English Parsers: Some Information based observations

Akshar Bharati, Amba Kulkarni
Department of Sanskrit Studies,
University of Hyderabad, Hyderabad, India
apksh@uohyd.ernet.in
Sriram Chaudhury
Language Technologies Research Center,
Inernational Institute of Information Technology,
Hyderabad, India
sriram_c@research.iiit.ac.in

Abstract

Last decade has seen introduction of several parsers for English ranging from rule based to statistical based. In recent years there is also a growing trend towards producing dependency output in addition to the constituency trees. The dependency format is preferred over the constituency not only from evaluation point of view but also because of its suitability for a wide range of NLP tasks. However there is no consensus among the dependency parser developers on the number of dependency relations and names of these relations.

Paninian Grammar (PG), the first dependency formalism, though is developed necessarily for Sanskrit, has potential to provide guidelines for producing the dependency output for English. We first summarize the issues involved with reference to English language parsing based on the dependency format output of the current English parsers. Next we highlight the Information theoretic point of view of Paninian Grammar. Fourth section contains guidelines for producing the dependency output for English, in the light of PG. We conclude with some suggestions for evaluation of existing parsers.

Key Words: Parsing, Paninian Grammar, Dependency Grammar, semantics, syntax, kaaraka roles, thematic roles

1 Introduction

Last decade has seen introduction of several parsers for English ranging from rule based to statistical. Within rule based again one sees parsers with a wide variety of formalisms such as Minipar (Lin, 1998) based on minimalism, enju parser (Yusuke and Junichi, 2005) and LKB parser (Copestake and Flickinger, 2000) based on HPSG, link parser (Sleator and Temperley, 1993) based on dependency grammar, XTAG parser (Joshi, 1985) based on Tree Adjoining Grammar, to name a few. There are around half a dozen statistical parsers for English viz. Collins (1999), Charniak (2000), Stanford parser (Klein and Manning, 2003), re-ranking parser (Johnson, 2005) and so on. The native output of all these parsers is naturally the formalisms they follow. In case of rule based parsers, it is the grammar formalism they are based on, and in case of statistical parsers, it is the Phrase structure trees, since they are trained on the Penn Treebank which is annotated using Phrase Structure Grammar (PSG).

Recent years has also seen a growing trend towards producing dependency output in addition to the constituency trees. The dependency format is preferred over the constituency not only from evaluation point of view (Lin, 1998) but also because of its suitability (Marneffe et al, 2006) for a wide range of NLP tasks such as Machine Translation (MT), information extraction, question answering etc. However no two dependency output formats match with each other. There is no consensus among the dependency parser developers on the number of dependency relations and names of these relations.

Paninian Grammar (PG), the first dependency formalism, though is developed necessarily for

Sanskrit, has potential to provide guidelines for producing the dependency output of English sentences. In fact, there have been attempts to apply PG to English. (Bharati et al, 1996, Bharati and Kulkarni, 2005).

We first summarize the issues involved with reference to English language parsing based on the dependency format output of the current English parsers. In the third section we highlight the Information theoretic viewpoint of PG, with special emphasis to English language. Fourth section contains guidelines for producing the dependency output for English, in the light of PG. Finally we give some suggestions for evaluation of different parsers.

2 Dependency format output: some issues related to English

A dependency relation is an asymmetric binary relation mapping a modifier to the modified. The word being modified is the head. A word may have several modifiers but can modify only one word. If there are n words in a sentence, n-1 relations are necessary and sufficient to describe the parsed output.

There is a very close relationship between the dependency grammar and the link grammar (Sleator and Temperley, 1993) on which is based the link parser. The relations in link parser, however, are not directional. The number of relations used in link parser is 106. Minipar also produces dependency format output and uses 59 relations. Carroll(Caroll et al, 1999) and King (King et al, 2003) have proposed a set of dependency relations. Marneffe et al (2006) have suggested modifications to these relations, largely based on practical considerations. The number of relations proposed by Marneffe are 47. Thus we see that there is a lot of variation among different parsed outputs with respect to the number of relations.

We looked at parsed outputs of different parsers for a wide range of sentences and recorded the phenomena where the parsed outputs differ. We also noticed certain cases where none of the parsers' performance was acceptable. The differences in their performance could be related to the issues summarized below.

a) Whether to treat function words such as prepositions, auxiliaries, etc. as words indicating relations thereby avoiding relations between these words with other content words or to treat these words at par with the content words?

This will have serious effect on the number of content words and the number of relations in a sentence.

b) The basic assumption of dependency grammar is that a modifier modifies only one word. In the following sentence

Ram went home and slept ---- (1)

Ram is a modifier of went as well as slept. Whether the parser should produce both the relations or only one?

Similarly in the sentences with missing wh -relativizer

I saw the man you love. --- (2)

The snake the mongoose attacked hissed loudly. -- (3)

whether the output should account for the missing wh-relativizer or not?

In case of subject and object control verbs such as

Ram persuaded Mohan to study well. ---(4)

Ram promised Mohan to study well. ---(5)

should the output account for the sharing of semantic roles by different verbs or not?

c) What should be the level of analysis – syntactic (specifying the subject, object relations), semantic (specifying the thematic roles), or something else?

d) Should the heads be decided semantically or syntactically? For example, in case of 'a cup of tea', the semantic head is tea, whereas the syntactic head is cup. In case of 'growth of industry', growth is both the semantic as well as syntactic head.

e) Should the sentences

Ram is good. --- (6)

and

Ram is a doctor. ---(7)

be treated alike, with semantic representation as good(Ram), and doctor(Ram) respectively, or should they be analyzed differently, reflecting the different underlying phrase structures?

To answer these questions, we look at English language from the 'information coding' point of view. We seek answers for the following questions.

i) What means does English uses to code the information about relations?

ii) What are the manners of coding the information, and finally,

iii) What is the semantic content of the relations?

3 Paninian Grammar

According to PG, a modifier may be classified into two major categories: samaanaadhikarana

(modifier and modified having the same locus), and vyadhikarana (modifier and modified have different loci).

Examples of samaanaadhikarana modifiers are

- a determiner modifying a noun (the boy)
- an adjective modifying a noun (good boy)

Examples of vyadhikarana modifiers are

- nominal expressions modifying a verbal root, also known as the kaaraka relations,
- a verb modifying another verb, etc.

Essentially, the samaanaadhikarana modifier and the corresponding modified head denote the same thing, and belong to the same word group[1]. So this kind of relation is a 'word-group-internal' relation. On the other hand the vyadhikarana modifier and the corresponding modified head belong to different word groups, and hence the relation involved here is 'across-the-word-group' relation. Further, the vyadhikarana modifiers are the building blocks of the parsed structure, with the samaanaadhikarana modifiers adding the flesh to this structure.

The most important vyadhikarana modifiers are the 'kaaraka' relations.

i) In Bharati et al (1998) it has been pointed out that English codes the kaaraka relations in position as well as through prepositions.

ii) Languages do not code all the kaaraka relations explicitly. For example, when a word has more than one kaaraka roles with respect to different verbs in the surface structure of a sentence, only one kaaraka relation is coded and other kaaraka relation need to be inferred from the language's grammatical rules (language conventions) or through the properties of lexical items. For example in sentence (1), it is the language convention which tells Ram is the subject of both the verbs went and slept. In sentences (2) and (3), it is the syntax of English which allows wh drop and thereby allow sharing of more than one kaaraka roles by the same nominal expression. In the sentence (5) the information that subject of 'study' is Ram, and in sentence (4) it is Mohan is coded in the meaning of lexical items promise and persuade respectively.

iii) According to PG, the kaaraka relations are the relations which map nominal expressions to verbal roots. These are syntactico-semantic relations. These indicate the optimum semantic analysis one can do using the language string and the language conventions alone without appealing to the world knowledge. Given the fact that present day computers are still not capable of handling the world knowledge, from computational point of view, therefore, it is a major milestone in the language analysis. One kaaraka relation may correspond to more than one thematic roles. For example, in the following sentences

Ram opened the lock with this key. ---(8)
This key opened the lock. ---(9)
The lock opened. ---(10)

Ram, *this key* and *the lock* are all 'karta', whereas their thematic roles are viz. agent, instrument and goal respectively. Similarly each semantic role may get realized into more than one kaaraka relations. For example, key in sentence (8) is karana kaaraka and in sentence (9) karta kaaraka. Lock is the karma kaaraka in sentences (8) and (9), whereas karta kaaraka in sentence (10).

To summarize,

i) English codes the kaaraka relations both in position as well as through the prepositions.

ii) Some relations are coded explicitly and some implicitly.

iii)The maximum semantics one can extract is the syntactico-semantic relations and not the thematic roles.

4 Guidelines for producing dependency output for English

We answer the issues raised in the second section, thereby leading to the guidelines for producing the dependency output for English.

a) In the light of earlier discussion, it is clear that we treat the prepositions connecting a noun with a verb or another noun as a *relation* rather than a content word. Further the auxiliaries together with the main verb form a 'semantic unit' leading to a word group with main verb as the head. Hence the auxiliaries should be grouped with the main verb, and there is no necessity of mentioning the internal relations.

b) Sentences (1) through (5) are all examples of kaaraka sharing and implicit encoding of the unspecified kaaraka relations. The implicit encodings are typically language grammar and lexicon specific and hence need to be made explicit in the parsed output.

c) On the basis of the discussion above, it is clear that, language codes only syntactico-semantic relations. So what one can extract from the language string alone is only kaaraka relations and not the thematic roles. But still a question remains to be answered, viz. How to extract the

1 Of course, there are cases where the words may belong to different word groups and still may have same locus, as in the case of 'He is a doctor'.

kaaraka relations from the syntactic relations? As has been pointed out by Bharati et al. (1998), the subject and object are the syntactic relations, whereas karta and karma are the syntactico-semantic relations. In active voice the occupant of the subject position, generally, corresponds to karta[2], and that of the object position corresponds to karma. It needs to be verified whether the object in English has the same 'semantic content' as that of karma as defined in PG. Till the detailed mappings from object to kaaraka roles are worked out, we map it to karma. The rules for assigning karta and karma role to the subject and object may be summarized as below:

If the verb is in active voice (with the exceptions listed below), the occupant of the subject position is karta, and that of object is karma.

If the verb is in passive voice, the occupant of the subject position is karma, and the by-object is the karta.

The exceptions are as follows:

In case of dummy there, the first noun group after the main verb is the karta.

In case of subject raising verbs such as 'seem' etc. the occupant of the subject position of seem is the karta of the subordinate verb with to infinitive.

d) Rules for determining the semantic head should be worked out for English, and one should provide the semantic heads and not the syntactic heads in the analysis.

e) In English the two sentences have different Phrase structures. But their semantic content is same. PG treats them in a uniform way, by postulating a samaanaadhikarana relation between Ram and good, and also between Ram and doctor. This in fact is an example of samaanaadhikarana modifier across the word groups!

5 Suggestions for evaluation of parsers

Parsers differ in their behavior with respect to the issues raised above. For example link parser treats prepositions as content words. It also treats sentences (6) and (7) differently. Stanford parser and Enju parser on the other hand try to do deeper semantic analysis leading to over-generalizations in some cases.

The differences among these parsers make it difficult to compare the parsers qualitatively. It is proposed that 'interfaces' based on the principles outlined above be developed to facilitate the comparison. These interfaces are also easy to use by a layman for understanding the 'parsed output' without any linguistic training (Bharati and Kulkarni, 2006).

In the light of above discussion the relations may be classified into three categories viz. word-group-internal relations, across-word-group-explicitly marked relations, and across-word-group-implicitly marked relations. The word-group-internal relations may be best handled by the constituency trees, whereas the across-word-group relations may best be handled by the dependency relations. Chunkers may be the reliable tools for marking the inter-word-grouping. The word grouper developed in-house performs better than the chunker on verb-auxiliary grouping. Handling the implicit relations involve some heuristic rules. These need to be, therefore, marked separately.

6 Conclusion

Interfaces based on PG are being developed for Link parser, Stanford Parser and Enju parser. For parsers producing only constituency output, we are using the Stanford parser's constituency to dependency format converter. The evaluation of these parsers following the guidelines mentioned above is underway.

Acknowledgment

Authors thank Dipti M Sharma for useful discussions on various issues related to English grammar. Amba Kulkarni also thanks G. Umamaheshwar Rao for useful discussions, and also for his comments and suggestions on the draft of the paper.

References

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

Akshar Bharati and Medhavi Bhatia and Vineet Chaitanya and Rajeev Sangal. 1998. *South Asian Language Review*, Creative Books, New Delhi.

Akshar Bharati and Amba P. Kulkarni. 2005. '*English from Hindi viewpoint: A Paaninian perspective*' In Platinum Jubilee conference of LSI at University of Hyderabad, Hyderabad, India. Dec 6-8, 2005

Akshar Bharati and Amba P. Kulkarni. 2006. '*Grammarian's shabdabodha for English Parsers*' In National seminar on 'Sanskrit for Innovation', Center for Advanced Studies in Sanskrit, Pune, India.

John Carroll, Guido Minnen, and Ted Briscoe. 1999. Corpus annotation for parser evaluation. In Pro-

2 However, it need not be so as has been pointed out by Bharati et, al. (2005)

ceedings of the EACL

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In Proceedings of NAACL-2000.

Euguen Charniak and Mark Johnson. 2005. '*Coarse-to-fine n-best parsing and MaxEnt descriminative reranking*' In Proceedings of the 43rd annual meeting of the ACL, pp. 173-180.

Michael Collins. 1999. Head-Driven Statistical Models for Natural Language Parsing. Ph.D. thesis, University of Pennsylvania.

Ann Copestake and Dan Flickinger (2000) An open-source grammar development environment and broad-coverage English grammar using HPSG In *Proceedings of the Second conference on Language Resources and Evaluation (LREC-2000)*, Athens, Greece.

Arvind K. Joshi. 1985. Tree Adjoining Grammar, In D. Dowty et.al. (eds.) Natural Language Parsing, Cambridge University Press.

Tracy H. King, Richard Crouch, Stefan Riezler, Mary Dalrymple, and Ronald Kaplan. 2003. The PARC 700 dependency bank. In 4th International Workshop on Linguistically Interpreted Corpora (LINC-03).

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. ACL 2003. pp. 423-430.

Dekang Lin. 1998. Dependency-based evaluation of MINIPAR. In Workshop on the evaluation of Parsing Systems, Granada, Spain.

Marie-Catherine de Marneffe and Bill MacCartney and Christopher D. Manning. 2006. 'Generating Typed Dependency Parses from Phrase Structure Parses' *To appear at LREC-06*.

Daniel D. Sleator and Davy Temperley. 1993. Parsing English with a link grammar. In Third International Workshop on Parsing Technologies.

MIYAO Yusuke, and TSUJII Jun'ichi. 2005. Probabilistic Disambiguation Models for Wide-Coverage HPSG Parsing In Proceedings of ACL-2005, pp. 83-90.