The Constituent Object Parser: Syntactic Structure Matching for Information Retrieval [1]

Douglas P. Metzler and Stephanie W. Haas

Department of Information Science
University of Pittsburgh

1. Introduction

There has long been interest in the idea of using syntactic information as part of an information retrieval strategy. People clearly gather information about the meaning of text (e.g., a sentence) both from the meanings of the individual words contained in it and from the structure in which the individual pieces are put together. Since syntax reflects this structure, it would seem that using syntactic information in addition to information about the presence of query terms would increase the performance of an information retrieval system. Conventional information retrieval systems do not employ syntactic information. They are primarily term based. Thesauri or similar methods can be used to generalize or specialize the actual terms included in the query. But only crude constraints on the organizational relationships between terms may be expressed, such as that they must appear in the same sentence, or within a given number of words of each other, or in a certain order. There is no way to express syntactic relationships, such as which term is the head of a phrase and which terms are its modifiers or what the scope of a modifier is. For example, consider the terms *junior* and *college*. A query requesting that they both appear in the same sentence could retrieve documents about *junior college* and *college juniors* as well as documents in which the two are barely related. (e.g., *high school juniors attending summer classes at a local college*). A user interested in documents about *junior college* may well have no interest in *college juniors* per se, and certainly does not want documents in which the two terms are not related at all, but that "accidentally" fit the query.

Unfortunately, developing the use of syntactic information in a workable system has proven difficult and such attempts have had mixed results at best (Salton & McGill, 1983, chp. 7, Salton, 1988). We believe that a major reason for the rather limited success enjoyed by previous attempts to use syntactic information to improve information retrieval performance is that they have not utilized the appropriate aspects of syntactic description. There seems to be relatively little to be gained from determining, for instance, whether two terms are in the same noun phrase, since, first, the noun phrase could express both the concept described in the query and a different concept that uses the same words (the "reverse" of the desired concept), and second, the target relationship can be expressed across noun phrase boundaries. We believe that the aspect of syntactic description that is most relevant to the semantic composition of larger linguistic entities is the hierarchical structure of these entities. There is a long history of work in cognitive psychology investigating the hierarchical nature of cognitive structures and processes in general, and our work is in part motivated by this tradition. (See e.g., Glass, Holyoak & Santa, 1979). Note, too, that the strongest psycholinguistic evidence for the cognitive reality of syntax is related to the hierarchical structural properties of these syntactic descriptions, rather than the fine grained details of the elements in the hierarchies. (see e.g., Clark & Clark, 1977, chp. 2)

Our approach is based on the relationships between terms which are expressed in hierarchical descriptions of sentences and phrases augmented with a single uniform relation, called *dependency*. This indicates which branch under any given node points to the basic concept at that node (the dominant branch) and which points to a modifier of the basic concept (the dependent branch). The intention to use these types of descriptions for pattern matching in the context of information retrieval placed constraints on the nature of the descriptions we developed, while in turn, the nature of these descriptions placed constraints on the development of our approach to parsing and the implementation of our parser. But at the same time, the simplicity of our descriptive formalism has permitted us to take advantage of some design decisions that make it easier to approach the building of a robust general natural language parser.

The Constituent Object Parser (COP) is designed to be used as a filter for conventional information retrieval systems. That is, COP is designed to analyze the structures of queries and the structures of the sentences in the

[1] This research has been supported by NSF grant IST-8520217, and equipment grants from Texas Instruments. Preliminary work on the project was supported by a grant form Online Computer Library Center (OCLC) Inc., Dublin, Ohio.

document surrogates that have been returned by a conventional system (or potentially in full text of such documents), in order to judge more precisely how relevant the document is likely to be to the query. Thus it is a precision enhancing mechanism, which indirectly can also help recall by allowing a more general conventional query to be used initially. This is in contrast to approaches (e.g., Sparck Jones & Tait, 1984) which attempt to use syntactic information to produce as many alternative phrasings of a query as possible even though most of these alternatives will not be found in the database, and also to approaches such as automatic indexing, which place all the emphasis on preprocessing the documents or surrogates, and attempt thereby to anticipate all of the query forms which might be relevant to each document. COP is inherently more efficient than these approaches because it uses a reasonably efficient initial (term based) retrieval approach to identify the candidates for more detailed matching and only processes these identified potential matches. COP is intended to capture exactly the level of syntactic detail which is pertinent to making the kinds of distinctions relevant for this purpose. By providing structural descriptions of the composite linguistic entities in a document and in a query, it is possible, in effect, to estimate the likelihood of the individual terms being used in similar ways in a document and query. Alternatively, the process can be viewed as gaining an implicit approximation or index to the similarities between the meanings of the composite entities without doing a semantic analysis. (The latter is not yet feasible for anything approaching the breadth of coverage of a typical information retrieval system and worse yet, can not be made domain, i.e. database, independent.) This paper describes the constraints that the information retrieval problem places on the nature of the linguistic representations intended to address this issue, the nature of the information retrieval motivated linguistic representations we have been developing, and the effects these representations have had on our approach to parsing including how COP handles ambiguity, various types of incomplete information, and the problem of reference.

2. Scope and Dependency

The syntactic descriptions produced by COP are summarized in only two syntactic relations, scope and dependency (Moulton & Robinson, 1981). The dependency relation indicates which branch is the dominant term (this generally coincides with the syntactic "head" of the phrase) and which term is dependent. In the example above, if the user is interested in *college juniors*, he or she wants the system to retrieve documents in which *juniors* is dominant over *college*; that is, the topic is *juniors* modified by *college*. The same modification relationship can be expressed in many different syntactic forms, between which the dependencies remain constant. For example, the topic *college juniors* could also be expressed as *juniors at/in college* or *juniors who are in college* or *juniors attending college*. Since only the dependency between the terms is used, rather than more detailed syntactic information such as the type of constituent in which the modifier appears, the precise phrasing does not matter. Scope refers to whether or not two terms (or larger constituents of the tree) actually have a dependency relation between them. Dependency relations are transitive, and thus it is possible to have distant terms related by dependency. But in general, not all pairs of terms in a structure have a dependency relation defined between them. (Figure 1.)

Figure 2 shows several noun phrases in which the head of the phrase is the dominant concept, and the modifiers are dependent. Note that similar concepts have similar structural descriptions, while non-matching concepts have contrasting structural descriptions. The same idea applies at the sentence level as well. Figure 3 shows how similar structural descriptions capture the relatedness of active, passive, interrogative and relative clause expressions of the same underlying propositional content. Notice that the basic subject, verb and object components of the underlying form are distinguishable in that the ("deep") subject is the most dominant concept of the three, the ("deep") object is the least dominant, and the verb is intermediate. [2] Note also that although other branches may exist between these constituents, this will not influence the relationships between them, and hence it will not influence the ways in which the matching process interprets their relatedness. Since we are concerned with producing scope and dependency trees as outputs, the COP grammar builds trees that are almost entirely binary. This enables us to determine the scope and dependency relationships between each piece of a constituent token as it is built. The basic matching procedure exploits this simple structural formalism to determine whether concepts are being used in compatible ways in a query and an abstract. The system can rank a match according to the likelihood that two terms are in the appropriate dependency relationship, the likelihood that they are in the opposite dependency relationship, or the fact that there is no defined dependency relationship between the terms.

We do not claim that the use of COP to enhance information retrieval would improve the precision of a search for all possible queries or databases. It is most likely to improve the precision of a search when the same query terms can appear in more than one scope and dependency relationship in a meaningful way in the same

[2] It is unusual in linguistics to consider the subject as dominant over the verb. This is motivated here by the notion that subjects are more important as indices to knowledge than are verbs (even if the verb is more central linguistically to the nature of the sentence), and also by the fact that it is easier (computationally) to keep the subject / object mapping distinct when they are separated by a branch for the verb.

database or document collection. For instance, the sentences *The plants eat the insects* and *The insects eat the plants* describe opposite concepts, and therefore have opposite scope and dependency relationships. Figure 3 shows that in the first sentence, *plants* is dominant over *insects*, while figure 4 shows that in the second sentence, *insects* is dominant over *plants*. Further, many documents on these topics might appear together (for example in an agricultural database). COP would be effective in screening out the unwanted documents by ranking documents containing the correct scope and dependency relationships above those containing the incorrect relationships. COP is also useful if the query terms could appear together in the same document without really being related, especially where the terms in the correct relationship mean something quite specific. For example, the terms *natural, language* and *processing* could appear in documents about data processing languages, (e.g., *natural approaches to developing data processing languages*), or even compiler design, yet they describe a precise concept in the phrase *natural language processing*. This phrase can not, however, be treated as a fixed string or idiom, since it might appear as *computer processing of natural language* or *techniques for processing natural language*. As these examples illustrate, the structural analysis supplied by COP, like other information retrieval techniques, can be augmented significantly by the use of thesaurus techniques or frame-based inheritance mechanisms to expand or contract the conceptual scope of a given query. Syntactic structure and thesaurus relations appear to be inherently orthogonal, and both capabilities should be provided in a full information retrieval implementation. For instance, the user should be able to obtain a match between *locusts eat corn* and *insects eat plants*. COP is less useful in situations where the terms mean the same or almost the same thing regardless of their scope and dependency relationships, as in *teaching English* and *English teaching*, or where the terms are not likely to appear in the same document at all unless the query topic is being addressed. One of the issues we are beginning to investigate is how well users can anticipate whether or not syntactic structures will be useful in formulating a particular query.

In addition to helping specify a query more precisely, COP allows the user to express a query in natural language phrases or sentences, rather than restricting the search to combinations of terms, particularly indexing terms. This has obvious potential benefits in enabling naive users to formulate useful queries. It is also useful in searching for concepts that do not fit well into the established structure of a given field or database.

3. Design Decisions

The previous sections argued that relatively simple hierarchical descriptions of the organization of sentence and phrase structure can (at least sometimes) be used to more precisely indicate the intended meanings of a query than can the terms alone or other sorts of structures (e.g., Boolean combinations) composed of terms. In order to put this idea to work it is necessary to build a system capable of producing these structures from the natural language expressions found in documents or surrogates, and in the natural language queries that this approach to information retrieval permits. Although it might appear possible simply to extract the hierarchical structure from the syntactic descriptions provided by an existing method of syntactic parsing, we have not done so for a number of reasons. Existing parsers are not designed for handling the kinds of structures (syntactic and pseudo-syntactic) that are found in abstracts. Further, many parsers take advantage of the syntactic and semantic characteristics of their intended domain, rendering them much less effective in other domains. Because we can take advantage of the nature of the intended use of the outputs of our parser without taking advantage of any particular domain, we have taken the position that it is better to build a parser for this purpose from scratch, rather than to try to adapt an existing approach. The basic point is that although our parser is simpler in many regards than other current approaches, it is not simply subsumed by them. (That is, its processes and structures are not simply a collapsing of finer grained operations and descriptions utilized in other approaches.) The intended application of this parser affects its design in several ways.

First, since we can anticipate the ways in which the outputs will be utilized, we can postpone some of the processing to the utilization stage, where it can be recast as an interpretation problem rather than one of structure building. For instance, we can ignore certain kinds of ambiguity (such as prepositional phrase attachment), and produce as output a simple canonical representation, if we build into the utilization procedures the capability of unpacking these canonical representations and recognizing the potential ambiguities that might be present.

Second, the parser's most important task is to produce a scope and dependency structure for any natural language input it might encounter. COP does not have to make a judgement regarding the input's well-formedness or grammaticality, nor must it explain such ungrammaticality, as theories of grammar must do. As a result, the parser "over-accepts" language that other systems might reject.

Third, we assume that in the domain of information retrieval any information about a document is more useful than no information. For example, even if the matching procedures can not definitely affirm that a document's scope and dependency relationships match those of a query, the information that at least they are not in the opposite relationship, and therefore definitely irrelevant, is still helpful. For some types of syntactic structures, this

is the most scope and dependency information that is available without calling on semantic information.

In sum, these design considerations and the system characteristics to which they led, result in a system which deals with some very complex issues in natural language processing, albeit at a relatively shallow level of analysis. These phenomena, e.g., ambiguous modification structures, conjunction, and intra-sentence pronoun reference, are in fact frequently considered to be outside the realm of syntax. It is possible to treat them here within syntax because of (1) our ability to tolerate incorrect analyses, and (2) our ability to anticipate the uses made of the syntactic descriptions.

3.1. Ambiguity

One of the major ways in which the intended application of COP influenced its design and permitted us to solve a difficult natural language processing problem concerns the handling of ambiguity. This is done by using canonical representations and post-processing matching procedures, rather than by attempting to derive a single unambiguous representation during parsing itself.

Many constructions in language, such as prepositional phrase attachment and adverbial modification, can cause a combinatorial explosion of possible syntactic structures. In the famous example, *I saw the man on the hill* produces two possible parses; one attaches the prepositional phrase to *the man*, and the other attaches it to the verb. But *I saw the man on the hill with the telescope* has six possible structures. These ambiguities (which can amount to hundreds of possibilities for a sentence of only average length), create great difficulties for natural language processing systems which are constrained to choose only a single, correct, interpretation, or for systems which attempt to produce all possibilities (e.g., as output from a syntactic parser intended to be filtered by a separate semantic component). Since we can anticipate the ways in which the outputs of COP will be interpreted, (i.e., by the dependency matching procedures), we can allow the parser to produce a single canonical representation for a sentence, that in effect captures all of the alternative ambiguous interpretations, as long as the matching processes can correctly interpret the representations. "Correctly interpret" in this regard does not mean determining the uniquely correct representation that the writer intended. It means recognizing the potential ambiguities that are implicitly encoded in the canonical representation so that matches with any one of the potential ambiguous alternatives can be recognized.

3.1.1. Prepositional Phrases

In terms of the representations produced and utilized by COP, the general issue of attachment is essentially one of scope. A modifier, such as a prepositional phrase, is always dependent on the constituent which it is modifying, and scope indicates what that dominant structure is. In figure 5a, the prepositional phrase *under 35 years old* is attached to the concept *delegates* whereas in figure 5b it is attached to *candidate*. Similarly, in *I saw the man on the hill with the telescope*, the question is whether the telescope is the instrument of the *seeing* action, or whether it is identifying which man is being seen, or which hill is the location. In COP, no attempt is made to determine which interpretation is correct in a given instance. Rather, prepositional phrases are always given the widest possible scope, by forcing their attachment to the noun or verb immediately to their left, a strategy which is known as rightmost attachment (Frazier & Fodor, 1978). This is similar to the method used in EPISTLE (Jensen & Heidorn, 1983) and PEG (Jensen & Binot, 1988). The matching procedure is then allowed to retrieve any possible dependency relations by recursively retrieving any (transitively represented) dependencies implicitly encoded in the tree representations. Since this procedure gives the prepositional phrases the widest possible scope, any errors that are made in matching are overpermissive. At worst, a prepositional phrase will be construed to modify a concept that it was not intended to modify, and the matching procedures will score the match too highly.

3.1.2. Adverbs

The problem of the scoping of adverbs is similar to that of the scoping of prepositions, and the parser treats adverbs in much the same way that it treats prepositions, except that they are treated as features (of other constituents) rather than as independent constituents (Metzler, Haas, Cosic & Wheeler, in press). For instance, in *The cop probably could have killed the robber*, what is probable (but not certain) might be the *robber* (the cop certainly could have killed someone, probably the robber), or the killing, (the act certainly involved the robber but its outcome wasn't certain), or even parts of the auxiliary verb (e.g., emphasis on *could* as opposed to *should*). During the parse, adverbs are attached as low in the parse tree as possible. They are therefore given the widest possible scope, in the same way as prepositional phrases. Premodifying adverbs are attached to the verb immediately to their right. So, in the example sentence, *probably* is given scope over the entire verb phrase. This will match any query (however unlikely) concerning *probably killed* as well as *probably killed robbers*. Postmodifying adverbs are given scope back to (and including) the verb. Therefore, the parses of *The robber probably ran into the woods*, *The robber ran probably into the woods* and *The robber ran into the woods probably* all give the adverb the same scope, and will all match queries about the robber's probable destination (as well as the other potentially "probable" elements of the sentence).

3.1.3. Conjunction and Ellipsis

The problems posed by conjunction are similar to those of prepositional phrases and adverbs. In general, it is not possible to identify a unique scope and dependency interpretation for these structures. However, it is possible to determine whether the representation of the material in an abstract can be construed as matching a particular structure identified in a query, in other words, whether the two representations can *possibly* have the same interpretation. The ambiguity involved in conjunctions is handled in two ways. When it is possible, the alternative interpretations are collapsed into a single canonical representation. This is generally used when building conjunctions of three or more similar constituents to avoid a multitude of different "bracketings" of the structures. When this approach is not possible, we resort to multiple representations of the alternative structures. (Metzler, Haas, Cosic & Weise, 1988).

Conjunctions are specially marked in the scope and dependency representations so that a query may attempt to match any number of the conjuncts in the conjunction, or all of them. The right and left conjuncts are first parsed separately, receiving the appropriate dependency structures. For instance, for the conjunction *the cop who followed the robber and the dog whom the robber followed*, each relative clause is treated in its normal fashion. (See Section 4.) The two conjuncts have neither scope nor dominance over each other. The conjoined constituent itself is treated the same way as an unconjoined constituent for scope and dependency relations with other constituents. Structures that modify or embed the conjunction can be construed as relating to the entire conjoined constituent or to either of the constituents, unless a particular relationship is blocked (e.g., by number agreement or rules concerning determiners). For instance, *increasingly dangerous* in the phrase *increasingly dangerous felonies and accidents*, can be construed as modifying *accidents* as well as *felonies*, although this is not necessarily intended by the user of the phrase. (See figure 6.)

With sentences containing ellipsis, producing the correct scope and dependency is complicated by having to identify the constituent or piece of a constituent that has been omitted, and fitting it into its place in the sentence. For instance, in the sentence *The African bees attacked the herds and the locusts the crops*, the fact that it is the verb *attacked* that is missing from the right conjunct must be determined first. Then the correct scope and dependency structure, with *attacked* dominant over *the crops* and *the locusts* dominant over the entire verb phrase, can be built. While it would be possible to have independent rules to describe phrases containing ellipsis, rather than reconstructing such phrases from the other conjunct, we have not done so for two reasons. First, this would require a large set of special-purpose rules in order to recognize all the possible types of elliptical structures. Second, the reconstruction would still be necessary in order to match a complete query on an elliptical conjunct. In the preceding example, for instance, the query *the locusts attacked the crops* should match the right (incomplete) conjunct. This decision regarding the immediate interpretation of ellipsis further illustrates the relationship between the design of COP and its intended use for information retrieval since it is dependent on our knowledge of the use of the representations and our tolerance of imperfect analyses.

3.2. Over-acceptance

One of the general design decisions taken as a result of the constraints from the information retrieval domain is embodied in what we call the Minimal Specification Principle. We attempt to minimize the level of detail incorporated in the parser and the grammar by including no more than is necessary to make the distinctions required by our sorts of structural descriptions. We are aided in this regard by another point which follows from our objective. Since we are not concerned so much with defining syntax as with rendering a sensible structural description of anything that might be encountered by the parser, we are not very concerned with the fundamental linguistic objective of distinguishing precisely between the grammatical sentences of a language and all non-grammatical strings not accepted by that language. In other words, we can permit a degree of "over-acceptance" in our system. When a distinction is required only to make sure a non-grammatical string is rejected, (rather than to assure that a correct structure is built), we can often ignore that distinction. Unlike other natural language processing applications such as natural language interfaces or machine translation, the parser does not "fail" if it can not produce a unique and perfectly correct interpretation of an input. In effect, all that is required is that the structures produced by the parser and the matching procedures that utilize them produce a document ranking that is positively correlated with users' relevancy judgements. Obviously, the higher that the correlation is in any particular situation, the more useful is the system.

The parser over-accepts ungrammatical sentences in two ways. First, since the grammar uses only limited semantic information, the parser over-accepts syntactic structures that could be ruled out using semantics. For instance, the sentences *The cop saw the robber to arrest* and **The cop understood the robber to arrest* have the same sequence of lexical types. However, the first sentence is grammatical, while the second is, if not ungrammatical, at least infelicitous. The parser will produce scope and dependency structures for both sentences.

The second type of over-acceptance occurs because the parser does not contain many of the sorts of constraints which are used by modern syntactic theories such as Government and Binding Theory [GB] (van Riemsdijk & Williams, 1986), Generalized Phrase Structure Grammar [GPSG] (Gazdar, Klein, Pullum & Sag, 1985) and Lexical Functional Grammar [LFG] (Kaplan & Bresnan, 1982) to explain the ungrammaticality of expressions which COP is not likely to encounter in any case. (In fact, if COP did encounter such an expression, its task would be to interpret it sensiblely, rather than to reject it.) An example of these constraints is that on the extraction of objects from adjunct clauses in wh-questions. Many grammars have some way of rejecting sentences such as **What evidence did the cop arrest the robber after seeing*, where the fronted noun phrase is the object of the adjunct *after seeing*, not the direct object of the verb. (*Which robber did the cop arrest after seeing the evidence* is an acceptable sentence.) GB theory uses boundary constraints that prevent the adjunct noun from being extracted. GPSG uses restrictions on the occurrence of SLASH to block such sentences. The COP grammar does not have any such extraction constraints, and so will produce a parse structure for such sentences, if they are encountered.

Another type of distinction that must be made in theoretical grammars concerns differences between similar appearing surface structures. For instance, GB theory makes generalizations about the difference between the deep structures of sentences such as *John is easy to please* and *John is eager to please*. COP does not distinguish between these two structures so it is not particularly useful in extracting the relationship between the subject of these sentences and *pleasing* as a verb (i.e., who is doing the pleasing), but the trees do readily capture the relationships between the subjects and *easy* or *eager*, and the relationships between these two adjectives and *please* as an infinitive. Moreover, since dependency is not defined between the subjects and infinitive complements of these sentences, COP is neutral rather than wrong in describing these relations, and would not block the appropriate matches.

LFG addresses a similar issue by distinguishing between verb types in several ways, including the types of objects or complements that can follow the verb, and the types of generalizations that can be made about the verb (such as whether it can passivise). For example, *The mayor promised the party to follow* and *The mayor persuaded the party to follow* have the same sequence of constituents, but LFG gives these sentences analyses that differ in whether the subject or the object controls *to follow*. In the first sentence, the subject controls the complement, and the sentence can not be passivised (**The party was promised by the mayor to follow*). In the second sentence, the object controls the complement, and the passive version is grammatical (*The party was persuaded by the mayor to follow*) (Sells, 1985, 166). Because COP does not deal with the semantic distinctions between these two verbs, it does not treat them differently. Rather, it deals with this situation more as an attachment ambiguity, and produces two versions for each of these verbs, as well as for a sentence containing the same sequence of constituents with a different verb in which the infinitival phrase would clearly be modifying the object, as in *The mayor found the party to follow*. We do not use a canonical structure here, as we do with prepositional phrases and adverbs (see sections 3.1.1 and 3.1.2), in part because these structures do not have the same potential for combinatorial explosion.

3.3. Inexact Matching

The entire information retrieval process is in a sense probabilistic. The presence of key words in an abstract, or even index terms in a document record does not guarantee that a document will be relevant to a user's information need, but it does substantially raise the likelihood. Similarly, the structural matching which COP provides can be useful even when it is not based on completely certain information. This point is central to this project because there are many structures in natural language that are extremely difficult to analyze precisely or completely. In many cases this would require semantic and pragmatic analyses that are beyond present capabilities in artificial intelligence. In other cases it would require a great deal more specificity in the grammar. Whether such specificity could be achieved in a very general parser, and whether it would be worth the computational costs, are certainly problematic.

The general goal of our matching procedure is to determine whether it is possible (or how likely it is) that the terms of two linguistic expressions are used (or related) in similar ways, and hence, whether the two aggregate concepts are similar. We are currently exploring several ranking procedures to measure the degree of match, but they have in common one design principle. We accept false positive matches but avoid building structures or matching procedures that would produce false negative results. (This is a design heuristic, it is not strictly necessary.) We have already seen examples of inexact matching in the discussions of ambiguity. For instance, our scope and dependency structures will permit the matching procedure to find modification relations that may not have been intended by the original author, but they will never fail to find a relation that was intended. Thus, among a set of documents each containing a particular pair of terms, one of which can modify the other, it is possible to subdivide the set into (1) those for which there is evidence for the sought after relation, (2) those in which there is no evidence for the relation, and (3) those in which the terms appear in a relation other than the one which is sought.

Another example of such inexact structures and matching involves bitransitive verbs, such as *write*, that take both a direct and an indirect object. The verb phrase can appear in two forms, *write a proposal to NSF* and *write NSF a proposal* and it would be desirable to produce the same dependency for both forms. However, it is very difficult to distinguish a bitransitive verb in the first form from a transitive verb followed by one object and a prepositional phrase using only syntactic information; for example, *write a proposal to NSF* and *write a proposal at the university*. The first phrase can undergo "dative movement" to produce *write NSF a proposal*, while the second can not. The dependency trees for these two forms are shown in figure 7. The first form is that of either a bitransitive verb structure without dative movement or a transitive verb structure with a prepositional phrase, while the second shows a bitransitive structure with dative movement. In both forms both of the objects are dependent on the verb, and these relations can be accurately retrieved from the trees. In the first form, the second object is dependent on the first, whereas in the second form, the two objects are not in each others' scope. Thus there is a slight loss of information, or ability to match, between the two forms of bitransitive verb phrases, concerning the relationship between the two objects, but again, there is no case in which the appropriate relationship is contradicted. The relationship between the objects captured in the first form is the same as that which is appropriate for the prepositional phrase case, and so matching between these two linguistic expressions will be accurate.

The distinctions between verbs based on the types of complements they accept, (e.g., persuade vs promise), which was discussed in the previous section, also illustrates this point. The difference in this example is in the relationship of the nouns (*mayor* and *party*) with the complement *to follow*; with *persuade*, the object noun will do the following, with *promise*, the subject noun will follow. Since COP is not basing any of the structure building procedures on the semantics of verbs like *persuade* or *promise*, it is not useful in making distinctions concerning the relations between the subject and object nouns and the verb complement itself. The dependency tree will represent a dominance relation between the subject noun and the verb complement, which is normally a false positive for the verb *persuaded* in which the object noun is carrying out the action of the complement. Thus, if the query is looking for documents where the subject is doing the following, the abstract will be confirmed as discussing it. Usually there is no definitive relationship recorded between the object noun and the verb complement, so the document ranking will not be directly affected either way, if a match is sought for these terms. Note however, that in no case are the trees recording relationships that contradict the correct ones (false negatives).

4. Intra-sentence Anaphoric Reference

The issue of reference resolution is a difficult problem in natural language processing that has important repercussions for information retrieval, especially in systems dealing with full text. Natural language is full of anaphora, in which a phrase or pronoun refers to a previously mentioned entity. While the anaphoric term itself often contains some syntactic clues about its referent, it requires a great deal of information about semantics and discourse structure to identify it unambiguously enough for full text processing, including information retrieval. For example, in the sentence *He caught the robber*, syntactic information reveals that the pronoun *he* refers to a male, animate being (possibly a dog or a person). However, in the sentence *It was impossible*, it is very difficult to gather any clues about the referent of *it*; it is a vague pronoun that yields few clues about its referent. It could refer to a person, a place, an event or a situation.

There are a wide range of anaphoric reference problems, many of them concerning issues such as discourse structure and general world knowledge. COP does not deal with these complex issues, but it does deal with one important class of intra-sentence reference, that involving relative clauses. These constructions involve an augmentation of the basic binary tree representations which significantly extends the system's ability to capture the appropriate dependency relations between the words of complex sentences.

In phrases containing relative clauses, the relative pronoun inside the relative clause refers to something mentioned in the phrase embedding the relative clause. This structure causes complications in the structure building processes because the intra-clausal dependency relations may not correspond to the inter-clausal ones. For instance, in *the judge who was questioned by the press*, the *judge* is the head of the noun phrase and is dominant over the rest of the phrase, which includes *the press*. But inside the relative clause, *who*, which should refer to *the judge*, should be dominated by the *press* which is the agent of the passive clause. However, with only the relative pronoun *who* in the relative clause, there is no direct relationship at that level between *judge* and *press*. These problems; the lack of the intra-clausal relation's referent, and the reversal between the intra and inter-clausal relations, are dealt with by additional structure building procedures. During the parse, a pointer-placing procedure places a pointer from the head of the noun phrase, *judge*, to the relative pronoun, *who*. (See figure 8a.) In this structure *judge* is appropriately the head of the entire noun phrase, and is thus available to participate correctly in any surrounding relationships, but the correct relationship between *judge* and *press* inside the relative clause is also available for matching, via the pointer established between *judge* and *who*. When the dependency tree

structure is collapsed into the strict list structure, the relative pronoun is replaced by its antecedent to preserve this information.

Reduced relative clauses (relative clauses without a relative pronoun) are treated in a fashion analogous to the treatment of full relative clauses since the retrieval problem is similar. In a noun phrase such as *the judge questioned by the press, judge* is dominant over the entire noun phrase, but there is an additional problem in that within the relative clause, there is no relative pronoun for the head noun to be linked with and for *press* to have dominance over. So an additional mechanism is required which places a "trace" in the position which would have held the relative pronoun if one had been there, and links it to the head of the noun phrase, (as the pronoun would have been linked). During the collapsing process, the trace, like the pronoun in the full relative clause case, is replaced with the noun to which it is linked so that it is available for the matching procedure. (See figure 8b.)

5. Conclusions and Future Directions

This paper has described how syntactic information can be used to improve information retrieval performance, and the nature of the syntactic analyses we have been developing for this purpose. Our discussion in this paper, and much of our work on this project to date, has been rather abstract. We have concentrated on building the required parsing techniques, and even our (still rather limited) empirical studies of the system's utility have focused on the abstract question of whether the precision of a particular query can be improved with these techniques. But information retrieval is a dynamic process, and much of our future work will look at the role that these sorts of syntactic analyses and matching processes can have in this dynamic process. For instance, it is possible that in some cases the advantage provided by the ability to frame a query as a natural language expression might be found more in the time it takes to formulate a satisfactory query and/or in the time it takes one to become adept in using a retrieval system, than in differences in the effectiveness of the final queries themselves. In order to test these sorts of hypotheses we will have to integrate this syntactic filter with an existing information retrieval system so that syntactic filtering can be performed readily at each step of the query refinement process. Among the issues we will be looking at then are the following: (1) the ease of user of natural language queries for naive users; (2) the effects of syntactic analyses on the query formulation process and "browsing"; (3) the ability of users (naive and expert) to anticipate when this syntactic matching procedure will be useful; (4) the use of syntactic analyses to identify complex terms that should be considered as units by other sorts of analyses, particularly statistical information retrieval methods and automatic indexing methods.

6. Bibliography

Clark, H. & Clark, E. (1977). *Psychology and Language.* New York: Harcourt Brace Jovanovich, Inc.

Frazier, L. & Fodor, J. D. (1978). The sausage machine: A new two-stage parsing model. *Cognition, 6,* 291-325.

Gazdar, G., Klein, E., Pullum, G. & Sag, I. (1985). *Generalized Phrase Structure Grammar.* Oxford: Basil Blackwell.

Glass, A., Holyoak, K. & Santa, J. (1979). *Cognition.* Reading, Massachusetts: Addison-Wesley.

Jensen, K. & Binot, J. (1987). Disambiguating prepositional phrase attachments by using on-line dictionary definitions. *Computational Linguistics, 13,* 3-4, 251-260.

Jensen, K. & Heidorn, G. (1983). The fitted parse: 100% parsing capability in a syntactic grammar of English. *Proceedings of the Conference on Applied Natural Language Processing,* 93-98.

Kaplan, R. & Bresnan, J. (1982). Lexical-Functional Grammar: A formal system for grammatical representation. In J. Bresnan (Ed.). *The Mental Representation of Grammatical Relations.* Cambridge: The MIT Press.

Metzler, D., Haas, S., Cosic, C. & Weise, C. (1988). *Conjunction, Ellipsis and Other Discontinuous Constituents in the Constituent Object Parser* (Tech. Rept.). The University of Pittsburgh, Department of Information Science.

Metzler, D., Haas, S., Cosic, C. & Wheeler, L. (in press). Constituent Object Parsing for Information Retrieval and Similar Text Processing Problems. *Journal of the American Society for Information Science.*

Moulton, J. & Robinson, G. (1981). *The Organization of Language.* Cambridge: Cambridge University Press.

Salton, G. (1988). *On the Use of Syntactic Procedures for the Content Analysis of Natural Language Texts.* Paper presented at the American Society for Information Science Mid-Year Meeting, Detroit.

Salton, G. & McGill, M. (1983). *Introduction to Modern Information Retrieval.* New York: McGraw-Hill, Inc.

Sells, P. (1985) *Lectures on Contemporary Syntactic Theories.* Stanford: Center for the Study of Language and Information.

Sparck Jones, K. & Tait, J. (1984). Automatic search term variant generation. *Journal of Documentation, 40,* 1, 50-66.

van Riemsdijk, H. & Williams, E. (1986). *Introduction to the Theory of Grammar.* Cambridge: The MIT Press.

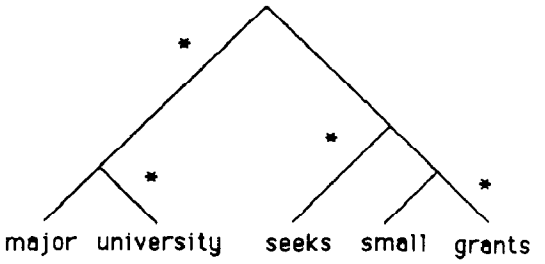


Figure 1: Dependency tree for ***Major university seeks small grants.*** "*" indicates the dominant branch at each node. ***University*** is transitively dominant over all other words in the sentence. ***Major*** and ***grants*** are unrelated.

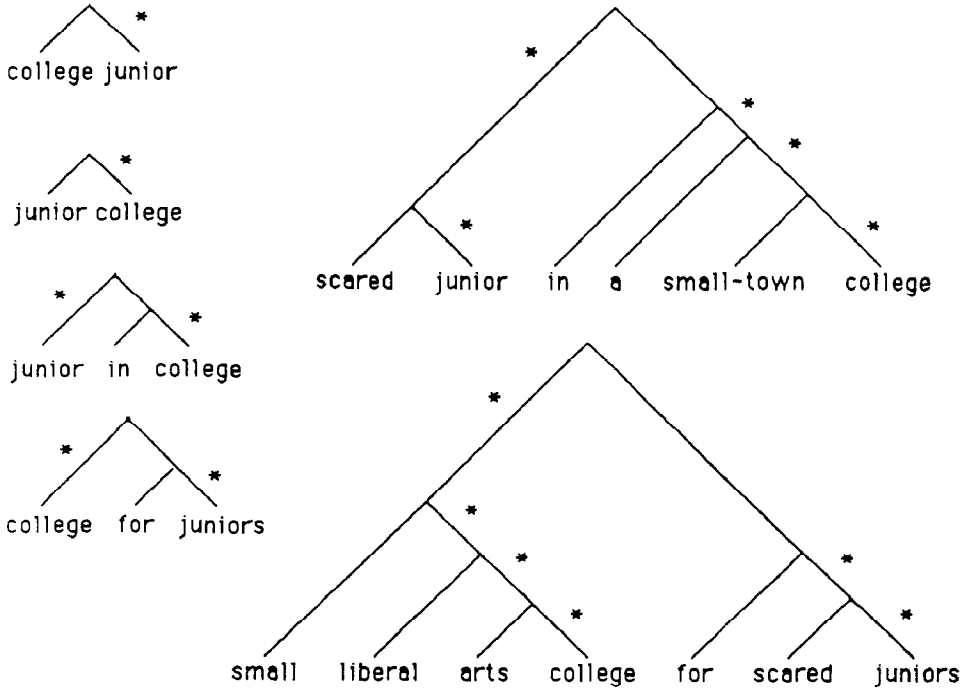


Figure 2: Noun phrase dependency trees.

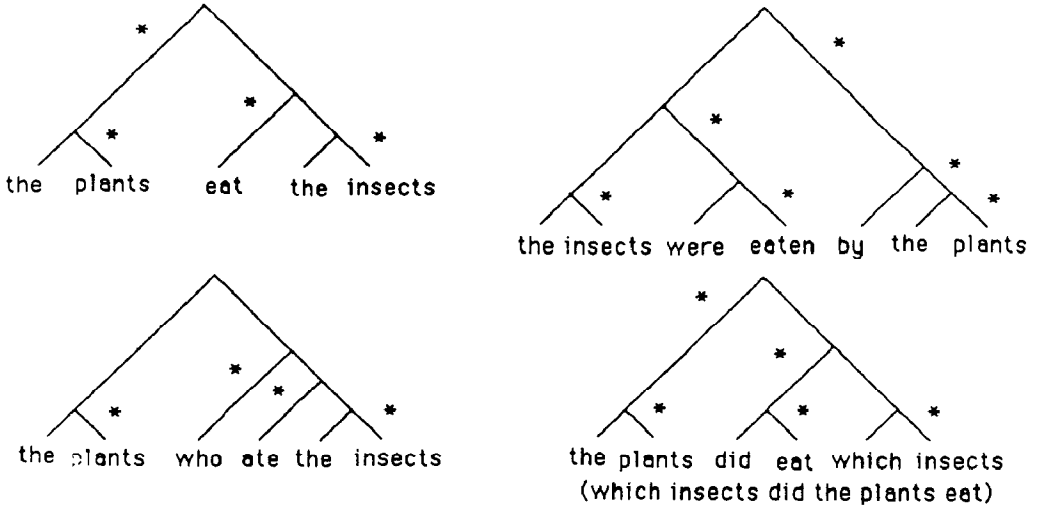


Figure 3: Sentence dependency trees.

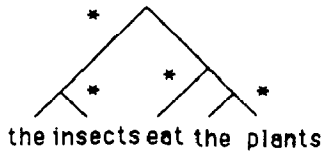
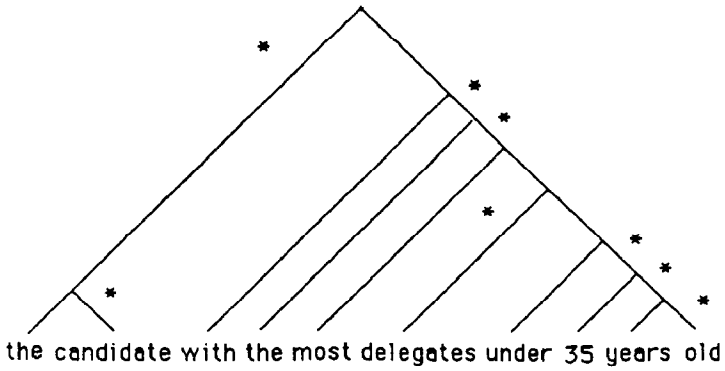


Figure 4: Dependency tree for *The insects eat the plants.*



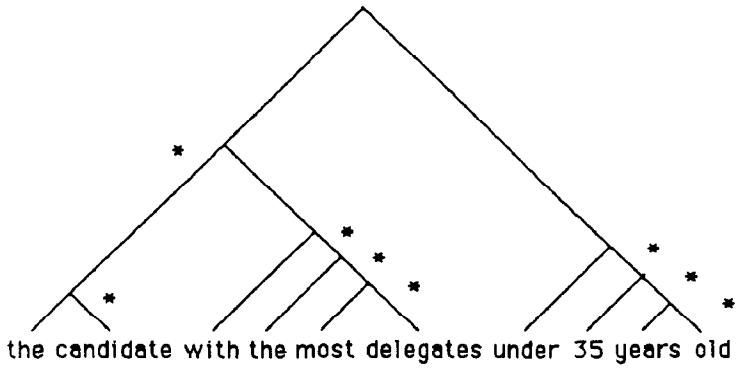
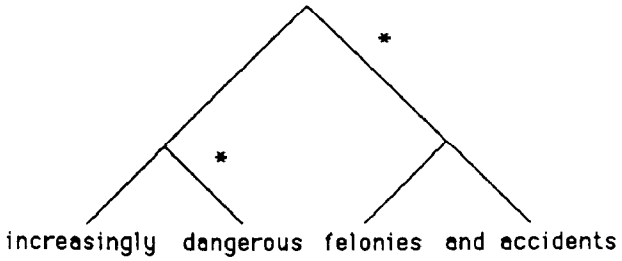


Figure 5: a. The prepositional phrase is attached to *delegates.* b. The prepositional phrase is attached to *candidate.*



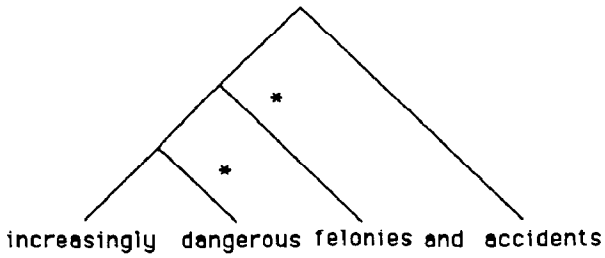


Figure 6: Two interpretations of *increasingly dangerous felonies and accidents.* The accidents may or may not be increasingly dangerous.

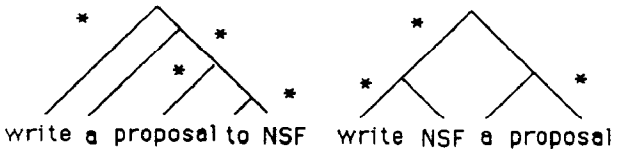
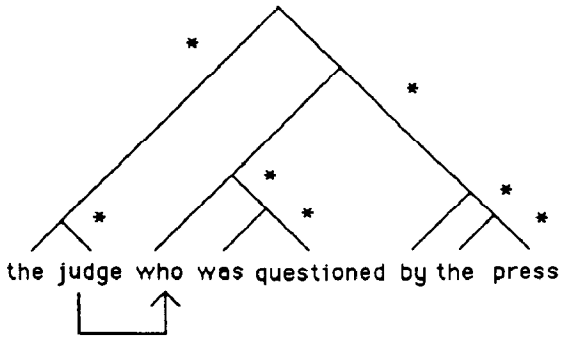


Figure 7: A bitransitive verb phrase without and with dative movement.



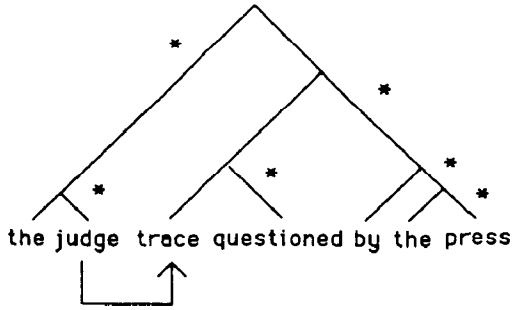


Figure 8: a. Dependency for *the judge who was questioned by the press,* showing the pointer from *judge* to *who.* b. Dependency tree for *the judge questioned by the press,* showing the pointer from *judge* to the trace.