

```
S (NP-SBJ-1 I)
   (VP fell
      (S-CLR (NP-SBJ *-1)
          (ADJP-PRD asleep))
      (PP-LOC on
          (NP the lobby floor))))
```

When annotating the verb 'fell', the small clause (marked as S-CLR above) should not be decomposed, which means that the trace [NP*-1] is not part of the argument Arg1 of the verb 'fell'. As the Propbank annotation below shows, the Arg1 argument of 'fell' is the NP 'I', rather than the chain [NP*-1] -> I, whereas the trace [NP*-1] is part of ArgM-PRD:

Propbank annotation:
Rel: fell
Arg1:   I
ArgM-PRD: [NP-SBJ *-1] asleep

NOT:
Rel: fell
Arg1:   [NP*-1] -> I
ArgM-PRD: asleep

Verbs like expect, allow and others are analyzed as having a clause as its argument. In this case, Propbank annotations follow Treebank analyses of these sentences, where the clausal complement is being marked as Arg1:

John expected Mary to come.

Propbank annotation:
Rel: expected
Arg0:   John
Arg1:   Mary to come

If such sentences are passivised, as shown below, then the Arg1 argument is a concatenation of the subject and the clausal complement.

Mary is expected [*] to come

Propbank annotation:
Rel: expected
Arg1:   [Mary][* to come]

Similar analysis applies to verbs like 'seem' and 'appear', which are known as raising verbs. For example, the NP 'everyone' below is not the argument of the verb 'seems', but rather this sentence can be paraphrased as 'It seems that everyone dislikes Drew Barrymore.'

Everyone seems to dislike Drew Barrymore

Treebank annotation:
(S (NP-SBJ-3 Everyone)
   (VP seems
      (S (NP-SBJ *-3)
         (VP to
            (VP dislike
               (NP Drew Barrymore))))))

Since the logical argument of the verb *seems* is a clause, Propbank annotation of this

sentence involves concatenation of the subject and the complement:

> Rel: seems
> Arg1:   [Everyone][* to dislike Drew Barrymore]

And, finally, another class of verbs which follows this analysis is the class of aspectual verbs like *continue* and *start*, which take events as their arguments.

> *New loans continue [*] to slow.*

> Rel: continue
> Arg1:   [New loans][* to slow]

3. Task 2: Annotation of modifiers (ArgMs).

The following types of modifiers are being used in PropBank:

DIR: Directionals
LOC: Locatives
MNR: Manner
EXT: Extent
REC: Reciprocals
PRD: Secondary Predication
PNC: Purpose
CAU: cause
DIS: discourse
ADV: adverbials
MOD: modals
NEG: negation

3.1. Directionals (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, "walk along the road" is a directional, but "walk around the countryside" is a location.
Directional modifiers are also used for some particles, as in back up.

> *Workers dumped large burlap sacks of the imported material into a huge bin , poured in cotton and acetate fibers and mechanically mixed the dry fibers in a process used [*] [*] to make filters .*
> ARG0:   Workers
> REL: dumped
> ARG1:   large burlap sacks of the imported material

ARGM-DIR: into a huge bin

What sector is [*T*-46] stepping forward [*-2] to pick up the slack ? " he asked [*T*-1]
ARG1:   [*T*-46]   →   What sector
REL: stepping
ARGM-DIR: forward
ARGM-PNC: [*-2] to pick up the slack

No one wants the U.S. to pick up its marbles and go home , " Mr. Hormats says [*T*-1] .
ARG1:   the U.S.
REL: go
ARGM-DIR: home

That response annoyed Rep. Markey , House aides said [0] [*T*-1] , and the congressman snapped back that there had been enough studies of the issue and that it was time for action on the matter .
ARG0:   the congressman
REL: snapped
ARGM-DIR: back
ARG1:   that there had been enough studies of the issue and that it was time for action on the matter

3.2. Locatives (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, as '[in his speech]-LOC he was talking about …".

The percentage of lung cancer deaths among the workers at the West Groton , Mass. , paper factory appears [*-1] to be the highest for any asbestos workers studied [*] in Western industrialized countries , he said [0] [*T*-2] .
ARG1:   [*]   →   any asbestos workers
REL: studied
ARGM-LOC: in Western industrialized countries

Areas of the factory [*ICH*-2] were particularly dusty where the crocidolite was used [*-8] [*T*-1] .
ARGM-LOC: [*T*-1]   →   where
ARG1:   [*-8]   →   the crocidolite
REL: used

In his ruling , Judge Curry added an additional $ 55 million [*U*] to the commission 's calculations .

ARGM-LOC: In his ruling
ARG0:   Judge Curry
REL: added
ARG1:   an additional $ 55 million [*U*]
ARG2-to: to the commission 's calculations

3.3. Manner Markers (MNR)

Manner adverbs specify how an action is performed. For example, "works well with others" is a manner. Manner tags should be used when an adverb be an answer to a question starting with 'how?'.

*Among 33 men who [*T*-4] worked closely with the substance, 28 [*ICH*-1] have died -- more than three times the expected number.*
ARG0:   [*T*-4] → who -> 33 men
REL: worked
ARGM-MNR: closely
ARG1-with: with the substance

Workers dumped large burlap sacks of the imported material into a huge bin, poured in cotton and acetate fibers and mechanically mixed the dry fibers in a process used [*] [*] to make filters.
ARG0:   Workers
ARGM-MNR: mechanically
REL: mixed
ARG1:   the dry fibers
ARGM-LOC: in a process used [*] [*] to make filters

The next morning, with a police escort, busloads of executives and their wives raced to the Indianapolis Motor Speedway , [-1] unimpeded by traffic or red lights .*
ARGM-TMP: The next morning
ARGM-MNR: with a police escort
ARG0:   busloads of executives and their wives
REL: raced
ARG1-to: to the Indianapolis Motor Speedway
ARGM-ADV: [*-1] unimpeded by traffic or red lights

3.4. Temporal markers (TMP)

Temporal ArgMs show when an action took place, such as "in 1987", "last Wednesday", "soon" or "immediately". Also included in this category are adverbs of frequency (eg. often always, sometimes (with the exception of 'never', see NEG below), adverbs of duration (for a year/in an year), order (eg. first), and repetition (eg. again)..

*A form of asbestos once used [*] [*] to make Kent cigarette filters has caused a high percentage of cancer deaths among a group of workers exposed [*] to it more than 30 years ago , researchers reported [0] [*T*-1] .*
ARG1:   [*]   -> A form of asbestos
ARGM-TMP: once
REL: used
ARG2-PNC: [*] to make Kent cigarette filters

Four of the five surviving workers have asbestos-related diseases, including three with recently diagnosed cancer.
ARGM-TMP: recently
REL: diagnosed
ARG2:   cancer

3.5. Extent Markers (EXT)

ArgM-EXT indicate the amount of change occurring from an action, and are used mostly for
-        numerical adjuncts like "(raised prices) by 15%",
-        quantifiers such as "a lot"
-         and comparatives such as "(he raised prices) more than she did.".

*PS of New Hampshire shares closed yesterday at $ 3.75 [*U*], off 25 cents, in New York Stock Exchange composite trading .*
ARG1:   PS of New Hampshire shares
REL: closed
ARGM-TMP: yesterday
ARGM-EXT: at $ 3.75 [*U*], off 25 cents,
ARGM-LOC: in New York Stock Exchange composite trading

*``An active 55-year-old in Boca Raton may care more about Senior Olympic games, while a 75-year-old in Panama City may care more about a seminar on health, " she says [*T*-1].*
ARG0:   An active 55-year-old in Boca Raton
ARGM-MOD: may
REL: care
ARGM-EXT: more
ARG1-about: about Senior Olympic games
ARGM-ADV: while a 75-year-old in Panama City may care more about a seminar on health

Rep. Jerry Lewis , a conservative Californian , added a provision of his own, intended [*] to assist Bolivia , and the Senate then broadened the list further by

[*-1] including all countries in the U.S. Caribbean Basin initiate as well as the Philippines - [*-1] backed [*] by the powerful Hawaii Democrat Sen. Daniel Inouye .
ARG0:   the Senate
ARGM-TMP: then
REL: broadened
ARG1:   the list
ARGM-EXT: further
ARGM-MNR: by [*-1] including all countries in the U.S. Caribbean Basin initiate as well as the Philippines
ARGM-ADV: [*-1] backed [*] by the powerful Hawaii Democrat Sen. Daniel Inouye

3.6. Reciprocals (REC)

These include reflexives and reciprocals such as *himself, itself, themselves, together, each other, jointly, both*, which refer back to one of the other arguments.

But voters decided that if the stadium was such a good idea someone would build it himself, and rejected it 59% to 41% [*U*].
ARGM-ADV: if the stadium was such a good idea
ARG0:   someone
ARGM-MOD: would
REL: build
ARG1:   it
ARGM-REC: himself

But while history can suggest what [*T*-1] is reasonable [0] [*] to expect [*T*-2] there 's no guarantee that the past will repeat itself.
ARG1:   the past
ARGM-MOD: will
REL: repeat
ARGM-REC: itself

3.7. 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.

Typical examples include
-      Resultatives: as in 'The boys pinched them dead' or
       'She kicked the locker lid [*] shut
-      Depictives
-      'as'-phrases, e.g. 'supplied as security in the transaction' and other cases of

secondary predication

> *Pierre Vinken , 61 years old , will join the board as a nonexecutive director Nov. 29 .*
> *ARG0:*   *Pierre Vinken , 61 years old ,*
> ARGM-MOD: will
> REL: join
> ARG1:   the board
> ARGM-PRD: as a nonexecutive director
> ARGM-TMP: Nov. 29

> *Prior to his term , a teacher bled to death in the halls , [*-1] stabbed [*-2] by a student .ARGM-TMP:*   *Prior to his term*
> ARG1:   a teacher
> REL: bled
> ARGM-PRD: to death
> ARGM-LOC: in the halls
> ARGM-ADV: [*-1] stabbed [*-2] by a student

> *This wage inflation is bleeding the NFL dry, the owners contend [*T*-1] .*
> ARG0:   This wage inflation
> REL: bleeding
> ARG1:   the NFL
> ARGM-PRD: dry

> *[*-2] Glamorous and pure-voiced as ever , Ms. Collins sang Joni Mitchell 's `` For Free '' -- about an encounter with a street-corner clarinetist , to which Mr. Stoltzman contributed a clarinet obligatto [\*T\*-1] -- and Mr. Douglas 's lush setting of a Gaelic blessing , `` Deep Peace . ''*
> ARGM-PRD: [*-2] Glamorous and pure-voiced as ever
> ARG0:   Ms. Collins
> REL: sang
> ARG1:   Joni Mitchell 's `` For Free '' -- about an encounter with a street-corner clarinetist , to which Mr. Stoltzman contributed a clarinet obligatto [\*T\*-1] -- and Mr. Douglas 's lush setting of a Gaelic blessing , `` Deep Peace

3.8. Purpose clauses (PNC: purpose, not cause)

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

> *More than a few CEOs say [0] the red-carpet treatment tempts them to return to a heartland city for future meetings .*
> ARG1:   them

REL: return
ARG4-to: to a heartland city
ARGM-PNC: for future meetings

In a disputed 1985 ruling , the Commerce Commission said [0] Commonwealth
Edison could raise its electricity rates by $ 49 million [*U*] [*-1] to pay for the
plant .
ARG0:   Commonwealth Edison
ARGM-MOD: could
REL: raise
ARG1:   its electricity rates
ARG2-by: by $ 49 million [*U*]
ARGM-PNC: [*-1] to pay for the plant

3.9. Cause clauses (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':

> *Pro-forma balance sheets clearly show why Cray Research favored the spinoff*
> *[*T*-1] .*
> ARGM-CAU: [*T*-1]   →   why
> ARG0:   Cray Research
> REL: favored
> ARG1:   the spinoff

> *However , five other countries -- China , Thailand , India , Brazil and Mexico --*
> *will remain on that so-called priority watch list as a result of an interim review ,*
> *U.S. Trade Representative Carla Hills announced [0] [*T*-1] .*
> ARGM-DIS: However
> ARG1:   five other countries -- China , Thailand , India , Brazil and Mexico
> --
> ARGM-MOD: will
> REL: remain
> ARG3-on: on that so-called priority watch list
> ARGM-CAU: as a result of an interim review

3.10. Discourse Markers (DIS)

These are markers which connect a sentence to a preceding sentence.
Examples of discourse markers are: *also, however, too, as well, but, and, as we've seen*
before, instead, on the other hand, for instance, etc.

Note that conjunctions such as *but* or *and* are only marked in the beginning of the sentence.

But for now , they 're looking forward to their winter meeting -- Boca in February .
ARGM-DIS: But
ARGM-TMP: for now
ARG0:   they
REL: [looking] [forward]
ARG1-to: to their winter meeting -- Boca in February

Do not mark *and*, *or*, *but*, when they connect two clauses in the same sentence.

Another type of discourse markers includes vocatives, which are marked as VOC in Treebank:

> *Treebank annotation:*
> (S (NP-VOC Kris),
>   (NP-SBJ *)
>   (VP go
>   (ADVP-DIR home)))

> *Propbank annotation:*
> ArgM-DIS: Kris
> Rel: go
> Arg0:   [*]
> ArgM-DIR: home

Vocative NPs in imperative sentences as shown above should not be tagged as chains, i.e. Arg0: [*] -> Kris. in order to make annotation consistent with other examples of Vocative NPs, which do not include traces:

> *I ai n't kidding you, Vince*
> ArgM-DIS: Vince
> Rel: kidding
> Arg0:   I
> Arg1:   you
> ArgM-NEG: n't

And, finally, the class of Discourse markers includes interjections such as 'oh my god' 'ah', and 'damn'

> *I might point out that your inability to report to my office this morning has not ah limited my knowledge of your activities as you may have hoped.*
> ArgM-DIS: ah
> Rel: limited
> Arg1:   my knowledge of your activities
> Arg0:   your inability to report to my office this morning
> ArgM-ADV: as you may have hoped

3.11. Adverbials (ADV)

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.

1.   Temporally related (modifiers of events)
   Treasures are just lying around, <u>waiting to be picked up</u>
2.   Intensional (modifiers of propositions)
   Probably, possibly
3.   Focus-sensitive
   Only, even
4.   Sentential (evaluative, attitudinal, viewpoint, performatives)
   Fortunately, really, legally, frankly speaking,
   clauses beginning with 'given that', 'despite', except for, 'if'

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

In some cases, modifiers like 'happily' can be ambiguous between MNR and ADV interpretations, as shown below:

> *She sang happily.*
> ArgM-MNR: happily

> *Happily, she sang.* (paraphrasable as 'I am happy that she sang')
> ArgM-ADV: happily

3.12. Modals (MOD)

Modals are: *will, may, can, must, shall, might, should, could, would*. "Phrasal modals" such as "going (to)", "have (to)" and "used (to)" are also included, although unlike the regular modals, these are also annotated as verbs in their own right, where they take their own Negation and Adverbial markers, but not any numbered arguments. Thus, in the sentence "John does not have to run", "have" is a modal adjunct of "run", but "not" is a negation adjunct of "have", and not of "run".

3.13. Negation (NEG)

This tag is used for elements such as "not", "n't", "never", "no longer" 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 'never', which could be marked as either TMP or NEG, the NEG tag should be used.

3.14. Stranded (STR)

Stranded tag is not a modifier tag, meaning that it does not mark a certain type of modifier and does not appear in the final, or released version, of Propbank annotations. The use of the tag is motivated by technical reasons. Specifically, STR tags are used when the tool doesn't allow you to combine everything that should be one argument. There are typically two kinds of stranded situations. First, an argument is a chain that points to something that cannot be selected as one node, e.g.:

Arg1:   *1 -> *2 -> that -> [X] [Y] [Z]

Second, the argument involves concatenating a chain with something else, e.g.:

Arg1:   [* -> 0 -> X] [Y] [Z]

In both cases, the chain should be constructed, which refers to the main element from the concatenated constituent. This chain is labeled as a numbered argument ArgN. The remaining constituents are being tagged as ArgN-STR:

*[Two members] [of the group] wanted *1 to continue *2 running.*

Ideally:
Arg0:   *2 -> *1 -> [Two members][of the group]
rel: running

In practice:
Arg0:   *2 -> *1 -> [Two members]
Arg0-STR: [of the group]
rel: running

Likewise,

*John wanted *1 to appear *2 to be funny.*

Ideally:
Arg1:   [* -> John][*2 to be funny]
rel: appear

In practice:
Arg1:   * -> John
Arg1-STR: [*2 to be funny]
rel: appear

Another example:

Italy 's Foreign Ministry said [0] it is investigating exports to the Soviet Union by an

Ing . C. Olivetti & Co. subsidiary called [*] OCN-PPL that [*T*-1] makes numerically controlled machine tools.

ARG0-STR: called [*] OCN-PPL
ARG0:   [*T*-1] -> that -> an Ing . C. Olivetti & Co. subsidiary
rel: makes
ARG1:   numerically controlled machine tools

3.15. Special cases

3.15.1. Modification in complex clauses

When annotating ArgMs, please make sure that they modify the verb being annotated, and not another verb in the sentence. For example, in the following sentence, ArgM-TMP modifies the verb *come*, but not *ask*:

> *[About 5 years ago]-TMP, Handley came to ask me if he could see the tattered register.*

In the next example, ArgM-TMP modifies the verb *interested*, but not *awakened*.

> *It awakened [RNR] and, [for a moment]-TMP, interested [RNR] him*

3.15.2. Multiple modifiers with the same tag

As a general rule, try to follow Treebank constituency, whenever it is possible. This means that ArgMs should not be decomposed or concatenated.

For example, if there is no constituent in Treebank which contains *yesterday* and *at 5 pm*, then they should be analyzed as separate ArgMs:

> *He was in the library yesterday at 5pm*

> ArgM-TMP: yesterday
> ArgM-TMP: at 5 pm

> NOT:
> ArgM-TMP: [yesterday][at 5 pm]

If there is a constituent which contains both modifiers, then such constituent should be selected.

The following examples illustrate two ArgM-Adv and ArgM-DIS in the same sentence:

> *Of all places in Poland Andrei hated Lublin the most.*

ArgM-Adv: Of all places in Poland
ArgM-Adv: the most

NOT:
ArgM-ADV: [Of all places in Poland][the most]

Oh, well, you can't really blame Lolotte.
DIS: Oh
DIS: well

NOT:
ArgM-DIS: [Oh]{well]

Another consequence of the rule 'follow the syntax' is that in the following sentence, 'all three' should be marked as ArgM-ADV, rather than concatenated with the argument 'they'.

They were [all three]-ADV bent over a shabby riding boot.
Rel: bent
Arg1:   they
ArgM-ADV: all three

NOT:
Rel: bent
Arg1:   [they][all three]

Likewise, annotators should not decompose ArgMs which are analyzed as constituents in Treebank, even if they can be semantically decomposed:

[*] To kayo him and maybe or maybe not kill

ArgM-ADV: maybe or maybe not
Rel: kill

A more complicated situation is when there are two arguments with the same semantic role. This usually happens when the argument is a location or a direction. For example, the Arg2 argument of the verbs 'locate' and 'put' is a location. In the following examples, there are two constituents which could be viewed as locations (e.g. 'at the curb' and 'in front of an apartment house' for the first sentence).

The rule of thumb which was adopted for such cases is that the constituent closest to the verb is marked as a numbered argument, and the other one as ArgM:

I located the car parked [*] at the curb in front of an apartment house.

Arg2: at the curb
ArgM-LOC: in front of an apartment house

She put the slipper neatly by its mate at the foot of the bed

    Arg2-LOC: by its mate
    ArgM-LOC: at the foot of the bed

3. 15. 3. Multiple Propositions

Multiple propositions are mostly being used in the case of gapping or ellipsis:

    John not only went to the store but also__ to the bank.

In the second clause, the verb is missing, but the gap is understood as referring to the verb in the first clause. Sentences of this type are annotated as having two propositions:

    Propbank annotation (2 propositions):
    Arg1: John
    ArgM-Dis: not only
    Rel: went
    Arg4: to the store

    Arg1: John
    ArgM-Dis: but also
    Rel: went
    Arg4: to the bank

Some instances of conjunction can also be annotated as having multiple annotations, but only if it is not possible to annotate it as one proposition.

    John went to the store and then to the beach.

In this case, the ArgM-DIS 'then' modifies the second conjunct only, which can only be captured by having 2 propositions:

    Propbank annotation (2 propositions):
    Rel: went
    Arg0: John
    ArgM-DIR: to the store

    Rel: went
    Arg0: John
    ArgM-DIR: to the beach
    ArgM-DIS: then

Do not decompose conjunctions into separate propositions, if a conjoined phrase denotes

an argument or a modifier:

> *John went to the store and to the beach.*

Rel: went
Arg0: John
ArgM-DIR: to the store and to the beach

> *An existentialist is a man who perceives himself only as "esse'', as existence without substance.*

Arg0: who -> a man
Rel: perceives
Arg1: himself
ArgM-PRD: only as "esse'', as existence without substance

4. Task 3: Coreference (for empty categories)

One of the most difficult issues that you will confront when annotating coreference is deciding whether a given NP is an argument of the verb, which should therefore be coindexed in a chain, or an implicit or arbitrary argument.

The following examples illustrate typical cases of annotator errors with respect to coreference annotation:

*Long-debated proposals [*] to simplify the more than 150 civil penalties and make them fairer and easier [0] [*] to administer [*T*-1] are in the House tax bill.*

1. ARG0:   [*] -> proposals
2. rel: make
3. ARG1:   them
4. ARG2-PRD: fairer and easier [0] [*] to administer [*T*-1]

Indexing for the most part has involved [-2] simply buying [*RNR*-1] and then holding [*RNR*-1] stocks in the correct mix [*] to mirror a stock market barometer, such as Standard & Poor 's 500-stock index, and match its performance.*

1. ARG0:   [*-2] -> Indexing
2. ARGM-ADV: simply
3. ARGM-DIS: then
4. rel: holding
5. ARG1:   stocks in the correct mix -> [*RNR*-1]

According to these annotations, 'the proposals are making penalties fairer', and 'indexing is holding stocks in the correct mix', which is certainly not the correct analysis of these

sentences.

The correct analysis is shown, where the argument Arg0 is an arbitrary subject (the empty category in understood as arbitrary PRO), which could be substituted by 'someone' in these examples.

      1. ARG0:   [*]
      2. rel: make
      3. ARG1:   them
      4. ARG2-PRD: fairer and easier [0] [*] to administer [*T*-1]

      1. ARG0:   [*-2]
      2. ARGM-ADV: simply
      3. ARGM-TMP: then
      4. rel: holding
      5. ARG1:   stocks in the correct mix -> [*RNR*-1]

More difficult examples involve ambiguous interpretations, as shown below:

The patent for Interleukin-3 covers materials and methods used [*] [*] to make the human blood cell growth factor via recombinant DNA technology.

Annotator 1:
ARG2:   [*] -> [*] -> materials and methods
rel: make
ARG1:   the human blood cell growth factor
ARGM-MNR: via recombinant DNA technology

Annotator 2:
Arg0: [*] -> [*]
rel: make
ARG1:   the human blood cell growth factor
ARGM-MNR: via recombinant DNA technology

Both annotations are correct interpretations of this sentence, since the trace can either refer to the Arg0 argument (i.e. the agent of making, which is arbitrary or implicit in this case), or the Arg2 argument (which corresponds to the instrument).

The rule adopted for such cases is similar to the rule used for resolution of ambiguous argument labels, and is based on the hierarchy of arguments, as discussed in section 2.2. above. According to this rule, the highest ranked argument should be selected, where Arg0 >> Arg1 >> Arg2 >>... . In this example, the trace should be interpreted as suggested by the annotator 2.