COORDINATION IN RECONNAISSANCE-ATTACK PARSING*

Michael B. KAC
Department of Linguistics
University of Minnesota
Minneapolis, MN 55455, USA

Thomas C. RINDFLESCH
Department of Linguistics and
Academic Computing Services and Systems
University of Minnesota
Minneapolis, MN 55455, USA

Abstract

A proposal for recognizing coordinate structures using the 'reconnaissance-attack' model is presented. The approach concentrates on distinguishing predicate coordination from other types of coordination and suggests that low-level structural cues (such as the number of predicates, coordinators, and subordinators occurring in the input string) can be exploited at little cost during the early phase of the parse, with dramatic results. The method is tested on a text of 16,000 words.

0. Introduction

Coordinate structures are difficult to parse in part because of the problem of determining, in a given case, what kinds of constituents are being coordinated. The examples in (1) will illustrate:

(1) a. John hits Fred and the other guys.
b. John hits Fred and the other guys attack him.
c. When John hits Fred and the other guys attack him.

Many variations on this theme are possible, to the point where serious doubts are raised regarding the efficacy in this domain of conventional parsers of either the top-down or bottom-up variety. In such parsers, it is necessary either to invoke backtracking to undo the effects of incorrect hypotheses or to store large numbers of alternatives until local indeterminacies are resolved. In this paper, we will suggest an alternative approach based on the 'Reconnaissance-Attack' model described in Kac et al. 1986 (and more fully in Rindflesch forthcoming), designed to skirt many of the problems associated with more traditional designs.

*The work presented here was supported under Control Data Corporation Grant #86M102 to the University of Minnesota (Jeanette Gundel, Larry Hutchinson and Michael Kac, Principal Investigators). Special thanks are due to Nancy Hedberg and Kari Swingle for their assistance on the project, and to Walling Cyre, technical liaison with CDC. The authors are listed in alphabetical order.

Our proposal is theoretical in two senses. On the one hand, it does not present a detailed picture of an actual parsing algorithm, being intended rather to show that a significant body of linguistic data supports the contention that rapid, early resolution of local structural indeterminacies of the kind exemplified in (1) is feasible in the vast majority of cases. On the other hand, it is also based on a significant idealization, namely that each word belongs to only one syntactic category. Our intent is, in part, to show the applicability to a difficult parsing problem of a technique which can be found in other AI domains (Kowalski 1979) but which seems to have been little exploited in work on natural language processing[1].

1. Theoretical Background

In a Reconnaissance-Attack parser, no structure-building is attempted until after an initial 'overflight' of the entire sentence has been made, directed at obtaining information, provided by low-level structural cues, which can then be exploited in narrowing the range of available options at a later point. (We assume here that the cues used are present in a minimally analyzed string, by which we mean one about which the only structural information available concerns the relative order and category membership of the individual words.) It is of the utmost importance to bear in mind that in this approach, if a given case cannot be resolved at a given point in the parse, there is no guessing as to which type of coordination might obtain and hence no need to backtrack for the purpose of undoing the effects of erroneous hypotheses; rather, the parser simply defers the decision to a later phase at which more structural information is available. Note as well that this is not 'bottom-up' parsing in the usual sense either, since where more than one possibility is logically available, the parser makes no attempt to represent them all and cull out the false positives later on; there is a strict principle of 'altruism avoidance' (that is, never undertaking computational effort without a guaranteed payoff) which compels the parser to give no answer at all during

[1]The approach described in Sampson 1986, while quite different in its actual character, is nonetheless similar in spirit to what we are proposing.

a particular phase if more than one answer is possible in principle given the information available to that point. (If, at the end of the process, unresolved indeterminacies remain, ambiguity is predicted.)

Intuitively, the difference between Reconnaissance and Attack is that Reconnaissance constitutes the gathering of information while Attack constitutes anything which involves decision-making. More formally, Reconnaissance can be viewed as a series of parameter-setting operations each of which is done independently of any of the others while Attack requires simultaneous access to all parameters.

It is worth noting that there does not appear to be any reason to exclude in principle the possibility of hybrid models in which principles of the sort we shall develop below are invoked prior to the application of a parser along the lines of those described in e.g. Dahl and McCord 1983 or Fong and Berwick 1985. Our principal contention is that whatever choices are made about how to go about 'parsing proper' (that is, actually building a syntactic representation for an input sentence), there is an advantage to having certain global structural information already available rather than starting 'blind'.

Following Kac 1978 and 1985, we subsume under a single rubric of 'predicate coordination' the coordination of verbs, VP's, and S's on the rationale that common to all three types is that they have the effect of rendering predicates 'equiordinate' (that is, so related that neither is sub- or superordinate to the other). In e.g.

(2) I believe that John likes Mary and Harry admires Sue.

the verbs *likes* and *admires* are both subordinate to *believe* but neither is subordinate to the other. Similarly, in a sentence like (1b) above, *hits* and *attacks* are both 'topmost' in the ordination scheme. (For a more detailed development of the theory of ordination relations, see Rindflesch forthcoming.) In this approach a distinction is made between STRICT and LOOSE coordination (two coordinate expressions are strictly so if separated by at most a conjunction, loosely coordinate otherwise, as in e.g. *John and Mary ran* vs. *John ran, and Mary (too)*) and also between PRIMARY and SECONDARY coordination. The primary coordinates in a coordinate structure are the largest coordinate expressions (e.g. the S's in sentential coordination), while the secondary coordinates are smaller expressions contained in the primary ones taken (by the theory) to be coordinate by virtue of the coordination of the containing expressions; for example, the predicates of coordinate sentences (both VP's and V's) are secondary coordinates in a sentential coordination.

For purposes of parsing, we assume that the first task is to coordinate WORDS rather than the larger expressions containing them; that is, secondary coordinates are sought first, and the primary coordinates in which they appear are identified later. This is consistent with the overall theoretical approach, described in more detail in Rindflesch op. cit., which is much more akin to dependency syntax than to phrase structure analysis. (See also Kac and Manaster-Ramer 1986.)

2. A Sketch of the Parsing Strategy

In this paper, our focus will be on determining, from a minimally analyzed string, whether or not a given instance of *and* or *or* enters into a predicate coordination as defined above. (A longer paper giving full details of the approach is in preparation.)

In the earliest stages of parsing a given sentence containing a coordinating conjunction, each conjunction is identified as either (a) definitely involved in a predicate coordination, (b) as definitely not involved in such a coordination, by virtue of failing certain necessary conditions for being so involved, or (c) as of indeterminate status which must be resolved (if possible) in a later phase of the parse. The following principles are invoked for this purpose:

Applied early in Attack:

(3) LIMITS CONSTRAINT (Rindflesch forthcoming)
The number of predicate-coordinating conjunctions in a sentence must be smaller than the number of verbs.

(4) POSITION CONSTRAINT (Kac 1978, 1985)
If a coordinating conjunction conjoins expressions X and Y, it lies somewhere between X and Y.

Applied late in Attack:

(5) MAIN PREDICATE CONSTRAINT
There is at least one predicate in every sentence which is not subordinate to any other predicate in that sentence.

(6) EQUIORDINATION CONSTRAINT
If two predicates are coordinate then they are also *equiordinate.*

The principles (3-6) are all rather straightforward, even common-seasical; it is nonetheless not entirely uninteresting to learn that they form the basis for an extremely effective parsing strategy.

Reconnaissance involves a single pass through the current string, the first steps being lexical lookup and counting and indexing all categories. The information gained from this counting and indexing is then used to eliminate impossible structures, via a check for compatibility with the principles (3-6) above.

In order to deal with coordination two ancillary lists, called POTENTIAL COORDINATION LISTS, are associated during Reconnaissance with each conjunction which occurs in the input string. One of these, PCL-L, contains words which occur to the left of the conjunction with which the list is associated; each of these words could thus potentially serve as the left-hand member of a coordination effected by that conjunction. The other list, PCL-R, serves a similar function for words which occur to the right of the conjunction. Two elements can be coordinated only if one occurs in PCL-L for a given conjunction and the other occurs in PCL-R for that conjunction

The constraints which apply early in Attack presuppose no information beyond what is gathered during Reconnaissance and are used to eliminate words in the input string as candidates for inclusion in these lists (on the assumption that it is best to eliminate as much as possible as early as possible on the basis of the least possible amount of information and thus enhance the efficiency of the parser). The remaining constraints remove words from the lists. In the early stages of the parse, each of these lists may be quite long, but as the parse proceeds, elements are deleted by the invocation of the Attack principles, until, for well formed input strings, each list contains only elements which, on some admissible reading of the input, can enter into a coordination effected by the associated conjunction. (In ambiguous cases such as *John believes the boys and the girls believe Fred*, each list would have more than one member.) In unambiguous cases, it can be determined that a conjunction is definitely involved in predicate coordination if both its PCL-L and its PCL-R contain exactly one predicate and no other word, and a conjunction is definitely not involved in predicate coordination if either of its PCL's does not contain any verb at all. The coordination status of a conjunction is indeterminate with regard to predicate coordination when, although both PCL's contain a verb, one (or both) of them contains at least one additional word.

A natural question to ask at this point is whether the strategy just described is not just bottom-up parsing of the familiar sort. The answer is no, for at least two reasons. First, the PCL's do not hold fully specified analyses of substrings of the input; they contain only words which, on the basis of information so far available, cannot be excluded from consideration as potential coordinates of the conjunction associated with a given pair of lists. Nor do the lists hold potential conjunct pairs. (Suppose, for example, that PCL-L and PCL-R respectively hold words A, B and C and X and Y. There is an obvious difference between the two lists and the six conjunct pairs derivable from them, that is, <A, X>, <A, Y>, <B, X> ...)

Reconnaissance consists of a single pass through the input string, during which, after lexical lookup, each word is indexed, a count is kept of the number of tokens of each category which occurs in the input string, and the PCL's are created for each conjunction. After Reconnaissance, if there are any conjunctions, the PCL's are filled subject to the Limits Constraint and the Position Constraint. The Limits Constraint is applied only when PCL-L is filled, and the Position Constraint is applied only when PCL-R is filled. PCL-L is filled first. A word is put into PCL-L if and only if its index is less than the index of the conjunction with which the PCL-L is associated and the number of words of this category in the string is greater than one (when this second condition is met the Limits Constraint is satisfied). Thus when *hits* is encountered while the parser is attempting to fill PCL-L for the conjunction in (1a), *hits* is not put into PCL-L since there is only one verb in the string. It can accordingly be determined that the conjunction is not coordinating predicates in (1a), since there will be no verb in either of the PCL's.

In order to satisfy the position constraint when PCL-R is filled, a word is put into PCL-R if and only if its index is greater than the index of the current conjunction and there is already a word in the PCL-L for the current conjunction which has the same category as the word being considered for inclusion in the PCL-R for this conjunction. For example, in processing

(7) John and Martha know Fred likes Dora

The parser does not put either *know* or *likes* into PCL-R because there are no verbs in PCL-L.

As will be discussed below, in the vast majority of cases in at least one domain the type of coordination occurring in a sentence can be determined solely on the basis of these straightforward principles. In these cases, the structure encountered is similar to that seen in (1a). In order to determine whether predicates are being coordinated in structures like those seen in (1b)

and (1c) it is necessary to have somewhat more information about the input string.

The additional information required to deal with strings such as (1b) and (1c), only one of which involves predicate coordination despite the fact that the two are nearly identical, concerns the relationships which obtain between predicates in a complex sentence. These relationships are enforced by constraints (5-6) above, in conjunction with

(8) MULTIPREDICATE CONSTRAINT
Every predicate in a multipredicate sentence must be in an ordination relationship with another predicate in the same sentence.

The task of the parser confronted with polypredicational examples of the type in which we are interested is to distinguish coordination of predicates, as in (1b), from sub-/superordination, as in (1c). During the Attack phase of the parse, we capitalize on the fact that it is possible to resolve certain indeterminacies about the structure of a sentence on the basis of only incomplete information about the ordination relations which obtain in the sentence. This depends on the fact that ordination relations can exist only in the presence of ORDINATION RELATION SIGNALS (ORS's). While space does not permit a complete discussion of ORS's here, some examples are subordinators (e.g. complementizers and subordinating conjunctions) and the marking of verbs like *know* and *believe* as allowing predicational objects. Here we will concentrate on subordinators. Each subordinator in a sentence must be associated with a verb in that sentence, and this association causes that verb to be necessarily subordinate to some other predicate. The fact which is of value in parsing coordinate structures is that this can be known even before the superordinate partner of the subordinate predicate has been identified. For example in (1c) even before anything else is known about the structure of the sentence, it can be determined that the subordinator *when* is associated with *hits* and that therefore *hits* will have to be subordinate to some other predicate in that sentence.

As noted above, the parsing principles applied during Attack remove words from the PCL's. In the parse of (1b), while there are nouns and verbs in both PCL's at the beginning of Attack, all the nouns are removed, as Attack proceeds, from both PCL's, leaving only the verbs to be coordinated. The way in which Attack accomplishes this is as follows.

There is more than one predicate in (1b) and thus the predicates have to be in an ordination relation in order to satisfy the Multipredicate Constraint. This relation cannot be subordination, since no subordinating ORS is present; assuming coordination to be the only other possibility, and given that there is a coordinating conjunction between the two predicates, we conclude that the predicates are in fact coordinate. In order to satisfy all of the constraints Attack must therefore remove *John* and *Fred* from PCL-L leaving *hits* as the sole member of that list. It must also remove *guys* and *him* from PCL-R leaving *attack* as the only word in that list. The configuration of these lists thus indicates that the only possible coordinates in (1b) are *hits* and *attack*.

These same principles determine that predicate coordination cannot obtain in (1c). As Attack begins, PCL-L for the conjunction in this string contains *John, hits,* and *Fred.* PCL-R contains *guys, attack,* and *him.* Since there is more than one predicate in this string, the predicates will have to be in an ordination relationship, but it will have to be a relationship of subordination rather than coordination. *Hits* will have to be subordinate to some predicate in this sentence by virtue of the fact that it is associated with the subordinator *when.* (We do not state the means by which this is established here; see Rindflesch op. cit. for details.) Since *hits* is necessarily non-main, any predicate coordinated with it would also have to be non-main, by the Equiordination Constraint. Therefore it is not possible to coordinate *attack* with *hits* in (1c) since such a construal would cause the Main Predicate Constraint to be violated. The only possible ordination relationship which can obtain between the predicates in (1c) is one in which *hits* is subordinate to *attack.* Therefore, *hits* must be removed from the PCL-L and *attack* must be removed from the PCL-R. From this it can at least be determined that (1c) does not involve predicate coordination.

3. Empirical Support for the Approach

To test the effectiveness of the strategy described above, we subjected to analysis a corpus of nearly 16,000 words (15,985 to be exact). The texts used were specifications and design requirements (5 in all) applying to hardware manufactured by Control Data Corporation, supplied to us in machine-readable form. Each text was run through a concordance program which identified all tokens of *and* and *or*; and for each token of each conjunction, the containing sentence was then analyzed (by hand). A total of 431 tokens of the two conjunctions occurred in the corpus, 362 of them in complete sentences (as opposed to section heads or fragments, which were ignored). As noted earlier, we did not, in undertaking the analysis, take into account the fact that there is widespread category-label ambiguity ('CLA')

in English; this represents a significant idealization of the data, but it is not a cheat. The problem with regard to coordination with which we are concerned is that even in cases where no CLA occurs, problems of the sort exemplified by (1) arise. That the overall problem is even worse than we make it out to be does not invalidate our claims, though it means -- and we are fully aware of this -- that the account is incomplete.

Of the conjunctions occurring in complete sentences, the type of coordination in which each was involved was correctly ascertainable via application of the five constraints in 91 % of the total number of cases, given only the information made available by Reconnaissance plus the ORS-verb associations made early in Attack. 82 % of the total number of cases were correctly identified solely on the basis of the Limits Constraint and the Position Constraint. Of the remaining cases, at least 51 % submit to resolution during the Attack phase on the basis of the comparatively low-level structural information concerning ordination relations (Main Predicate, Equiordination, and Multipredicate Constraints). (This figure is conservative in that further principles may be identified in the future which would improve performance.)

4. Examples

We conclude with an analysis of some sentences from the corpus, to illustrate the approach in more detail. The discussion here concentrates on our stated goal of determining for any conjunction what kinds of expressions are being coordinated. A large number of the sentences in the corpus, with respect to coordination, have a structure resembling

(9) A single sector single port buffer will provide speed matching between the host interface and the controller.

In this sentence, there is only one predicate (*will provide*) and furthermore there is no predicate to the right of the conjunction. Either the Limits Constraint or the Position Constraint can therefore determine solely on the basis of information determined during Reconnaissance that there is no predicate coordination in (9).

The somewhat more complex structure of (10) can also be handled without difficulty.

(10) The primary purposes of the special functions are to support diagnostic analysis, data recovery, and download capabilities.

Although there are two predicates in (10) (*are* and *to support*), The Position Constraint correctly predicts that they cannot be coordinate since they are not separated by the conjunction in this sentence.

Sentences containing more than one conjunction submit to the principles we propose in this paper, as illustrated by

(11) The primary structures and relationships of these memory blocks are illustrated in Figure 11 and are defined more precisely in later sections.

The first conjunction in (11) does not effect predicate coordination, while the second does. The Position Constraint assures the correct analysis for the first conjunction: PCL-L for the first conjunction will not contain a verb since there are no verbs to the left of this conjunction; consequently, no verb will be put in the corresponding PCL-R, thus precluding predicate coordination for the first conjunction in (11). When the PCL's are filled for the second conjunction in (11), they will both contain nouns as well as predicates; hence either could potentially be coordinated. However, since there are two predicates in (11) (*are illustrated* and *are defined*) and since there are no subordinating ORS's in the sentence, the predicates in fact must be coordinate in order to satisfy the Multipredicate Constraint.

Although the PCL's for the conjunction *or* in (12) will initially contain both nouns and verbs, the correct analysis of this sentence does not involve predicate coordination.

(12) When switch position 1 is set to the "off" position, a 2 byte or a 16 bit word will be available on the data bus bits 0-F.

The analysis of (12) is similar to the analysis of (1c). There are two predicates in the string (*is set* and *will be available*), one of which (*is set*) is necessarily non-main due to its association with the subordinating conjunction *when*. Were these predicates to be coordinated they would both be non-main by the Equiordination Constraint. Therefore, the only way the Multipredicate Constraint and the Main Predicate Constraint can be satisfied is to consider there to be no predicate coordination in this sentence.

References

Dahl, V. and M.C. McCord. 1983. Treating coordination in logic grammars. Am. J. Comp. Ling. 9.69-81.

Fong, S. and R.C. Berwick. 1985. New approaches to parsing conjunctions using Prolog. Proceedings of the Twenty-Third Annual Meeting of the Association for Computational Linguistics. 118-126.

Kac, M.B. 1978. Corepresentation of Grammatical Structure. Minneapolis and London University of Minnesota Press and Croom Helm.

---------. 1985. Constraints on predicate coordination. Indiana University Linguistics Club.

---------. 1986. Parsing without (much) constituent structure. Proceedings of the Eleventh International Conference on Computational Linguistics. 156-158.

---------, T.C. Rindflesch and K.L. Ryan. 1986. Reconnaissance-attack parsing.Proceedings of the Eleventh International Conference on Computational Linguistics. 159-160

Kowalski, R. 1979. Algorithm = logic + control. Communications of the ACM. 22.424-436.

Rindflesch, T.C. Forthcoming. University of Minnesota Dissertation.

Sampson, G. 1986. A stochastic approach to parsing. Proceedings of the Eleventh International Conference on Computational Linguistics. 151-155.