PROCESSING WORD ORDER VARIATION WITHIN A MODIFIED ID/LP FRAMEWORK

Pradip Dey
University of Alabama at Birmingham
Birmingham, AL 35294, U.S.A.

ABSTRACT

From a well represented sample of world languages Steele (1978) shows that about 76% of languages exhibit significant word order variation. Only recently has this wide-spread phenomenon been drawing appropriate attention. Perhaps ID/LP (Immediate Dominance and Linear Precedence) framework is the most debated theories in this area. We point out some difficulties in processing standard ID/LP grammar and present a modified version of the grammar. In the modified version, the right hand side of phrase structure rules is treated as a set or partially ordered set. An instance of the framework is implemented.

1. Introduction

From a well represented sample of world languages Steele (1978) shows that about 76% of the languages exhibit significant word order variation[1]. Until recently this widespread phenomenon was not given proper attention in natural language processing. The primary goal of this study is to develop computationally efficient and linguistically adequate strategies for parsing word order variation. The strategies are implemented in a network based parser. At first we characterize the basic problem at an abstract level without going into details of the problem in any specific language (in Section-2). Then, in Section-3, the details of the problems in a specific language, namely, Hindi, are presented.

The Immediate dominance and linear precedence (ID/LP) framework, developed by Gazdar and Pullum, is one of the most debated theories in the study of word order variation (Pullum 1982, Uszkoreit 1982, Shieber 1983, Barton 1985). The basic idea behind ID/LP framework is to separate immediate dominance from linear precedence in rewrite rules. Pullum (1982) expresses this via a metagrammar. The modified version presented in this paper expresses this directly in the object grammar eliminating the need for a metagrammar. It treats the right hand side of a PS (Phrase Structure) rule as a set or partially ordered set. Parsing with this type of rule can proceed by checking set membership.

2. The Word Order Problem in General

The word order problem is the problem of processing the whole range of word order variation occurring in natural languages. Some Australian languages such as Warlpiri show extreme word order variation (Hale 1983). Hindi, Japanese and German also allow considerable word order variation. In this section we develop descriptive formalisms and parsing mechanisms that are adequate for the whole range of word order variation.

Consider a grammar that allows a node labeled S to have daughters labeled $, O, and V in any linear order, and nothing else. Such a grammar can be presented with a set of rules such as that given in (2.1).

(2.1) $S \rightarrow \$OV,\quad S \rightarrow \$VO,\quad S \rightarrow O\$V,$
$S \rightarrow OV\$,\quad S \rightarrow V\$O,\quad S \rightarrow VO\$$

The problem with a grammar such as that given in (2.1) is that it needs too many rules to capture word order variation (in this case free word order). For 5 'words' such a grammar will need 5! = 120 rules. With the increase in the number of words, such a grammar will grow factorially. That is, for *n* number of words it will need *n!* rules.

There is a convenient way of 'collapsing' rules in GPSG (Generalized Phrase Structure Grammar) of Gazder (1981). It uses metarules that operate on basic rules to generate derived rules which then function as basic rules in derivations. Thus, (2.1) can be abbreviated as (2.2).

(2.2) Basic rule: $[_S\ \$\ O\ V]$

Metarule: $[_S ....X...Y...] \Rightarrow [_S ....Y...X...]$

where X and Y range over $, O, V.

Within GPSG Pullum (1982) suggests another solution which also involves a metagrammar. He suggests that a grammar such as (2.1) can be expressed via a metagrammar that treats immediate dominance and linear precedence separately. Pullum's theory is known as ID/LP analysis[2]. According to this theory grammar (2.1) "would be specified by means of the metagrammar" given in (2.3). Similarly, the metagrammar given in (2.4) "determines" the grammar shown in (2.5). In (2.3) and (2.4) immediate dominance statements are given under a, and linear precedence statements are given, under b. In the case of (2.3) however the set of linear precedence statements is empty. In the case of (2.4) $ < O means 'if any rule introduces $ and O, $ linearly precedes O'.

(2.3)a. $\{ S \rightarrow \$, O, V \}$ b. $\{ \emptyset \}$

(2.4)a. $\{ S \rightarrow \$, O, V \}$ b. $\{ \$ < O \}$

(2.5) $\{ S \rightarrow \$OV,\quad S \rightarrow \$VO,\quad S \rightarrow V\$O \}$

An important advantage of ID/LP analysis is that it can account for word order variation in a general way, capturing "analytical intuition, often hinted at in the literature, that fixing constituent order "costs" in the same way that having special NP case-marking rules or verb agreement rules does" (Pullum 1982: 211). The main disadvantage of the standard ID/LP framework is that it is difficult to process (Shieber 1983, Barton 1985).

The alternative solution proposed in this study treats the right hand side of a rule as a set[3]. Thus, the grammar in (2.1) can be presented in this format either as (2.6a) or as (2.6b). The latter rule is to be understood under the node admissibility condition.

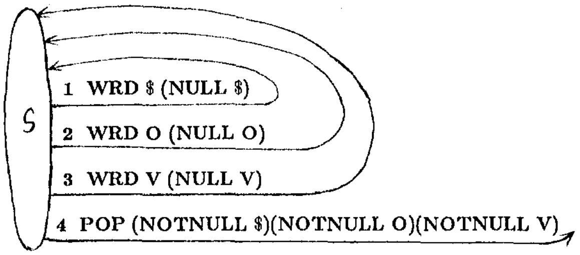
(2.6)a. $S \rightarrow \{ \$, O, V \}$ b. $\{_S\ \$, O, V \}$

Since the right hand side of the rule is a set, the order of $, O and V does not matter. In parsing, this solution has definite advantages. Firstly, the factorial growth of rules is eliminated. Secondly, parsing can proceed by checking set membership or set difference. That is, instead of 'ordered match' the parser has to do 'unordered match'. The precise way of doing it will vary from parser to parser. We describe one way of implementing it in the ATN (Augmented Transition Network (Woods 1970, Finin and Hadden 1977)) formalism.

Consider the ATN fragment presented below in (2.7) for the grammar given in (2.6). Conditions on arc are given in LISP like structures within parenthesis. Thus (null $) means 'if

$-register is empty'. By the arc WRD $ (null $), the 'word' $ will be accepted if no $ has previously been found. (In natural language, one can assume $ = Subject, O = Object, V = Verb, and use PUSH arc in place of WRD in the following diagram.)

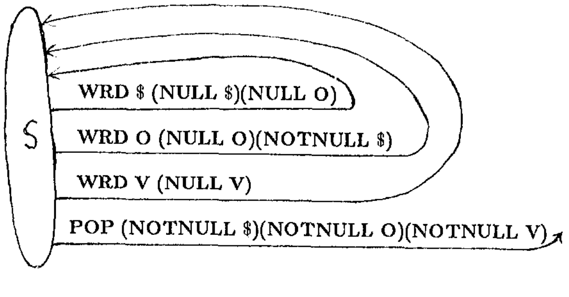
(2.7)



(2.7) parses any strings generated by (2.1) by scanning the input from left to right and checking the set membership. Thus, in recognition and parsing, (2.7) correctly reflects (2.6). Suppose, the input string is VO$. One way to see how a sentence is parsed is to trace through the analysis of the sentence as arc sequence. This string is accepted by the arc sequence (3, 2, 1, 4). The ATN given in (2.7) can be said to have conditioned multiple loops. For convenience of reference, we shall refer to ATN structures such as (2.7) as 'set-loops'. Further restrictions on set-loops (such as (2.7)) can be imposed and all constituent order variations can be parsed simply by imposing additional conditions on arcs. Thus, an ATN parser such as (2.8b) can parse the language generated by the grammar given in (2.5). Formally, (2.5) is presented with a partially ordered set such as (2.8a) in the proposed framework. The partial ordering is specified as a constrain after "/", as in a context-sensitive rule contexts are specified after "/".

(2.8)a. $\{_S \ \$, O, V \} / \$ < O$

(2.8)b.


S
WRD $ (NULL $)(NULL O)
WRD O (NULL O)(NOTNULL $)
WRD V (NULL V)
POP (NOTNULL $)(NOTNULL O)(NOTNULL V)

Suppose that $, O, and V are nonterminals which are further expanded by appropriate rewrite rules. Right hand side of such expansions can also show word order variations as shown in (2.9).

(2.9) $\$ \Rightarrow \{a, b, c\}, \quad O \Rightarrow \{d, e\}, \quad V \Rightarrow \{f, g, h\}$

So far, we have described parsing strategies for constituent order variations. However, in natural language we often find a discontinuous constituent. That is, an element can be moved out of its constituent (topicalization in English would be a good example if VP is a constituent) which can be described by categories with holes (eg. VP/NP). In cases such as this, VIR arcs in combination with hold lists are used in ATN (Bates 1978). Alternatively, temporary registers can be used to parse discontinuous constituents. Temporary registers are particularly suitable to handle large number of 'misplaced' words that cannot be handled by usual HOLD lists in combination with VIR arcs. We would like to apply the general strategies described above to the case of Hindi which shows considerable word order variation.

3. Word Order in Hindi

In Hindi, the order of the major constituents such as $ (Subject), O (direct Object), I (Indirect object), and V (Verb (+aux)) is free. For example, out of the four constituents present in (3.11.1), we can make twenty four variants of the same sentence, all of which are perfectly good in Hindi as is obvious from (3.11.1-24).

(3.11)1. mohan ne raam ko sev diaa thaa. ($IOV)
Mohan ag Ram to apple gave was
"Mohan gave the apple to Ram."

2. mohan ne raam ko diaa thaa sev. ($IVO)

3. mohan ne diaa thaa raam ko sev. ($VIO)

4. mohan ne diaa thaa sev raam ko. ($VOI)

:

24. diaa thaa sev mohan ne raam ko. (VO$I)

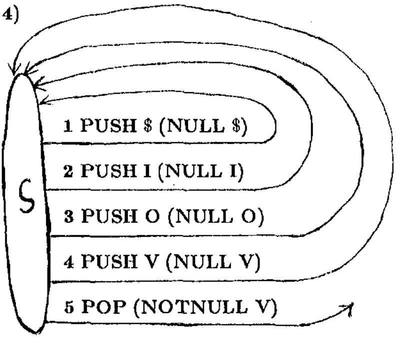
To capture the freedom of order of $, I, O, and V in sentences such as (3.11) we can have a rule such as (3.12) in the grammar of Hindi. The V alone can stand as a sentence in Hindi since it is highly inflected (see Kachru 1980). Hence (3.13) is more appropriate for Hindi where $, I, and O are given within paratheses to show their optional occurrence.

(3.12) $\{_S \ \$, I, O, V \}$

(3.13) $\{_S \ (\$), (I), (O), V \}$

We have been referring to rules such as (3.13) as set rules. An ATN fragment, such as (3.14) would be appropriate for (3.13).

(3.14)



(Assume appropriate subnets for $, I, O, V)

Suppose we are parsing (3.11.1) *mohan ne raam ko sev diaa thaa* "Mohan gave the apple to Ram". It is accepted by the arc sequence (1, 2, 3, 4, 5). The sentence given in (3.11.24) is accepted by the arc sequence (3, 1, 4, 2, 5). (3.14) captures constituent order variation in Hindi in a general way. However, it is to be noted that sentences such as (3.11.1) have bi-transitive (or double transitive) V. We have to impose more conditions on arc 5, POP, to parse intransitive and transitive sentences. Informally, the conditions are: (1) If the V is intransitive then the I and O must be empty. (2) If the V is transitive then the I must be empty. We have implemented a large parser of Hindi with wide coverage of construction types including relative clauses, interrogatives, passives, dative subjects, compound verbs and gapping which interact with word order variation (see Dey 1982, 1984).

Word order variation in Hindi is fairly restrictive. Thus, in the sentences of (3.11) the main verb must precede the AUX. (3.15) is unacceptable because it violates this restriction.

(3.15) * thaa diaa sev mohan ne raam ko.
was gave apple Mohan ag Ram to

Similarly, (3.16) violates the restriction that the case elements must follow the noun (Verma 1970).

(3.16) * ne mohan ram ko sev diaa thaa.
ag Mohan Ram to apple gave was

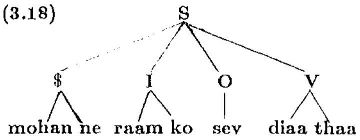
As obvious from (3.15-16), all rules of Hindi are not 'set-rules'. Thus, the subject-NP, $, cannot be expanded by (3.17a); rather, we use the usual (3.17b).

(3.17)a. $\{_{\$}\ \mathrm{NP},\ \mathrm{K}\ \}$ b. $[_{\$}\ \mathrm{NP}\ \ \mathrm{K}\ ]$

(Assume $[_{\mathrm{NP}}\ \textit{mohan}\ ]$, $[_{\mathrm{K}}\ \textit{ne}\ ]$)

In the modified ID/LP framework we allow strict order rules such as (3.17b), free order rules such as (3.14) and partial order rules such as (2.8a). We also allow notions like subject and object. That means the grammar is an annotated PS grammar. The parsing strategy suggested above for this grammar has an important consequence. It does not recognize VP (that dominates V, O, I) as a constituent. It advocates a 'flat' structure for sentences as shown in (3.18).

(3.18)



"Mohan gave the apple to Ram."

It should be noted that actual structural representations should be given with more details. Some parse trees given by the parser are presented below:

(3.19) (parse (mohan ne raam ko sev diaa thaa))

```
(S (NP-subj (NP (DET nil) (ADJ) (N mohan)) (K-ag ne))
   (NP-ind (NP (DET nil) (ADJ) (N raam)) (K-dat ko))
   (NP-obj (NP (DET nil) (ADJ) (N sev)))
   (VX (ADV) (V diaa (AUX thaa))))t
```

(3.20) (parse (diaa thaa sev raam ko mohan ne))

```
(S (NP-subj (NP (DET nil) (ADJ) (N mohan)) (K-ag ne))
   (NP-ind (NP (DET nil) (ADJ) (N raam)) (K-dat ko))
   (NP-obj (NP (DET nil) (ADJ) (N sev)))
   (VX (ADV) (V diaa (AUX thaa))))t
```

It is to be noted that though case words like *ne* and *ko* often help to identify subjects, objects etc. the parser must use semantic information in order to identify them in sentences such as the ones given in (3.21-22) (see Dey 1984).

(3.21.) (parse (mohan anDaa khaataa hai))
Mohan egg eats is
"Mohan eats an egg"

```
(S (NP-subj (NP (DET nil) (ADJ) (N mohan)) (K-ag nil))
   (NP-ind nil (K-dat nil))
   (NP-obj (NP (DET nil) (ADJ) (N anDaa)))
   (VX (ADV) (V khaataa (AUX hai))))t
```

(3.22) (parse (anDaa mohan khaataa hai))
"Mohan eats an egg"

```
(S (NP-subj (NP (DET nil) (ADJ) (N mohan)) (K-ag nil))
   (NP-ind nil (K-dat nil))
   (NP-obj (NP (DET nil) (ADJ) (N anDaa)))
   (VX (ADV) (V khaataa (AUX hai))))t
```

4. Concluding Remarks

Processing word order variation with new techniques within the modified ID/LP framework seems to be revealing. But, it is not context-free unlike other ID/LP based parsers. Detailed comparison of ID/LP based parsers is a subject of further research.

Footnotes:

[1] I am grateful to A. K. Joshi, A. Kroch, T. Finin, D. Hindle, S. Gambhir, K. Reilly, B. Kaemmerer, K. Ryan, H. Bullock and the anonymous COLING-86 referees for their helpful suggestions and comments.

[2] See Uszkoreit (1982) for an implementation of ID/LP framework.

[3] The right hand side of a rule should be treated as a restricted set rather than as a pure set. The restriction can be stated as follows: a member of a set can occur only once in the set unless specified otherwise. Thus, though formally the following two sets are equal, under the restrictions imposed they are not equal: $\{\ \$,\ \mathrm{O},\ \mathrm{V}\ \} \neq \{\ \$, \mathrm{O}, \mathrm{V}, \$, \mathrm{V}\ \}$

References:

Barton, G. E. Jr. 1985. "On the Complexity of ID/LP Parsing" Computational Linguistics, 11, 205-218

Bates, M. 1978. "The Theory and Practice of Augmented Transition Network grammars". In L. Bolc (ed.) Natural Language Communication with Computers. Spring Verlag, Berlin: 191-259.

Dey, P. 1982. "A Parser for Hindi". Presented to 4th South Asian Languages Round Table, Syracuse, 1982.

-----1984. Computationally Efficient and Linguistically Adequate Parsing of Some Natural Language Structures. Ph.D. diss., University of Pennsylvania.

Finin, T. and G. Hadden 1977. "Augmenting ATNs". In the Proceedings of the 5th IJCAI.

Gambhir, V. 1980. Syntactic Restrictions and Discourse Functions of Word Order of Standard Hindi. Ph.D. diss., University of Pennsylvania.

Gazdar, G. 1981. "Unbounded Dependencies and Coordinate Structure", Linguistic Inquiry 12, 155-184.

Hale, K. 1983. "Warlpiri and the Grammar of Non-configurational Languages," Natural Language and Linguistic Theory, 1. 5-48.

Kachru, Y. 1980. Aspects of Hindi Syntax. Delhi: Monohar.

Pullum, G. K. 1982. "Free Word Order and Phrase Structure Rules," NELS, 12, 209-222.

-----1983. "Context-freeness and the Computer Processing of Human Languages," Proc. of the 21st ACL Conference.

Shieber, S. 1983. "Direct Parsing of ID/LP Grammars," Linguistics and Philosophy 7:2.

Steele, S. 1981. "Word Order Variation: A Typological Study," in J. Greenberg (ed.) Universals of Language, Vol. 4. Stanford, CA: Stanford University Press.

Uszkoreit, H. 1982. "A Framework for Parsing Partially Free Word Worder," Proceedings of the 21st ACL Conference.

Verma, M. K. 1971. The Structure of Noun Phrase in English and Hindi. Delhi: Motilal Banarsidas.

Woods, W.A. 1970. "Transition Network Grammars for Natural Language Analysis," Comm. of ACM 13, 591-606.