Dependency Annotation Scheme for Indian Languages

Rafiya Begum, Samar Husain, Arun Dhwaj, Dipti Misra Sharma, Lakshmi Bai and Rajeev Sangal
Language Technologies Research Center,
IIIT, Hyderabad, India.
{rafiya,samar}@research.iiit.ac.in,
{dipti,lakshmi,sangal}@iiit.ac.in

Abstract

The paper introduces a dependency annotation effort which aims to fully annotate a million word Hindi corpus. It is the first attempt of its kind to develop a large scale tree-bank for an Indian language. In this paper we provide the motivation for following the Paninian framework as the annotation scheme and argue that the Paninian framework is better suited to model the various linguistic phenomena manifest in Indian languages. We present the basic annotation scheme. We also show how the scheme handles some phenomenon such as complex verbs, ellipses, etc. Empirical results of some experiments done on the currently annotated sentences are also reported.

1 Introduction

A major effort is currently underway to develop a large scale tree bank for Indian Languages (IL). The lack of such a resource has been a major limiting factor in the development of good natural language tools and applications for ILs. Apart from that, a rich and large-scale tree bank can be an indispensable resource for linguistic investigations. Some notable efforts in this direction for other languages have been the Penn Tree Bank (Marcus et al., 1993) for English and the Prague Dependency Bank (Hajicova, 1998) for Czech.

It is well known that context free grammar (CFG) is not well-suited for free-word order languages (Shieber, 1985); instead dependency framework appears to be better suited (Hudson, 1984; Mel'Cuk, 1988, Bharati et al., 1995). Also, the dependency framework is arguably closer to semantics than the phrase structure grammar (PSG) if the dependency relations are judiciously chosen. In recent times many research groups have been shifting to the dependency paradigm due to this reason. Modern dependency grammar is attributed to Tesnière (1959). In a dependency analysis, there is no hierarchical arrangement of phrases (or substrings) like in phrase structure grammar. Rather, we just have words connected via dependency relations between them.

Prague Dependency Bank (PDT) for Czech (which has relatively free word order) is one such large-scale effort which implements a three-tier annotation scheme and annotates morphological information, analytical and tectogrammatical level annotations at these three levels. Out of the three levels, the analytical and tectogrammatical level are dependency based. The tectogrammatical level tries to capture the deep-semantics of the sentence; the annotation at this level is very rich and is linked to the other two lower levels. Other major efforts in the dependency framework are Alpino (van der Beek et. al, 2002) for Dutch, (Rambow et. al, 2002) for English, TUT (Bosco and Lombardo, 2004) for Italian, TIGER (Brants et. al, 2002) (combines dependency with PSG) for German. In this paper we describe an approach to annotate ILs using the Paninian[1] model. The paper is arranged as follows, Section 2 gives a brief overview of the

[1] Paninian theory was formulated by Panini about two thousand five hundred years ago for Sanskrit. It evolved with the contributions of grammarians that followed.

grammatical model and the motivation for following the framework. Section 3 talks about the chosen corpus and the annotation procedure. In Section 4 we discuss some dependency relations. Section 5 describes the evaluation procedure. We report the empirical results of experiments done on the annotated data in Section 6. Section 7, concludes the paper.

2 Grammatical Model

ILs are morphologically rich and have a relatively flexible word order. For such languages syntactic subject-object positions are not always able to elegantly explain the varied linguistic phenomena. In fact, there is a debate in the literature whether the notions 'subject' and 'object' can at all be defined for ILs (Mohanan, 1982). Behavioral properties are the only criteria based on which one can confidently identify grammatical functions in Hindi (Mohanan, 1994); it can be difficult to exploit such properties computationally. Marking semantic properties such as thematic role as dependency relation is also problematic. Thematic roles are abstract notions and will require higher semantic features which are difficult to formulate and to extract as well. So, thematic roles are not marked at this juncture. On the other hand, the notion of *karaka* relations (explained shortly) provides us a level which while being syntactically grounded also helps in capturing some semantics. What is important to note here is that such a level can be exploited computationally with ease. This provides us with just the right level of syntactico-semantic interface. The experiments conducted on the present annotated text provide empirical evidence for this claim (section 6). Paninian grammar is basically a dependency grammar (Kiparsky and Staal, 1969; Shastri, 1973). In this section we briefly discuss the Paninian model for ILs and lay down some basic concepts inherent to this framework.

The main problem that the Paninian approach addresses is to identify syntactico-semantic relations in a sentence. The Paninian approach treats a sentence as a series of modifier-modified relations. A sentence is supposed to have a primary modified (the root of the dependency tree) which is generally the main verb of the sentence. The elements modifying the verb participate in the action specified by the verb. The participant relations with the verb are called *karaka*. The appropriate mapping of the syntactic cues helps in identifying the appropriate karakas ('participants in an action'). The framework is inspired by an inflectionally rich language like Sanskrit; it emphasizes the role of case endings or markers such as post-positions and verbal inflections.

There are six basic karakas, namely; *adhikarana* 'location', *apaadaan* 'source', *sampradaan* 'recipient', *karana* 'instrument', *karma* 'theme', *karta* 'agent'. We must note here that although one can roughly map the first four karaka to their thematic role counterpart, karma and karta are very different from 'theme' and 'agent' respectively (see section 4.1.1).

In our annotation scheme, we use chunks as a device for modularity. A chunk represents a set of adjacent words which are in dependency relations with each other, and are connected to the rest of the words by a single incoming dependency arc. The relations among the words in a chunk are not marked for now and hence allow us to ignore local details while building the sentence level dependency tree. Thus, in our dependency tree each node is a chunk and the edge represents the relations between the connected nodes labeled with the karaka or other relations. All the modifier-modified relations between the heads of the chunks (inter-chunk relations) are marked in this manner. Intra-chunk relations can be marked by a set of rules at a later point. Experiments have been conducted with high performance in automatically marking intra-chunk dependencies.

Using information such as karakas based on some vibhaktis (post-positions) and other information like TAM (tense, aspect and modality) of the main verb seems very well suited for handling free word order languages. Other works based on this scheme like (Bharati et al., 1993; Bharati et al., 2002; Pedersen et al., 2004) have shown promising results. We, therefore, propose the use of dependency annotation based on the Paninian model in the Indian context.

3 Annotation Procedure and Corpus Description

The annotation task is planned on a million word Hindi corpora obtained from CIIL (Central Institute for Indian Languages), Mysore, India. It is a representative corpus which contains texts from various domains like newspaper, literature, government reports, etc. The present subset on which the dependency annotation is being performed has already been manually tagged and chunked. Currently the annotation is being carried out by 2 annotators, who are graduate students with linguistic knowledge. The tool being used for the annotation is part of *Sanchay* (Singh, 2006) which is a collection of tools and APIs for South Asian languages.

4 Scheme

There are a total of 28 relations (see, *http://ltrc.deptagset.googlepages.com/home*) which we encode during the annotation. The total number of relations in the framework is few which has a direct bearing on the parser based on this framework (both, rule based and statistical). We briefly discuss some of these relations in this section.

4.1 Dependency relations

As mentioned earlier, the proposed scheme uses the dependency relations from Paninian grammar. Section 4.1.1 below shows some karaka relations, section 4.1.2 shows some other relations;

4.1.1 *karaka* relations

(1) *raama phala khaataa hai*
'Ram' 'fruit' 'eat' 'is'
'Ram eats fruit'

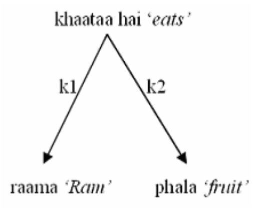

Figure1.

(2) *raama chaaku se saiv kaattaa hai*
'Ram' 'knife' -*inst* 'apple' 'cut' 'is'
'Ram cuts the apple with a knife'

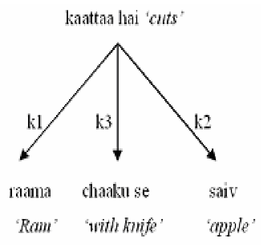

Figure2.

Examples (1), and (2) above show some simple cases which have *karaka* relations such as k3 (karana*;* 'instrument'), k1 (karta), k2 (karma) (the term karta and karma can be roughly translated as 'agent' and 'theme'). One must note here that the notion of karta, karma, etc, is not equivalent to that of the 'agent', 'theme' thematic roles (although they might map to them sometimes). The reason for this divergence in the two notions (karaka and thematic role) is due to the difference in what they convey. Thematic role is purely semantic in nature whereas the karaka is syntactico-semantic. Examples (3), illustrates this point,

(3) chaabhi ne darvaazaa kholaa
'key' '-*erg*' '*door*' '*opened*'
'The key opened the door'

In the above examples *chaabhi* is k1 (*karta*), whereas it takes instrument thematic role. While the karaka relations are primarily motivated via verbal semantics, syntactic cues like postpositions and verbal morphology play an important role too. For example in (3) above, the ergative case '*ne*' provides a strong cue to identify *karta*. Panini defines *'karta'* as *'svatantra karta'* which can be translated as 'the participant which is the most independent in a given action'. In (3) 'key' has such a property. When the speaker uses 'key' in (3), he/she intends to elevate the role of 'key' in the action of opening and does not communicate the actual agent of the action. The speaker uses 'key' as the independent participant in the act of opening. Hence, 'key' is the *karta* (see Bharati et al., 1995, pp. 65-73, for a more detailed discussion).

4.2 Special Cases

(a) **POF (Part of relation)**

Conjunct verbs form a very challenging case for analysis in Indian languages. They have been extensively analyzed in the past. Some notable attempts have been (Greaves, 1983; Kellogg, 1972; Mohanan, 1994; Butt, 2004). The example below shows a N+V conjunct verb usage;

(4) *raama ne mujhase prashna kiyaa*
'Ram' -*erg* 'me-*inst*' 'question' 'do'
'Ram asked me a question.'

In example (4), *prashna kiyaa* is a conjunct verb and behaves as a single semantic unit. These verbs can also be discontiguous as in (5),

(5) *raama ne mujhase prashna pichle saal kiyaa*
'Ram' -*erg* 'me-*inst*' 'question' 'last' 'year' 'did'
'Ram asked me a question last year.'

In the above example above a normal conjunct verb sequence *prashna kiyaa* is disjoint, making it rather difficult to annotate. In fact, practically anything can come between the disjointed elements. Ideally, the noun/adjective + verb sequence of the conjunct verb is placed in one chunk. Keeping this in mind, example (6) below is even more problematic,

(6) *maine usase ek prashna kiyaa*
'I-*erg*' 'him-*inst*' 'one' 'question' 'did'
'I asked him a question'

The noun *prashna* 'question' within the conjunct verb sequence *prashna kiyaa* is being modified by the adjective *ek* 'one' and not the entire noun-verb sequence; the annotation scheme should be able to account for this relation in the dependency tree. If *prashna kiyaa* is grouped as a single verb chunk, it will not be possible to mark the appropriate relation between *ek* and *prashna*. To overcome this problem it is proposed to break *ek prashna kiyaa* into two separate chunks, [*ek prashna*]/NP[2] [*kiyaa*]/VG[3]. The dependency relation of *prashna* with *kiyaa* will be **POF** ('Part OF' relation), i.e. the noun or an adjective in the conjunct verb sequence will have a POF relation with the verb. This way, the relation between *ek* and *prashna* becomes an intra-chunk relation as they will now become part of a single NP chunk. What makes such a sequence unique is the fact that the components which make up a conjunct verb are chunked separately, but semantically they constitute a single unit.

The proposed scheme has the following advantages:

(i) It captures the fact that the noun-verb seque ence is a conjunct verb by linking them with an appropriate tag, this information is extremely crucial syntactically.

(ii) It allows us to deal with the modifier-modified relation between an adjective and its modified noun, as in example (6), which is a frequent phenomenon.

The tree below shows the proposed solution, where the adjective *ek* modifies the noun *prashna* instead of the entire *prashna kiyaa*, which would have been the case had we not separated *prashna kiyaa* into two separate chunks.

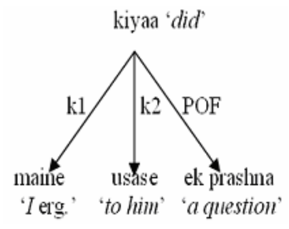

Figure3.

(b) **ccof ('conjunct of' relation) and ellipses**

In the case of coordinating conjunction like *aur* 'and' , the conjunct becomes the root and takes the two conjoined elements as children, the relation marked on the edges is **ccof** (conjunct of). This analysis captures the fact that neither of the conjoined elements is the head. (The head of the two (or more) conjoined elements lies in the conjunct, and may be so computed when needed.) The elements participating in the coordination can belong to various categories, such as, noun, adjective, adverbs etc; they can also be entire clauses, participles, etc. Other conjunct and punctuations which act like conjuncts are annotated similarly.

When one or more element from a sentence is dropped, it is called ellipses. A null element marked with a special tag 'NULL' is introduced in cases of ellipses, where without inserting it the tree cannot be drawn. Null_NP, Null_VG, Null_CCP etc mark different kinds of ellipses.

In this section, we have briefly discussed some of the relations and showed their actual usage using some sentences. The number of tags in the proposed scheme is not very large. A limited set of tags helps immensely in developing high-performance parsers (both rule based and statistical) and other related applications. We should note here that our tag-set, although small, is not a de-

[2] Noun Phrase
[3] Verb Group

limiting factor and is not a compromise on the semantics, as these 28 relations are enough to fully parse the sentences in the language.

5 Evaluation

To make sure that the quality of the annotated corpus is good, the annotators cross-validate each other's work. A senior team member finally checks the annotated corpus (of both the annotators) to ensure that the errors are minimized. Note that such a setup is only temporary, we need such a thorough validation because we are still in the process of revising the guidelines. Once the guidelines become stable, the annotators won't need to cross-validate. Of course, the task of final validation will still continue.

6 Experiments

Some preliminary experiments were conducted on a corpus of 1403 Hindi sentences that have been fully annotated. The aim was to access;

1. Whether the syntactic cues can be exploited for better machine learnability.
2. Whether certain generalization can be made for a constraint parser.
3. How far would the automatic annotation help the annotators?

We found a strong co-relation between most vibhakti-karaka occurrences (shaded cells in Table 1). k7 ('place') for example, overwhelmingly takes *mem* post-position, k3 (*karana*) takes *se* in all the cases. Of course, there are some competing relations which show preference for the same post-position. In such cases only the post-position information will not be sufficient and we need to take into account other syntactic cues as well. These syntactic cues can be TAM (tense, aspect and modality) of the verb, verb class information, etc. For example, in case of *karata karaka* (k1), the following heuristics help resolve the ambiguities seen in Table 1. These heuristics are applied sequentially, i.e. if the first fails then the next follows. Note that the heuristics mentioned below are meant only for illustrative purpose. The cues mentioned in the heuristics will finally be used as features by an efficient ML technique to automate the task of annotation.

H1: k1 agrees in gender, number and person with the verb if it takes a nominative case,
H2: k1 takes a *ko* case-marker if the TAM of the verb has *nA*,
H3: It takes a *kaa/ke/ki* if the verb is infinitive,
H4: It takes a *se* or *dvaara* if the TAM of the verb is passive
H5: It takes a *ne* case-marker if the verb is transitive and the TAM is perfective

Table-2 shows the results when the heuristics were tested on the annotated corpus to test their effectiveness.

V[5] / Kr[4]	kA	se	par	mem	ne	ko
k1	53	5	0	0	344	127
k2	174	70	5	5	0	280
k3	0	51	0	0	0	0
k4	0	9	0	0	0	77
k5	1	97	0	0	0	0
k7	1	4	38	141	0	0
k7p	45	11	72	302	0	0
k7t	13	12	6	31	0	7

Table 1. karaka-vibhakti correlation

	Total	Correct Identified	%
H1	1801	1461	81.1
H2	127	35	27.5
H3	53	19	35.1
H4	10	10	100
H5	344	330	95.9

Table 2. Heuristics for k1 disambiguation

The field 'Total' in Table-2 gives us the number of instances where a particular heuristic was applied. For example, there were 1801 instances where k1 appeared in a nominative case and H1 correctly identified 1461 instances. H1 failed due to the errors caused by the morphological analyzer, presence of conjuncts, etc. Of particular interest are H2 and H3 which didn't work out for large number of cases. It turns out that H2 failed for what is understood in the literature as dative subjects. Dative subjects occur with some specific verbs, one possible solution could be to use such verbs for disambiguation. Automatic identification of conjunct verbs is a difficult problem; in fact, there isn't any robust linguistic test which can be

[4] karaka relations (see, http://ltrc.deptagset.googlepages.com/home)
[5] vibhakti (post-position)

used to identify such verbs. Similar heuristics can be proposed for disambiguating other karaka based on some syntactic cues. Based on the above results one can safely conclude that arriving at some robust generalization (like, karaka-vibhakti correlation) based on the syntactic cues is in fact possible. This can help us immensely in building an efficient parser for Hindi (and other ILs). It goes without saying that there exists a lot of scope for automating the annotation task.

7 Conclusion

In this paper we have introduced an ongoing effort to annotate Indian languages with dependency relation. We stated the motivation behind following the Paninian framework in the Indian Language scenario. We discussed the basic scheme along with some new relations such as ccof, POF, etc. We also showed the results of some experiments conducted on the annotated data which showed that there is a strong co-relation between vibhakti-karaka relations.

Acknowledgement

Thanks are due to Prof. Ramakrishnamacharyulu who has been guiding us throughout the development of the proposed scheme.

References

Akshar Bharati and Rajeev Sangal. 1993. Parsing Free Word Order Languages in the Paninian Framework, *ACL93: Proc. of Annual Meeting of Association for Computational Linguistics*.

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

Akshar Bharati, Rajeev Sangal, T Papi Reddy. 2002. A Constraint Based Parser Using Integer Programming, In *Proc. of ICON-2002: International Conference on Natural Language Processing, 2002*.

Cristina Bosco and V. Lombardo. 2004. Dependency and relational structure in treebank annotation. In *Proceedings of Workshop on Recent Advances in Dependency Grammar at COLING'04*.

S. Brants, S. Dipper, S. Hansen, W. Lezius and G. Smith. 2002. The TIGER Treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*.

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

Edwin Greaves. 1983. *Hindi Grammar*. Asian Educational Services, New Delhi, pp. 335-340

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

R. Hudson. 1984. *Word Grammar*, Basil Blackwell, 108 Cowley Rd, Oxford, OX4 1JF, England.

S. H. Kellogg. 1972. *A Grammar of the Hindi Language*. Munshiram Manoharlal, New Delhi, pp. 271-279.

P. Kiparsky and J. F. Staal. 1969. 'Syntactic and Relations in Panini', *Foundations of Language* 5, 84-117.

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

I. A. Mel'cuk. 1988. *Dependency Syntax: Theory and Practice*, State University, Press of New York.

K. P. Mohanan. 1982. Grammatical relations in Malayalam, *In* Joan Bresnan (ed.), *The Mental Representation of Grammatical Relations*, MIT Press, Cambridge.

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

M. Pedersen, D. Eades, S. K. Amin, and L. Prakash. 2004. Relative Clauses in Hindi and Arabic: A Paninian Dependency Grammar Analysis. In *COLING 2004 Recent Advances in Dependency Grammar*, pages 9–16. Geneva, Switzerland.

O. Rambow, C. Creswell, R. Szekely, H. Taber, and M. Walker. 2002. A dependency treebank for English. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation.*

Anil Kumar Singh. 2006. *http://sourceforge.net/projects/nlp-sanchay*

Charudev Shastri. 1973. *Vyakarana Chandrodya (Vol. 1 to 5)*. Delhi: Motilal Banarsidass. (In Hindi)

S. M. Shieber. 1985. Evidence against the context-freeness of natural language. In *Linguistics and Philosophy*, p. 8, 334–343.

L. Tesnière. 1959. *Eléments de Syntaxe Structurale*. Klincksiek, Paris.

L. van der Beek, G. Bouma, R. Malouf, and G. van Noord. 2002. The Alpino dependency treebank. *Computational Linguistics in the Netherlands*.