Chapter 7

Natural Language Processing

0 PREVIEW

The documents and user queries under consideration in information retrieval are often available as natural language formulations. It is important therefore to be aware of the automatic methods currently used to process natural language texts. This chapter describes the state of the art in the automatic processing of natural language material with emphasis on applications in information retrieval.

The various levels of linguistic methods are examined first and the role of linguistic methods in information retrieval is described. This is followed by a general examination of modern language understanding systems. The components of language processing systems are then covered in detail with emphasis on the syntactic process which is of greatest interest in information retrieval. The main features of several grammatical models are described, including phrase structure grammars, transformational grammars, and augmented transition network grammars. This is followed by a discussion of applications of syntactic analysis in information retrieval.

The full scope of language understanding may not be needed in information retrieval. Language understanding is, however, an essential component in the design of question-answering systems. The chapter closes with a description of

linguistic methods useful for question answering. The structure of knowledge representation systems is covered and the language processing component of certain well-known experimental question-answering systems is described to provide an example of the potential of the currently available automatic language processing techniques.

1 COMPONENTS OF NATURAL LANGUAGE SYSTEMS

A Interest in Natural Language Processing

This chapter is devoted to a study of the problems which arise when natural language data are processed by computers. In particular, the main approaches to natural language processing are covered, and an attempt is made to provide a state-of-the-art view of different efforts in this area.

One may want to question the wisdom of examining linguistic procedures in an information retrieval context, particularly when the material in the earlier chapters makes it clear that linguistic methods play only a minor role in retrieval at the present time. The fact is however that a large part of the information stored in bibliographic retrieval systems consists of natural language data, and that many users would prefer, given the choice, to approach a retrieval system by using natural language formulations of their information needs. Furthermore, even if the currently usable language processing techniques appear inadequate for full utilization under operational retrieval conditions, there is always the hope that new developments may render the linguistic techniques more attractive in the future.

Precisely what does one expect to gain in using linguistic approaches in the retrieval context? The most immediate aim is surely the possible use of free language formulations by retrieval system users, both for the submission of initial query statements and for the various interactive processes in which queries or documents are adjusted based on information obtained from the user population. The use of natural language search statements could raise the efficiency as well as the effectiveness of the retrieval operations by making possible the formulations of precise requests that correctly reflect user needs and by simplifying the user-system interactions.

A second application of natural language processing is the use of complex analysis techniques for the content representation of the input documents. Indeed, when the analysis system is confined to the use of single words for the content description of queries and documents, a user query dealing with "computational complexity" and indexed by the terms "compute" and "complex" is just as likely to cover extraneous topics such as "computation with complex numbers" as the actual subject area of interest. Of course, phrases can be automatically assigned to documents and search requests by using term co-occurrence statistics and word adjacency operators; but the statistical techniques are imperfect. In particular, they do not distinguish between cases such as "blind Venetian" and "Venetian blind." However, if accurate linguistic techniques were usable to combine single terms into meaningful larger units, then com-

plete *structured index descriptions* might be generated consisting, for example, of noun-verb-noun combinations or of sentence units of even larger scope. Substantial improvements in text analysis and hence in retrieval should result.

Another important problem in retrieval is the construction of synonym dictionaries and thesauruses in which related words are grouped into affinity classes. Under current conditions thesauruses are constructed manually, or automatically by using word co-occurrence information. However, if linguistic descriptions were available to characterize the individual text units, a thesaurus class might be defined as the set of words occurring in similar contexts in the documents of a collection.

In addition to the free-language query submission and automatic indexing applications, a variety of extensions to the normal information retrieval process come easily to mind in which sophisticated language analysis techniques based on the use of deductive and contextual information obtainable from texts would play a major role. The following possibilities may be cited in this connection [1–5]:

1 Automatic question-answering systems might be designed where the system is expected to give explicit answers to incoming search requests ("What is the boiling point of water?" Answer: 100 degrees Celsius), as opposed to merely furnishing bibliographic references expected to contain the answer.

2 Automatic abstracting systems could become practical where the document texts are automatically reduced to abstract-length excerpts that inform the reader about the content of the corresponding documents.

3 Foreign language documents could be treated automatically making it possible, for example, to extend the automatic indexing techniques to include documents originally available in languages other than English.

4 Sophisticated treatment of the full text of natural language documents could be considered as required, for example, for text analysis and text concordance generation.

Because of the difficulties that are inherent in a complete linguistic analysis of natural language texts, many of these problems are currently approached by creating a simplified situation—for example, by restricting the allowable discourse area to a narrow topic slice, or by imposing limitations on the variety of natural language forms that are actually handled by the system. These restrictions might be given up in the future, assuming sufficient gains in understanding the natural language phenomena.

B Levels of Language Processing

It is customary to recognize several different levels of language processing. These may be characterized as the phonological, morphological, lexical, syntactic, semantic, and pragmatic levels.

The *phonological* level deals with the treatment of speech sounds as needed, for example, for the handling of speech understanding or speech gen-

eration systems. This level of linguistic processing is not of immediate interest in the retrieval context and is not discussed further in this volume.

The *morphological* level of linguistic processing is concerned with the processing of individual word forms and of recognizable portions of words. The recognition and removal of word suffixes and prefixes and the generation of word stems (used earlier to enhance search recall) are based on morphological knowledge.

The *lexical* level deals with the procedures operating on full words. In information retrieval this covers operations such as common word deletion, dictionary processing of individual words, and the replacement of words by thesaurus classes. In syntactic parsing applications where an attempt is made to obtain a structural description of a sentence, a preliminary lexical operation normally identifies a set of linguistic features (for example, noun, adjective, preposition, etc.) for each text word to be used later in the main syntactic analysis process.

The *syntactic* level is designed to group the words of a sentence into structural units such as prepositional phrases, and subject-verb-object groupings that collectively represent the grammatical structure of the sentence. A syntactic analysis is normally based on the surrounding structure in which the individual words are embedded in a sentence and on the use of syntactic features characterizing the individual words. Most currently available syntactic analysis systems are sufficiently advanced to permit the recognition of the principal structural characteristics of English text.

The *semantic* level adds contextual knowledge to the purely syntactic process in order to restructure the text into units that represent the actual meaning of a text. Thus a syntactic analysis for the utterance "John is easy to please" would designate "John" as the subject of the sentence, whereas the semantic process would designate "John" as the complement, as indicated by the semantic interpretation, "It is easy for someone (unnamed) to please John." A variety of such preestablished semantic characterizations of the terms are normally used to obtain satisfactory semantic interpretations.

Finally, the *pragmatic* level uses additional information about the social environment in which a given document exists, about the relationships that normally prevail in the world between various entities, and about the world-at-large to help in the text interpretation. For example, knowledge of the size of pigs and pens, respectively, and of the normal habitat of pigs permits an unambiguous interpretation of the sentence, "The pig is in the pen."

The morphological and lexical operations were examined in Chapter 3 as part of the automatic indexing operations. The principal interest in the present discussion is then confined to the syntactic, semantic, and pragmatic levels of linguistic processing.

In dealing with the various linguistic procedures, one must consider the degree to which the individual levels are easily recognized and independent of each other. Unhappily this question, like so many others in language processing, is controversial. All knowledgeable observers agree that automatic

language processing raises complicated issues, that the use of the context in which individual words occur is essential for automatic text interpretation, and that syntax and semantics are related in that semantic knowledge is needed to avoid ambiguities in the syntax, whereas syntactic information helps in producing useful semantic output [6].

Unfortunately, beyond the general realization that the language analysis task is difficult, there is little agreement about how to handle the job. It is unclear which levels of language processing are most important and how the corresponding techniques are best applied. One school of thought points to the fact that human processes are highly integrated and concludes that a language analysis system must necessarily be based on the global use of many different approaches at each point in the process:

> Understanding is a completely integrated process; the idea of building modular systems (where, for example, the syntactic phase is isolated from the semantic) has hampered advances in parsing, because the full range of our knowledge should obviously be available to help disambiguate, find appropriate word senses, and just as importantly help us know what to ignore [7].

This viewpoint is supported by the fact that in some of the existing question-answering systems dealing with restricted topic areas, a sophisticated syntactic analysis system proves unnecessary, since the few semantic patterns accepted by these systems are completely understood and procedures to handle these patterns can be provided in advance.

Many other observers prefer the modular viewpoint of language analysis. This uses a large dictionary storing linguistic features for the words in the language, and a grammar designed to produce a rigorous analysis of the possible sentences in the language. A semantic analysis system is then added to transform the syntactic output into formalized units of meaning. Most people still feel that a syntactic analysis system is essential when unrestricted natural language input is processed [8]. Since the syntax alone can be helpful in certain retrieval tasks, whereas the semantic processing is not as yet well understood, a good deal of emphasis is placed on syntax in the present chapter.

C Language Understanding Systems

The earliest uses of computers for language processing date back to the 1950s when programs were written that could translate short excerpts of text from one language to another—often from Russian or German into English [9,10]. Substantial effort was invested in the construction of mechanized dictionaries useful for the translation of large unrestricted texts. But when these dictionaries were actually put to the test by using them on large samples of input, it turned out that the translation task became too difficult. New programs had to be written to take care of problems that had not been met in considering the earlier text samples, and the new modifications interfered with the earlier programs which were originally thought to be adequate. When it proved impossi-

ble to reach a steady-state condition after some 10 or 15 years of work, the machine translation work was generally given up in favor of more fundamental work in language analysis and understanding [11].

Thus small text samples were intensively studied in an attempt to determine all the information contained in or derivable from the sample, and systems were built that could automatically answer direct questions dealing with the specific and restricted topic areas covered by particular text samples. Unfortunately, the lessons learned earlier through the machine translation work had to be learned all over again: while it was comparatively simple to deal with 250-word text samples and 300-word dictionaries, the extensions to larger portions of the language were neither assured nor forthcoming even after substantial effort.

While the general understanding of how human beings analyze language has been increased by recent psychological experiments, it has not been possible so far to translate this understanding into automatic programs that can satisfactorily process large samples of unrestricted natural language texts [12].

The early approaches to language analysis tended to be word-oriented. Each word was assumed to represent one or more well-defined units of meaning, and when words were combined the available grammars more often than not produced a number of acceptable output analyses. For example, given standard dictionary entries for the words "time" and "flies" as

Time: singular noun or transitive verb
Flies: plural noun or transitive verb or intransitive verb

many standard grammatical analysis systems would produce at least two acceptable results for the sentence, "Time flies":

1 A declarative statement in which "time" is interpreted as a noun subject of the sentence and "flies" is the intransitive verb
2 An imperative sentence where "time" is interpreted as a transitive verb and "flies" is the plural noun complement

The first interpretation is of course the one normally assumed whereas the latter must be termed semantically improbable because flies are not the type of insects that are normally timed. For longer sentences, such as for example the well-known "Time flies like an arrow," the number of acceptable syntactic outputs could be much larger than two.

Refinements were introduced into the early syntactic analysis systems in the form of transformations defined on the standard syntactic output. Thus given an accepted syntactic interpretation for the sentence "John hit the ball" another acceptable analysis would be produced for the passive form "The ball was hit by John." Formally defined transformations could then account for equivalences between active and passive moods, and between declarative and the corresponding interrogative forms of a sentence.

Even though the complexity of the syntactic analysis systems grew rapidly, it became clear that many issues of syntactic "well-formed-ness" were not explainable by syntax alone, and that a sentence-by-sentence analysis based purely on syntax was not sufficient to analyze many texts. For example, it is not possible using simple syntactic considerations to explain the fact that the sentence "John and his sister went to Paris" is acceptable, whereas "John's sister and he went to Paris" is not.

Clearly some semantic features must be incorporated in the syntactic process before a sentence can finally be interpreted. For example, by recognizing semantic usages for ambiguous words such as "ball," and taking into account the context in which a given word occurs, it becomes possible to produce correct interpretations for sentences such as "He played with the ball" and "He went to the ball" [13].

One direct link between syntax and semantics is provided by the introduction of *case grammar* [14]. In case grammar a sentence is considered as a tenseless *proposition* consisting of a verb and related noun phrases and embedded sentences, plus a *modularity* specifying tense, mood, and aspect. By using a dictionary to store partial semantic characterizations, or *cases*, for each noun, and then classifying the verbs according to which cases could be related to them, one could then specify the permissible patterns in the language by using a finite set of relations between verbs and nouns (also known as *case frames*). Typical cases for nouns are agent (A), object (O), instrument (I), indirect object (D), and so on. Typical case frames for the word "write" could be

(A)—with (I)
(A)—(O)
(A)—to (D)
(A)— (O) to (D) with (I)

as in "The author writes with a pen," "The author writes a letter," "The author writes to his publisher," "The author writes a letter to his publisher with a pen." It was found experimentally that a relatively small number of case frames could account for a vast number of different sentences in the language. The case frames could also effectively constrain the accepted utterances to include meaningful sentences only.

In recent years, most of the interest in computational linguistics has been devoted to the integration of various linguistic approaches into complete language processing systems. A complete language processing system of the kind useful in information retrieval can then be viewed as a three-part structure [1]:

1 First a standardized, formal representation is constructed of the meaning of each sentence or unit of discourse; typically units of meaning are extracted for each component from a dictionary, and the components are then assembled into a formal sentence representation using the constraints imposed by the syntax.

2 This formally restricted input is then compared with a stored *knowledge base* in order to augment the initial descriptions and to identify additional relationships between entities.

3 Finally, a desired task is performed which uses the combined information provided by the available input augmented by the stored knowledge; for example, inference procedures may be used to derive and formulate answers to requests for information.

A feeling exists that knowledge may be organized around *conceptual entities* such as objects, relations, events, and scenes, with associated descriptions and procedures. Given such conceptual structures the reasoning process—for example, the process needed to generate answers in response to incoming queries—consists in a matching and recognition process. The objects and events provided at the input (for example, the incoming user queries) are compared against stored knowledge structures. Then special "reasoning strategies" including inferencing capabilities are used to derive a desired output. Many people feel that the information included in the stored knowledge structures should be clustered so that similar objects may appear in close proximity. In this way related elements can share common properties and the comparison and search process in the knowledge base may be simplified [15].

Beyond these general perceptions about knowledge representation, there is no agreement about what structures are actually needed and how they are to be represented in an operational system. One unsolved problem concerns the usefulness of representing knowledge by a small number of *primitive* concepts. If useful semantic primitives could be identified, the construction and manipulation of the knowledge base would be substantially simplified. All processes could then be reduced to the manipulation of only those primitive entities for which specific rules would be established.

In the so-called conceptual dependency model only six conceptual categories are recognized, including real world objects, real world actions, attributes of objects, attributes of actions, times, and locations. These categories are related in only 16 different ways [16]. In situations where the number of primitive concepts is small, the reasoning and inferencing processes are relatively simple. Since only a small number of entities are available in each category, the possible events that may occur can be established and stored in advance. For example, five "acts" are allowed in conceptual dependency describing physical actions that people can perform (termed "propel," "move," "ingest," "expel," and "grasp"). For each act, a small set of inferences is defined as true with varying degrees of certainty when the particular act occurs.

Many people, however, do not believe that it is possible or realistic to represent knowledge using semantic primitives. Instead it is suggested that human beings use a variety of redundant descriptions to represent each object, and that the particular description used in a given instance depends on circumstances. The holistic view of knowledge representation is based on a redundant characterization of each object consisting in part of comparisons with other ob-

jects. When enough comparisons with other objects are stored, a given entity is then assumed to be completely described. A table might, for example, be characterized in part by saying that it is more similar to a chair than to a lamp. As more information is obtained about an object, the new data are simply added to the existing descriptions.

Regardless of one's views about the primitive components in a knowledge base, it becomes necessary to make a choice about the structures actually used to represent knowledge and the types of knowledge to be represented by them. Two principal structural representations are described in the literature. The first is the semantic graph or *semantic net* [17]. Such a graph consists of nodes and of branches, or links connecting certain pairs of nodes. In a semantic graph the nodes are used to represent the main concepts of interest, including objects, events, assertions, actions, abstractions, and functions, as well as attributes and characteristics of objects. Links between nodes identify hierarchical and contextual relations between the concepts, as well as other special information relating to the concepts. The nodes of a particular semantic net might, for example, represent the individual members of a given family, and the links between pairs of nodes could identify the relationship between certain members of the family, such as husband-wife or father-son. Other nodes representing properties of certain persons, such as age, profession, or marital status, could be linked to the nodes representing the corresponding persons. The hierarchical arrangement of topic classes contained in a conventional library classification system represents a form of semantic net in which the relationships between concepts is restricted to hierarchical inclusion.

The other type of knowledge structure that is much discussed is variously known as a "frame," "script," or "schema" [6,18,19]. A frame is a complicated data structure, typically representing a complex object such as a living room, a birthday party, or a restaurant. In particular, the frame would include the collection of knowledge associated with the concept being represented, including both the information that is always true of a given environment—for example, the fact that a restaurant contains tables and chairs, and customers eating food—and also information that may be true only in particular instances —for example, that a certain person would be located in a certain restaurant. A frame may be viewed as an extended notion of *case*, in the sense that the frame circumscribes an area of discourse by defining a context within which certain things are allowed to happen.

A script is a frame that also contains the timing sequence in which events are expected to happen. Certain portions of the script could store events that might be expected to happen—for example, that after having eaten in a restaurant, the customer is expected to pay the bill. In addition prescriptions of what would happen if the expectations were not confirmed could also be stored—for example, the fact that a customer who refused to pay the bill would be asked by the restaurant owner to wash dishes.

Given a stored collection of frames or scripts, the process of understanding a new statement implies that the new information must fit with the collection of

knowledge already stored. In particular, the interpretation of a sentence would depend on the contents of certain frames, and might also lead to the modification of certain frames in accordance with the new available knowledge. Since the frames specify sequences of events, a comparison of a particular event with the knowledge stored in the frames makes it possible to draw inferences. Thus, the inference might be that the customer who does not pay the bill at the restaurant forgot a wallet or lost it, or that it was stolen.

At the present time, the specific structure of a frame is not completely understood, and no established methods exist for searching collections of frames, updating the stored information, and drawing appropriate inferences. Nor is there any agreement about what stored knowledge base is actually needed and how this store should be represented. There is, however, agreement that a complete language processing system must be based not only on linguistic techniques (such as lexical analysis, parsing, and the application of semantic rules) but also on a large variety of stored knowledge about the particular topic and the situation being discussed. Further the generally available "common sense" know-how about time, events, states, actions, motivations, and beliefs must be available. A complete automatic language understanding system will not become practical until satisfactory solutions are provided for the construction and manipulation of the required knowledge structures.

Fortunately, a good deal can be accomplished in various applications areas, including information retrieval in particular, without totally mastering the language understanding problem. The remainder of this chapter is concerned with language processing methods that might actually be used in various practical situations.

2 LANGUAGE PROCESSING AND INFORMATION RETRIEVAL

Before describing the language processing methods to be used in information retrieval, it is important to point out that a variety of views have been expressed concerning the importance of linguistic methods in information retrieval. On the one hand, some individuals are convinced that to retrieve items "about" certain subjects, it is necessary to use all available facts pertaining to these items. This operation necessarily requires an analysis of meaning which is not substantially different in information retrieval from other areas of language understanding. In particular, a desirable indexing, or content analysis, approach would then consist of translating the document or query into some formal language consisting of concepts and relationships between the concepts. This introduces the notion of a semantic network and of translations from one language (the input) to another (the formalized index descriptions). Necessarily then, it is argued the full power of language processing tools is needed in information retrieval [20].

To support this view, one can point to the growing use of semantic tools in analyzing linguistic structures—semantic markers are increasingly included as entries in dictionaries and thesauruses, and relationship indicators specified in

semantic nets are usable to form phrases and to disambiguate terms. Even when complete networks with semantic relationships are not available, some syntactic tools such as the previously mentioned case grammar utilize word identifiers (agent, object, instrument) that have semantic connotations.

The opposite view about the importance of language analysis in retrieval comes to very different conclusions. In particular, a good deal of evidence points to the importance and usefulness of statistical, probabilistic, or vector space techniques of the kind discussed in Chapters 3 and 4 for both indexing and classification. On the other hand, no comparable evidence now shows that linguistic methods are effective in retrieval. The reason may be that a fundamental difference exists between information retrieval on the one hand and certain other language processing tasks on the other. In retrieval one needs to render a document retrievable, rather than to convey the exact meaning of the text. Thus, two items dealing with the same subject matter but coming to different conclusions are treated identically in retrieval, that is, either they are both retrieved or they are both rejected. In a question-answering or language translation situation, these documents would of course be treated differently. This amounts to a qualitative difference between document retrieval on the one hand and question-answering or language translation systems on the other. For example, to answer a specific question about an apple it is helpful to have some detailed knowledge about apples. To retrieve documents about apples, it may be unnecessary to understand precisely what the concept of apple actually entails. Instead, it may be sufficient to detect rough similarities between documents and concepts—for example, it might be enough to know that an apple is more similar to a pear than to an elephant. The notion of vector matching and vector similarity computation was used extensively in the earlier chapters for classification and retrieval purposes.

This view of information retrieval rejects the notion that information retrieval is simply an early stage of more refined question answering [21]. Question answering makes it necessary to understand the topic area in order to permit the generation of inferences leading to specific answers. Information retrieval, on the other hand, simply provides references to the users designed to fill an information need. To quote from Sparck Jones:

> The whole idea of meaning representation is dubiously relevant to document retrieval in anything like its present form . . . one can get quite good (retrieval) results with simple terms and weights [22].

This is not to say that some well-established linguistic procedures could not lead to improvements in retrieval effectiveness. This idea is further explored in the next few sections.

3 SYNTACTIC ANALYSIS SYSTEMS

Among the many kinds of syntactic analysis systems, three stand out as especially important: the phrase structure grammars, which are believed to model

many of the basic structural properties of the language elements, the transformational grammars, which account for syntactically distinct representatons of some semantically equivalent fragments, and finally the transition network grammars, which are most often used in modern automatic language processing systems. These three syntactic models are briefly presented in this section.

*A Phrase Structure Grammars

Many observers feel that a purely syntactic analysis of the language does not do very much for people interested in language processing. In this view syntax is an adjunct to a more complete analysis process. However, syntax is better understood than most other linguistic procedures, and it is actually used in some retrieval systems. Linguists are noted for their work on *generative* grammars. These are grammars designed to generate grammatically correct sentences. One of the main properties of a useful grammar is *simplicity*, in the sense that a small grammar should account for the generation or the analysis of a large number of sentences. A particular type of grammar, known as "phrase structure" grammar, is simple and is usable both for sentence generation and for sentence analysis. This type of grammar is therefore examined first.†

Consider "rewrite rules" of the form

$$S \rightarrow A + B \tag{1}$$

which stands for "the variable S can be rewritten as A followed by B." The symbol "+" separates the variables. When presenting rewrite rules, capital letters denote *nonterminal* elements, that is, elements that can be rewritten further, by appearing on the left side of some rewrite rule.

The rewrite rule

$$S \rightarrow \text{john} + \text{ran} \tag{2}$$

says that the symbol S can be rewritten as "john" followed by "ran." S denotes the *sentence* symbol. The symbols "john" and "ran" are both *terminal* elements (that cannot be rewritten further) as indicated by the lowercase letters. Thus, expression (2) generates the sentence "john ran."

In principle, it is possible to use one rewrite rule for each sentence to be generated. Thus one could write

$$S \rightarrow \text{john} + \text{ran} \,|\, \text{sally} + \text{jumped} \,|\, \ldots \tag{3}$$

where the bar | stands for the logical connective OR. The result would be a huge grammar which would not be parsimonious or simple. Consider the sentences

1. the man hit the ball
2. the man hit the man

† The examples used in the sequel are taken from lecture notes prepared by Justin Fisher.

3. the ball hit the ball
4. the ball hit the man
5. the man took the ball
6. the man took the man
7. the ball took the ball
8. the ball took the man

ignoring for the moment the inherent semantic anomalies. The eight sample sentences could in principle be generated by eight rewrite rules of the form S → sentence. If the word "rock" were added to the vocabulary, 10 additional sentences could be produced:

9. the man hit the rock
10. the ball hit the rock
11. the rock hit the man
12. the rock hit the ball
13. the rock hit the rock
14. the man took the rock
15. the ball took the rock
16. the rock took the man
17. the rock took the ball
18. the rock took the rock

Suppose now that the following grammar had been used for the original eight sentences

$$\begin{aligned} S &\rightarrow NP + VP \\ NP &\rightarrow T + N \\ T &\rightarrow \text{the} \\ N &\rightarrow \text{man} \mid \text{ball} \\ VP &\rightarrow V + NP \\ V &\rightarrow \text{hit} \mid \text{took} \end{aligned} \tag{4}$$

where NP stands for "noun phrase," VP represents "verb phrase," and T, V, and N are symbols, respectively, for "article," "verb," and "noun." When the grammar (4) is used for sentence generation, the addition of the word "rock" requires only one new rule, namely

$$N \rightarrow \text{rock}$$

instead of the 10 additional rules in the direct system of sentence generation.

It turns out that it is useful to represent the sentence generation in tree

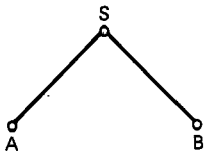


Figure 7-1 Rewrite rule S → A + B.

form. The simple rewrite rule (1) would appear in tree form as shown in Fig. 7-1. In the direct sentence generation system, the *derivation tree* for sentence 1 would then appear as shown in Fig. 7-2. On the other hand, when the grammar (4) is used to generate sentence 1, the derivation tree appears as in Fig. 7-3. The tree of Fig. 7-3 is more complicated than that of Fig. 7-2, but it tells something about the structure of the sentence. In particular, Fig. 7-3 exhibits the *constituent phrases* of the sentence—for example, the fact that "the" and "man" go together to form the noun phrase "the man." For this reason a tree of the type shown in Fig. 7-3 is known as a "phrase structure tree," or "phrase marker," and a grammar such as (4) is known as a "phrase structure grammar" [23–25]

A phrase structure grammar is used to exhibit the constituent phrases of the sentences. In the case of sentence 1 the constituent structure is represented as

((the man) (hit(the ball))) (5)

which corresponds quite well to one's intuition about this sentence. That is, a segmentation such as

((the) (man hit the) (ball))

would be curious because the initial "the" is obviously associated with "man," etc. A good grammar should then produce a constituent structure which conforms to linguistic intuition.

A grammar such as that of expression (4) is known as a "context-free" phrase structure grammar because the nonterminal symbols of the left-hand side of the rewrite rules can be replaced by the right sides of the rules regardless of the context in which these symbols may appear. That is, no contextual restrictions apply to any rewrite rule.

Consider now the sentences.

19. john phoned mary
20. john phoned up mary
21. john phoned mary up

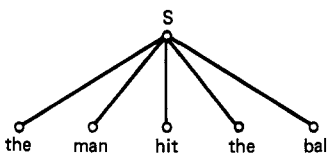


Figure 7-2 Direct sentence generation.

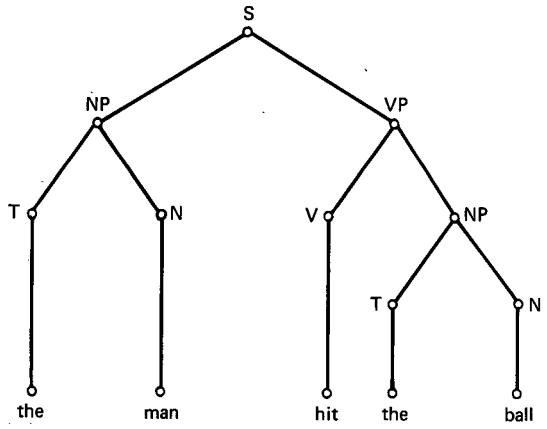


Figure 7-3 Phrase structure sentence generation.

The word "up" is dependent on "phoned," and "phoned up" intuitively is a constituent phrase. The first two sentences, 19 and 20, are easily handled by the context-free grammar:

$$\begin{aligned} S &\rightarrow NP + VP \\ NP &\rightarrow \text{john} \mid \text{mary} \\ VP &\rightarrow V + NP \\ V &\rightarrow \text{phoned} \mid \text{phoned} + \text{up} \end{aligned} \tag{6}$$

But to handle sentence 21, one needs additional rules such as

$$\begin{aligned} VP &\rightarrow V' + NP + PART \\ V' &\rightarrow \text{phoned} \\ PART &\rightarrow \text{up} \end{aligned} \tag{7}$$

Use of (7) produces the phrase marker of Fig. 7-4, corresponding to the consti-

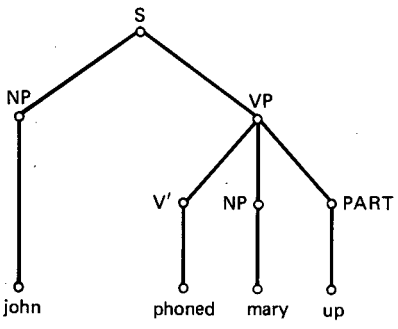


Figure 7-4 Phrase marker for "John phoned Mary up."

tuent structure (john(phoned mary up)). Intuitively "phoned mary up" is a constituent, but a context-free phrase structure grammar is unable to exhibit the phrase. This shows that context-free grammars do not reflect the full complexity of the language. In this case, they cannot handle *discontinuous constituents* such as "up" in "phoned up."

Another problem is due to the agreement expected in the number of subject and verb. Consider the following grammar:

$$
\begin{aligned}
S &\rightarrow NP + VP \\
NP &\rightarrow T + N \\
T &\rightarrow \text{the} \\
N &\rightarrow \text{man} \mid \text{men} \mid \text{ball} \mid \text{balls} \\
VP &\rightarrow V + NP \\
V &\rightarrow \text{have} \mid \text{has}
\end{aligned}
\tag{8}
$$

This generates sentences such as "the man has the ball" or "the men have the ball." Unfortunately, the grammar also generates "the man have the ball," "the men has the ball," and so on. This problem can be fixed by making distinctions between singular and plural noun phrases, labeled NP_s and NP_p, respectively, and between singular and plural verb phrases and verbs, denoted VP_s, V_s, VP_p, and V_p.

A new grammar can now be generated that will ensure agreement in number between subject and verb:

$$
\begin{aligned}
S &\rightarrow NP_s + VP_s \mid NP_p + VP_p \\
NP_s &\rightarrow T + N_s \\
T &\rightarrow \text{the} \\
N_s &\rightarrow \text{man} \mid \text{ball} \\
NP_p &\rightarrow T + N_p \\
N_p &\rightarrow \text{men} \mid \text{balls} \\
VP_s &\rightarrow V_s + NP_s \mid V_s + NP_p \\
V_s &\rightarrow \text{has} \\
VP_p &\rightarrow V_p + NP_s \mid V_p + NP_p \\
V_p &\rightarrow \text{have}
\end{aligned}
\tag{9}
$$

The new grammar (9) can now distinguish between the two trees of Fig. 7-5.

Problems such as the discontinuous constituents and the subject-verb agreement led to the development of more powerful grammars than simple phrase structure systems, including in particular the transformational grammars.

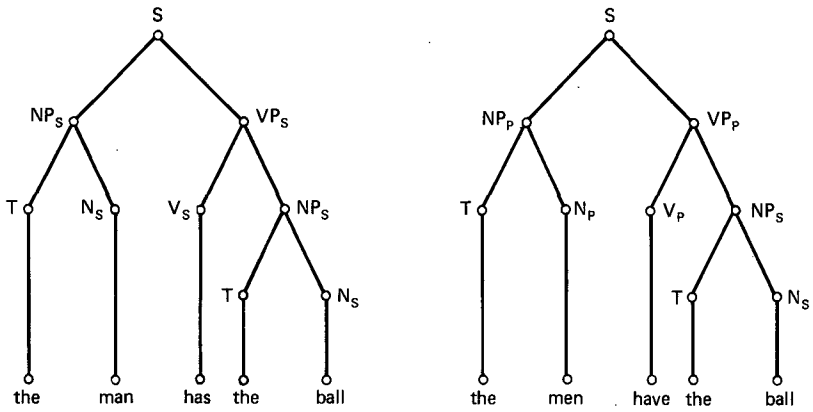


Figure 7-5 Phrase markers illustrating subject-verb agreement.

B Transformational Grammars

The basic innovation in the transformational grammars is the introduction of *context-sensitive* rewrite rules of the type

$$w\ A\ x \rightarrow w\ \gamma\ x \tag{10}$$

where A is a nonterminal variable of the grammar and γ is a string of terminal or nonterminal characters. The rule specifies that when the variable A appears in the context w and x (that is, is preceded by w and followed by x, where either w or x might be unspecified), then A can be replaced by the string γ.

A context-sensitive rewrite rule could be added to grammar (6) to generate sentence 21 as follows:

$$\text{phoned} + \text{up} + \text{NP} \rightarrow \text{phoned} + \text{NP} + \text{up} \tag{11}$$

leading to the following progression of transformations for the generation of sentence 21:

$$\begin{aligned} \text{S} &\rightarrow \text{NP} + \text{VP} \\ &\rightarrow \text{john} + \text{VP} \\ &\rightarrow \text{john} + \text{V} + \text{NP} \\ &\rightarrow \text{john} + \text{phoned} + \text{up} + \text{NP} \\ &\rightarrow \text{john} + \text{phoned} + \text{NP} + \text{up} \\ &\rightarrow \text{john} + \text{phoned} + \text{mary} + \text{up} \end{aligned} \tag{12}$$

Subject-verb agreement can similarly be handled by introducing contextual symbols sing and pl, standing for singular and plural, and rewrite rules such as

$$
\begin{aligned}
\text{N} &\rightarrow \underline{\text{man}} \mid \underline{\text{ball}} \\
\text{V} &\rightarrow \underline{\text{have}} \\
\underline{\text{sing}} + \underline{\text{have}} &\rightarrow \text{has} \\
\underline{\text{pl}} + \underline{\text{have}} &\rightarrow \text{have} \\
\underline{\text{man}} + \underline{\text{sing}} &\rightarrow \text{man} \\
\underline{\text{man}} + \underline{\text{pl}} &\rightarrow \text{men} \\
\underline{\text{ball}} + \underline{\text{sing}} &\rightarrow \text{ball} \\
\underline{\text{ball}} + \underline{\text{pl}} &\rightarrow \text{balls}
\end{aligned}
\tag{13}
$$

where the underline under man, ball, have indicates that these are nonterminal symbols covering varying forms of the corresponding terms.

A more convincing argument for the need for transformational grammars can perhaps be made by giving examples of related sentences that intuitively ought to be handled by a practical grammar:

22. Chomsky proved the theorem
23. the theorem was proved by Chomsky
24. Chomsky did not prove the theorem
25. did Chomsky prove the theorem?
26. was the theorem proved by Chomsky?
27. the theorem was not proved by Chomsky
28. did not Chomsky prove the theorem?
29. was not the theorem proved by Chomsky?

It is obvious that all these sentences can be generated from the first one by suitable sequences of transformations, including in particular active-passive, positive-negative, and declarative-interrogative transformations. A feature list for the eight sentences is shown in Table 7-1.

Assuming that sentence 22 can be generated by a rule such as

$$\text{S} \rightarrow \text{NP}_1 + \text{V} + \text{ed} + \text{NP}_2$$

where NP_1 and NP_2 denote specific instances of particular noun phrases, it is easy to devise context-sensitive transformation rules that will generate the transformed sentences:

Active-passive:

$$\text{NP}_1 + \text{V} + \text{ed} + \text{NP}_2 \rightarrow \text{NP}_2 + \text{was} + \text{V} + \text{ed} + \text{by} + \text{NP}_1$$

Positive-negative:

$$\text{NP}_1 + \text{V} + \text{ed} + \text{NP}_2 \rightarrow \text{NP}_1 + \text{did not} + \text{V} + \text{NP}_2$$

Declarative-interrogative:

$$\text{NP}_1 + \text{V} + \text{ed} + \text{NP}_2 \rightarrow \text{did} + \text{NP}_1 + \text{V} + \text{NP}_2$$

Table 7-1 Feature List for Sentences 22 to 29

Active	Positive	Declarative	Sentence number
√	√	√	22
x	√	√	23
√	x	√	24
√	√	x	25
x	√	x	26
x	x	√	27
√	x	x	28
x	x	x	29

The remaining rules, such as passive-interrogative or negative-interrogative, are equally simple to generate.

The language analysis, or *recognition* process, using a phrase structure grammar is straightforward since it consists in a sequential application of the rewrite rules to the initial sentence symbol (S), and terminates when the complete sentence of terminal symbols has been generated and no nonterminals remain.

The recognition process using a transformational grammar is more complex. A transformational grammar may be separated into two parts. The first is known as the *base component* of the grammar that generates the so-called deep structure of a sentence reflecting the actual syntactic and semantic interpretation of the input. Second, there is the *transformational component* that operates on the output of the base component and generates the *surface structure* of the sentence reflecting the actual phonetic representation. The sentence *generation* process using a transformational grammar is outlined in Fig. 7-6. Sentences that are semantically identical but structurally different will exhibit the same deep structure but different surface structures [26].

To utilize a transformational grammar for the analysis of natural language input, it is necessary to reverse the process of Fig. 7-6 as follows:

1 A standard parsing system is used first to obtain one or more trees exhibiting the *surface* structure of the input.

2 *Reverse* transformations must then be applied to the surface structures to obtain the underlying *deep* structure.

This process must be repeated for each initial surface tree and for all alternative intermediary trees obtained when more than one reverse transformation applies.

Unfortunately, while the number of transformations that produce a surface structure from the underlying deep structure is normally small, this is not true when the process is reversed. Experience indicates that the reverse process ex-

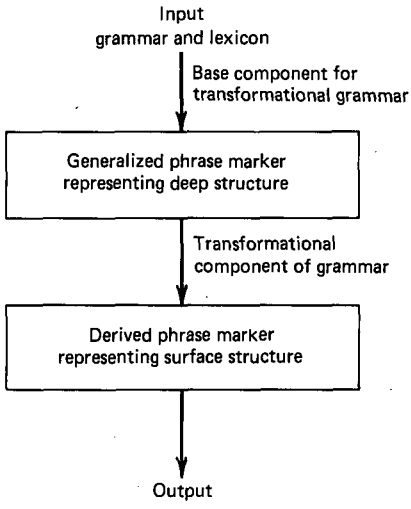


Figure 7-6 Sentence generation using transformational grammar.

plodes except when the number of generated surface trees is very small or when enough information is provided for each node in the surface trees to select only those inverse transformations that are likely to lead to correct deep structures. There is some hope that for sufficiently restricted topic areas appropriate statements of surface structure/deep structure relations might be generated so as to render a transformational analysis useful in practice [8,27].

**C Augmented Transition Network Grammars

The augmented transition network (ATN) grammars offer all the facilities inherent in transformational grammars. In addition their structure is sufficiently simple to render a practical application reasonable. As a result, ATN grammars have been chosen for incorporation into most practical language processing systems [28–30].

Several grammatical systems including the ATN operations are based on the notion of a *finite state machine* represented as a graph. A finite state graph consists of *nodes* and *branches* between certain pairs of nodes. Each node symbolizes a state of the machine, and the branches represent transitions from one state to another. A transition from state A to state B takes place when the symbol attached to branch AB occurs at the input.

Consider as an example the simple finite state graph of Fig. 7-7. The start is assumed to be in state S. If the next input symbol is "a," a transition is made from S to state Q_1. If the initial input symbol is not "a," the recognition process fails because no other path is provided for leaving state S. Assuming that the first symbol is in fact an "a," two possibilities are open in state Q_1: either the next input symbol is a "b," in which case the machine stays in state Q_1, or the recognition process ends at the pop exit. The pop marker does not represent a new state but simply designates a compulsory exit from the graph. When the pop is reached after the last input symbol has been read, the input is *ac-*

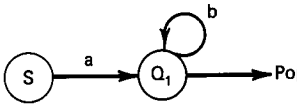


Figure 7-7 Simple finite state graph.

cepted, that is, a correct analysis has been obtained. It is clear from the graph of Fig. 7-7 that the only sentences actually accepted are of the form abbb . . . b, also written ab^n for some value of $n \geq 0$.

Machines such as that of Fig. 7-7 are not capable of accepting all context-free languages. Suppose, however, that a collection of graphs were available and that it were possible to jump from one graph to another by labeling certain branches with the start state of some graph. An example of such a network is given in Fig. 7-8. It may be seen that once arrived in state Q_1, the current status can be saved and the graph can be started again at state S. A *push-down store*, or *stack*, is used to store the current status before proceeding back to the beginning state. A stack uses a last-in first-out discipline, very much like the usual stack of plates in a cafeteria line. That is, only the top item is accessible (the one last placed onto the stack). In the example of Fig. 7-8, the branch in state Q_1 is labeled push S to indicate that the current state (Q_1) is "pushed" onto the stack before going back to state S, unless a "b" is recognized at the input, in which case the alternative transition to state Q_2 would be made without going back to S.

When the pop symbol is reached and the stack is empty, the input is accepted. If the stack is not empty when a pop is reached, the top symbol is read from the stack and a return is made to the state from which the original push occurred. It can be verified that the graph of Fig. 7-8 accepts the input sentence a^nb^n for $n > 0$, that is, a string of a's followed by a string of the exact same number of b's. Such a sentence is context-free.

Consider as an example the input aabb. The details of the analysis are presented in Table 7-2. The first "a" is recognized, causing a transition from S to Q_1. Before returning to S to accept the second "a," the current state (Q_1) is stored in the stack. When the second "a" is read from the input, a second transition occurs from S to Q_1. When the first "b" is read, a transition is made to Q_2 from where the pop exit is reached. Since the stack is not empty, the stack contents reveal that a return to state Q_1 is in order which now makes possible the acceptance of the final "b" symbol.

In practical transition networks, there may be ambiguity in the node labels. That is, from a given state several possible transitions to other states may be possible. Then it is wise to arrange the possible paths according to the probability that a given path will lead to a correct acceptance of the input. The most

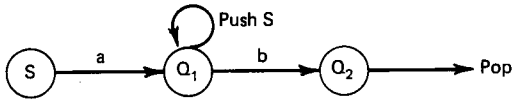


Figure 7-8 Finite state graph recognizing context-free languages.

Table 7-2 Recognition Process of aabb by Finite State Graph of Fig. 7-8

Current state	Input string remaining to be recognized	Stack contents
S	aabb	—
Q_1	abb	—
S	abb	Q_1
Q_1	bb	Q_1
Q_2	b	Q_1
Pop	b	Q_1
Q_1	b	—
Q_2	—	—
Pop	Accept	

probable path is then taken first. If that path does not lead to a satisfactory termination—for example, the input may be exhausted before the final pop is reached—a *backtracking* mechanism may be used to return to the point of ambiguity in order to try the next alternative path.

Consider as an example the finite state graph of Fig. 7-9, and assume that the input is abb. It is clear that if the upper path is taken from S to Q_1 when the "a" is read, a "jam" occurs in the sense that the pop exit is reached with a final "b" still unrecognized. The lower path of Fig. 7-9 will, however, properly recognize the input sentence.

A first sample transition network grammar for English is presented in Fig. 7-10. This consists of three graphs whose initial states represent sentences (S), noun phrases (NP), and prepositional phrases (PP), respectively. The transitions from one graph to another are indicated by push labels as before: from the S-graph one can reach the NP-graph in three different ways from states S, Q_2, and Q_4, respectively; the PP-graph can also be reached from state Q_5 of the S-graph, and transfers are provided both ways between the NP- and the PP-graphs.

Suppose that the sentence to be recognized starts with a noun phrase, a push would then occur from state S to the NP-graph. Assuming that the noun phrase were actually recognized, one of the pop exits would be reached in the NP-graph after states Q_7 or Q_8. The highest entry in the stack would read

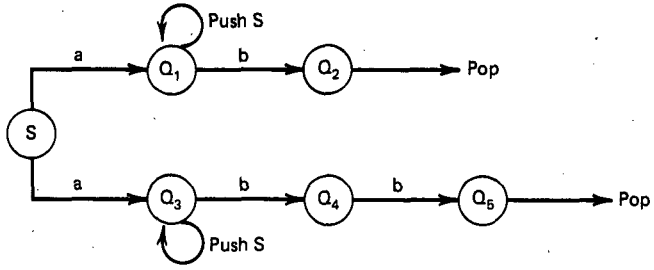
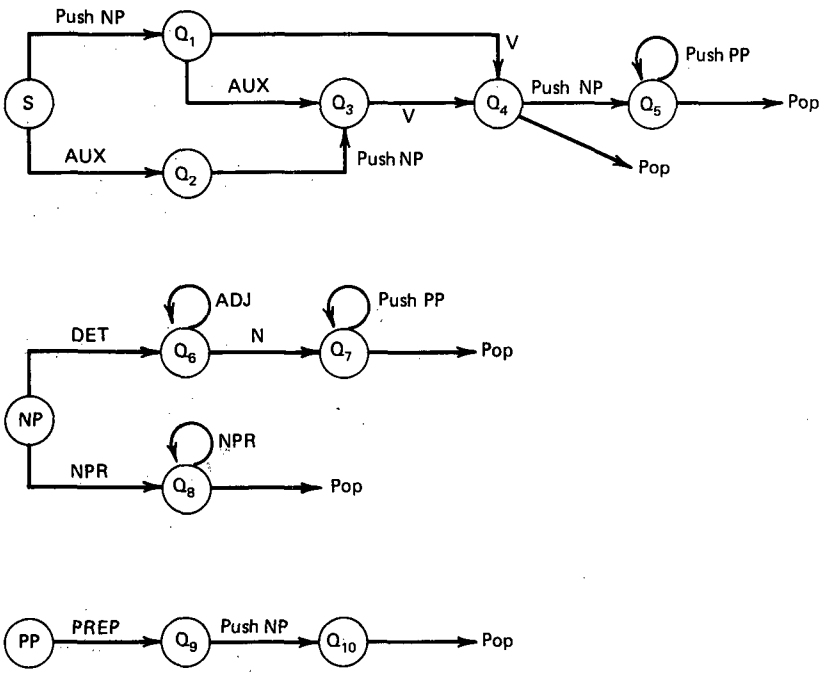


Figure 7-9 Finite state graph with ambiguous transition.



S: sentence AUX: auxiliary verb N: noun
NP: noun phrase ADJ: adjective NPR: proper name
PP: prepositional phrase DET: determiner PREP: preposition

Figure 7-10 A simple transition network grammar for English.

S(NP), indicating that the original transfer came from state S and that a noun phrase was expected to occur. If a noun phrase is found, then the conditions of the arc from S to Q_1 are fulfilled, and a return can be made from the NP-graph to state Q_1 of the S-network. The push label thus indicates that a transition is in order to another state; at the same time it designates the return state from which the original recognition process must be resumed after the temporary transfer. The return must be made to the state pointed to by the push branch.

The complete recognition process is shown in Table 7-3 for the sample input "the tall man in the Stetson is John Wayne" using the grammar of Fig. 7-10. It may be noted that a backtracking jump is made from state Q_6 on line 9 to state Q_8 on line 10, because the input fails to substantiate either the adjective or the noun required at state Q_6. Instead "Stetson" is classified as a proper name (NPR) which is correctly recognized by the alternative path from the NP node.

The transition network grammars actually generate a phrase structure tree instead of merely providing the output "accept" or "reject." When a particular graph is started, a node is created with the name of the graph. Each time an

Table 7-3 Recognition Process of "The Tall Man in the Stetson Is John Wayne" Using Grammar of Fig. 7-10

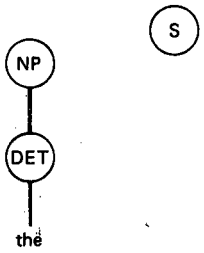
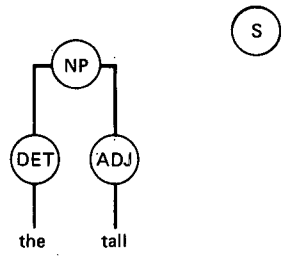
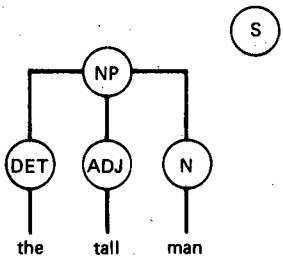
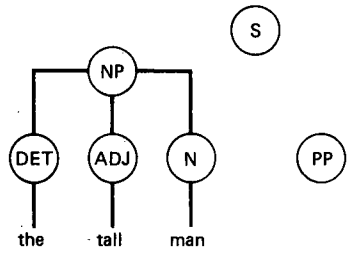
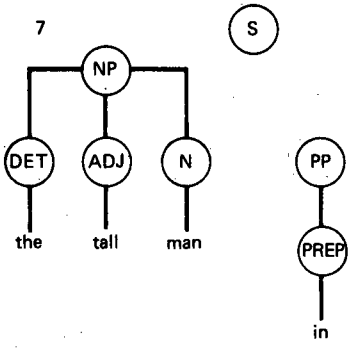
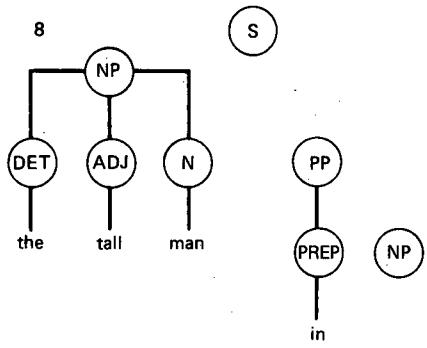
Current state		Input string remaining to be recognized	Stack contents
1	S	the tall man in the Stetson is John Wayne	—
2	NP	the tall man in the Stetson is John Wayne	S(NP)
3	Q_6	tall man in the Stetson is John Wayne	S(NP)
4	Q_6	man in the Stetson is John Wayne	S(NP)
5	Q_7	in the Stetson is John Wayne	S(NP)
6	PP	in the Stetson is John Wayne	S(NP)Q_7(PP)
7	Q_9	the Stetson is John Wayne	S(NP)Q_7(PP)
8	NP	the Stetson is John Wayne	S(NP)Q_7(PP)Q_9(NP)
9	Q_6	Stetson is John Wayne	S(NP)Q_7(PP)Q_9(NP)
10	Q_8	is John Wayne	S(NP)Q_7(PP)Q_9(NP)
11	Pop	is John Wayne	S(NP)Q_7(PP)Q_9(NP)
12	Q_{10}	is John Wayne	S(NP)Q_7(PP)
13	Pop	is John Wayne	S(NP)Q_7(PP)
14	Pop	is John Wayne	S(NP)
15	Q_1	is John Wayne	—
16	Q_4	John Wayne	—
17	NP	John Wayne	Q_4(NP)
18	Q_8	Wayne	Q_4(NP)
19	Q_8	—	Q_4(NP)
20	Pop	—	—
21	Q_5	—	—
22	Pop	Accept	—

input word is recognized (that is, when a transition is completed) an appropriate terminal node of the tree is created. When a pop is reached, the currently active subtree is attached to the node representing the next higher level subtree to which the process now returns. The tree building process for the sample sentence of Table 7-3 is illustrated in detail in Fig. 7-11.

The problems previously mentioned for the standard context-free phrase structure grammars are of course still present with transition networks of the type illustrated in Fig. 7-10. In particular the grammar could equally well accept the sentence "the tall man in the Stetson are John Wayne." This kind of problem can be taken care of by a larger grammar of the same type which distinguishes plural noun phrases (P.NP) from singular noun phrases (S.NP), plural verbs (P.V) from singular verbs (S.V), and so on. This solution produces an unwieldy grammar and requires a great deal of backtracking.

The solution is to use an *augmented* transition network grammar which adds to the basic apparatus a set of *storage registers*, tests on the content of the storage registers, and *actions* consisting either of storage register settings or of structure building operations that change the structure of the output tree. The storage registers are used principally to store information about the number or type of a noun or the tense of a verb. For example, when analyzing "the tall

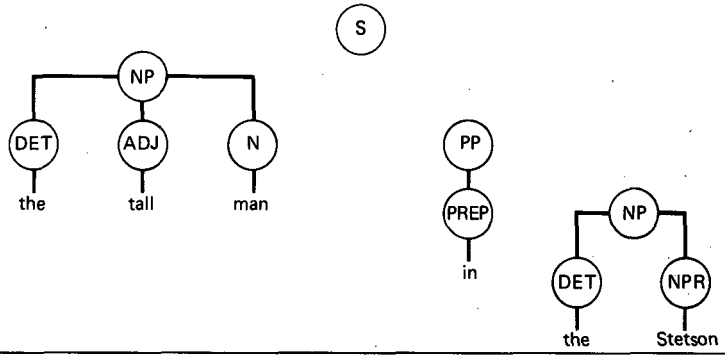
Figure 7-11 (*opposite*) Tree building process for sample sentence of Table 7-3.

Line number in Table 7-3	
1	
2	
3	
4	
5	
6	
7	
8	

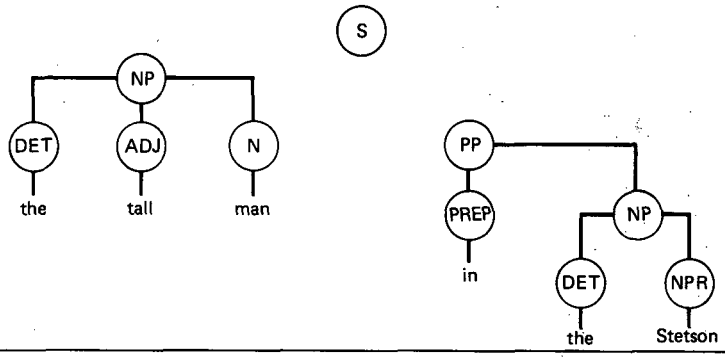
(Continued)

Line number
in Table 7-3

9-10

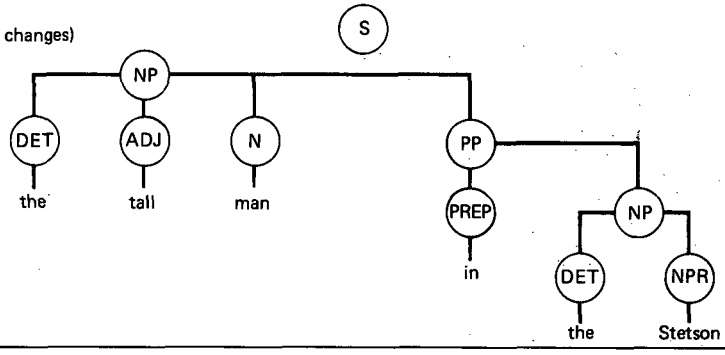


11

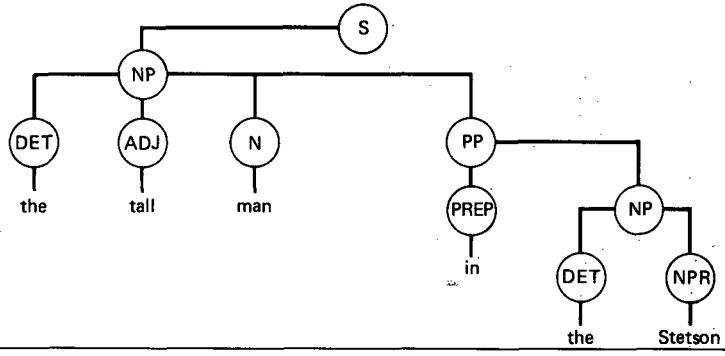


12 (no changes)

13



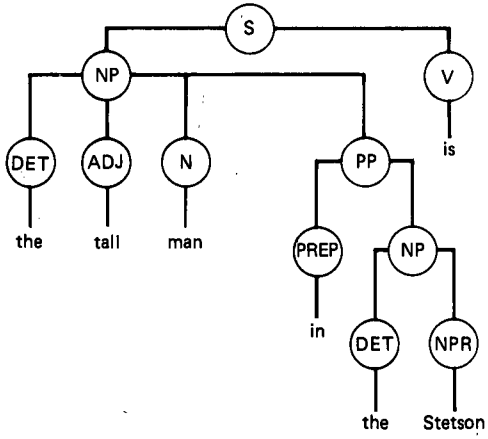
14



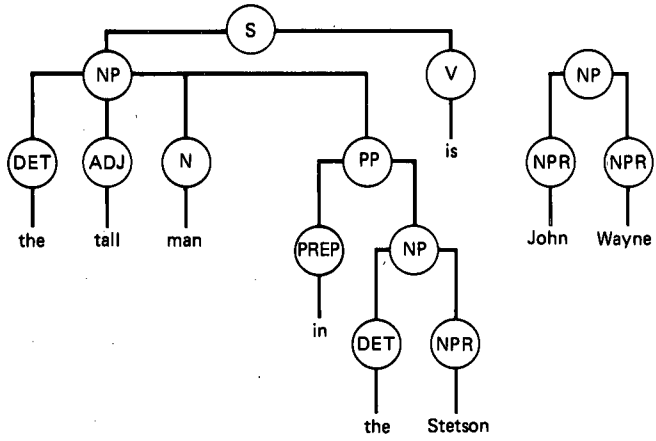
Line number
in Table 7-3

15 nothing changes

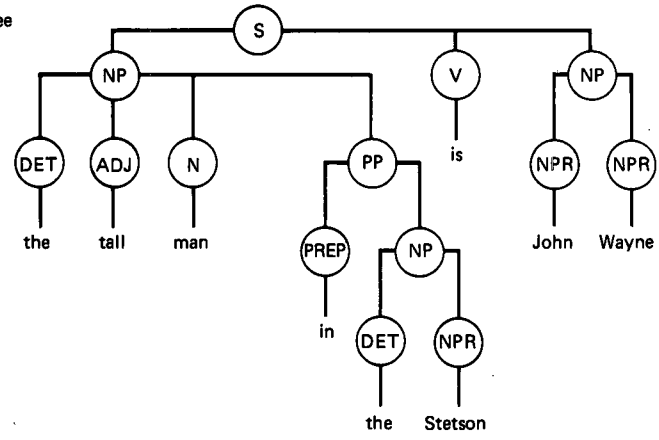
16



19



20 final tree



man," a "singular" identifier is stored in a register to specify the number of "man." Later on when the verb is analyzed, a test is performed prior to making the corresponding transition on the graph to ensure that the number of the verb currently being processed is the same as the number of the noun previously stored away in the register. The result is that the fragment "is John Wayne" would be accepted whereas "are John Wayne" would be rejected.

The structure building operations may be quite complex as required, for example, to rearrange a phrase structure tree following the analysis of a passive sentence. Alternative grammatical analysis systems have been suggested including some that scan a sentence in right-to-left instead of left-to-right order, some based on simultaneously following several analysis paths instead of backtracking when an ambiguity is detected, and some where the input is scanned many times instead of only once. Many people feel that ATN grammars are simpler to deal with operationally than other syntactic formalisms. Several of the question-answering systems mentioned later in this chapter use ATN grammars to handle the syntactic analysis part.

4 SYNTACTIC ANALYSIS IN INFORMATION RETRIEVAL

Syntactic analysis methods can be used in standard bibliographic retrieval systems in two main ways: on the one hand, a syntactic identification may enhance the indexing operation by making possible the assignment to the documents and queries of syntactically correct phrases replacing the single terms that are normally used. On the other hand, it may be possible by using syntactic approaches to obtain a more detailed view of the document contents, leading to directed retrieval activities that would take into account individual document portions such as sentences and paragraphs.

The latter possibility has led to the so-called passage retrieval, where attempts are made to retrieve individual passages or sentences of documents rather than complete documents only [31–33]. Passage retrieval is based on the analysis of the full text of documents, the aim being to retrieve either *answer reporting* passages, that is, passages from which an answer to a question can effectively be inferred, or alternatively *answer indicative* passages which indicate that the same document also contains an answer reporting passage. Passage retrieval may be advantageous because answers to questions could be immediately available instead of merely references to answers.

The basic idea in passage retrieval is the construction of detailed queries followed by the retrieval of all passages that contain all aspects of the query. A syntactic analysis system might be used to control the detailed comparison between queries and document passages. In addition, a thesaurus that expands the original query by including synonyms and other related words is also needed.

Connected passages might also be retrieved by choosing appropriate answer reporting, or answer indicative passages, and then adding follow-up sentences starting with connecting terms, such as "however," "these," or "on

the other hand." The follow-up sentences should also exhibit additional matches with the query. In tests performed with experimental collections, the passage retrieval technique produced fairly high recall results exceeding 60 percent in many cases [31–32].

A possibly more immediate use of syntactic techniques in information retrieval is provided by its application to the indexing task, and particularly to the choice of noun phrases or prepositional phrases for indexing purposes. One possibility consists in using a simplified syntactic analysis that assigns one or more syntactic markers to each text word, and defining an indexing phrase as consisting of sets of contiguous words representing a specified sequence of syntactic markers [34–37].

One particular technique for the detection of indexing phrases uses the following basic indexing procedure [34–35]:

1 A *recognition* dictionary is used to assign *one* of 16 possible syntactic categories to each word (one particular category identifies the corresponding word as a "throw-away" word to be disregarded).

2 A *format* dictionary stores a total of 77 permissible syntactic formats for the phrases to be assigned to the text; these formats range in complexity from a single noun (N) to a sequence of five nouns (ZZZZZ) none of which is sufficiently meaningful to be permitted to stand alone.

3 The indexing cycle consists in accumulating sequences of non-throw-away words of up to five words from the input, the length of a sequence being determined by the occurrence of delimiters such as punctuation signs. The syntactic markers corresponding to the input words are obtained from the recognition dictionary, and the format dictionary is used to determine whether the input sequence of markers corresponds to one of the permissible formats in the dictionary. If so, the corresponding phrase is accepted as an indexing phrase. If no match occurs with one of the stored formats, the candidate phrase is shortened by deletion of one word and the operation is repeated.

In tests comparing this simple automatic indexing process against a conventional manual indexing method, slightly lower recall but slightly higher precision were obtained for the machine process compared with the manual methodology, indicating again that conceptually quite simple automatic methods can be competitive with the conventional manual procedures [36].

An alternative method of generating indexing phrases consists, of course, in performing a full syntactic analysis of a text, or text excerpt such as an abstract, and in assigning those phrases as index terms whose components exhibit specified syntactic relations between them [38]. Such a process was used in an early implementation of the SMART system in the hope of avoiding ambiguities of the "Venetian blind" versus "blind Venetian" type.

In the SMART system, a dictionary of *criterion trees* was used to record the allowable syntactic patterns between phrase components. A criterion tree is a structure including term specifications, syntactic indicators, and syntactic relations obtaining between certain terms. A typical criterion phrase specifica-

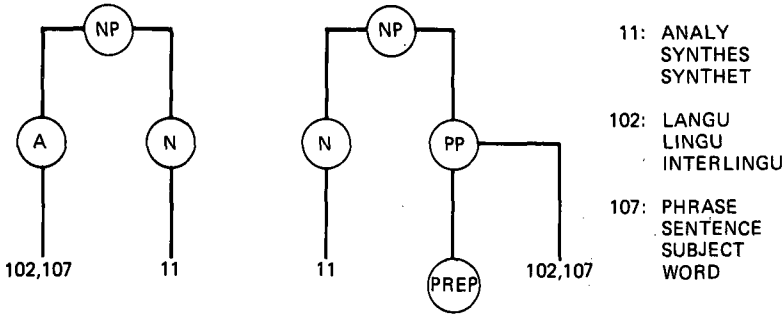


Figure 7-12 Sample criterion phrase specification.

tion is shown in Fig. 7-12. The example of Fig. 7-12 exhibits the syntactic relations which are specified between the word classes. Each word class is given a numeric code (11, 102, and 107 in the example), and a dictionary is used to substitute one or more actual words or word stems for each class. The left-hand tree of Fig. 7-12 covers phrases such as language analysis, linguistic analysis, and interlingual synthesis. The right-hand tree represents analysis of words, synthesis of language, analysis for sentences, etc.

The flexibility of the criterion tree process stems from three main characteristics:

1 Word stems rather than individual words are used in the dictionary as class entries; a single word stem represents many words.

2 Class numbers, rather than words or word stems are attached to the nodes of the syntactic trees.

3 A variety of syntactic connection patterns is provided between the word classes.

As a result, a small criterion tree dictionary can account for a large number of potential indexing phrases.

When the criterion tree dictionary is used for indexing purposes, the index terms and phrases are generated using the normal indexing process. A context-free phrase structure grammar is then used to determine the syntactic structure of the indexing phrases. Finally, a check of the preconstructed criterion tree dictionary reveals whether the actual syntactic pattern in the phrases matches an entry in the tree dictionary. If so a phrase is accepted; otherwise it is rejected.

The existing experimental evidence unfortunately shows that the criterion tree dictionary does not operate as well as expected [39]. In fact, the statistical phrase construction methods described in Chapters 3 and 4 would normally outperform the syntactic phrase procedures. Since the syntactic processing is expensive to perform, there is obviously no point in going that route unless substantial improvements in recall and precision are obtainable. The disappointing results of the early tests may be due to two principal causes:

1 A phrase structure grammar was used to perform the syntactic analysis, rather than a more sophisticated transformational or ATN grammar; it is conceivable that a more refined syntactic process could provide more accurate retrieval results.

2 The syntactic restrictions imposed on the indexing phrases are often very confining, with the result that satisfactory phrases are rejected because the components do not obey the preestablished restrictions.

An example may be used to illustrate the second argument. The loosest kind of syntactic connection between two or more phrase components is a requirement to have the components appear in a common subtree of the syntactic phrase marker. Unfortunately, even such a loosely formulated syntactic restriction will cause the rejection of some phrases that appear essential for content identification. Consider a sentence such as "Experts in *linguistics* may study sentence generation and *analysis*." Most syntactic analysis systems would not place the terms "linguistics" and "analysis" into a common subtree, and hence the phrase "linguistic analysis" would not be assigned to a document containing that sentence, even though no other phrase would appear to be more pertinent.

The conclusion is that the role of linguistic methodologies in general and of syntactic analysis in particular is still unresolved for information retrieval. Before reaching a final conclusion in this area, it is wise to wait for the appearance of more sophisticated language analysis methods that are at the same time sufficiently efficient to permit incorporation into operational retrieval frameworks. Such methods should then be thoroughly evaluated to determine their actual value in information retrieval.

5 LINGUISTIC METHODS IN QUESTION ANSWERING

**A Knowledge Representation

While some question exists about the usefulness of language processing in information retrieval, most experts feel that the linguistic methodology cannot be bypassed in question-answering systems. It is not possible in the present context to examine in detail the components and structure of modern question-answering systems. A brief look at some of the more notable features must suffice.

Consider first the problem of knowledge representation. It was seen earlier that stored knowledge structures are needed in many language processing systems. They are required both for the representation of knowledge in the topic area under discussion and also to supply "common sense" knowledge that is normally available to human beings.

Before examining the various types of knowledge structures, it may be useful to look briefly at some of the characteristics of question-answering systems. Such systems may be considered to be extensions of the normal data base management and document retrieval systems in the following sense:

1 The data records used as a basis for question answering may in principle be more general and of greater scope than those common in business processing or in bibliographic retrieval. In any case, they are not normally restricted to the simple tabular format common in data base systems, or to the processing of bibliographic records alone.

2 The queries allowed in question answering are also more general than those common in data base systems or in document retrieval. In data base systems the normal query specifies the values of certain attributes attached to the records—for example, a request for personnel records could refer to items whose profession equals engineer, whose age is 33, with length of service exceeding 10 years. In document retrieval, queries consist of keyword identifiers possibly interconnected by Boolean operators. In question-answering systems, on the other hand, a greater variety of queries may be allowed. For ease of interaction one would also like to allow natural language queries, and furnish natural language answers.

3 External knowledge intrudes because the answering process in a question-answering system depends on a knowledge of the social context and on the prevailing conversational framework in addition to the normally required subject knowledge. In particular, the problem of determining the *focus* of a query plays a role in general question answering, when it does not normally in data base management or in document retrieval. That is, in question answering it becomes necessary to decide why users are asking a question in addition to ascertaining what they want to know before formulating an answer. (A user who asks, "Why did you fly to Stockholm?" does not want a reply stating "because it was too far to walk," even though such an answer might be formally correct.)

The knowledge structures needed to cope with this expanded environment normally take the form of *semantic nets*. These consist of *nodes* to represent the concepts, events, characteristics, and values of interest in a system, as well as *branches* specifying the relationships between nodes. The branches may be labeled with "case" labels, such as agent, object, instrument, source, or destination, to simplify the interpretation of the graph [40–45]. A typical semantic network representing the operation of bolting two objects together using nuts and bolts is shown in Fig. 7-13. The branches of Fig. 7-13 designate timing information as well as whole-part and subset-superset relationships.

A wide variety of semantic nets has been introduced in the literature. Two main types may be distinguished, the *logical* networks and the *conceptual* networks. In the logical network, the graph is interpreted as a logical statement about properties holding for various entities. The basic primitives in a logical network are *predicates* defined for the entities under discussion. A predicate is a logical function of one or more variables producing a truth value (true or false) when the variables are replaced by appropriate actions, objects, instruments, and so on.

Typical predicates defined for a given area of study could be "is a member of," "is the father of," "is a prime number." When the variable *slots* in a predicate are filled with appropriate entities, a predicate becomes a *proposition*,

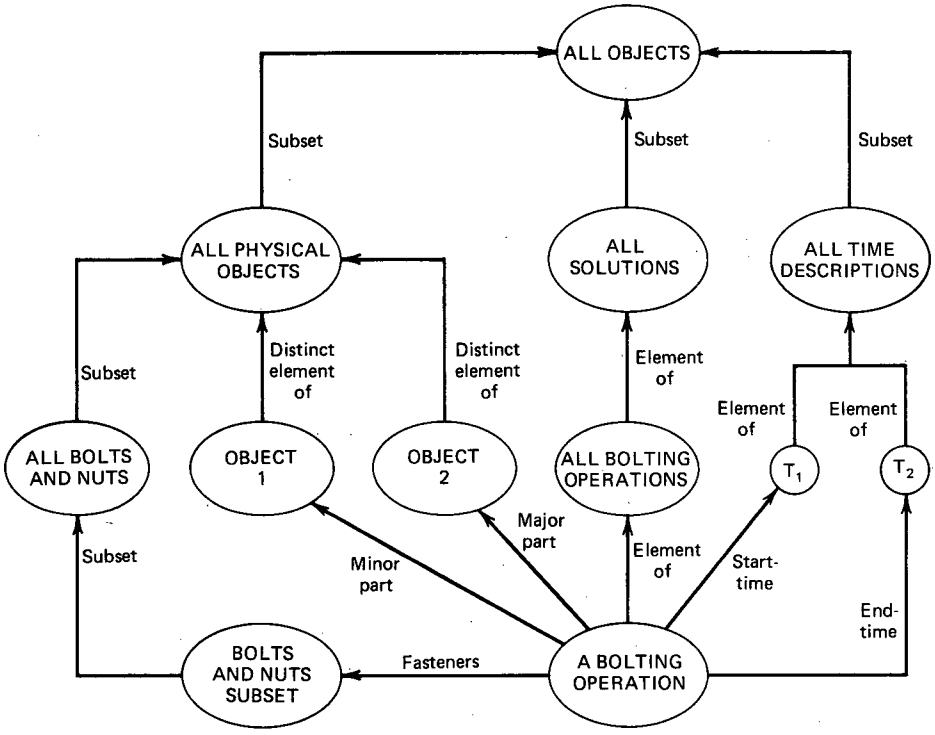


Figure 7-13 Sample semantic network (describing a bolting operation using nuts and bolts to connect two objects). (*Adapted from reference 46.*)

that is, a statement with a truth value. Typical propositions involving the predicates introduced earlier are "John is the father of Mary," "17 is a prime number," "x is a member of class y." Figure 7-14 illustrates the logical net representation of a sample proposition.

Two or more distinct propositions can be combined into a compound proposition using logical relationships, such as the Boolean connectives AND, OR, NOT. Thus, given two propositions such as "x is a member of y" and "y is a member of z," a new proposition would be stated as "x is a member of y AND y is a member of z." More generally, using a logical model, a semantic network can be provided with a precise semantic interpretation, and with the inference rules and procedural steps necessary to construct answers in response to certain questions. In other words, since the components of the logical net include all the apparatus of the *predicate calculus*, the question-answering process effectively becomes a theorem-proving task guided by precise logical rules.

Consider a query such as "is x a member of z?" Given the stored propositions, "x is a member of y" and "y is a member of z," the compound proposition "x is a member of y AND y is a member of z" is generated. From this the formal inference rules may be usable to derive a new proposition "x is a member of z" which can then be used to answer the question affirmatively.

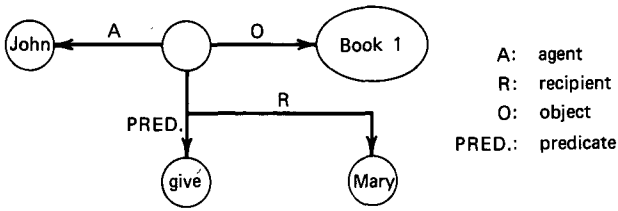


Figure 7-14 Typical representation of a proposition as a logical network, "Agent (John) gives object (book 1) to recipient (Mary)." (*From reference 47.*)

It is clear that given a proper choice of predicates and logical characteristics describing the data, a wide variety of different kinds of information can be represented. For example, normal index terms, values of attributes, as well as hierarchical and other relationships between entities may be represented. A logical network could then serve as a specification of the semantic and/or syntactic characteristics of many types of records, including commercial files, document collections, and other kinds of artifacts.

However, for some purposes the logical framework may be somewhat confining for the representation of general knowledge. It may then be convenient to allow for more general *conceptual networks*, where once again the nodes represent the basic concepts and entities of interest. The relationships between concepts represented by the branches are then essentially open-ended. A variety of relationships represented by portions of conceptual semantic nets are presented in Fig. 7-15.

In one formalism, the following components are recognized [45]:

1 *Concepts* that are considered as the essential constants or parameters of the world
2 *Events* that represent the actions which occur in the topic area under consideration
3 *Characteristics* that are used to modify concepts, events, or other characteristics
4 *Values* that represent the attributes attached to individual records such as a particular weight of a person or an address of an individual

In many semantic networks a distinction is made between *generic* concepts that represent abstractions or classes of events, from particular instances of events. A generic statement such as "physical objects carry a weight" would then be treated differently than a specific instance such as "Joe Smith weighs 150 pounds."

Among the operations that must be defined for semantic nets are the basic *search* functions that relate individual instances to characterizations on the generic level. *Node-creating* operations can add information to a network. *Network transformation* rules alter the network in accordance with newly available information, or combine individual events, concepts, and characteristics into sequences of events, or *scenarios*. The network transformations and exten-

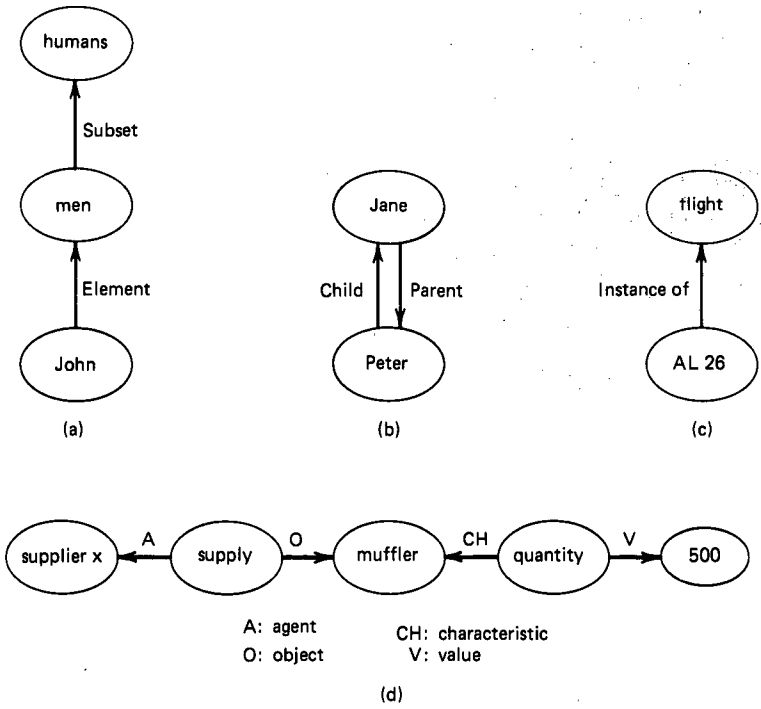


Figure 7-15 Typical relationships included in semantic nets. (a) "John is a member of the human species." (b) "Peter is the son of Jane." (c) "AL 26 is a flight." (d) "Supplier × supplies 500 mufflers."

sions may then serve a role in content representation similar to that previously mentioned for frames, scripts, or schemes.

The basic problem with the current perceptions of knowledge representation is that they lack *uniformity*. In other words, no *accepted* theory of knowledge representation exists. Many of the structures appear to be produced by ad hoc definitions. Rules used to manipulate the structures are sometimes uncertain and inadequately motivated. The structures are thus hard to extend to new discourse areas or to altered processing environments. Since an adequate theory of knowledge representation is essential for question-answering purposes, it is not surprising that the question-answering problem still lacks a general solution.

B Question-Answering Environment

From what has already been said it should be clear that a number of quite sophisticated processing steps are required in order to produce direct answers to questions. At the least, it is necessary to thoroughly comprehend the subject area under discussion. In addition the context surrounding the question-answering interplay must be understood. Finally, a complete language analysis

system capable of transforming user-formulated queries into answers to questions must be available.

Most existing question-answering systems perform tasks of substantial sophistication in a restricted semantic environment. Typically, a microworld is chosen in which only a small number of entities and a small number of relations are recognized. This automatically implies that the dictionaries which specify the syntactic and semantic properties of the concepts are of limited size, and that the syntactic and semantic patterns that occur (or that are allowed to occur) are also restricted to those of a well-defined domain of discourse. Alternatively, a larger area of discourse may be tolerated, or the dependence of the syntactic and semantic systems on the discourse area may be less rigid, provided that the resulting ambiguities can be resolved externally, possibly by interaction with the user during the course of the question-answering process [48].

In many systems the more difficult problems in language analysis are disregarded. The system may not, for example, include methods for the interpretation of conjunctions (and, or, but, etc.), the proper resolution of quantification (all, every, each, some, etc.), the handling of anaphoric expressions (where pronouns are used to refer to antecedents in the text that may be ambiguous), the processing of ellipses (where parts of the text are omitted because the intended meaning is clear from the context), and the interpretation of polysemantic words, such as "base," whose meaning must be clarified by the context.

Typically, a special-purpose environment is chosen where many of the aforementioned difficulties can be bypassed. Examples are the use of special kinds of texts such as pathology data and medical diagnostic summaries [49–52] or of formatted data structures such as occur in data base management [53–59]. In either case, the conversion of an information request into a formalized statement of user needs may be simpler to carry out than in totally unrestricted question-answering environments.

The language analysis features incorporated into some existing question-answering systems are described in the remainder of this chapter.

*C Linguistic Features in Question Answering

One of the earliest operational question-answering systems was LSNLIS, designed to answer questions about chemicals and rock samples brought back from the moon by the astronauts [53–54]. Like other existing question-answering systems, the LSNLIS system is characterized by a discrete knowledge area and by circumscribed user interests. The system is modular in that the syntactic component is separated from the semantic interpretation, and that in turn is distinct from the data base retrieval component, as seen in Fig. 7-16.

The user queries are first subjected to a standard syntactic analysis using an ATN grammar. The semantic rules are tightly bound to the subject matter and are designed to produce formal representations of the meaning of the queries. These formal representations are then compared with the stored data

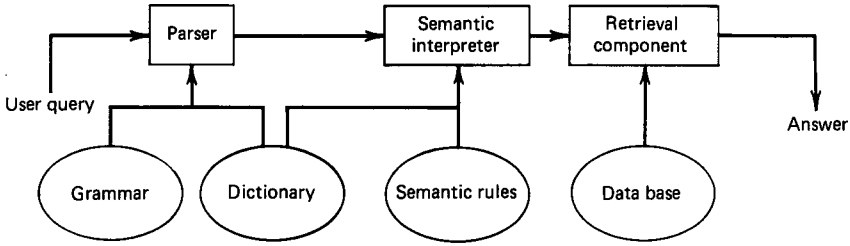


Figure 7-16 Basic operations of the LSNLIS system.

base consisting of information about the composition of the various rock samples found on the moon, the amount of material found of each type of mineral, the location where each sample was found, and so on. Eventually, appropriate answers are extracted from the data base.

The semantic interpretation rules constitute possibly the most interesting part of the system. Two types of information are used as input, consisting first of portions of the syntactic phrase markers produced by the syntactic analysis, and second of certain semantic features attached to the terminal nodes of the syntactic trees (that is, semantic classification data for some of the words in the query texts).

Consider, as an example, the semantic interpretation rule "author of" presented in detail in Fig. 7-17. The input consists of two syntax tree excerpts identified, respectively, as S:NP-V and S:V-NP, representing the usual subject-

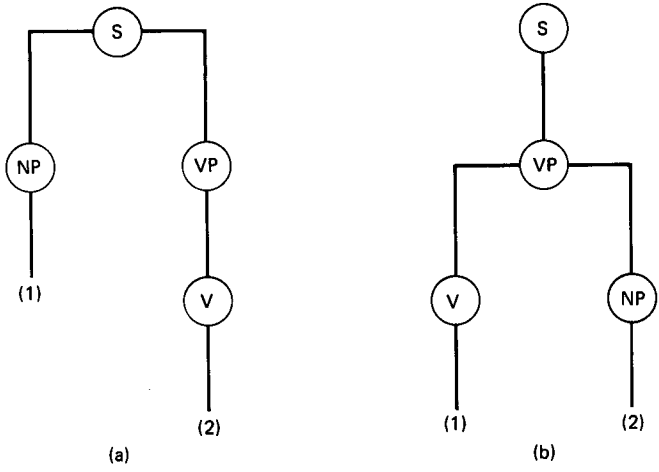


Figure 7-17 Semantic interpretation in the LSNLIS system. (a) Subject-verb tree—S:NP-V. (b) Verb-object tree—S:V-NP. (c) Semantic rule "author of."

verb and verb-object components. Assuming that the dictionary processing reveals that the word attached to node (1) of the subject-verb tree is a member of the class of "persons," that is, an animate human being, while nodes (1) and (2) of the verb-object tree correspond, respectively, to some form of the word "write" and to a member of the class of "documents," the conclusion is that node (1) of the NP-V tree is the "author of" node (2) of the V-NP tree. The rule of Fig. 7-17 then makes it possible to answer questions such as "who wrote document x?" or "list the authors writing documents that concern rock samples."

Many of the existing question-answering systems are based on an organization similar to LSNLIS in the sense that the syntactic information produced by an ATN analysis is used with semantic information attached to the nodes of the syntactic tree (for example, information about the "cases" of the concepts attached to certain nodes) to derive the semantic interpretations of the user queries. When the queries to be handled by the system are tightly circumscribed and the discourse area is well defined, the number of different syntax trees and the variety of the needed semantic interpretation rules often proves sufficiently small to permit the construction of practical question-answering systems.

Because of the restricted discourse areas, many of the semantic ambiguities arising in the language-at-large are effectively eliminated in the practical question-answering environment. Thus the word "sample" in the LSNLIS context may safely be assumed to refer to a lunar sample without further analysis. Nevertheless, a substantial amount of effort can be spent to adapt the question-answering environment to the sometimes sloppy habits of the users. Thus, some systems include routines for spelling correction, ellipsis substitution, and the resolution of pronoun reference [55–57]. Ellipsis in the question-answering context normally implies that the user proposes a series of questions without repeating all components. The user might state: "identify all samples in which glass was found"-- "what about chromite?" and the system would interpret the second part as "identify all samples in which chromite was found." Pronoun reference is normally resolved by looking for a likely referent in the preceding context. Thus "identify the composition of all samples containing glass—list *their* identification numbers" would be interpreted as "list the identification numbers of all samples containing glass."

A substantial amount of work has also gone into the perfection of the dialogue between system and user during the question-answering process. In some systems a clarifying dialogue is in fact initiated with the user when an ambiguity arises. The hope is that the user will resolve the ambiguity by letting the system know which of several alternatives may be correct. The same strategy can be used to get the user to certify the correctness of a final interpretation of the query by the system. If the user returns a "yes," the system proceeds to the retrieval phase; a "no," on the other hand, would mean that the user's interest was misinterpreted. The user would then submit a new formulation of the query [57–59].

Several question-answering systems utilize a syntactic analysis component that is not based on the ATN formalism. The REQUEST system, for example, is based on a transformational grammar [27], and the LSP project uses a "string analysis" system [50–52]. The latter is noteworthy because its somewhat formalized input, consisting of reports on medical test results or medical diagnostic summaries, makes it possible to reduce the sentences of natural language text to a tabular format which serves as a basis for the retrieval component. The system includes

1 A syntactic analysis component
2 A system of transformations to regularize the syntactic structure of each sentence
3 A reduction of the text to tabular format
4 A retrieval component that uses the entries in the columns of the tables to answer user queries.

A typical tabular format for an excerpt of a diagnostic report is shown in Table 7-4. Such a table might not be easily generated for general texts that do not exhibit the kind of stereotyping and uniformity of language that apparently occurs in some scientific disciplines.

In several question-answering systems, the linguistic processing is not as tightly bound to the domain of discourse as expected. Inevitably, a substantial amount of ambiguity must then be accepted, because restrictive semantic rules are no longer included in the process. The ROBOT system uses a relatively general system of linguistic processing, initially based on an ATN grammar [60]. The tight semantic interpretation rules are then replaced by an interesting verification system that uses the stored data base itself to help in the disambiguation of concepts that require elaboration.

Specifically, given a query such as "list the names of all Chicago secretaries," the system will search the data base to determine whether "Chicago" might be an attribute of secretary. When "Chicago" is discovered as a component of the address of some secretaries, the query is reinterpreted to mean "list the names of all secretaries located in Chicago." A request for "green Fords," where "green" is listed both as a color and as the name of a person, and Ford identifies the name of a person or alternatively an automobile manufacturer, is eventually interpreted correctly because the data base contains entries only for records where car manufacturer equals Ford, and car color equals green, whereas no records exist in the data base for the other three possible interpretations.

An alternative to using the data base for semantic verification consists of building an elaborate knowledge component into the question-answering system. This knowledge system in the form of scripts or frames can then be used not only for query interpretation but also for the generation of answers to questions [61,62]. The answer generation problem in unrestricted question-answering systems presents many unresolved problems. The obvious answer to a

Table 7-4 Tabular Format for Typical Diagnostic Summary

"Patient first had sickle cell anemia diagnosed at age 2 years when he complained of leg pain; he was worked up and diagnosis was made."

			Treatment		Patient state				Time	
	Conjunction	Patient	Instrument	Verb medical	Verb pattern	Body part	Sign	Diagnosis	Prep.	Reference point
1.		patient		first had diagnosed				sickle cell anemia	at	age 2 years
2.	when	he			complained of	leg	pain			
3.		he		was worked up						
4.	and			diagnosis was made						

Adapted from reference 51.

question is almost never adequate—for example, a simple yes/no answer to a user query will be found confining in most instances. Instead users look for answers that take into account the focus of the questions and offer appropriate elaborations and rationalizations. The time is not yet at hand when knowledge encoded in frames or scripts can be used efficiently for the interpretation of queries and the generation of system responses in unrestricted automatic question-answering systems [63].

6 SUMMARY

It has become evident in recent years that an unrestricted automatic natural language information system must necessarily incorporate a complete language understanding system. The latter in turn should be based on an acceptable theory of language, and on prestored knowledge covering the area of discourse under consideration, the general world knowledge normally assumed, and the psychological context of a given interaction in question answering.

Since these basic cornerstones of a full language processing system are not close to being under control, the choice on the part of practitioners interested in making use of language processing tools is twofold. On the one hand, an attempt can be made to use sophisticated linguistic tools in a restricted discourse area where the semantic difficulties can be resolved by the context and much of the ambiguity is automatically absent. This is the path followed by researchers who design question-answering systems in restricted topic environments. On the other hand, some of the language processing tools that appear to be well understood—notably syntactic analysis—could be used to improve the effectiveness of some classes of information systems. The use of syntactic methods in information retrieval may become attractive before long, especially in automatic indexing applications to control the phrases assigned to documents and queries for content identification. Syntactic analysis systems could also be useful for incorporation into full-text processing systems—for example, for automatic abstracting and passage retrieval.

Current indications are that more comprehensive linguistic theories may be needed before sophisticated language processing tools will actually be usable in many general-purpose automatic information systems.

REFERENCES

[1] T.R. Addis, Machine Understanding of Natural Language, International Journal of Man-Machine Studies, Vol. 9, No. 2, March 1977, pp. 207–222.

[2] K. Sparck Jones and M. Kay, Linguistics and Information Science, Academic Press, New York, 1973.

[3] C.A. Montgomery, Linguistics and Information Science, Journal of the ASIS, Vol. 23, No. 3, May–June 1972, pp. 195–219.

[4] K. Sparck Jones and M. Kay, Linguistics and Information Science—A Postscript, in Natural Language in Information Science, D.E. Walker, H. Karlgren and M. Kay, editors, FID Publication 551, Skriptor, Stockholm, 1977, pp. 183–192.

[5] F.J. Damerau, Automated Language Processing, in Annual Review of Information Science and Technology, Vol. 11, M.E. Williams, editor, American Society for Information Science, Washington, DC., 1976, pp. 107–161.
[6] G. Silva and C.A. Montgomery, Knowledge Representation for Automated Understanding of Natural Language Discourse, Computers and the Humanities, Vol. 11, No. 4, July–August 1977, pp. 223–234.
[7] R.C. Schank, M. Lebowitz, and L. Birnbaum, An Integrated Understander, American Journal of Computational Linguistics, Vol. 6, No. 1, January–March 1980, pp. 13–30.
[8] N. Sager, Computational Linguistics, in Natural Language in Information Science, D.E. Walker, H. Karlgren, and M. Kay, editors, FID Publication 551, Skriptor, Stockholm, 1977, pp. 75–100.
[9] A.G. Oettinger, Automatic Language Translation, Harvard University Press, Cambridge, Massachusetts, 1960.
[10] Y. Bar Hillel, Language and Information, Addison-Wesley Publishing Company, Reading, Massachusetts, 1964.
[11] Automatic Language Processing Advisory Committee, Language and Machines, National Academy of Sciences, Publication 1416, Washington, DC., 1966.
[12] D.E. Rumelhart, Introduction to Human Information Processing, John Wiley and Sons, New York, 1977.
[13] J.J. Katz and J.A. Fodor, The Structure of Semantic Theory, Language, Vol. 39, No. 2, 1963, pp. 170–210.
[14] C.J. Fillmore, The Case for Case, in Universals in Linguistic Theory, E. Bach and R.T. Harris, editors, Holt Rinehart and Winston, New York, 1968, pp. 1–88.
[15] D.G. Bobrow and T. Winograd, An Overview of KRL—A Knowledge Representation Language, Cognitive Science, Vol. 1, No. 1, January 1977, pp. 3–46.
[16] R.C. Schank, Conceptual Dependency Theory, in Conceptual Information Processing, Chapter 12, North Holland Publishing Company, Amsterdam, 1975, pp. 22–82.
[17] N.V. Findler, editor, Associative Networks—Representation and Use of Knowledge by Computers, Academic Press, New York, 1979.
[18] M. Minsky, A Framework for Representing Knowledge, in The Psychology of Computer Vision, P. Winston, editor, McGraw-Hill Book Company, New York, 1975, pp. 211–277.
[19] R.C. Schank and R.P. Abelson, Scripts, Plans, Goals and Understanding, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1977.
[20] J.C. Gardin, On the Relation between Question-Answering Systems and Various Theoretical Approaches to the Analysis of Text, in The Analysis of Meaning, M. MacCafferty and K. Gray, editors, Aslib, London, 1979, pp. 206–220.
[21] S.E. Robertson, Between Aboutness and Meaning, in The Analysis of Meaning, M. MacCafferty and K. Gray, editors, Aslib, London, 1979, pp. 202–205.
[22] K. Sparck Jones, Problems in the Representation of Meaning in Information Retrieval, in The Analysis of Meaning, M. MacCafferty and K. Gray, editors, Aslib, London, 1979, pp. 193–201.
[23] N. Sager, Syntactic Analysis of Natural Language, in Advances in Computers, Vol. 8, Academic Press, New York, 1967, pp. 153–188.
[24] S. Kuno and A.G. Oettinger, Multiple-Path Syntactic Analyzer, in Information Processing—62, North Holland Publishing Company, Amsterdam, 1963, pp. 306–311.
[25] S. Kuno, Automatic Syntactic Analysis, in Seminar in Computational Linguistics,

A.W. Pratt, A.H. Roberts, and K. Lewis, editors, Public Health Service Publication 1716, Government Printing Office, Washington, D.C., 1968, pp. 19–41.

[26] N. Chomsky, Aspects of the Theory of Syntax, MIT Press, Cambridge, Massachusetts, 1965.

[27] W.J. Plath, REQUEST: A Natural Language Question-Answering System, IBM Journal of Research and Development, Vol. 20, No. 4, July 1976, pp. 326–335.

[28] R. Grishman, A Survey of Syntactic Analysis Procedures for Natural Language, American Journal of Computational Linguistics, Vol. 13, No. 5, 1976, Microfiche 47.

[29] W.A. Woods, Transition Network Grammars for Natural Language Analysis, Communications of the ACM, Vol. 13, No. 10, October 1970, pp. 591–606.

[30] M. Bates, The Theory and Practice of Augmented Transition Network Grammars, in Natural Language Communication via Computers, L. Bolc, editor, Lecture Notes in Computer Science, Springer Verlag, Berlin, 1978, pp. 191–260.

[31] J. O'Connor, Retrieval of Answer Sentences and Answer Figures by Text Searching, Information Processing and Management, Vol. 11, No. 5/7, 1975, pp. 155–164.

[32] J. O'Connor, Data Retrieval by Text Searching, Journal of Chemical Information and Computer Sciences, Vol. 17, 1977, pp. 181–186.

[33] I. Steinacker, Indexing and Automatic Significance Analysis, Journal of the ASIS, Vol. 25, No. 4, July–August 1974, pp. 237–241.

[34] P.H. Klingbiel, Machine Aided Indexing of Technical Literature, Information Storage and Retrieval, Vol. 9, No. 2, February 1973, pp. 79–84.

[35] P.H. Klingbiel, A Technique for Machine-Aided Indexing, Information Storage and Retrieval, Vol. 9, No. 9, September 1973, pp. 477–494.

[36] P.H. Klingbiel and C.C. Rinker, Evaluation of Machine-Aided Indexing, Information Processing and Management, Vol. 12, No. 6, 1976, pp. 351–366.

[37] J.M. Carroll, W. Fraser, and G. Gill, Automatic Content Analysis in an On-Line Environment, Information Processing Letters, Vol. 1, No. 4, June 1972, pp. 134–140.

[38] G. Salton, Automatic Phrase Matching, in Readings in Computational Linguistics, D.G. Hays, editor, American Elsevier Publishing Company, New York, 1966, pp. 169–188.

[39] G. Salton, The SMART Automatic Document Retrieval System—An Illustration, Communications of the ACM, Vol. 8, No. 6, June 1965, pp. 391–398.

[40] R.J. Brachman, What's in a Concept: Structural Foundations for Semantic Networks, International Journal of Man-Machine Studies, Vol. 9, No. 2, March 1977, pp. 127–152.

[41] J.R. Abrial, Data Semantics, in Database Management, J.W. Klimbie and K.I. Koffernan, editors, North Holland Publishing Company, Amsterdam, Holland, 1974, pp. 1–60.

[42] J.W. Sowa, Conceptual Graphs for a Data Base Interface, IBM Journal of Research and Development, Vol. 20, No. 4, July 1976, pp. 336–357.

[43] G.G. Hendrix, Encoding Knowledge in Partitioned Networks, Technical Note 164, Stanford Research Institute, Menlo Park, California, June 1978.

[44] H.J. Schmid and J.R. Swenson, On the Semantics of the Relational Data Model, ACM SIGMOD Conference Proceedings, Association for Computing Machinery, New York, 1975, pp. 211–223.

[45] N. Roussopoulos and J. Mylopoulos, Using Semantic Networks for Data Base

Management, Proceedings of the Conference for Very Large Data Bases, Association for Computing Machinery, New York, 1975, pp. 144.

[46] B.J. Grosz, The Representation and Use of Focus in Dialogue Understanding, SRI Technical Note 151, Stanford Research Institute, Menlo Park, California, July 1977.

[47] N. Cercone, Morphological Analysis and Lexicon Design for Natural Language Processing, Computers and the Humanities, Vol. 11, No. 4, July–August 1977, pp. 235–258.

[48] T. Winograd, Understanding Natural Language, Academic Press, New York, 1972.

[49] G.S. Dunham, M.G. Pacak, and A.W. Pratt, Automatic Indexing of Pathology Data, Journal of the ASIS, Vol. 29, No. 2, March 1978, pp. 81–90.

[50] N. Sager, Sublanguage Grammars in Science Information Processing, Journal of the ASIS, Vol. 26, No. 1, January–February 1975, pp. 10–16.

[51] R. Grishman and L. Hirschman, Question Answering from Natural Language Medical Data Bases, Artificial Intelligence, Vol. 11, 1978, pp. 25–43.

[52] N. Sager and R. Grishman, The Restriction Language for Computer Grammars of Natural Language, Communications of the ACM, Vol. 18, No. 7, July 1975, pp. 390–400.

[53] W.A. Woods and R.M. Kaplan, The Lunar Sciences Natural Language Information System, Report No. 2265, Bolt Beranek and Newman, Cambridge, Massachusetts, September 1971.

[54] W.A. Woods, R.M. Kaplan, and B. Nash-Webber, The Lunar Sciences Natural Language Information System, Report No. 2378, Bolt Beranek and Newman, Cambridge, Massachusetts, 1972.

[55] G.G. Hendrix, E.D. Sacerdoti, D. Sagalowicz, and J. Slocum, Developing a Natural Language Interface to Complex Data, ACM Transactions on Database Systems, Vol. 3, No. 2, June 1978, pp. 105–147.

[56] W.A. Martin, Some Comments on EQS, A Near Term Natural Language Data Base Query System, Proceedings ACM 1978 Annual Conference, December 1978, pp. 156–164.

[57] D.L. Waltz, An English Language Question Answering System for a Large Relational Data Base, Communications of the ACM, Vol. 21, No. 7, July 1978, pp. 526–539.

[58] E.F. Codd, R.S. Arnold, J.M. Cadiou, C.L. Chang, and N. Roussopoulos, Rendezvous: Version 1, Report RJ 2144, IBM Research Laboratory, San Jose, California, January 1978.

[59] E.F. Codd, Seven Steps to Rendezvous, Report RJ 1333, IBM Research Laboratory, San Jose, California, January 1974.

[60] L.R. Harris, User Oriented Database Query with the ROBOT Natural Language Query Sytem, International Journal of Man-Machine Studies, Vol. 9, 1977, pp. 697–713.

[61] D.G. Bobrow, R.M. Kaplan, M. Kay, D.A. Norman, H. Thompson, and T. Winograd, GUS—A Frame-Driven Dialog System, Artificial Intelligence, Vol. 8, No. 2, April 1977, pp. 155–173.

[62] W. Lehnert, Problems in Question Answering, Sixth International Conference on Computational Linguistics, Ottawa, Canada, 1976.

[63] B. Shneiderman, Software Psychology: Human Factors in Computer and Information Systems, Winthrop Publishers, Cambridge, Massachusetts, 1980.

BIBLIOGRAPHIC REMARKS

The following references can be used to obtain an overview of several linguistic techniques used in automatic information processing:

L. Bolc, editor, Natural Language Communication via Computers, Lecture Notes in Computer Science, Springer Verlag, Berlin, 1978.
R. Rustin, editor, Natural Language Processing, Courant Computer Science Symposium No. 8, Algorithmics Press, New York, 1973.

The following references provide a deeper view of various methods currently used for knowledge representation:

D.G. Bobrow and A. Collins, editors, Representation and Understanding, Academic Press, New York, 1975.
N.V. Findler, editor, Associative Networks—Representation and Use of Knowledge by Computers, Academic Press, New York, 1979.
R.C. Schank and R.P. Abelson, Scripts, Plans, Goals and Understanding, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1977.

Problems arising in interfacing linguistic techniques and information retrieval are covered in the following references:

K. Sparck Jones and M. Kay, Linguistics and Information Science, Academic Press, New York, 1973.
D.E. Walker, H. Karlgren, and M. Kay, editors, Natural Language in Information Science, FID Publication 551, Skriptor, Stockholm, 1977.

EXERCISES

7-1 In Chapter 4, four different grammatical interpretations were given for the sentence, "Time flies like an arrow."

- **a** List the various facts of world knowledge that appear to be needed in a language analysis system capable of recognizing the sample sentence.
- **b** Assuming that a context-free grammar is used for recognition purposes, generate a set of semantic rules based on the stored knowledge that are capable of eliminating the extraneous analyses (leaving only the correct interpretation) when added to the normal context-free recognition process.

7-2 Explain the similarities and differences between a context-free grammar, a context-sensitive grammar, a transformational grammar, and an ATN grammar. What do the initials ATN stand for? Explain the function of the concepts represented by the A, T, and N, respectively.

7-3 Consider a TN grammar capable of recognizing character strings consisting of A's and B's only. Generate a TN grammar that accepts all input strings exhibiting an equal number of A's and B's regardless of the ordering of the letters, and rejects all other strings. Thus AABB, ABAB, the null string, etc., would be accepted; however, AAB, ABABC, A, etc., would be rejected.

7-4 Consider the following segment of a phrase structure grammar for English:

(1) $S \rightarrow NP + VP$
(2) $VP \rightarrow V$
(3) $NP \rightarrow DET + ADJ^* + N$

The asterisk * following ADJ indicates that a variable number of adjectives are acceptable.

a Draw a transition network corresponding to the three rules.
b Show the individual steps needed to recognize the sentence "The big gray hippo wallows."
c How can the grammar and network be extended to accept the sentences "The big gray hippo wallows mightily" and "The big gray hippo mightily wallows"?
d Illustrate the recognition of one of the earlier sentences by giving the phrase markers corresponding to each step of the recognition process.

7-5 Consider the first two paragraphs of the current chapter.

a Choose indexing phrases (noun phrases, prepositional phrases) to represent the content of these paragraphs.
b Generate a set of criterion phrases together with the corresponding thesaurus (as in Fig. 7-12) to represent the phrases specified under part a.
c Specify the noun-phrase portion of a context-free grammar capable of recognizing the corresponding phrases in conjunction with the criterion phrase dictionary of part b.
d How would you extend the criterion phrase dictionary and the grammar to handle the content of the third paragraph of this chapter, in addition to the first two paragraphs? What does this imply about the ease of extending limited linguistic recognition systems to wider subject areas?