

AN LR CATEGORY-NEUTRAL PARSER WITH LEFT CORNER PREDICTION

Paola Merlo

University of Maryland/ Université de Genève

Faculté des Lettres

CH-1211 Genève 4

merlo@divsun.unige.ch

Abstract

In this paper we present a new parsing model of linguistic and computational interest. Linguistically, the relation between the parser and the theory of grammar adopted (Government and Binding (GB) theory as presented in Chomsky(1981, 1986a,b) is clearly specified. Computationally, this model adopts a mixed parsing procedure, by using left corner prediction in a modified LR parser.

ON LINGUISTIC THEORY

For a parser to be linguistically motivated, it must be *transparent* to a linguistic theory, under some precise notion of transparency (see Abney 1987). GB theory is a modular theory of abstract principles. A parser which encodes a modular theory of grammar must fulfill apparently contradictory demands: for the parser to be explanatory it must maintain the modularity of the theory, while for the parser to be efficient, modularization must be minimized so that all potentially necessary information is available at all times.¹ We explore a possible solution to this contradiction. We observe that linguistic information can be classified into 5 different classes, as shown in (1), on the basis of their *informational content*. These we will call IC Classes.

- (1) a. Configurations: sisterhood, c-command, m-command, \pm maximal projection ...
- b. Lexical features: \pm N, \pm V, \pm Funct, \pm c-selected, \pm Strong Agr ...
- c. Syntactic features: \pm Case, $\pm\theta$, $\pm\gamma$, \pm barrier.
- d. Locality information: minimality, binding, antecedent government.
- e. Referential information: \pm D-linked, \pm anaphor, \pm pronominal.

¹On efficiency of GB-based systems see Ristad(1990), Kashkett(1991).

This classification can be used to specify precisely the amount of modularity in the parser. Berwick(1982:400ff) shows that a modular system is efficient only if modules that depend on each other are compiled, while independent modules are not. We take the notion of *dependent* and *independent* to correspond to IC Classes, in that primitives that belong to the same IC Class are dependent on each other, while primitives that belong to different IC Classes are independent from each other. We impose a modularity requirement that makes precise predictions for the design of the parser.

Modularity Requirement (MR) *Only primitives that belong to the same IC Class can be compiled in the parser.*

RECOVERING PHRASE STRUCTURE

According to the MR, notions such as headedness, directionality, sisterhood, and maximal projection can be compiled and stored in a data structure, because these notions belong to the same IC Class, *configurations*. These features are compiled into context-free rules in our parser. These basic \bar{X} rules are augmented by λ rules licensed by the part of Trace theory that deals with configurations. The crucial feature of this grammar is that nonterminals specify only the \bar{X} projection level, and not the category. The full context-free grammar is shown in Figure 1.

The recovery of phrase structure is a crucial component of a parser, as it builds the skeleton which is needed for feature annotation. It must be efficient and it must fail as soon as an error is encountered, in order to limit backtracking. An LR(k) parser (Knuth 1965) has these properties, since it is deterministic on unambiguous input, and it has been proved to recognize only valid prefixes. In our parser, we compile the grammar shown above into an LALR(1) (Aho and Ullman 1972) parse table. The table has been modified

$X'' \rightarrow Y'' X'$	specification
$X'' \rightarrow X' Y''$	
$X' \rightarrow X Y''$	complementation
$X' \rightarrow Y'' X$	
$X' \rightarrow Y'' X'$	modification
$X' \rightarrow X' Y''$	
$X'' \rightarrow Y'' X''$	adjunction
$X'' \rightarrow X'' Y''$	
$X \rightarrow \text{empty}$	empty heads
$X'' \rightarrow \text{empty}$	empty Xmaxs

Figure 1: Category-Neutral Grammar

in order to have more than one action for each table entry.² Three stacks are used: a stack for the states traversed so far; a stack for the semantic attributes associated with each of the nodes; a tree stack of partial trees. The LR algorithm is encoded in a *parse* predicate, which establishes a relation between two sets of 5-tuples, as shown in (2).³

$$(2) T_i \times S_i \times A_i \times C_i \times PT_i \rightarrow T_j \times S_j \times A_j \times C_j \times PT_j$$

Our parser is more elaborate and less restrictive than a standard LR parser, because it imposes conditions on the attributes of the states and it is nondeterministic. In order to reduce the amount of nondeterminism, some predictive power has been introduced. The cooccurrence restrictions between categories, and subcategorization information of verbs is compiled in a table, which we call Left Corner Prediction Table (LC Table). By looking at the current token, at its category label, and its subcategorization frame, the number of choices of possible next states can be restricted. For instance, if the current token is a verb, and the LR table allows the parser either to project one level up to V' , or it requires to create an empty object NP, then, on consulting the subcategorization information, the parser can eliminate the second option as incorrect if the verb is intransitive.

RESULTS AND COMMENTS

The design presented so far embodies the MR, since it compiles only dependent features in two tables off-line. Compared to the use of partially or fully instantiated context-free grammars, this

²This modification is necessary because the grammar compiled into the LR table is not an LR grammar.

³In (2) T_k is an element of the set of input tokens, S_k is an element of the set of states in the LR table, A_k is an element of the set of attributes associated with each state in the table, C_k is an element of the set of chains, i.e. displaced element, and PT_k is an element of the set of tokens predicted by the left corner table (see below).

Grammar	Instantiated	\bar{X}
Number of Rules	51	16
Number of States	46	14
Shift/reduce conflicts	224	24
Reduce/reduce conflicts	270	36

Figure 2: Numbers

organization of the parsing algorithms has been found to be better on several grounds.

Consider again the \bar{X} grammar that we use in the parser, shown in Figure 1. One of the crucial features of this grammar is that the nonterminals are specified only for level and headedness. This version of the grammar is a recent result. In previous implementations of the parser, the projections of the head in a rule were instantiated: for instance $NP \rightarrow YP N'$. Empirically, we find that on compiling the partially instantiated grammar the number of rules is increased proportionately to the number of categories, and so is the number of conflicts in the table. Figure 2 shows the relative sizes of the LALR(1) tables and the number of conflicts. Moreover, on closer inspection of the entries in the table, categories that belong to the same level of projection show the same reduce/reduce conflicts. This means that introducing unrestricted categorial information increases the size of the table *without* decreasing the number of conflicts in each entry, i.e. without reducing the nondeterminism in the table.

These findings confirm that categorial information can be factored out of the compiled table, as predicted by the MR. The information about cooccurrence restrictions, category and subcategorization frame is compiled in the Left Corner (LC) table, as described above. Using two compiled tables that interact on-line is better than compiling all the information into a fully instantiated, standard context-free grammar for several reasons.⁴ Computationally, it is more efficient.⁵ Practically, manipulating a small, highly abstract grammar is

⁴Fully instantiated grammars have been used, among others, by Tomita(1985) in an LR parser, and by Dorr(1990), Fong(1991) in GB-based parsers.

⁵It has been argued elsewhere that for context-free parsing algorithms, the size of the grammar (which is a constant factor) can easily become the predominant factor for all useful inputs (see Berwick and Weinberg 1982). Work on compilation of parsers that use GPSG seems to point in the same direction. The separation of structural information from cooccurrence restrictions is advocated in Kilbury(1986); both Shieber(1986) and Phillips(1987) argue that the combinatorial explosion (Barton 1985) of a fully expanded ID/LP formalism can be avoided by using feature variables in the compiled grammar. See also Thompson 1982.

much easier. It is easy to maintain and to embed in a full-fledged parsing system. Linguistically, a fully-instantiated parser would not be transparent to the theory and it would be language dependent. Finally, it could not model some experimental psycholinguistic evidence, which we present below.

PSYCHOLINGUISTIC SUPPORT

A reading task is presented in Frazier and Rayner 1987 where eye movements are monitored: they find that in locally ambiguous contexts, the ambiguous region takes *less* time than an unambiguous counterpart, while a slow down in processing time is registered in the disambiguating region. This suggests that selection of major categorial information in lexically ambiguous sentences is delayed.⁶ This delay means that the parser must be able to operate in absence of categorial information, making use of a set of category-neutral phrase structure rules. This separation of item-dependent and item-independent information is encoded in the grammar used in our parser. A parser that uses instantiated categories would have to store categorial cooccurrence restrictions in a different data structure, to be consulted in case of lexically ambiguous inputs. Such design would be redundant, because categorial information would be encoded twice.

CONCLUSION

The module described in this paper is implemented and embedded in a parser for English of limited coverage, but it has some shortcomings, which are currently under investigation. Refinements are needed to compile the LC table automatically, to define IC Classes predictively instead of by exhaustive listing. Finally, a formal proof is needed to show that our definition of *independent* and *dependent* is always going to increase efficiency.

ACKNOWLEDGEMENTS

This work has benefited from suggestions by Bonnie Dorr, Paul Gorrell, Eric Wehrli and Amy Weinberg. The author is supported by a Fellowship from the Swiss-Italian Foundation.

⁶For instance, in the sentences in (3), (from Frazier and Rayner 1987) the ambiguous target item, shown in capitals in (3)a, takes less time than the unambiguous control in (3)b, while there is a slow down in the disambiguating material (in italics).

- (3) a. The warehouse FIRES numerous employees each year.
 b. That warehouse fires numerous employees each year.

REFERENCES

- Abney Steven 1987, "GB Parsing and Psychological Reality" in MIT Parsing Volume, Cognitive Science Center.
- Aho A.V. and J.D. Ullman 1972, *The Theory of Parsing, Translation and Compiling*, Prentice-Hall, Englewood Cliffs, NJ.
- Barton Edward 1985, "The Computational Difficulty of ID/LP Parsing" in *Proc. of the ACL*.
- Berwick Robert 1982, *Locality Principles and the Acquisition of Syntactic Knowledge*, Ph.D Diss., MIT.
- Berwick Robert and Amy Weinberg 1982, "Parsing Efficiency, Computational Complexity and the Evaluation of Grammatical Theories", *Linguistic Inquiry*, 13:165-191.
- Chomsky Noam 1981, *Lectures on Government and Binding*, Foris, Dordrecht.
- Chomsky Noam 1986a, *Knowledge of Language: Its Nature, Origin and Use*, Praeger, New York.
- Chomsky Noam 1986b, *Barriers*, MIT Press, Cambridge MA.
- Dorr Bonnie J. 1990, *Lexical Conceptual Structure and Machine Translation*, Ph.D Diss., MIT.
- Fong Sandiway 1991, *Computational Properties of Principle-based Grammatical Theories*, Ph.D Diss., MIT.
- Frazier Lyn and Keith Rayner 1987, "Resolution of Syntactic Category Ambiguities: Eye Movements in Parsing Lexically Ambiguous Sentences" in *Journal of Memory and Language*, 26:505-526.
- Kashkett Michael 1991, *A Parameterised Parser for English and Warlpiri*, Ph.D Diss., MIT.
- Kilbury James 1986, "Category Cooccurrence Restrictions and the Elimination of Metarules", in *Proc. of COLING*, 50-55.
- Knuth Donald 1965, "On the Translation of Languages from Left to Right", *Information and Control*, 8.
- Phillips John 1987, "A Computational Representation for GPSG", DAI Research Paper 316.
- Ristad Eric 1990, *Computational Structure of Human Language*, MIT AI Lab, TR 1260.
- Shieber Stuart 1986, "A Simple Reconstruction of GPSG" in *Proc. of COLING*, 211-215.
- Thompson Henry 1982, "Handling Metarules in a Parser for GPSG" in *Proc. of COLING*.
- Tomita Masaru 1985, *Efficient Parsing for Natural Language*, Kluwer, Hingham, MA.