Paninian Grammar Framework Applied to English

Akshar Bharati, Medhavi Bhatia, Vineet Chaitanya, Rajeev Sangal
Department of Computer Science and Engineering
Indian Institute of Technology Kanpur
sangal@iitk.ernet.in

February 1996

Abstract

[Published in South Asian Language Review, Creative Books, New Delhi, 1998.]

Computational Paninian Grammar framework (PG) has been successfully applied to modern Indian languages earlier, using which anusaaraka machine translation system has been built (Narayana, 1994). In this paper, we show that PG can also be applied to English resulting in an elegant computational grammar.

First, we generalize the notion of vibhakti to include position of the word in a sentence along with its case and associated preposition, if any. This allows us to use the familiar PG notions of karaka chart, karaka chart transformation, and sharing rules (Bharati et al., 1995) to account for the English actives and passives, lexical control, infinitives, etc. A transformation of the karaka chart and the vibhakti therein, very naturally accounts for what is called movement.

Second, we introduce a new vibhakti called TOPIC position (which corresponds to the first position in a clause) and a new operation called join for connecting a relative clause to its head. These two together handle long distance dependency in relative clauses and wh-questions, raising, tough-movement, pied-piping, etc. The karakas with TOPIC vibhakti appear at the beginning of the clause.

This paper establishes that PG is more general than hitherto considered, and can be used to explain not just free word order languages but also positional languages. Further reseearch is continuing on this and related aspects.

1 PG for Indian Languages - A Review

The Paninian framework considers *information* as central to the study of language. When a writer (or a speaker) uses language to convey some information to the reader (or the hearer), he codes the information in the language string[1]. Similarly, when a reader (or a hearer) receives a language string, he extracts the information coded in it. The computational Paninian Grammar framework (PG) is primarily concerned with: how the information is coded and how it can be extracted.

Two levels of representation can be readily seen in langauge use: One, the actual language string (or sentence), two, what the speaker has in his mind. The latter can also be called as the meaning. Paninian framework has two other important levels: karaka level and vibhakti level (Figure 1).

The surface level is the uttered or the written sentence. The vibhakti level is the level at which there are local word groups based on case endings, preposition or postposition markers.[2]

[1] By string we mean any of: word, phrase, sentence, paragraph, etc.

[2] For positional languages such as English, it would also include position or word order information.

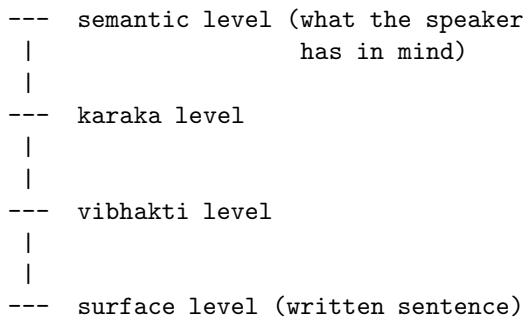


Figure 1: Levels in the Paninian model

A noun group is a unit containing a noun (or a pronoun, proper name, etc.), its vibhakti and possibly some adjectives. Vibhakti for verbs includes the verb form and the auxiliary verbs.

At the karaka level, we have karaka relations and verb-verb relations etc. Karaka relations are syntactico-semantic (or semantico-syntactic) relations between the verbs and other related constituents (typically nouns) in a sentence. They capture a certain level of semantics which is close to thematic relations but different from it. But this is the level of semantics that is important syntactically and is reflected in the surface form of the sentence(s).

The vibhakti level abstracts away from many minor (including orthographic and idiosyncratic) differences among languages. The topmost level relates to what the speaker has in his mind. This may be considered to be the ultimate meaning level. Between this level and vibhakti level is the karaka level. It includes karaka relations and a few additional relations such as taadaarthya (or purpose). One can imagine several levels between the karaka and the ultimate level, each containing more semantic information. Thus, karaka level is one in a series of levels, but one which has relationship to semantics on the one hand and syntax on the other.

As mentioned earlier, vibhakti for verbs can be defined similar to that for the nouns. A head verb may be followed by auxiliary verbs (which may remain as separate words or may combine with the head verb). Such information consisting of the verb ending, the auxiliary verbs, etc. is collectively called vibhakti for the verb. The vibhakti for a verb gives information about tense, aspect and modality (TAM), and is, therefore, also called the TAM label. TAM labels are purely syntactic determined from the verb form and the auxiliary verbs.

The grammar gives the mapping between the levels. For example, PG specifies a mapping between the karaka level and the vibhakti level, and the vibhakti level and the surface form.

It has been shown earlier (Bharati et al., 1995) that the Paninian grammar is particularly suited to free word order languages. It gives a mapping between karaka relations and vibhakti, and uses position information only secondarily. As the Indian languages have (relatively) free word order and vibhakti, they are eminently suited to be described by Paninian Grammar.

1.1 Karaka-Vibhakti Mapping

The most important insight regarding the karaka-vibhakti mapping is that it depends on the verb and its tense aspect modality (TAM) label. The mapping is represented by two structures: default karaka chart and karaka chart transformation. The default karaka chart for a verb or a class of verbs gives the mapping for the TAM label known as basic. It specifies the vibhakti

permitted for the applicable karaka relations[3] for the nouns etc. when the verb has the basic TAM label. In other words, when the verb in a sentence has the basic TAM label, then for each of the nouns in the sentence with a given karaka relation with the verb, their vibhakti is given by the the karaka chart. The basic TAM label chosen for Hindi is *taa_hei* and roughly corresponds to present indefinite tense. Any other TAM label could have been chosen as basic without any problem. (The TAM labels are purely syntactic in nature and can be determined by looking at the verb form and the associated auxiliary verbs, etc.) For TAM labels other than basic, there are karaka chart transformation rules. Thus, for a given verb with some TAM label in a sentence, appropriate karaka-vibhakti mapping can be obtained using its default karaka chart and the transformation rule depending on its TAM label.

The default karaka chart for three of the karakas is given in Figure 1.1. This explains the

Karaka	*Vibhakti*	*Presence*
Karta	ϕ	mandatory
Karma	ko or ϕ	mandatory
Karana	se or dvaaraa	optional

Figure 2: Default karaka chart

vibhaktis in sentences A.1 to A.2 as described below.

Take for example, the following modifier-modified structure:

```
        piiTa (beat) [present]

    k1          k2

    Raama       Mohana
```

Using the karaka chart, we get the following vibhaktis:

raama [ϕ], mohana [ko], piiTa [taa_hei]

which yield the following sentences:

```
A.1  raama mohana  ko  piiTataa  hei.
     Ram   Mohan  -ko  beat      is
     (Ram beats Mohan.)

A.2  mohana  ko  raama  piiTataa  hei.
     Mohan -ko   Ram    beat      is
     (Ram beats Mohan.)
```

Note that no order is specified by the grammar among the nouns.

It is important to reemphasize that the transformation depends on TAM label which is purely syntactic, and not on tense, aspect and modality which are semantic. The TAM label can be determined for Hindi and other Indian languages by syntactic forms of the verb and its

[3]What karaka relations are permissible for a verb, obviously depends on the particular verb. Not all verbs will take all possible karaka relations.

auxiliaries without the need to refer to any semantic aspects. The specification for obtaining TAM labels is given by a finite state machine as described in Bharati et al. (1995; Chap. 4).

The Paninian theory outlined above (i.e., karaka charts and karaka chart transformations) can be used to generate (or analyze) the sentences given above as we have seen. However, there are additional constraints that would disallow the following sentences to be generated, for example[4]:

```
B.1   *ladake  ne  raama  ne laDakii  ko  kitaaba dii.
       boy    -ne  Ram   -ne girl    -ko  book    gave
       (*The Ram the boy gave a book to the girl.)

B.2   *laDake  ne  laDakii  ko kitaaba  phoola  dii.
       boy    -ne  girl    -ko book     flower gave
       (*The boy gave a book a flower to the girl.)
```

The constraints are given below:

1. Each mandatory karaka in the karaka chart for each verb group, is expressed *exactly once*. (In other words, a given mandatory karaka generates only one noun group with the specified vibhakti in its karaka chart unlike in B.1.)

2. Each optional karaka in the karaka chart for each verb group, is expressed *at most once*.

3. Each source word group satisfies some karaka relation with some verb (or some other relation). In other words, there should no unconnected source word group in a sentence, otherwise, the sentence becomes bad as in B.2.

Karaka charts are based on the idea of aakaankshaa and yogyataa. A karaka chart for a verb expresses its aakaankshaas or demands, and specifies the vibhaktis that must be used (i.e., yogyataa) with word groups that satisfy the demands. The same ideas can be used to handle noun-adjectives, noun-noun relations, verb-verb relations etc, each of which can be viewed as a demand-satsifaction pair.

1.2 Complex Sentences

Let us now consider the generation of sentences with more than one verb group. We begin with an example. Suppose the speaker (or writer) wants to express the fact that

```
raama ne   mohana ko  phala  khaakara      bulaayaa.
Ram  erg.  Mohan dat  fruit  having-eaten  called
(Ram called Mohan having eaten the fruit.)
```

This can be expressed by the structure at the karaka level shown in Figure 3.

One way of realizing it in a sentence is to choose *kara* TAM label for *khaa* (eat). This label specifies the temporal precedence relation with its parent node *bulaa* (call).[5]

The vibhaktis for the nouns can now be generated using the karaka chart (i.e., the default karaka chart together with karaka chart transformation rule for *kara* shown in Figure 4). In this example, the transformed karaka chart shows that karta of *khaa* is not expressed. The vibhaktis for the nouns in Figure 3 are as follows:

[4] A '*' before a sentence indicates that it is not a good sentence.

[5] However, it comes packaged, so to say, with the karaka sharing constraint that karta of *khaa* must be the same as that of its parent. Since this constraint is satisfied, the choice of TAM as *kara* is acceptable. More on this later.

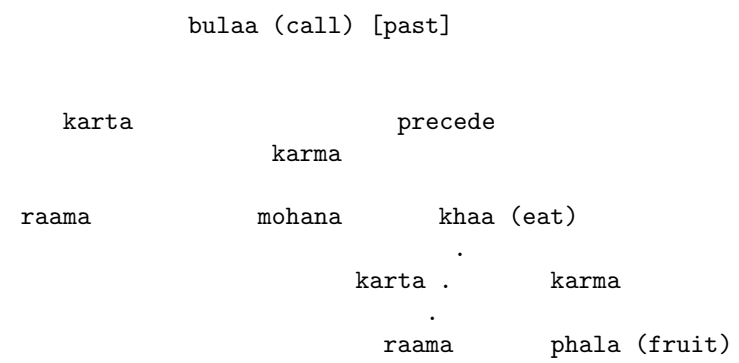


Figure 3: Modifier-modified relations for a complex sentence (shared karakas shown by dotted lines)

raama [ϕ], mohana [ko], phala [ϕ], khaa [kara], bulaa [taa_hei]

The generated sentence reads as:

raama mohana ko phala khaakara bulaataa hei.

The karaka chart transformation rules are given in Figure 4. The karaka sharing rule is given

TAM label	*Transformation*
kara	Karta must not be expressed.
	Karma is optional.
naa	Karta and karma are optional.
taa_huaa	Karta and karma are optional.

Figure 4: Transformation rules for complex sentences

in Rule S1.

Rule S1: Karta of intermediate verb with TAM label *kara* is the same as the karta of the verb modified by the intermediate verb.

1.3 Constraint Based Parsing

The Paninian theory outlined above can be used for building a parser. First stage of the parser takes care of morphology. For each word in the input sentence, a dictionary or a lexicon is looked up, and the associated grammatical information is retrieved. In the next stage, local word grouping takes place, in which based on local information certain words are grouped together yielding noun groups and verb groups. These are the word groups at the vibhakti level (i.e., typically each word group is a noun or verb with its vibhakti, TAM label, etc.). These involve grouping post-positional markers with nouns, auxiliary verbs with main verbs etc. Rules for local word grouping are given by finite state machines. Finally, the karaka relations among the elements are identified in the last stage called the *core parser* (Bharati et al., 1995; Chap. 6).

The task of the core parser to identify karaka relations. It requires karaka charts and transformation rules. For a given sentence after the word groups have been formed, each of the noun groups is tested against each row (called *karaka restriction*) in each karaka chart for each of the verb groups (provided the noun group is to the left of the verb group whose karaka chart is being tested). When testing a noun group against a karaka restriction of a verb group, vibhakti information is checked, and if found satisfactory, the noun group becomes a candidate for the karaka of the verb group.

The above can be shown in the form of a constraint graph. Nodes of the graph are the word groups and there is an arc labeled by a karaka from a verb group to a noun group, if the noun group satisfies the karaka restriction in the karaka chart of the verb group. (There is an arc from one verb group to another, if the karaka chart of the former shows that it takes a sentential or verbal karaka.) The verb groups are called demand groups as they make demands about their karakas, and the noun groups are called source groups because they satisfy demands.

As an example, consider a sentence containing the verb khaa (eat):

```
E.1  baccaa   haatha  se   kelaa    khaataa  hei.
     child    hand -se     banana   eats
     (The child eats the banana with his hand.)
```

Its word groups are marked and khaa (eat) has the same karaka chart as in Figure 1.1. Its constraint graph is shown in Figure 4.

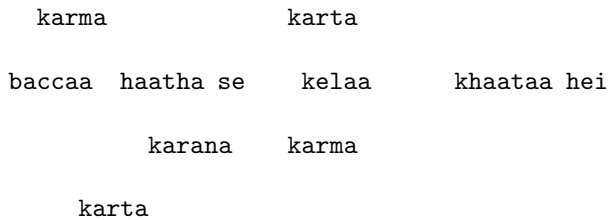


Figure 5: Constraint graph for sentence E.1

A parse is a sub-graph of the constraint graph containing all the nodes of the constraint graph and satisfying the following conditions:

C1. For each of the mandatory karakas in a karaka chart for each demand group, there should be *exactly one* outgoing edge labelled by the karaka from the demand group.

C2. For each of the desirable or optional karakas in a karaka chart for each demand group, there should be *at most one* outgoing edge labelled by the karaka from the demand group.

C3. There should be *exactly one* incoming arc into each source group.

Efficient methods based on bipartite graph matching are known for finding solution graphs.

If several sub-graphs of a constraint graph satisfy the above conditions, it means that there are multiple parses and the sentence is ambiguous. If no sub-graph satisfies the above constraints, the sentence does not have a parse, and is probably ill-formed.

With the constraints (C1, C2 and C3) specified above, the parsing problem reduces to bipartite graph matching[6] and assignment problems (Bharati et al. (1995; Chap. 6)). These have efficient solutions even in the worst case.

[6]We are indebted to Somnath Biswas for suggesting the reduction.

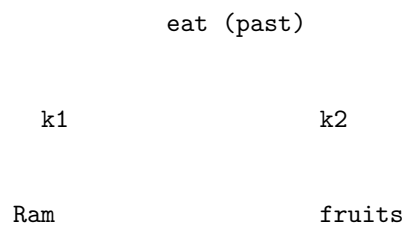


Figure 6: An example modifier-modified structure

2 Paninian Framework Applied to English

2.1 Basic Issues

We have already discussed in the last section that PG gives primacy to information and its coding. Thus, the karaka-vibhakti mapping for a language specifies how the karaka information can be coded in the vibhakti in that language. The vibhakti for a free word order language, say an Indian language, consists of case and post-positional markers. It is natural, therefore, to include position information as part of the vibhakti in a positional language such as English.

We introduce the notion of generalized vibhakti. The generalized vibhakti includes relative position of a constituent besides case endings, post-positions and prepositions. As was just mentioned, in a (relatively) fixed word order language like English or French, the relative position plays an important role, whereas in a (relatively) free word order language such as an Indian language or Japanese, it is the other markers that are more important. Thus, a language may choose to have any of these devices or a combination thereof for encoding semantic information.

There are two important vibhaktis for English based on word order:

Subject: This is the pre-verbal position, that is the position to the immediate left of a verb group.

Object: This is the post-verbal position, that is, the position to the immediate right of a verb group.

These do not have any other implications regarding tree structure. Thus, they should not be confused with subject and object terms in the generative enterprise or some other formalism in linguistics where these terms are defined with respect to the trees.

Take for example the modifier-modified structure in Figure 6. The karaka chart for eat is given in Figure 7 where 'acc' means that the word (which is karma) must be in accusative case. For karana no position is specified in the vibhakti, so that it can occur anywhere as long as vibhakti of other karakas are not affected. (If we had wanted to place a constraint that it must occur to the right of object, we could have written its vibhakti as ' ¿ obj'.) The following vibhakti level representation is produced, when we try to generate it for the modifier-modifier structure using the karaka chart for eat.

Ram [subj] fruits [obj] ate

No order is explicitly shown in the above. Finally, when we generate the sentence we get:

Ram ate fruits.

Karaka	*Vibhakti*	*Presence*
Karta (k1)	subj	mandatory
Karma (k2)	obj + acc(accusative)	mandatory
Karana (k3)	prep.: with	optional

Figure 7: Default karaka chart for English verb *eat*

where Ram and fruits have taken their (positional) vibhakti. (Determiners and adjectives are handled at the level of local word grouping. They are ignored here for simplicity, and it will be assumed that they can be produced when needed by local word grouping.)

Let us introduce a more pictorial notation for karaka charts containing generalized vibhakti. In the new notation, the karaka chart for *eat* given earlier would appear as:

```
                                    *
______  eat   ______[acc]   with_______
  k1            k2              Opt k3
```

An underline indicates that a constituent is to appear there. The placement of the line shows its relative position with respect to its neighbour unless a '*' is marked above it which indicates that there is no constraint on word order for that constituent. The prepositions and case are shown along with the underline. The default case is nominative, if no case is shown. The relation that the constituent will have with the verb *eat* is written below the line.

Optional karaka relations are marked by *opt* and this indicates that it is not mandatory that the constituent with the given karaka relation and vibhakti be present.

For the modifier-modified structure in Figure 8, the following sentences can get generated

```
                    eat [past]

karta               karma           karana

  Ram           fruits              his hands
```

Figure 8: Another example of modifier-modified structure

using the above karaka chart:

Ram ate the fruits with his hands.
With his hands, Ram ate the fruits.

Note that *with his hands* cannot come between *Ram* and *ate* or between *ate* and *fruit* because then the adjacency relation between karta and the verb and karma and the verb (specified by subj. and obj. vibhakti, respectively) would be violated.

2.2 Passives

For passives, we have the following karaka chart:

TAM label	*Transformation*
Passive	k2 subj.; k1 opt, prep: by; *

Figure 9: Karaka chart transformation for passive TAM

```
                                *                *
_______  eat[passive]    by_____[acc]    with_____
  k2                        opt k1           opt k3
```

The karaka chart transformation rule given in Figure 9 generates the above chart from the basic or default chart.

Pictorially the transformation rule can also be shown as:

```
                          passivization                          *
____  eat  ______[acc]    ===========>>   _____  eat[passive]  by_____[acc]
 k1          k2                            k2                   Opt k1
```

The passive voice sentence gets generated when the verb in the modifier-modified tree is marked as passive semantically (meaning thereby that the role of the karta is de-emphasized).

As an example, we have a modifier-modified structure and the generated sentence as shown in Figure 10.

```
                bring [past,+passive]

              k1            k2


             she               books

          The books were brought by her.
```

Figure 10: Modifier-modified structure for a passive

TAMs in English are the counterpart of those in Indian languages: They specify tense, aspect and modality (compositely) associated with the verb, and are given by the verbal ending and the auxiliary. Past-passive TAM, for example, means that the auxiliary verb *be* is with the appropriate tense (*were*) followed by the participle form of the main verb (*brought*). As defined earlier, a TAM label of a verb can also be called as the vibhakti of the verb.

2.3 Lexical Control

We will now look at some verbs that take another verb as their argument, and share some karakas with it. In the following sentence, for example:

Ram promised Mohan to go to the party.

go is the karma-vishaya (abbreviated as k2-v)[7] of *promise*, and *Ram* is the karta of *promise* as well as *go* in Figure 11.

[7]Karma-vishaya is a karma which gives the topic or subject related to the main verb.

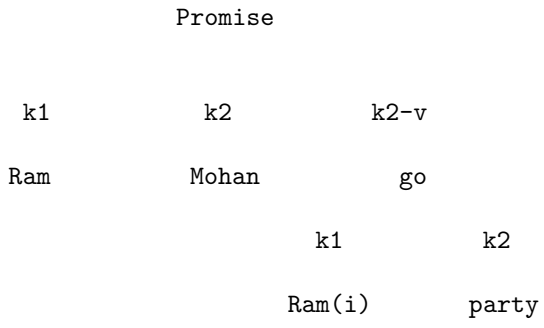


Figure 11: Modifier-modified structure with shared karta in lexical control

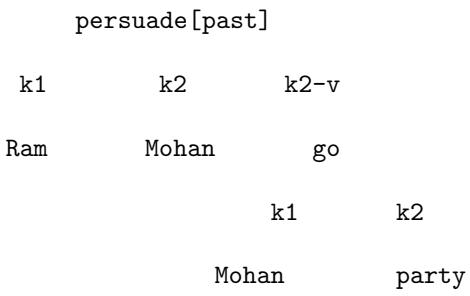


Figure 12: Modifier modified structure for *persuade*

Consider another sentence where the karta of the modifier verb is the same as the karma of the modified verb:

Ram persuaded Mohan to go to the party.

This is generated from the structure shown in Figure 12.

The karaka charts for *promise* and *persuade* are given in Figure 13. They are the similar to karaka charts shown for other verbs except that they differ in the karaka control or k-control. *Promise* is said to be a *karta-control* verb whereas *persuade* is a karma-control verb. The former has its k-control equal to karta, and the latter has k-control as karma.

The modified verb, shares its karta with a karaka of the modified verb depending on its k-control. It is important to note that the control is independent of the voice (or TAM) of the verb. For example, *Mohan* is still the karta of *go* in the following sentence:

Mohan was persuaded by Ram to go to the party.

_____	promise	_____[acc]	_____[inf]
k1		k2	k2-v (k-control sharing)
_____	persuade	_____[acc]	_____[inf]
k1		k2	k2-v (k-control sharing)

Figure 13: Karaka charts for *promise* and *persuade*

In the standard linguistics analysis, one talks of *persuade* as an object-control verb. But this is not quite right because when passivized, it acts as a subject-control verb. Alternatively, one will have to say that *persuade* is on object-control verb when active and a subject-control verb when passive. PG analysis is simpler and more elegant.

2.4 Infinitives

In this section, we look at complex sentences with infinitives indicating purpose relations. The modifier verb is an infinitive (TAM equal to infinitive) and specifies a purpose for the modified verb. For example, *cut* is the purpose for the main verb in the following sentence:

Mohan brought a knife to cut fruit.

There is a relation at the karaka level called the *taadarthya* or *purpose*. Using it the karaka charts can indicate a purpose relation between two verbs (usually the main verb and another verb):

```
                                    *            *            *
_____  bring  ______[acc]  with_____[acc]  in_____[acc]  to___[v,inf]
  k1            k2             opt k3       opt k7       opt purpose
```

The karaka chart specifies that the argument verb must take the infinitive TAM.

If we begin with the default karaka chart of *cut* verb with its TAM equal to simple present:

```
______  cut   ________[acc]
  k1            k2
```

then, for the infinitive form of *cut*, the karaka chart can be obtained by karaka chart transformation rule for the infinitive TAM shown in Figure 14. This yields the karaka chart:

TAM	Transformation
infinitive	karta must not be expressed (sharing rule T1), or karta has prep.: *for*, [acc]

Figure 14: Karaka chart transformation for infinitive TAM

```
For______[acc]  cut[inf]  _______
    opt k1                 opt k2
```

It also places a constraint that if karta is not expressed, it must be the same as the karta of the main verb.[8] More generally this can be stated as the sharing rule T1.

Sharing rule T1: If the karta of a modifier verb is not expressed, it must be the same as the karta of the modified verb.

With the above karaka charts we can generate a sentence for the modifier-modified structure in Figure 15. The generated vibhakti structure is:

[8] In the example sentence, rule T1 applies. With *purpose* relation, however, karta could be shared with other karakas also. For example in the following sentence:

Ram gave him a toy to play.

karta of *play* is possibly shared with sampradana of *give*, namely *him*. But not so for the modifier verb *win* in the following sentence:

Ram gave him a toy to win his trust.

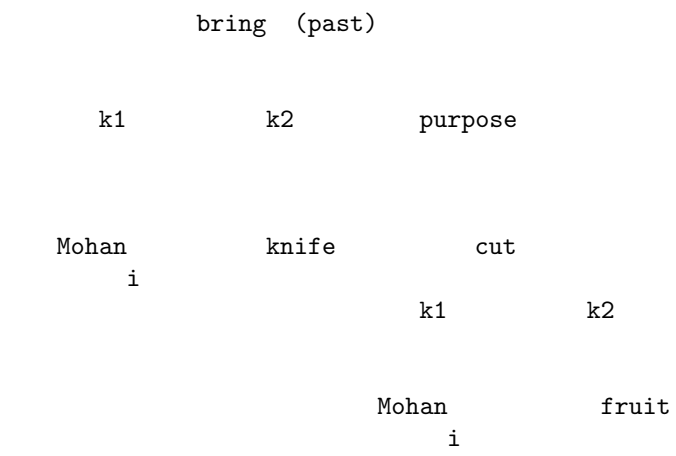


Figure 15: Modifier-modified tree with a shared karta

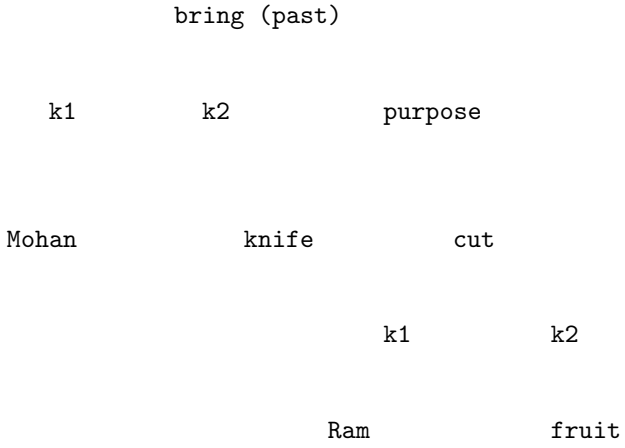


Figure 16: Modifier-modified structure without shared karta

Mohan [subj of bring] knife [obj (bring)] fruit [obj (cut)] brought cut[inf]

and finally the sentence:

Mohan brought a knife to cut fruit.

or

To cut fruit, Mohan brought a knife.

If the kartas for the two verbs were different, we could not drop the karta of the *cut* verb. For example, for the structure in Figure 16, the karaka charts would permit the generation of the following sentence:

Mohan brought a knife for Ram to cut fruit.

but not one in which karta of cut is not explicitly expressed:

** Mohan brought a knife to cut fruit.* (Bad sentence for the above tree.)

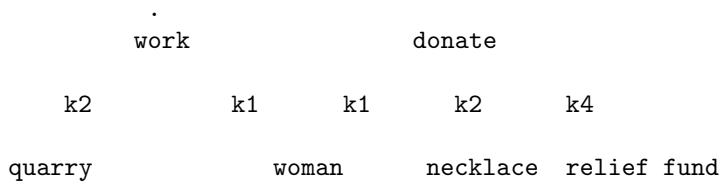


Figure 17: Modifier-modified tree for relative clause

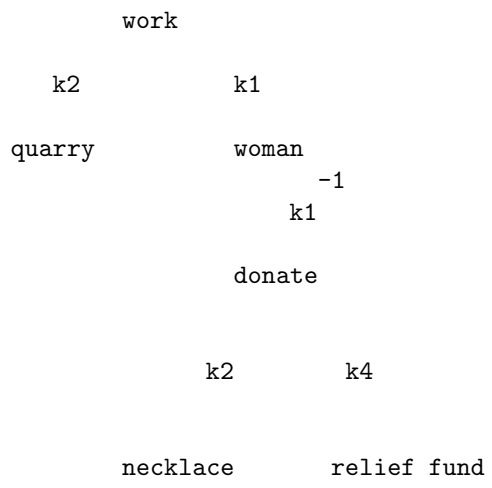


Figure 18: An alternate modifier-modified structure for relative clause

2.5 Relative clauses

Relative clauses allow us to refer to any participant in an action, and relate to another action (say, the main action) etc. In English, it requires putting a wh-word for the participant to be referred to and moving it to the beginning of the clause. For example, in the sentence below, *the woman* participates in two actions: *work* and *donate*. It occurs with the main verb *works* and is shown to be a participant in the action *donate* using a wh-word (*who*):

> *(S1) The woman [who donated the necklace to the relief fund] works in a quarry.*

Similarly, here is another example where the *necklace* is the entity referred to by the relative clause involving the action *donate*:

> *(S2) The necklace [which the woman donated to the relief fund] was bought by her husband.*

Note that *who* in the sentence S1 (or *which* in the sentence S2), is at the beginning of the clause, rather than, its normal position with respect to its verb *donate*.

The modifier-modified structure for the sentence S1 is given in Figure 17, where a dot marks the main action (*work*). Equivalently, the above structure can be drawn as a tree using $k1^{-1}$ relation as shown in Figure 18 (where a dot is not necessary as the tree has a single root node denoting the main action). When a verb takes a sentence as its argument in a relative clause,

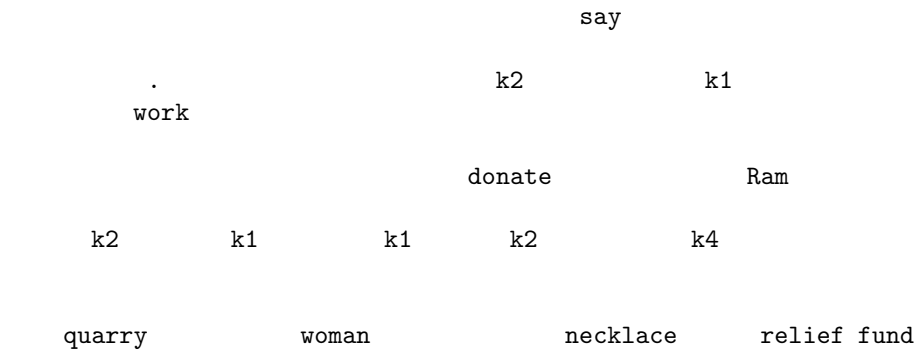


Figure 19: Modifier-modified tree M3 for an embedded sentence in relative clause

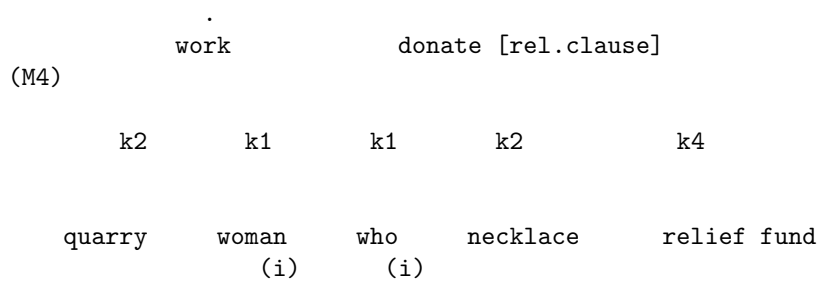


Figure 20: Breaking into two modifier-modified trees for a relative clause

the wh-element still moves to the front, but the rest of the order is not disturbed in the clause. For example, when *say* is introduced, we have:

> *(S3) The woman [who Ram said donated the necklace to the relief fund] works in a quarry.*

The modifier-modified structure (M3) in Figure 19 corresponds to the sentence S3. Note that *the woman* is still a participant in the two actions: *work* and *donate*; except now, *donate* is itself an argument of another action *say*. The positions of the arguments of *say* remain undisturbed, in the sentence S3, only the positions of arguments of donate are altered. In particular, *who*, the referred argument of *donate*, occurs in the first place in the clause.[9]

We now present a theory in the Paninian spirit for generating the above kind of sentences with relative clauses. We look at the theory in two steps: first for clauses with only one verb (simple clauses) and then for non-simple clauses.

Whenever we have a modifier-modified structure which has more than one root (as in Figures 17 and 19), we break the structure up into trees as shown in Figure 20. For a shared node (i.e., a node with two parents), its link to a parent which is not under a root node marked by a dot, is broken and instead a new node consisting of wh-element is created and attached to the broken link. The parent (*donate*) is marked to be of relative clause type.

[9]If there are several ancestor nodes of *donate*, they would all maintain normal positions of their arguments, and the referred argument of *donate* would be further away from its normal position. Therefore, this is called long distance movement or long distance dependency.

The karaka chart for a verb marked to be a relative clause type has a vibhakti called TOPIC for one of its karaka. This vibhakti has the constraint that it occurs in the first position of the clause obtained after substitution operation. Thus, the following is the karaka chart for *donate* [rel. clause]:

```
   [TOPIC]                                          *
[+wh]____   donate[rel.clause]   _____[acc]   to________
     k1                            k2              k4
```

M4 trees generate the following representations at vibhakti level:

(a) woman(i)[subj(works)] works quarry[prep:in]
(b) who(i)[TOPIC] donate[past] the necklace[obj (donate)] relief fund[prep:to]

Strings with ordered elements can be produced out of the above representations. Finally, the two strings are merged together by looking at coindexing across them and performing the join operation. In the join operation, the coindexed item in the dependent string immediately follows the other coindexed item and the dependent string is inserted there. In the past example, we first get the following strings:

(a) The woman(i) works in the quarry.
(b) who(i) donated the necklace to the relief fund

After the join operation we have:

The woman who donated the necklace to the relief fund works in a quarry.

In a similar way if we generate a sentence for the structure M3, then the representations at tree level, vibhakti level and strings are as given in Figure 21.

Note that the TOPIC vibhakti forces *who* to appear as the first element in the string. Joining operation produces the correct sentence.

2.6 Wh-questions

Wh-questions are handled by a similar mechanism. An example Karaka chart for a wh-question regarding the karta is given as follows:

```
   [TOPIC]                                    *
[+wh]_____  donate[wh-Q]  ________[acc]   to______
     k1                      k2              k4
```

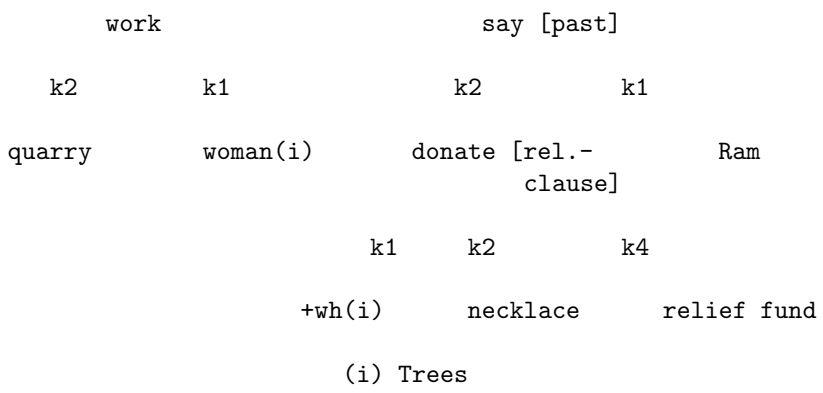
For the modifier-modified structure in Figure 19, this generates (ignoring the rules for placement of *did*):

Who did Ram say donated the necklace to the relief fund?

2.7 Sentential Subjects

Many verbs take a sentence as an argument. For example, *tell, say, think, expect* take a sentence as their karma. In active voice, the vibhakti assigned to karma is object position. For example:

Ram said/expected that the boy will come to the party.



(i) Trees

(a) woman(i)[subj] works quarry [obj+prep:in]
(b) Ram[subj(say)] say[past] who(i)[TOPIC] donate[past] necklace[obj(donate)] relief fund[prep:to]

(ii) Vibhakti representation

(a) the woman(i) works in the quarry
(b) who(i) Ram said donated the necklace to the relief fund

(iii) Strings

Figure 21: Three level representation for M4

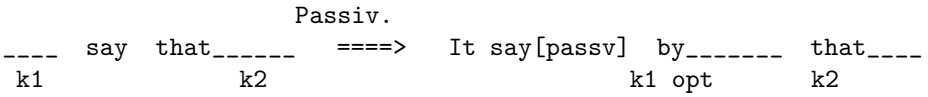


Figure 22: Karaka chart transformation for passives of *say* verbs

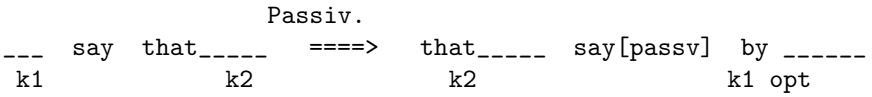


Figure 23: Karaka chart transformation same as for usual passives

When such verbs are passivized, there is a tendency in English not to make these sentential arguments as subjects.[10]

Passivization for such verbs produces a karaka chart in which the subject position is filled by a dummy or pleonastic element (*it*):

It is said that the boy will come to the party.
It is expected by Ram that the boy will come to the party.

The karaka chart transformation rule for passivization of such verbs is shown in Figure 22.

The usual passivization rule shown in Figure 23 is less preferred (or not acceptable). and would produce a bed sentence such as:

**That the boy will come to the party is said.*
?That the boy will come to the party is expected.

2.8 Raising

Raising is a peculiar phenomenon in English which occurs with a small number of verbs that take sentential arguments. (Such verbs are called raising verbs or exceptional case-marking verbs.) We illustrate it by means of an example sentence involving *expect*, and the modifier-modified structure given in Figure 24.

Note that an argument of the embedded verb *come* has moved to the subject position of *expect*. In Paninian terms, a karaka of an embedded verb is given a vibhakti (position) that seems to be defined with respect to the main verb.

The karaka charts for *expect* and *come* that handle this phenomenon are shown in Figure 25. Note that for raising to take place, the embedded verb must be in the infinitive form.

The infinitive form can also occur without raising as shown in Figure 26 to generate the marginally acceptable sentence:

It is expected for the boy to come to the party.

The TOPIC constraints block the generation of the following bad sentences:

**The boy it is expected for to come to the party.*
Is expected the boy to come to the party.

[10]This might flow from basic principles relating to sentence processing by human mind which prefer that the head of the main sentence (i.e., the verb) not be postponed indefinitely.

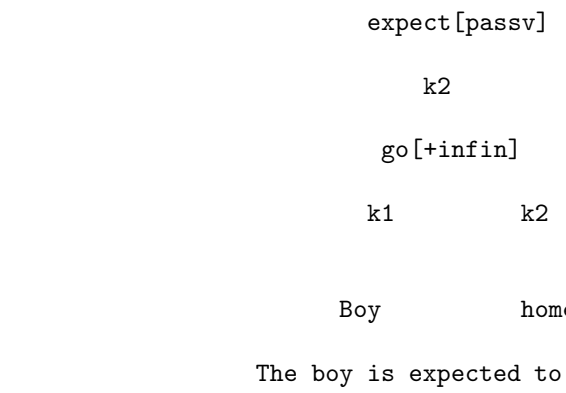


Figure 24: Modifier-modified structure involving *expect*

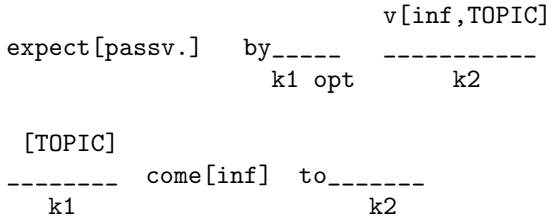


Figure 25: Karaka charts for *expect*, a raising verb

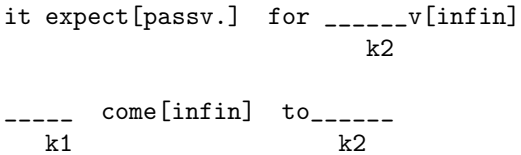


Figure 26: Karaka charts for *expect* without raising

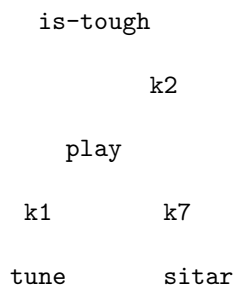


Figure 27: Modifier-modified structure involving *tough*

```
                        [TOPIC]
for______  play[inf]  ________  on________
   k1 opt                k2          k7

                                   [TOPIC]
for______  play[inf]  ________  on________
   k1 opt                k2          k7

          v[+TOPIC]
be tough  _______
             k2
```

Figure 28: Karaka chart for *tough*

2.9 Tough-movement

There are certain adjectives which display a phenomenon similar to raising. For example, the following sentences:

It is tough to play this tune on the sitar.
This tune is tough to play on the sitar.

are generated for the modifier-modified structure in Figure 27. The karaka charts shown in Figure 28 are similar to those used for handling raising. By simply marking certain karakas as having the vibhakti TOPIC in the karaka chart for the embedded verb *play*, the tough-movement is handled.

If the tough-adjective is viewed as a passive form of *find tough* as in:

Ram finds it tough to play this tune on the sitar.
It is tough for Ram to play this tune on the sitar.

then in a possible analysis the usual rules for sharing of karta between the main verb (*find tough* or *is tough*) and the infinitive verb (*to play*) apply as discussed in an earlier section.

The tough movement can occur over sentences embedded several levels deep:

It is tough to believe Mohan can play this tune on the sitar.
This tune is tough to believe Mohan can play on the sitar.

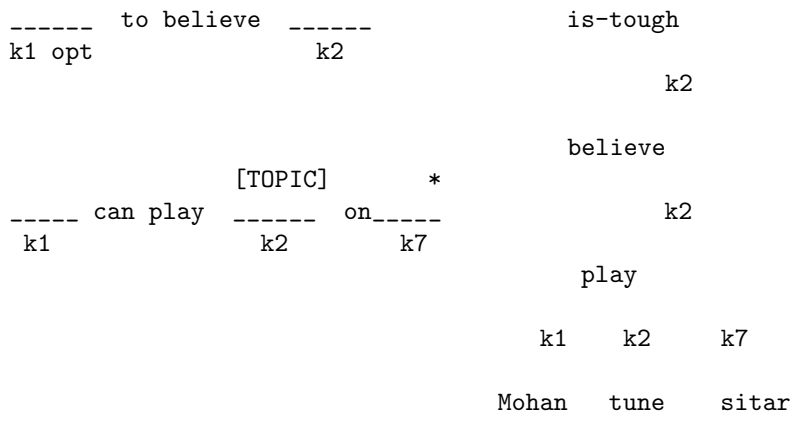


Figure 29: Karaka charts for *tough* with embedded sentences

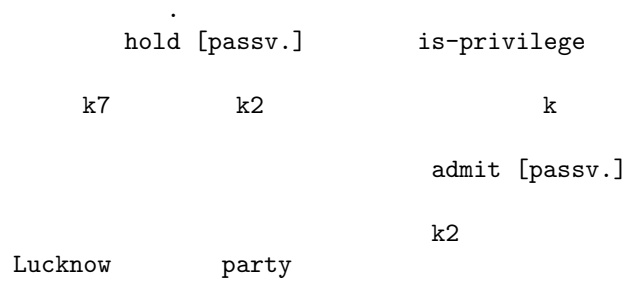


Figure 30: Modifier modified structure to illustrate pied-piping

However, exactly the same mechanism as outlined earlier handles it. Figure 29 shows the karaka charts used as before, except one of the vibhaktis is marked as TOPIC. Note that by simply making the vibhakti of k2 as TOPIC in an embedded sentence, everything else falls into place.

The following sentence which has a combination of shared karta between the main verb and the infinitive and tough movement is handled correctly as the reader can verify:

This tune is tough for Ram to believe Mohan can play on the sitar.

2.10 Pied Piping

In a normal relative clause, only the wh-element "moves" from its position to the TOPIC position. For example,

The party to which it was a privilege to be admitted was held in Lucknow.

In pied piping, a number of items move with the wh-element (or follow the pied piper) as shown below:

The party to be admitted to which was a privilege was held in Lucknow.

Modifier-modified structure in Figure 30 corresponds to the above sentences.

It is interesting to bring a contrast with Hindi. For the above modifier-modified structure, the following sentences could be generated in Hindi which respectively correspond to the two English sentences just given:

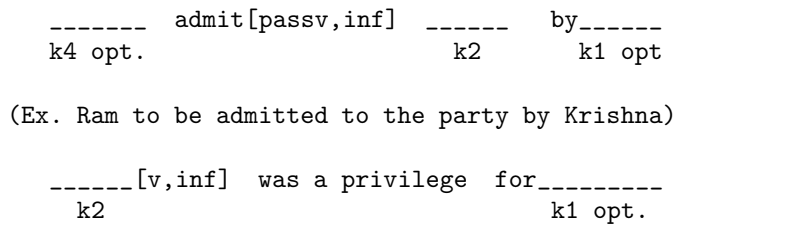


Figure 31: Karaka charts for pied piping

```
? bhavya paartii jisameM  aadara kii baata  thii pravesha diyaa jaanaa
  grand  party   in-which respect's  thing  was  to-be-admitted
                                  lakhanauu meM hotii thii.
                                  Lucknow   -in held  was
  (The grand party to which it was an honour to be admitted was held in Lucknow.)

  bhavya paartii jisameM pravesha diyaa jaanaa aadara kii baata thii
                                  lakhanauu meM hotii thii.
  (The grand party to be admitted to which was an honour was held in Lucknow.)
```

Note that what is called pied piping is the normal preferable sentence in Hindi[11].

The English sentence with pied-piping can be derived by relaxing the wh-movement (karaka chart wh-transformation) out of *admit* and performing normal passive karaka chart transformation for *is-privilege*. Again the normal rules for karaka sharing apply. Figure 31 has the relevant transformed karaka charts with some example strings that can be generated by them. Note that normal karaka sharing rules between main verb and infinitives apply if the optional k1 (karta) does not occur with the infinitive verb.

The above would not only correctly generate the sentence with pied piping given above, but would also generate correctly the following sentences from their respective modifier-modified structures:

For Ram to be admitted to the party was a privilege for Dasarath.
The party for Ram to be admitted to which was a privilege was held in Delhi.
The party to be admitted to which was a privilege for Ram was held in Delhi.
?For Krishna to admit Ram to the party was a privilege for Vasudeva.
The party to admit Ram to which was a privilege for Krishna was held in Vrindavan.

3 Conclusions

In this paper, we have presented how the computational Paninian Grammar formalism can account for English. The approach is information theoretic and uses the notion of vibhakti and karaka. Since vibhakti is a marker at the surface level, and English is a positional language, therefore, the position of a word in a sentence is also a part of its generalized vibhakti. To account for the long distance dependency, a special vibhakti called TOPIC is introduced which

[11] This possibly has to do with the free word order and verb final nature of Hindi together with ease of information extraction.

stands for the first position in a clause even though the sentence might be embedded several levels deep in the clause.

With the above generalization of vibhakti, the karaka charts and karaka chart transformations work same as before (but naturally, a transformation of vibhakti appears as movement). A new operation called join is introduced, besides the normal substitute operation, for connecting the relative clause modifier with a noun. The whole framework is based on information theoretic ideas and the extensions mentioned above are in the Paninian spirit.

Interestingly, the join operation is the same as the adjoin operation in Tree Adjoining Grammar (Joshi, 1985) provided the auxiliary trees have their foot node at either the extreme left or the extreme right leaf node.

There are a number of issues which are being looked at as part of the ongoing research. One of the issues for example, is how to handle double movement, e.g., tough and wh-movement in the same clause:

The instrument which this raaga is tough to play on was designed in Banaras.

This requires two TOPIC positions, which seem to be required in a definite order. Similarly, there are issues in why 'that' blocks movement. It seems to be related to the nature of modifier-modified structure and the vibhakti. In the paper, we have also not discussed subject-verb agreement. While no difficulty is anticipated, it needs to be spelt out.

At a more general level, we would also like to compare the adjoining operation in TAG with no restrictions on auxiliary trees with a suitable operation in PG.

Finally, we have not discussed parsing issues in this paper. Paninian parser for Indian languages has already been described earlier in Bharati et al. (1995). Parser for the new generalized vibhakti and the join operation needs to be worked out in detail. It is expected to be similar in spirit to before, as only the constraints have changed. However, a more efficient parsing algorithm is expected to be designed after the study of the particular nature of the constraints.

4 Acknowledgements

We would like to acknowledge the earlier work done by Rajesh Bhatt on the same problem of applying PG to English (Bhatt, 1993) where pre-cursors to some of the ideas given here were explored. The solution using TOPIC position for wh-movement and other phenomena related to long-distance dependency outlined in this paper, differs from the treatment outlined in Bhatia (1995), and was worked out after Medhavi Bhatia moved out of Kanpur. He should not be held responsible for errors, if any, in this part of the theory.

References

[1] Bharati, Akshar, Vineet Chaitanya, and Rajeev Sangal, Anusaraka or Language Accessor: A Short Introduction, In *Automatic Translation*, Thiruvananthpuram, Int. school of Dravidian Linguistics, 1994.

[2] Bharati, Akshar, Vineet Chaitanya, and Rajeev Sangal, *Natural Language Processing: A Paninian Perspective*, Prentice-Hall, New Delhi, 1995.

[3] Bhatia, Medhavi. *Paninian Theory Applied to English.* Dept. of CSE, IIT Kanpur, 1995. Bachelor's Thesis.

[4] Bhatt, Rajesh. *Paninian Theory for English.* Dept. of CSE, IIT Kanpur, 1993. Bachelor's Thesis.

[5] Joshi, Aravind K., Tree Adjoining Grammar: How much Context-Sensitivity is Required to provide reasonable Structural Description. In D. Dowty, L. Karttunen, and A. Zwicky (eds.), *Natural Language Parsing*, Cambridge University Press, Cambridge, UK, 1985.

[6] Narayana, V.N., *Anusarak: A Device to Overcome the Language Barrier*, Ph.D. thesis, Dept. of CSE, IIT Kanpur, January 1994.