AnnCorra : Building Tree-banks in Indian Languages

Akshar Bharati
Rajeev Sangal
Vineet Chaitanya
Amba Kulkarni
Dipti Misra Sharma
International Institute of Information Technology
Hyderabad, india
{sangal, vc, amba, dipti}@iiit.net

K.V. Ramakrishnamacharyulu
Rashtirya Sanskrit Vidyapeetha, Tirupati, India
kvrk@sansknet.org

ABSTRACT

This paper describes a dependency based tagging scheme for creating tree banks for Indian languages. The scheme has been so designed that it is comprehensive, easy to use with linear notation and economical in typing effort. It is based on Paninian grammatical model.

1.BACKGROUND

The name AnnCorra, shortened for "Annotated Corpora", is for an electronic lexical resource of annotated corpora. The purpose behind this effort is to fill the lacuna in such resources for Indian languages. It will be an important resource for the development of Indian language parsers, machine learning of grammars, lakshancharts (discrimination nets for sense disambiguation) and a host of other such tools.

2. AIMS AND OBJECTIVE

The aim of the project is to :

- develop a generalised linear syntacto- semantic tag scheme for all Indian languages
- annotate training corpus for all Indian languages
- develop parallel tree-banks for all Indian languages

To fulfill the above aim - a marathon task - a collaborative model has been concieved. Any collaborative model implies involvement of several people with varying levels of expertise. This case, becomes further complicated as the tag scheme to be designed has to be equally efficient for all the Indian languages. These languages, though quite similar, are not identical in their syntactic structures. Thus the tag scheme demands the following properties :-

- comprehensive enough to capture various sysntactic relations across languages.
- simple enough for anyone, with some background in linguistics, to use.
- economical in typing effort (the corpus has to be manually annotated).

3. AN ILLUSTRATION

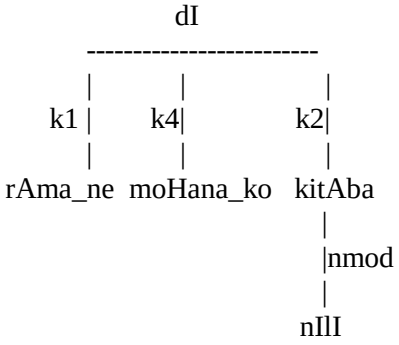
The task can be better understood with the help of an illustration. Look at the following sentence from Hindi

0:: **rAma ne moHana ko**
'Rama' 'ErgPostP' 'Mohan' 'PostP'

nIlI kitAba dI
'blue' 'book' 'gave'

'Rama gave the blue book to Mohan.'

Tree-1 is a representation of the above verb, argument relationship within the various constituents of sentence 0 -



Tree-1

Since the input for tagging is a text corpus and the marking has to be done manually, the tagging scheme is linearly designed. Therefore, Sentence 0 will be marked as follows -

rAma_ne/k1 moHana_ko/k4 [nIlI
'Ram postp' Mohan postp' 'blue '

kitAba]/k2 dI::v
'book' 'gave'

The markings here represent

- **'di'** ('give') is the verb node
- **'rAma_ne'** is the 'karta' or 'agent' (k1) of the verb 'dI',
- **'moHana_ko'** is 'sampraadana' or 'beneficiary' (k4) of verb 'dI' ('give')
- '**[nIlI kitAba]**' – (blue book) a noun phrase - is the 'karma' or 'object' (k2) of the verb.

The elements joined by an underscore represent one unit. Postpositions which are separated by white space in the written texts are actually the inflections of the preceding noun or verb units. Therefore, they are conjoined.

The modifier-modified elements are paranthesised within square brackets. Tags showing the name of the ARC (or branch) are marked by '/' immediately after the constituent they relate to. '/' is followed by the appropriate tagname.

Thus '/' specifies a relationship of a word or constituent with another word or constituent. In this case it is the relationship of verb 'dI' with the other elements in the sentence.

Tags denoting a type of node are marked by '::'. '::v' indicates that 'dI' is a verbal node.

The idea here is to mark only the specific grammatical information. Certain DEFAULT CONVENTIONS are left unmarked. For example, the adjective 'nIlI' ('blue') of 'kitAba' ('book) has been left unmarked in the above example since normally noun modifiers precede the noun they modify (adjectives precede nouns). Such DEFAULT CONVENTIONS save unnecessary typing effort.

4. GRAMMATICAL MODEL

It was quite natural to use Paninian grammatical model for sentence analysis (hence the tagnames) because :-

1) Paninian grammatical model is based on the analysis of an Indian language (Sanskrit) it can deal better with the type of constructions Indian languages have.

2) The model not only offers a mechanism for SYNTACTIC analysis but also incorporates the SEMANTIC information (nowadays called dependency analysis). Thus making the relationships more transparent. (For details refer Bharati (1995).)

Following tags (most of which are based on Paninian grammatical model) have been used in the above example.

k1 : kartaa (subject or agent)
k2 : karma (object)
k4 : sampradaana (beneficiary)
v : kriyaa (verb)

Obviously the task is not an easy one. Standardization of these tags will take some time. Issues, while deciding the tags, are many. Some examples are illustrated below to show the kind of structures which the linear tagging scheme will have to deal with.

4.1. Multiple Verb Sentences

To mark the nouns-verb relations with the above tags in single verb sentences is a simple task. However, consider the following sentence with two verbs :-

1: **rAma ne khAnA khAkara**
am' 'postp' 'food' 'having_eaten'

pAnI pIyA
'water' 'drank'

`Ram drank water after eating the food.`

Sentence 1 has more than two verbs - one non-finite (khAkara) and one finite (piyA). The finite verb is the main verb. Noun 'khAnA' is the object of verb 'khAkara', whereas noun 'pAnI' is the object of verb 'piyA'. 'k2' is the tag for object relation in our tagging scheme. Co-indexing becomes the obvious solution for such multiple relations. Since there are two verbs the tagging scheme allows them to be named as 'i' and 'j' (using notation 'i' and 'j'). By default 'i' refers to the main verb and any successive verb by other characters ('j' in the present case):

rAma_ne khAnA khAkara::vkr:j
'Ram_postp' 'food' 'having_eaten:j'

pAnI piyA::v:i
'water' 'drank:i'

This provides the facility to mark every noun verb relationship.

rAma_ne/k1>i khAnA/k2>j
khAkara::vkr:j pAnI/k2>i piyA::v:i

Fortunately, there is no need to mark it so "heavily". A number of notations can be left out, and the DEFAULT rules tell us how to interpret such "abbreviated' annotation. Thus, for the above sentence, the following annotation is sufficient and is completely equivalent to the above :

rAma_ne/k1 khAnA/k2
khAkara::vkr:j pAnI/k2 piyA::v

Even though there are two verbs, there is no need to name the verbs and refer to them. Two default rules help us achieve such brevity (without any ambiguity) :

(1) karta or k1 kaaraka always attaches to the last verb in a sentence (Thus 'rAma_ne/k1' attaches to the verb at the end).
(2) all other kaarakas except k1, attach to the nearest verb on the right. Thus 'khAnA/k2' attaches to 'khAkara' and 'pAnI/k2' attaches to 'piyA', their respective nearest verbs on the right.

4.2. Compound Units

Sometimes two words combine together to form a unit which has its own demands and modifiers, not derivable from its parts. For example, a noun and verb join together to operate as a single unit, namely as a verb. In the sentence 'rAma (Rama) ne (postp) snAna(bath) kiyA (did)', 'snAna' and 'kiyA' together stand for a verb 'snAna+kiyA' (bathed). Such verbal compounds are like any other verb having their own kaarakas.This sentence would be marked as follows :

rAma_ne/k1 snAna::v+ kiyA::v-
'Ram_postp' 'bath+' 'did-'

`Ram took a bath`

A 'v+' or a 'v-' indicates that the word 'snAna' or 'kiyA' are parts of a whole (a verb in this case). Taken together they function as a single verb unit. Such a device which may appear to be more

powerful was needed to mark the 'single unitness' of parts which may appear separately in a sentence. Thus, the above notation allows even distant words to be treated as a single compound. Such occurrences are fairly common in all Indian languages as illustrated in the following example from Hindi :

snAna::v+ to mEMne/k1
'bath' 'emph' 'I_erg'

subaHa_HI kara_liyA_thA::v-
'morning_emph' 'had_done'

I had bathed (taken a bath) in the morning itself.

'+'and ' - ' help in marking this relation explicitly. (a more detail description of the notation in 5.1)

4.3. Embedded Sentence

Tags are also designed to mark the relations within a complex sentence. Consider the example below where a complete sentence (having verb 'piyA' (drank)) is a kaaraka of the main verb 'kaHA' (said).

moHana ne kaHA ki {rAma
'Mohan' 'postp' 'said' 'that ' {'Rama'

ne pAnI khAnA khAkara
'postp' 'water' 'food' 'having eaten'

piyA}.
'drank}

(Mohan said that Ram drank water after having eaten the food)

The embedded sentence can be first marked as follows -

--------- **{rAma_ne/k1 pAnI/k2>j khAnA/k2 khAkara::vkr piyA::v:j}::s.**

The whole embedded sentence is the 'karma' (object) or k2 of 'piyA' (drank):

The relation of the embedded sentence relation as the object of the main verb is co-indexed in the following way :-

moHana_ne kaHA::v:i ki
'Mohan_postp' 'said' 'that'

rAma_ne/k1 pAnI/k2>j khAnA/k2
'Rama_postp' 'water' 'food'

khAkara::vkr piyA::v:j::s/k2>i
'having_eaten' 'drank'

Thus the device of naming the elements and co-indexing them with their respective arguments can be used most effectively.

5. TAGGING SCHEME

The tagging scheme contains : notations, defaults, and tagsets.

5.1. NOTATION

Certain special symbols such as double colon,underscore, paranthesis etc. are introduced first. Two sets of tags have been provided (to mark the crucial ARC and node information). However, apart from these symbols and tags, some special notation is required to explicitly mark certain disjointed, scattered and missing elements in a sentence. Following notation is adopted for marking these elements :-

5.1. 1. X+ ... X- : disjointed elements

As shown above (4.2), when a single lexical unit composed of more than one elements is separated by other intervening lexical units, its 'oneness' is expressed by using '+' on the first element in the linear order and '-' on the second element. '+' indicates to look ahead for the other part till you find an element with '-'. '-' suggests, 'an element marked '+' is left behind, to which it should get itself attached'.

Example - Verb **'snAna_karanA'** (to bathe) in Hindi can occur disjointedly

snAna to mEMne kiyA_thA
'bath' 'emph' 'I' 'did'

para phira gaMdA Ho_gayA
'but' 'again' 'dirty' 'became'

`Bathe I did , but got dirty again.'

'snAna_karanA' is one verb unit in Hindi. But its two components 'snAna' and 'karanA' can occur separately. Notation 'X+....X-' can capture the 'oneness' of these two elements. So '**snAna.karanA**' (‘bathe’) in the above sentence would be marked as follows :

snAna::v+ to mEMne
'bath' 'emph' 'I'

kiyA_thA::v- para phira gaMdA
'did' but' 'again' 'dirty'

Ho_gayA
'became'

Another example of 'scattered elements' is 'agara .... to' construction of Hindi.

agara tuma kaHate to mEM
'if' 'you' 'said' 'then' 'I'

A_ jAtA
'would_have_come'

`Had you asked I would have come'

‘agara' and 'to' together give the 'conditionality' sense. Though they never occur linearly together they have a 'oneness' of meaning. Their dependency on each other can also be expressed through 'X+....X-' notation.

agara::yo+ tuma kaHto::yo- mEM A_ jAtA (tag 'yo' is for conjuncts)

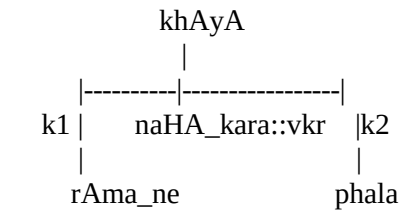
5.1.2. >i ....:i : explicitly marked dependency (:i is the head)

(a) Example -- The sentence 1a below has the dependency structure given in T-2

1a. **phala rAma ne**
'fruit' 'Rama' 'Ergpostp'

naHA_ kara khAyA
'having_bathed' 'ate'

' Rama ate the fruit after taking a bath'



T.2

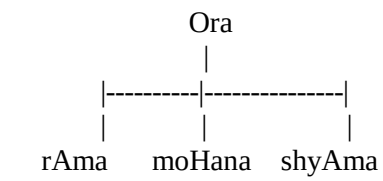
Default (5.2.5) states that all kaarakas attach themselves to the nearest available verb on the right. In (1a) above, the nearest verb available to 'phala' (fruit) is 'naHA_kara'. However, 'phala' (fruit) is not the 'k2' of 'naHA_kara'. It is the 'k2' of the main verb 'khA'. Therefore, an explicit marking is required to show this relationship. The notation '>i...:i' makes this explicit. Therefore,

phala/k2>i rAma_ne naHA_kara khAyA::v:i

Where 'khAyA' is the 'head', thus marked ':i' and 'phala' is the dependent element, thus marked '>i'. An element marked '>i' always looks for another element marked ':i'.

(b) Another example of such attachments which need to be marked explicitly is given below -

2a. **rAma, moHana Ora shyAma**
'Rama', 'Mohan' 'and' 'Shyama'
Ae
'came'



T-3

To show their attachment to 'Ora' (and) the three elements 'rAma','moHana', 'shyAma' have to be marked (as in 2b.) the following way in our linear tagging scheme.

rAma>i, moHana>i Ora::yo:i shyAma>i

The justification to treat 'Ora' as the head and show the 'wholeness' of all the elements joined by '>i' to ':i' is made explicit by the following examples-

rAma, Ora Haz, moHana Ora
'Rama' 'and"yeah', 'Mohana' 'and'

shyAma Ae_ the
'Shyama' 'had_come'

In this case there is an intervening element 'Ora HAz' (‘and_yeah) between 'rAma' and 'moHana' etc. So paranthesis alone will not resolve the issue of grouping the constituents of a whole. (By paranthesising, elements which are not part of the whole will also be included.) To avoid this the 'Ora' (and) has to be treated as a head.

5.1.3. 0 : explicit marking of an ellipted element (missing elements). Example -

rAma bAjZAra gayA, moHana
'Rama' 'market' 'went' 'Mohana'

ghara Ora Hari skUla
'home' 'and' 'Hari' 'school'

‘Rama went to the market, Mohana home and Hari to the school.’

The sentence above has two ellipted elements. The second and third occurrence of the verb 'gayA'(‘went’). To draw a complete tree the information of the missing elements is crucial here. Arguments 'moHana', 'ghara', 'Hari', and 'skUla' are left without a head, and their dependency cannot be shown unless we mark the 'ellipted' element.

rAma bAjZAra gayA, moHana
'Rama' 'market' 'went', 'Mohana'

ghara 0 Ora Hari skUla 0
'home' 'and' 'Hari' 'school'

In cases where this information can be retrieved from some other source (DEFAULT) it need not be marked. In the above case it need not be marked. However, there may be cases where marking of the missing element is crucial to show various relationships. In such cases it has to be marked. Look at the following example -

eka Ora sajjana
'one' 'more' 'gentleman'

kaHate_HEM bacce baDZe
'says' 'children' 'big'

Ho_gaye_ HEM kisI
'become' 'nobody'

kI bAta naHIM mAnate
'gen' 'saying' 'not' 'agree'

' One more gentleman says that the kids have grown older and do not listen to anybody.'

The above sentence does not have any explicit 'yojaka(conjunct)', between two sentences,

a) **bacce baDZe Ho gaye HEM** and `kids have grown older'

b) **kisI kI bAta naHIM mAnate** `do not listen to anybody'

Both these sentences together form the 'vAkyakarma(sentential object)' of the verb 'kaHate HEM' (‘say’).

So the analysis would be -

[eka Ora sajjana]/k1 kaHate_HEM::v:i
‘one’ ‘more’ ‘gentleman’ ‘says’
{{bacce/k1ud baDZe/k1vid
‘children’ ‘big’
Ho_gaye_HEM::v}::s {kisI_kI/6
‘become’ ‘nobody’_s’

bAta]/k2 naHIM::neg
‘words’ ‘not’
mAnate::v}::s}/k2>I
‘listen’

It appears to be a neatly tagged sentence. However, some crucial information is missing from this analysis. In the sentence the relationship between the two sentences within the larger sentential object is not expressed. The problem now is how to do it. Use of '>i...:i' notation can help express this. However, it needs the ':i' information and since there is no explicit 'yojaka' (conjunct) element between the two sentences it will not be possible to mark it. The information of the presence of a 'yojaka' (conjunct) which is the head of a co-ordinate structure is CRUCIAL here. Without its presence its dependency tree cannot be drawn. The notation '0' can be of help in such situations. '0' can be marked in the appropriate place. This will allow the tagging of the dependent elements. Therefore, the revised tagging would be -

[eka Ora sajjana]/k1 kaHate_HEM::v:i

{{bacce/k1ud baDZe/k1vid

ho_gaye_HEM::v}::s>j 0::yo:j

{kisI_kI/6 bAta]/k2 naHIM::neg

mAnate::v}::s>j}/k2>i

Here the information of missing conjunct has been marked by a '0'.

5.2. DEFAULTS

Apart from tagsets and special notations the scheme also relies on certain defaults. Defaults have been specified to save typing by the human annotator. For example, no sentence has to be marked ba a sentence tag till it is crucial for the dependency analysis. For example :

rAma ne yaHa socA ki
'Rama' 'postp' 'this' 'thought' 'that'

moHana AegA
'Mohana' 'would_come'

`Rama thought that Mohana would come'

This is a complex sentence where the subordinate sentence is the object complement of the verb 'socA'(‘thought’) . To indicate the relation of the subordinate clause with the main verb, it has to marked.

Similarly, within the square paranthesis, right most element is the Head. So there is no need to mark it. Postpositions's attachment to the previous noun is also covered by the default rule. There are other defaults which take care of modifier-modified relationships. In short, the general rules have been accounted for by defaults and only the specific relations have to be marked. Elements preceding the head within paranthesis are to be accepted as modifiers of the head. However, In case the number of elements within paranthesis is more than two (Head plus two) and one or more of them do not modify the head then it should be marked.

Example - **[HalkI nIlI kitAba**],
'light' 'blue' 'book'

Here, 'halkI'(‘light’) can qualify both 'nIlI'(‘blue’) and 'kitAba'(‘book’). In case it is modifying 'kitAba'(‘book’), say, in terms of light weight, then it should be left unmarked. But if it modifies 'nIlI'(‘blue’), in terms of light shade, then it SHOULD be marked by adding '>' on the right of the modifying element.

'halkI' [HalkI> nIlI kitAba].
‘light’ [‘light’> ‘blue’ ‘book’]

Let us look at another case where the dependency has to be explicitly marked. Participle form 'tA_HuA', in Hindi, can modify either a noun or a verb. For example take the Hindi sentence -

mEMne/k1 dODZate_Hue::vkr
'I_erg' 'running'

ghoDZe_ko/k2 dekhA::v
'horse' 'saw'

This ambiguous sentence may mean either the following :-
a**) mEMne dODZate_Hue::vkr>i ghoDZe_ko:i/k2 dekhA ;**

'I saw the horse while the horse was running'
Or

b) mEMne dODZate_Hue::vkr>i ghoDZe_ko/k2 dekhA::v:I

'While I was running I saw the horse'

There is no need to mark ':i' in sentence (a). However (b) will need explicit marking.

5.3.TAGSETS

The tagsets used here have been divided into two categories -
1) TAGSET-1 - Tags which express relationships are marked by a preceding '/' . For example kaarakas are grammatical relationships, thus they are marked '/k1', '/k2', '/k3' etc.
2) TAGSET-2 - Tags expressing nodes are marked by a preceding '::' verbs etc. are nodes, so they will be marked '::v',

Certain conventions regarding the naming of the tags are ;
k = kaaraka, -- all the kaaraka tags will begin with k-,
Therefore, k1, k2, k3 etc.
n = noun
v = verb -- eg. v, vkr etc.

6. CONCLUSIONS

A tagging scheme has been designed to annotate corpora for various Indian languages. The objective has been to use uniform tags for all the Indian languages thereby evolving a standard which can be followed for various syntactic analysis for machine processing. The scheme is yet to be yet implemented on corpora from various languages. Some trial workshops have been conducted to see its applicability in other Indian languages. However, once the actual task of tagging begins one may come across cases which are not covered by the present scheme. The idea is to provide a basic scheme which can later be improved and revised.

7. REFERENCES

Bharati, Akshar, Vineet Chaitanya and Rajeev Sangal, "Natural Language Processing: A Paninian Perspective", Prentice-Hall of India,
New Delhi, 1995.

Bharati, Akshar, Dipti M Sharma, Vineet Chaitanya, Amba P Kulkarni,
Rajeev Sangal, Durgesh D Rao, LERIL : Collaborative Effort for Creating
Lexical Resources, In Proc. of Workshop on Language Resources in
Asian Languages, together with 6th NLP Pacific Rim Symposium, Tokyo.