CONJUNCTIONS AND MODULARITY IN LANGUAGE ANALYSIS PROCEDURES

Ralph Grishman
Department of Computer Science
Courant Institute of Mathematical Sciences
New York University
New York, New York, U. S. A.

Summary

The further enrichment of natural language systems depends in part on finding ways of "factoring" the effects of various linguistic phenomena, so that these systems can be partitioned into modules of comprehensible size and structure. Coordinate conjunction has a substantial impact on all aspects of syntactic analysis -- constituent structure, grammatical constraints, and transformations. If the rules of syntactic analysis were directly expanded to accomodate conjunction, their size would increase severalfold. We describe below the mechanisms we have used to localize the effect of conjunction in our natural language analyzer, so that most of the rules of our grammar need not explicitly take conjunction into account.

Introduction

Progress in computational linguistics depends in part on identifying ways of decomposing complex components of the analysis procedure into more elementary constituents corresponding to separable or nearly separable aspects of linguistic structure. If this "factoring" is successful, the constituents will in sum be substantially simpler than the component they replace, thus clarifying the linguistic theory represented by the analysis procedure and paving the way for further enrichment of the language system.

A familiar example of such factoring began with early context-free natural language parsers. Such parsers tried to use a context-free grammar to describe both the constituent structure of sentences and the most basic grammatical constraints (e.g., number agreement between subject and verb and within noun phrases, subcategorization constraints). Because these grammatical constraints have a multiplicative rather than additive effect on the size of the grammar, this approach rapidly becomes unwieldy. In its place arose two component systems with a separate procedural restriction component for expressing such constraints.

We have devoted considerable effort to factoring out the effects of coordinate conjunction on language analysis. Conjunction greatly increases the number of different possible structures which the components of syntactic and semantic analysis must be able to process. If each component were simply expanded to accomodate all these additional structures, the resulting system would be huge and the essential function of the components greatly obscured. We have sought instead to isolate, as much as possible, the effects of conjunction within separate modules which modify the operation of the parser and restructure the result of the parse.

Our System in Brief

Over the past 15 years, members of the Linguistic String Project and Computer Science Department at New York University have developed a powerful set of tools for natural language analysis [1,2]. Our primary objective has been the automated retrieval of information from natural language texts; for the past several years we have been applying these tools to the construction of systems for gathering statistics and answering questions about hospital records (initially, radiology reports; currently, discharge summaries).

We have divided the task of answering questions about such texts into two subtasks [3]. The first of these is the automatic mapping of these texts into a tabular data base structure called an information format. We term this mapping procedure formatting [4]. The second subtask is the retrieval of information from this data base in response to natural language queries. Both subtasks -- the analysis of the texts and the analysis of the queries -- involve several stages of syntactic processing followed by several of semantic processing. The syntactic processing is similar in the two cases, although the semantic processing is quite disparate: in the formatting, it involves the mapping of sentence constituents into the format; in the question answering, it involves a translation into logical form (an exten-

sion of predicate calculus) and thence into a data base retrieval request. We shall focus in this paper on the effects of conjunction on the syntactic processing, although we shall also comment at the end on the interaction of the syntactic and semantic processing.

Syntactic processing is done into two stages: parsing and transformational decomposition. Parsing is done by a top-down context-free parser augmented by grammatical constraints expressed in an special-purpose procedural language, Restriction Language [2,5]. The resulting organization is similar to that of an augmented transition network (ATN), although sharper separation is maintained between the context-free grammar which defines the constituent structure and the restrictions which implement the grammatical constraints. In contrast to most ATNs, transformational decomposition is performed as a separate stage following the parse [6].* The decomposition regularizes the sentence structure by such operations as converting passive sentences to active, expanding relative clauses and other noun modifiers to full sentential structures, etc.

Incorporating Coordinate Conjunction

In this section we shall briefly consider the effect of coordinate conjunction on each aspect of syntactic processing, and describe how we have added modules to our processing components to account for these effects.

Constituent Structure

Let us consider first the problem of the new structures introduced by conjunction. The allowed patterns of conjoinings in a sentence are quite regular. Loosely speaking, a sequence of elements in the sentence tree may be followed by a conjunction and by some or all of the elements immediately preceding the conjunction. For example, if the top-level sentence structure is subject - verb - object and an "and" appears after the object, the allowed patterns of conjoinings include subject - verb - object - and - subject - verb - object ("I drank milk and Mary ate cake."), subject - verb - object - and - verb - object ("I drank milk and ate cake."), and subject - verb - object - and - object ("I drank milk and seltzer."). There are certain exceptions, known as gapping phenomena, in which one of the elements following the conjunction may be omitted; for example, subject - verb - object - and - subject - object ("I drank milk and Mary seltzer.").

We could extend the context-free component of our surface grammar to account for these patterns. For example, in place of the production

```
S   -> SUBJ VERB OBJ
```

we would have the set of productions

```
S   -> SUBJ CA1 VERB CA2 OBJ CA3
CA1 -> SUBJ CA1 |
       null
CA2 -> SUBJ CA1 VERB CA2 |
       VERB CA2 |
       null
CA3 -> SUBJ CA1 VERB CA2 OBJ CA3 |
       VERB CA2 OBJ CA3 |
       OBJ CA3 |
       null
```

(this does not include gapping). The trouble with coordinate conjunctions is that they can occur almost anywhere in the structure of a sentence. Thus the same changes which we made above to the definition of S would have to be made to all (or at least many) of the productions in the grammar. Clearly, such an extension to the grammar could increase its size by perhaps an order of magnitude.

One alternative is to automatically generate the additional elements and productions needed to account for conjunction as required during the parsing process. When a conjunction is encountered in the sentence, the normal parsing procedure is interrupted, a special conjunction node is inserted in the parse tree (such as the CAn nodes above), and the appropriate definition is generated for this conjunction node. This definition allows for all the alternative conjoined element sequences, like the definitions of the CAn shown above. Conjoinings not fitting the basic pattern, such as gappings, are still included explicitly in the grammar. An interrupt mechanism of this sort is part of the Linguistic String Project parser [1]. A similar mechanism is included in Woods' augmented transition network parser [8] and a number of other systems.

---

* The separation of constituent structure, restrictions, and transformations is another example of the modularity we try to achieve in our system. See Pratt [7] for a discussion of the modularity of augmented context-free analyzers.

Restrictions

The restrictions enforce grammatical constraints by locating and testing constituents of the parse tree. One of the simpler restrictions in the Linguistic String Project grammar is WSEL1, verb-object selection for noun objects. Verbs may be marked (in the dictionary) as excluding certain classes of noun objects; WSEL1 verifies that the object is not a member of one of these classes. For instance, the verb "eat" is coded as excluding objects of the class NSENT1, which includes such words as "fact", "knowledge", and "thought."* The sentence "John ate his thought." would therefore fail WSEL1 and be marked as ungrammatical by the parser.

Explicitly modifying each restriction to account for possible conjoined structures would expand that component several fold. Most restricions, however, apply distributively to conjoined structures -- a constraint is satisfied if it is satisfied separately by each conjunct. For example, when the object is conjoined (verb noun1 and noun2) verb-object selection must be satisfied both between verb and noun1 and between verb and noun2. Thus in "John ate meat and potatoes.", WSEL1 must separately check selection between "ate" and "meat" and between "ate" and "potatoes". This constraint can exclude incorrect analyses for some conjoined sentences. For instance, in "John ate his sandwich and thought about Mary.", it excludes the analysis where John ate his thought about Mary.

Our implementation takes advantage of the fact that most restrictions apply distributively. The restrictions are stated in terms of a set of grammatical routines which locate constituents of the parse tree; for example, the CORE routine locates the head noun of a noun phrase. In a conjoined context, these routines are in effect multi-valued; in "John ate meat and potatoes.", the CORE OF THE OBJECT has two values, "meat" and "potatoes". We achieve this effect through a non-deterministic programming mechanism which is invoked by the routines when a conjoined structure is encountered [2,9]. This mechanism automatically reexecutes the remainder of the restriction for each value of the routine (each conjunct). In this way, the effect of conjunction is largely isolated within these grammatical routines. Restrictions which do not distribute (such as number agreement) must still be explicitly modified for conjunction, but these represent a relatively small fraction of the grammar.

---

* NSENT1 is one of several noun classes defined in the Linguistic String Project grammar in terms of the types of sentential right modifiers they can take (such as "the fact that John is here").

Transformational decomposition

The transformations regularize the parse by incrementally restructuring the parse tree, and are therefore almost all affected by the possible presence of conjunctions in the portion of the tree they manipulate. Most of the transformations, however, only rearrange elements within a single sentential structure or noun phrase. We therefore chose the expand each conjoined structure into conjoined complete sentential structures or noun phrases at the beginning of transformational decomposition (for example, "John baked cookies and made tea." would be expanded to "John baked cookes and John made tea."); in this way most of the transformations are unaffected by the presence of conjunctions.

The rules for determining quantificational structure, however, must take account of the copying which occurs when expanding conjoined structures (for example, "Some people speak English and understand Japanese." is not synonymous with "Some people speak English and some people understand Japanese."). In simplest terms, quantifiers derived from noun phrases which are copied during conjunction expansion (such as "some people" in the last example) must be assigned wider scope than the logical connective derived from the conjunction. We do this by assigning a unique index to each noun phrase in the parse tree, copying the index along with the noun phrase in the transformations, and checking these indices during the scope analysis which is a part of the translation to logical form. Similar account must be taken of copied determiners and quantifiers in conjoined noun phrases (because, for example, "ten colleges and universities" is not necessarily synonymous with "ten colleges and ten universities").

Sentence Generation

As part of our question-answering system, we generate answers by translating from logical form (extended predicate calculus) into full English sentences [10]. There is a close parallel between the components for sentence analysis and sentence generation; in par-

ticular, the last major step in generation is the application of a set of generative transformations. In accordance with the basic symmetry between analysis and generation, the generative transformations operate on trees containing conjunctions only of full sentential structures and noun phrases. Conjunction reduction (changing, for example, "John ate cake and John drank milk." to "John ate cake and drank milk.") is performed at the end of the transformational cycle. Most of the generative transformations operate within a single sentential structure or noun phrase. As a result, the generative transformations, like the analytic transformations, are for the most part not affected by the presence of conjoined structures.

Discussion

In the preceding sections we have described the effect of coordinate conjunction on all the components of a syntactic analyzer. We have shown how we have been able to encapsulate the changes required to these components -- as an interrupt mechanism for our context-free parser; as a non-deterministic programming mechanism invoked by the routines used by the restrictions; as a set of expansion routines preceding transformational decomposition. We have thus avoided the need for pervasive changes which would have substantially enlarged, complicated, and obscured the original components. In addition, our approach has isolated and characterized the effect of conjunction in such a way that it may be carried forward to future systems and other groups.

Although modularity is generally regarded as a desirable objective, it is sometimes claimed that it imposes constraints on the communication between modules which will ultimately lead to unacceptable losses of efficiency. We would respond that some constraints are necessary if a complex system is to be manageable and comprehensible. If the mode of interaction is appropriately chosen and sufficiently powerful (such as the interrupt mechanism and the non-deterministic programming mechanism) the resulting system will be both clearly structured and reasonably efficient.

Acknowledgements

The author wishes to acknowledge the primary roles played by Naomi Sager and Carol Raze Friedman in the design of the conjunction mechanisms. Carol Friedman developed the routines and restrictions for conjunction and the conjunction expansion procedure. Ngo Thanh Nhan implemented the conjunction transformations for question analysis and the conjunction reduction routine for sentence generation.

This research has been supported in part by Grant No. N00014-75-C-0571 from the Office of Naval Research; in part by the National Science Foundation under Grants MCS78-03118 from the Division of Mathematical and Computer Sciences and IST-7920788 from the Division of Information Science and Technology; and in part by National Library of Medicine Grant No. LM02616, awarded by the National Institutes of Health, DHEW.

References

1. N. Sager, Syntactic Analysis of Natural Language. Advances in Computers 8, 153-188. Academic Press, New York (1967).

2. R. Grishman, N. Sager, C. Raze, and B. Bookchin, The Linguistic String Parser. AFIPS Conference Proceedings 42, 427-434. AFIPS Press, Montvale, New Jersey (1973).

3. R. Grishman and L. Hirschman, Question Answering from Natural Language Medical Data Bases. Artificial Intelligence 11, 25-43 (1978).

4. N. Sager, Natural Language Information Formatting: The Automatic Conversion of Texts to a Structured Data Base. Advances in Computers 17, 89-162. Academic Press, New York (1978).

5. N. Sager and R. Grishman, The Restriction Language for Computer Grammars of Natural Language. Communications of the ACM 18, 390-400 (1975).

6. J. Hobbs and R. Grishman, The Automatic Transformational Analysis of English Sentences: An Implementation. Int'l J. of Computer Mathematics, Section A, 5, 267-283 (1976).

7. V. Pratt, Lingol -- A Progress Report. Advance Papers Fourth Int'l Joint Conf. Artificial Intelligence, 422-428 (1975).

8. W. A. Woods, An Experimental Parsing System for Transition Network Grammars. Natural Language Processing, Courant Computer Science Symposium 8, 111-154 (1973).

9. C. Raze, A Computational Treatment of Coordinate Conjunctions. Am. J. Computational Linguistics, microfiche 52 (1976).

10. R. Grishman, Response Generation in Question-Answering Systems. Proc. 17th Annl. Meeting Assn. Computational Linguistics, 99-101 (1979).