Adapting Link Grammar Parser (LGP) to Paninian Framework
Mapping of Parser Relations for Indian Languages

by

Akshar Bharati, Dipti Misra Sharma, Sukhada

in

National Seminar on Computer Science and its Applications in Traditional Shastras (CSATS'09)
(*CSATS2009*)

Report No: IIIT/TR/2009/218

Centre for Language Technologies Research Centre
International Institute of Information Technology
Hyderabad - 500 032, INDIA
February 2009

Adapting Link Grammar Parser (LGP) to Paninian Framework & Mapping of Parser Relations for Indian Languages

Akshar Bharati, Dipti Misra Sharma
IIIT-Hyderabad, India
dipti@iiit.ac.in

Sukhada
IIIT-Hyderabad, India
sukhada@research.iiit.ac.in

Abstract

Machine translation system development activities in India are being carried out at many institutions. One of them is 'Anusaaraka project'[1] being developed at IIIT-H. Anusaaraka systems have been developed from Marathi, Telugu, Bengali, Kannada and Punjabi into Hindi. Currently, the work is going on English to Hindi Anusaaraka system. To develop such a system a large number of people are required to get involved in improvement of linguistic/grammatical knowledge. This is a big task and can be easily achieved if a large number of people participate in the activity. Our aim is to make the task of contributing linguistic resources for the system easy and simple so that those who want to participate in the task of developing resources to improve the output quality of machine translation systems can do it easily and also feel empowered at the end.

1 Introduction

1.1 Requisite Factors (anubandha chatushataya)

- The 'subject matter' (vishaya) of this work is to establish correspondence between Link Grammar Parser (LGP) relation and Paninian relations and to develop a tool which maps the LGP relations into adapted Paninian relations so that Sanskrit scholars can easily participate in Anusaaraka project[1] / work on other NLP applications.
- Indian linguists, Sanskrit scholars (who want to participate in machine translation (MT), common Indians and others who want to develop English to Indian language translation system are the 'eligible persons' (adhikaarii) to pursue the 'subject matter'.
- The 'affiliation' (sambandha) is between upakaarya (vishaya - which is to establish correspondence between LGP relations and Paninian relations) and upkaaraka (adhikaarii – the eligible persons).
- The 'purpose' (prayojana) is to build a system that helps Sanskrit scholars (vaiyaakarana) and common Indians to participate in the field of English to Indian language MT systems.

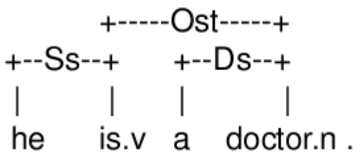
2 The Link Grammar Parser (LGP), An Introduction

The Link Grammar Parser is an English syntactic parser based on link grammar theory(The Parser is modeled on link grammar theory that is described briefly below. For more details one can refer to http://www.link.cs.cmu.edu/link/dict/index.html.) The entire LGP system is available for download on the web under BSD (Berkeley Software Distribution) license.

This (LGP) is similar to a dependency grammar as it is not taking abstract nodes like NP, VP etc. for establishing relations. However its not merely dependency as it lacks the head-child (visheshya-visheshana) relation. The structure assigned to a sentence by LGP is through links, between pair of words. Rather than thinking in terms of constituents (such as verb phrase, noun phrase), one must think in terms of relationships between pairs

[1] Anusaaraka is an English to Hindi language accessor and machine translation system. Anusaaraka has been developed as a collaborative project of CIF, IIIT-H, HCU and TCS.

of words. For example, look at LGP's output for the sentence *He is a doctor.*:



```
          +-----Ost-----+
+--Ss--+      +--Ds--+
|           |      |            |
he       is.v   a    doctor.n .
```

Figure – 1

In the above example (figure-1), pairs of words are linked with certain relations, e.g. "he" and "is" are linked with the relation "Ss" (subject); "is" and "doctor" are linked with "Ost" (object); and "a" and "doctor" are linked with "Ds".

3 Links:

As presented in the Link Grammar Parser documents, the total number of main link-types is 107. All these links are named in one or more superscripts (upper-case letters), such as A, B, BT, C, CO, SFI etc.. Apart from these main link types there are many other links with subscripts (lowercase letter) like Bs, Bp. Link types may have multiple subscript characters, such as Bsw, Bsm, Bsmt, Bpj etc.. A close count of all the links, including the subscripted ones gives a total of 410 links. Though most of the link names have mnemonic characters (like in 'SFI' link-name, 'S' is for subject, 'F' for filler and 'I' for inverse) but there are some links which do not have such mnemonic character. For example, 'K' link is used to connect verbs with particles but it does not have any character which says that this link joins verb with a particle.

4 What does LGP lack?

As mentioned in 1.1, our objective is to map LGP relations into Paninian relations. In Panini's grammar a sentence is treated as a series of modified – modifier (visheshya-visheshana vhaava) relations. The information regarding *visheshya-visheshaNa* (modified and modifier) is not available in the LGP links. i.e. by looking at the links, one can not understand whether the left end of the link is modified/modifier or the right end. In other words, although it represents the relation between two words, it does not show which of them is the head/parent and which of them is the modifier/child.

5 How do we fulfill this?

To show *visheshya-visheshaNa* (modifier and modified) relation among the parser links as well as the mapped relations, we have adopted a convention that we will write the *visheshya* (modified) on the left side of the *visheshaNa* (modifier), so that one can easily understand that the left side of the relation is modified and the right side of the relation is modifier. For example: The mapped relations are divided into two parts using "-" hyphen. The left hand side of the hyphen is always *visheshya* (modified) and the right hand side is *visheshaNa* (modifier). The convention implicitly interprets the modified-modifier relation from the link labels For example: in the relation "kriyA-upasarga" (verb-particle), *kriyA* (verb) is the head and the *upasarga* (particle) is the child. Which maps to the LGP's link "K" .

6 Using LGP in Anusaaraka system

As mentioned above LGP is available under BSD license for free download, this facilitates the following:

- Anusaaraka is also a system which is going to be released under GPL (BSD license is compatible with GPL) and availability of an English parser under BSD matches well with Anusaaraka requirements.

- Since LGP is available under BSD, it is possible to modify it for our needs such as modifying the dictionaries.

7 Why do we need to map LGP relations into Paninian relations?

It is mentioned above that our aim is to involve large number of people in developing/improving the Anusaaraka system. The system involves improvement of linguistic/grammatical knowledge. This is a large task and can be easily achieved if a large number of people participate in the activity of building required linguistic resources. For the people to be able to participate in this activity, it is important to do the following:

7.1 Simplify the procedure for incorporating linguistic information

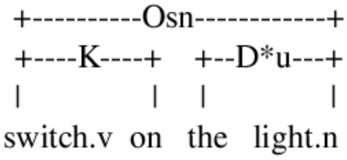
In the example “He is a doctor.” Link Parser’s output shows that ’doctor’ is the object of ’is’ but for us instead of ’object’ of ’is’; ’doctor’ has *samaanaadhikaraNa* relation with ’He’ i.e. ’He’ and ’doctor’ are referring to the same entity. Such cases are simplified and given the labels such as *subject-subject_samAnAXikaraNa*.

7.2 Use Familiar relation names

Because almost every Indian, learns some grammar for his/her language at school, which is almost similar to Paninian grammar, so the Paninian relation terms are familiar to most Indian people and thus are adaptable in less time and effort by common men, modern Indian linguists and Sanskrit grammarians.

The major problem while developing a machine translation system is WSD (word-sense-disambiguation). It requires a large set of rules. The task can be achieved if a large number of people get involved in this task. So, by adopting relation labels which are familiar to the contributors we can make the task easy for every Indian (interested in MT). The rules may often require the relation information between various words. For example, the disambiguation of the “switch” word in the following examples is a must to arrive at the correct Hindi translation.

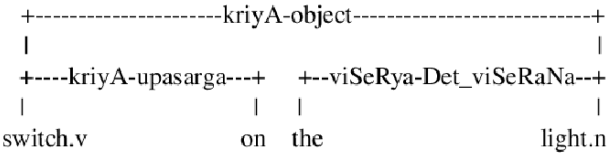
LGP’s output for the sentence ”Switch on the light.”:



```
+----------Osn------------+
+----K----+   +--D*u---+
|         |   |        |
switch.v  on  the  light.n
```

Figure – 2

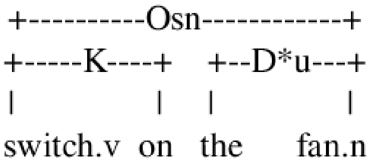
The above is the output that LGP provides us. Here, after mapping we get a ”kriyA-object” relation between ”switch” and ”light” and ”kriyA-upasarga” relation between ”switch” and ”on”. The sentence after mapping is given in Figure 3.



```
+---------------------kriyA-object---------------------------+
|                                                             |
+----kriyA-upasarga---+   +--viSeRya-Det_viSeRaNa--+
|                     |   |                        |
switch.v              on  the                 light.n
```

Figure –3

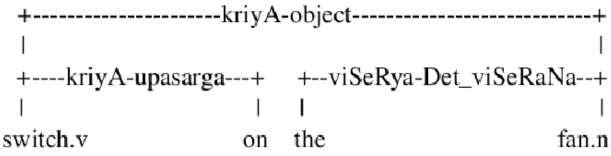
Now look at LGP’s output for the sentence ”Switch on the fan.”:



```
+----------Osn------------+
+-----K----+   +--D*u---+
|          |   |        |
switch.v  on   the    fan.n
```

Figure –4

Mapped output:



```
+---------------------kriyA-object---------------------------+
|                                                             |
+----kriyA-upasarga---+   +--viSeRya-Det_viSeRaNa--+
|                     |   |                        |
switch.v              on  the                  fan.n
```

Figure –5

After mapping we get a “kriyA-object” relation between ”switch” and “light” and “switch” and “fan”. Since we are getting the translation based on the label (here it is kriyA-object) which we have provided to the pair of words, which is now similar in both cases. Then the system obviously tries to translate the word ”switch” as either *jalaana* or *chalaana* which is not correct either. Then how does a machine know that both have different interpretations? To disambiguate the words we write rules which make use of the mapped relations and objects of the verbs as well. We say that if ”kriyA-object” relation is between ”switch” and ”light” then the Hindi meaning of ”switch” would be “jalaanaa” whereas if “kriyA-object” relation is between ”switch” and “fan” then the Hindi meaning of ”switch” would be “chalaanaa” . For such reasons we have mapped the LGP relations into Paninian relations.

8 Relation labels

Here we are trying to discuss some of the issues such as 'subject' and 'object' of English through Paninian view point. The major issues for our discussion are:

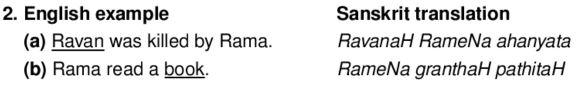
- Subject vs kartaa
- Object vs karma
- Problems in mapping determiners
- Miscellaneous

8.1 Subject vs kartaa

1. English example	Sanskrit translation
Rama reads.	*RaamaH pathati*
Rama will read.	*RaamaH pathishyati*

The subject 'Rama' in English examples (1) is translated as 'RaamaH' in Sanskrit which is *kartaa,* ending in prathamaa vibhakti . The original vibhakti for *kartaa* kaaraka is defined as tritiiyaa (ashtaa 2.3.18). In our examples *kartri-kaarakatvam* is abhihita (said) by the tip and that is the subject of our example. However, if we look at the passive construction, Ravana was killed by Rama (RavanaH RameNa ahanyata). Ravana in this sentence is still the subject whereas within the Paninian analysis it is not kartaa. Thus, the notion of *kartaa* does not always correspond to the notion of Subject. Also, English is a positional language, i.e. English has relatively fixed word order with word positions containing some grammatical information. Since the notion of 'subject' also contains 'positional' information, it is useful to retain this notion as such. Therefore, in our scheme we have decided to keep 'subject' as a relation label too.

8.2 Object vs karma



2. English example	Sanskrit translation
(a) Ravan was killed by Rama.	*RavanaH RameNa ahanyata*
(b) Rama read a book.	*RameNa granthaH pathitaH*

The subject *Ravana* in English example 2(a) is translated as *RavanaH* in Sanskrit which is *karma,* ending in *prathamaa vibhakti*. The original vibhakti for *karma* kaaraka is defined as *dvitiiyaa* (ashtaa. 2.3.2). In our examples 2(a-b) *karma-kaarakatvam"*is abhihita (said) by ta and kta respectivaly and that is the subject.

Thus, the subject and the object in English are based on position and can not be mapped into *kartaa* and *karma*.This is the reason we are not mapping 'subject' into *kartaa* or object into *karma*. It appears that 'subject' maps more closely to *abhihita*. However, as for now we are not making any such claims.

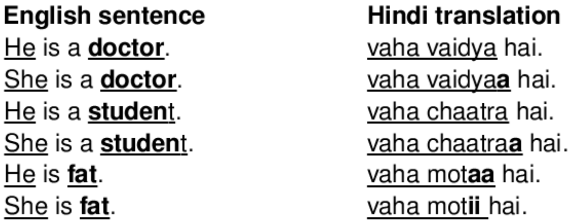
8.3 Problems in mapping determiners

English determiners also pose a problem for us. Determiners are noun modifiers. Noun modifiers can be more general including adjectives. Link parser, apart from providing link labels between words, also provides category labels of some word types.$SeeFig1.$. However it doesn't provide the category labels for determiners and other function words. if we decide not to map the determiner label to any Panian label or to map it to a more general label of viSesRaNa, some infomation will be lost. Since Anusaaraka believes in information preservation, it has been decided to keep the label as
viSeRya-det_viSeRaNa.
For example, the relation between 'the' and 'fan' in the sentence "Switch on the fan". $SeeFig-5.$

8.4 Miscellaneous

There are also many other links that can not be mapped directly into Paninian labels. For example, though "Ost" link's left end is a verb and the right end is an object. But instead of mapping the "Ost" link into "kriyA-object", we decided to map the "Ss" link's left end and the right end of "Ost" link, into
subject-subject_samAnAXikaraNa relation.

Because while looking from Hindi point of view, "Ost" link's left end goes with the subject and not with object. Such relations are very useful for Hindi generation. All the underlined words in the sentences below will have *subject-subject_samAnAXikaraNa* relation.

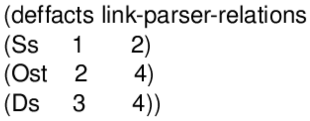


English sentence	Hindi translation
He is a **doctor**.	vaha vaidya hai.
She is a **doctor**.	vaha vaidya**a** hai.
He is a **studen**t.	vaha chaatra hai.
She is a **studen**t.	vaha chaatra**a** hai.
He is **fat**.	vaha mot**aa** hai.
She is **fat**.	vaha mot**ii** hai.

The words *doctorstudent* and *fat* are neutral in the English examples above. However, the Hindi counterpart changes its gender according to the gender of the subject. So, while generating Hindi we look for the relation *subject-subject_samAnAXikaraNa* and assign the similar gender of the subject to its samAnAXikaraNa. There are many other such cases like *visheshya-visheshana* etc. where gender, number, person and other problems are solved using these mapped labels.

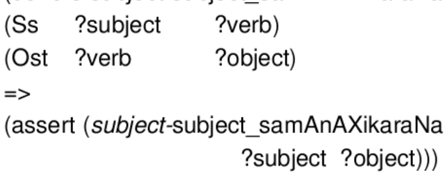
9 Implementation

We are using CLIPS (C Language Integrated Production System) an expert system shell, to convert Parser-relations into Paninian relations. Because of the ease of maintenance and modification of linguistics' knowledge base. For the sentence "He is a doctor." Link Parser's output in CLIPS' format is:



```
(deffacts link-parser-relations
(Ss     1       2)
(Ost    2       4)
(Ds     3       4))
```

To get the *subject-subject_samAnAXikaraNa* relation between "he" and "doctor", we have made a rule in CLIPS such as below:



```
(defrule subject-subject_samAnAXikaraNa
(Ss     ?subject        ?verb)
(Ost    ?verb           ?object)
=>
(assert (subject-subject_samAnAXikaraNa
                                 ?subject  ?object)))
```

The output of this rule is: (subject-subject_samAnAXikaraNa 1 4)

Which means that there exists a *subject-subject_samAnAXikaraNa* relation between the 1st word (He) and the 4th word (doctor) or we can say:

(subject-subject_samAnAXikaraNa He doctor)

10 Future Work

10.1 Classification of Links

So far we have developed a detailed mapping scheme between Link Grammar Parser 'links' and Panini grammar relation labels. However, to achieve a higher degree of generalization, it is important to group the link types. Therefore, we plan to clasify the links in future. The different types of classifications are as follows:

10.1.1 pratyaya sambandha

- **avyavahita pratyaya sambandha**
  This class will contain the links which form a direct LWG (local word grouping) group without any interruption with other words. For example in the figure below, the links 'PPf' and 'Pg*b' give a TAM (Tense, aspect and modality) group among the words *has*, *been* and *sleeping*, which occur directly one after the other in the sentence.

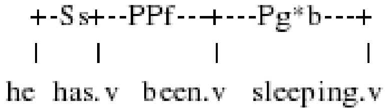
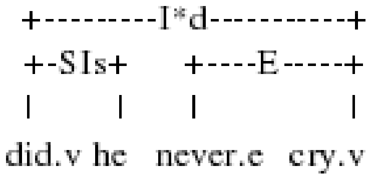


Figure-6

- **vyavahita pratyaya sambandha**
  In this class we will put those links which join the relative words together, though the words are interrupted with some other unrelated word in the sentence. In the example below "I*d" link's left node (did) and right node (cry) together will form a group, though they are interrupted by 'he'.



```
+---------I*d----------+
+-SIs+    +----E-----+
|     |    |         |
did.v he  never.e  cry.v
```

Figure-7

10.1.2 samaasa sambandha/ Compounds:

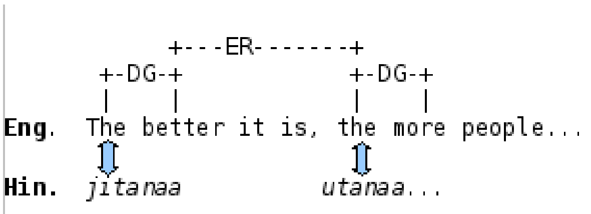
- **samaasa (compound)**
  Examples:
  The *income tax proposal* was rejected.

- **ardha-samaasa/Quasi compound**

  The name *ardha-samaasa* or quasi-compound is given for the words, which form roughly a phrase. For example in the sentence ”He was sitting *on a big black horse*.” the words *on a big black horse* have a single prepositoin *on* which is related to each words in the group.

10.1.3 arthanirdhaaraNaartha links

Some of the links can be used for word sense disambiguation. For example the left node of the ”DG” link will be translated as *jitanaa* and the right node as *utanaa* if ”ER” link is found with the left and the right node of the ”DG” link. See figure-8 below.



```
          +---ER-------+
     +-DG-+          +-DG-+
     |    |          |    |
Eng.  The better it is, the more people...
Hin.  jitanaa         utanaa...
```

Figure-8

10.1.4 adhyaahaarya shabda suuchaka links

English allows ellipsis of certain word in its sentences. e.g. there is an ellipsis of ”that” in the sentence ”It is likely he’ll do it.”(This sentence can be rewritten as ”It is likely *that* he’ll do it”) but while translating this sentence into Hindi, we need to insert ”ki” word in Hindi sentence, which is the translation of ”that”. For such insertions we take help of some link types like ”Ci” etc..

10.2 Reversibility Test

As we have mentioned above that at the same time we are preserving the information regarding English that LGP provides us. To test this, we are planning to build a tool which reverses the mapped relations back to their LGP links and vice verse.

10.3 Under Discussion

There are certain links like Vh, Vd, UN etc. which need a little discussion with scholars.

11 Conclusions

This is a task under development, so there may happen lots of modifications and improvements. We need a lot of feedback from all of you. Your feedback will help us to make the system more efficient.

12 Acknowledgement

We acknowledge the guidance of Dr. Vineet Chaitanya who provided the basic idea and its working module. In fact, the entire work has been done under his supervision.

References

- Link Grammar Parser
  http://www.link.cs.cmu.edu/link/dict/index.html
- Anusaaraka work-bench
  http://www.chinfo.org/Anusaaraka.html
  http://arxiv.org/abs/cs/0308019

- CLIPS
  http://clipsrules.sourceforge.net