QUESTIONS AND INFORMATION SYSTEMS

Edited by

Thomas W. Lauer
Eileen Peacock
Oakland University

Arthur C. Graesser
Memphis State University

LAWRENCE ERLBAUM ASSOCIATES, PUBLISHERS
1992 Hillsdale, New Jersey Hove and London

10 Interpretation of Textual Queries Using a Cognitive Model

Kathleen Dahlgren
Intelligent Text Processing, Inc., Santa Monica, CA

The volume of machine-readable text is growing exponentially. In news media, the government, medicine, law, and other fields, machine-readable texts or abstracts have been stockpiled for decades. Wire and news services make text available in real-time across worldwide networks. This information explosion has reached proportions in which institutions and busy professionals are unable to read and incorporate the unanalyzed glut of news, memos, and articles. This chapter explores the problem of computationally asking questions about text and of text. Current text retrieval systems are based on pattern matching of key words to the text words. However, these methods have proven to be inaccurate, with hit rates as low as 20% recall (Blair & Maron, 1985). Experienced users express dissatisfaction with these offerings in several areas: the types of questions that can be asked, the representation language of the query, and the relevance of replies. In the typical key word-based text retrieval system, the user begins a session with few key words and is inundated with irrelevant references. As the user adds more key words, the number of documents that contain all of the key words very rapidly declines to zero in a "cross-section" effect.

This chapter describes a text retrieval system, Interpretext. Interpretext was built by the author and colleagues who were originally at IBM (Dahlgren, McDowell, & Stabler, 1989; Lord & Dahlgren, 1989) and at Intelligent Text Processing. The goal of the system is to improve dramatically on existing text retrieval performance, both in user-friendliness and in accuracy, by introducing the findings of modern linguistics and cognitive science into software algorithms. Planned applications range from the less challenging tasks of text selection (archiving or clipping service) and text routing (real-time delivery of messages to all and only the interested parties), to the quite difficult task of text querying (retrieving facts from text). Text querying in another guise can be enhanced to perform textual data base updating, in which a program reads text and maintains a standard format data base.

Picture the busy professional, such as an official who digests world political events as they unfold and ensures that appropriate superiors are informed of important events to which the government must respond. The present scenario for such an official is that each morning he or she sits down to a desk full of printed pages of news and message items produced by a key word-based text retrieval system. Two-thirds of the stack is irrelevant to topic domain and must be tossed. Similarly, throughout the day the official weeds out irrelevant incoming clippings. For example, suppose the domain is Latin American terrorism. The domain is defined in a profile of key words intended to cull from the incoming news and messages those that concern Latin American terrorism. Unfortunately, the text retrieval program clips many irrelevant articles because they contain the words *Latin*, *American* and *terrorism*. On the other hand, relevant and important articles are missed because they do not happen to contain exactly those words. Unlike humans, the program cannot recognize that the mention of any Latin American country, major city, or government official classifies a news report as about Latin America.

Enhancements of such systems include the addition of more key words in boolean expressions, that is, those which include both "ANDing" and "ORing" of key words. In our example, the official might develop an "ORed" list of all Latin American countries, capitals, and major government officials, in effect saying that any article that contains the expressions *Latin America* OR *Mexico* OR *Rio de Janeiro* is relevant to our official's profile. This list could be "ANDed" with the expression *terrorism*, yielding a profile that culls all articles with both an expression from the Latin America list and the term *terrorism*. One of the known problems with this enhancement is that even highly educated professionals find working with booleans difficult and unnatural. One solution has been the introduction of knowledge engineers or text librarians as intermediaries who assist in accessing the data base. At one installation, 65 knowledge engineers are employed to formulate queries to a text retrieval system. Another enhancement has been the introduction of weightings to the key word queries, as in Verity. However, this solution again requires software engineers.

Interpretext addresses these problems in three significant ways. First, the interpretation of the incoming *target* text uses several levels of linguistic analysis to produce "deep natural language understanding," that is, the most refined level of computer computational understanding possible given the limitations of modern linguistics and computer science. Secondly, the query or profile is stated in English, making the system much more easily and directly usable by the busy professional. Thirdly, the system employs a cognitive model of both the query and the target text. The cognitive model is constructed by drawing on naive semantics, a representation of the world knowledge people use to disambiguate text.

The first section of the chapter describes the particular linguistic theory employed in Interpretext. The second section runs through the program architecture, which takes English as input and translates it several times, applying several different levels of linguistic analysis. The third section takes up illustrative examples of linguistic analysis at a few of the levels, and demonstrates their importance in achieving accurate and relevant replies to queries. The chapter briefly sketches the treatment of such problems in alternative systems.

COGNITIVE MODELS IN INTERPRETEXT

The particular approach in Interpretext is a model of natural language interpretation that incorporates all levels of linguistic processing that are known to exist in text interpretation by people: syntax, formal semantics, and world knowledge. The model addresses the interface between formal structures in language interpretation and memory structure (or the content of language). Other approaches are narrower. Either they can handle parsing (just syntactic form) OR world knowledge (just content, not form), as in Schank's (1982) conceptual dependencies and MOPs, OR formal syntax and formal semantics (just form, no content), as in the SRI approach (Shieber, van Noord, Moore, & Pereira, 1989). A representation of content requires a cognitive model that represents more than syntactic structure (Johnson-Laird, 1983; Morrow, Greenspan, & Bower, 1987). A cognitive model reflects both an interpretation of the explicit linguistic form and those inferences that must be drawn in order to interpret the form. The text in Table 10.1a is used to illustrate the claims of the chapter. In Table 10.1a, the reader's interpretation includes not only the structural assignment of *the U.N.* to subject of the first sentence constituent (S), but also an understanding that the U.N. is said to have accused Guatemala. Evidence that this inference is a required element of the interpretation lies in the next sentence, which begins with the pronoun *they*. To determine that *they* in the second sentence has *the U.N.* as antecedent rather than Guatemala, the reader must understand that both charging and citing are types of accusing, and that the citing activity was probably done as part of the charging activity. The reader can then infer that both activities have the same agent—*the U.N.* This is the kind of inferential information included in the cognitive model and not in structure-only models. Cognitive models are distinctly different from and more powerful than simple parse trees on the one hand, and semantic net representations on the other. The cognitive model is built on the output of the parser in the following levels:

1. Syntax (with syntactic disambiguation);
2. Sentential Formal Semantics;
3. Discourse Formal Semantics;
4. Cognitive Model.

TABLE 10.1
Sample Text and Levels of Representation

a. English Text: Guatemala was charged today by the U.N. They cited an illegal attack on a newsman.

b. Parse: s(np(det(the) & (un) & vp (v(charge) & np(n(guatemala))) & adv (today))

c. DRS for one sentence: *el, un, g*

charge6 (*el, un, g*)
today (*el*)

d. DRS for text: *el, un, g, e2, they1, e3, al*

charge6 (*el, un, g*)
today (*el*)
cite (*e2, they1, e3*)
attack2 (*e3*)
illegal (*e3*)
on (*e3, al*)
newsman (*al*)
they1 = un

e. Cognitive preference: constituency (*e1, e2*)

f. First Order Logic: ∃*el*(charge6 (*el, un, g*) & today (*el*)

g. Prolog: charge6 (*el, un, g*)

The Interpretext system translates the cognitive model further for processing purposes as follows:

5. First Order Logic;
6. Prolog;
7. Reasoner;
8. Relevance, Problem Solver, or Data base Update.

The ensuing subsections outline the analysis of Table 10.1a at each of the levels (1–8).

Parse

A parse tree for the first sentence analyzes the grammatical parts of the sentence and their roles relative to each other. Although the surface order of the words has *Guatemala* as subject, the simplified parse in Fig. 10.1 shows that the deep grammatical subject is *the U.N.* The serial form of such a parse looks like Table 10.1b.

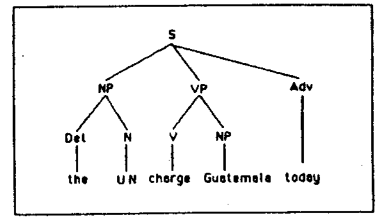

FIG. 10.1. A simplified parse tree.

Sentential Formal Semantics

A parse captures only the grammatical structure of a sentence. Another level of information conveyed by the form of linguistic elements are truth conditions. The formal semantic level provides a formula expressing the conditions in the actual world (or some possible world) that would have to exist in order for a sentence to be true. In the first sentence of Table 10.1a, there are entities (*the U.N., Guatemala*), an event (*of charging*), and a relation between them (*the U.N.* and *Guatemala* stand in a "charging" relation). A translation function based on Discourse Representation Theory of Kamp (1981) and Asher (1987) produces from the parse the Discourse Representation Structure (DRS) in Table 10.1c. The entities in the sentence are listed in the top part of the box. Each of these denotes an individual or set of individuals, an event or set of events in a world model. It might be noted that the world model is represented as an entity-attribute data base. The lower part of the box contains predictions on these entities and the relations among them. Thus, we see that event *el* is the one in which entity *un* (the U.N.) relates to entity *g* (Guatemala) as the charger in a charging event in which entity *g* is the chargee. Similarly, we find that event *el* occurred today (the day the text was written).

Discourse Formal Semantics

This level represents the truth conditional properties of the entire discourse. It tracks and equates entities across text as they are mentioned over and over again (this is called *anaphor resolution*). The DRS for both sentences of Table 10.1a contains the resolution of the anaphoric device *they* and shows that it corefers with *U.N.* to the entity *un*. This identity is expressed in the equation *they1=un*, as in Table 10.1d. Notice that entities (or instantiations), and sorts of things like *newsman* are distinguished in the DRS, as needed for knowledge-base reasoning (Brachman & Schmolze, 1985). Table 10.1d omits temporal equations also provided in the DRS, which indicate the time relations between events. In Table 10.1d, the equations would indicate that events *el* and *e2* overlap in time, as in Partee (1984).

The DRS provides a formal representation of the events described in the text, and a method for assigning truth conditions to the assertions of the text. In other words, given the first order representations in the DRS, whether the world conformed on that day to the assertions in the text can be checked by the application of formal logical rules to a model produced by a mapping from the DRS. If entities in the model of the actual world had the relations expressed in the DRS, then the assertions in the text are true. This level of representation is very important in computational text understanding, especially for the ultimate goal of extracting data base knowledge from text. The DRS is translatable into first order logic and then into a programming language or computational knowledge base. The assertions in the text can then be checked against data bases or used to update them, which is exactly the knowledge acquisition procedure in Interpretext.

Formal semantic properties of natural language include the capability of expressing unqualified assertions, such as "The guerrillas attacked the base." In addition, assertions can be qualified in a large number of ways. Models affect the force of assertions, as in "The guerrillas might attack the base." The latter sentence does not imply a world model in which the base has actually been attacked. Operators such as negation reverse the force of an assertion, as in "The guerrillas did not attack the base." Formal semantic properties such as these are treated in the Interpretext program, but will not be explored in depth in this chapter.

Cognitive Model

Another level of representation in the theory is the *cognitive model*, so called because it infers the intended content from what the text says directly and because it incorporates world knowledge. The clauses of a text are written with the intention that the reader draw some relationship between them. For purposes of economy, language is both highly ambiguous and highly telescopic. A given form of words in English can make a large number of different assertions, depending on the situation being referred to (Barwise & Perry, 1983). This ambiguity of the English language requires that the human interpreter reason about the likely situation to which the writer might be referring. So a mandatory aspect of text interpretation for humans is drawing inferences about the connections between the contents of individual sentences. The sentences in (1a) describe a likely situation, whereas those in (1b) do not. Furthermore, the second sentence in (1a) disambiguates the first sentence, which has a possible interpretation concerning finances rather than legalities.

(1a) "Guatemala was charged today by the U.N. They cited an illegal attack on a newsman."

(1b) "Guatemala was charged today by the U.N. They ate oranges."

The two events are unrelated. The reader cannot even decide whether "they" corefers with Guatemala or the U.N. In contrast, with (1a) the reader uses knowledge that charging and citing are related events to infer that the U.N. is the grammatical subject of both *cite* and *charge*. Furthermore, the human reader guesses that the charging event is the broader one, and that part of it was a subevent of citing. The representation of the cognitive model for (1a) as built by Interpretext is pictured in Table 10.1d, with the added inference 1e that the two events of citing and charging cohere under the relation "constituency." This inference states that the citing event *e2* occurred as part of the charging event *e1*.

Other inferences are added to the cognitive model to break longer texts into coherent segments (not paragraphs, which may or may not coincide with coherent segment breaks). In addition, the cognitive model contains an indication of which event is the topic event of the text, and which are the main participants in the topic event (see Dahlgren, 1989). In summary, the cognitive model contains a list of discourse entities (individuals and events) and a list of predicates. The latter represent: properties of the discourse entities; relations between those entities directly expressed in the text; temporal equations; coherence relations between events inferred from both the text and world knowledge; and textual structure (topic and segments).

First-Order Logic

The cognitive model in Interpretext is translated into first-order logic formulas. First-order logic has the important advantage, as a representational language, that its proof properties have been well-known for a century. Translation of English to first-order logic results in a representation to which standard proof methods can be applied. In contrast, translation to a special knowledge representation language requires special reasoning modules that become large, slow, and unpredictable in their behavior. The first-order logic form for the first sentence of Table 10.1a is shown in Table 10.1f. This means "there is an event *e1* such that the individual *un* (in the world model) charged the individual *g* (in the world model) during the event *e1*, and *e1* occurred today." The cognitive model is translated to first-order logic.

Prolog

The first-order logic form is an intermediate representation between the cognitive model and Prolog. First-order forms contains quantifiers $\exists$ (there exists) and $\forall$ (for all). The translation eliminates these because they introduce infinite chains of reasoning. The translation function transforms first-order logic (e.g., Table 10.1f) into Prolog (e.g., Table 10.1g). The Prolog form is the data base used by the reasoning module to answer questions. The Reasoner and higher level modules are explored in the Architecture section.

NAIVE SEMANTICS

The approach to lexical knowledge in Interpretext is called Naive Semantics (NS) and is fully described and defended in Dahlgren (1988). NS is fundamental to the construction of the cognitive model. It is based on the observation that a reader draws on world knowledge to disambiguate and clarify text, selecting the most plausible interpretation from among the infinitely many possible ones. Although extensive world knowledge can, in principle, affect the choice of an interpretation, NS is just the shallow layer of knowledge speakers must use in production and interpretation of language. The nature of the required shallow layer of knowledge is carefully and projectibly defined. Lacking this knowledge, they are incompetent in the language. Operationally, NS knowledge consists of characteristics of objects and the implications of events produced by subjects in psycholinguistic experiments.

An informal example of a naive semantic concept is the following description of the typical lawyer, drawn from a psychological study of concepts (Dahlgren, 1985).

> If someone is a lawyer, typically they are well-dressed, use paper, books and brief cases in their job, have a high income and high status. They are well-educated, clever, articulate, knowledgeable, as well as contentious, aggressive, and ambitious. Inherently lawyers are adults, have gone to law school, and have passed the bar. They practice law, argue cases, advise clients, and represent them in court. Conversely, if someone has these features, he/she probably is a lawyer. (p. 57)

Concepts are called naive in NS because they are not always objectively true and bear only a distant relation to scientific theories. They are probabilistic rather than formulaic in force. In other words, if the meaning of *lawyer* is that a lawyer typically negotiates settlements, then a sentence "John is a lawyer" only probabilistically implies that John negotiates settlements. The success of communication with a vague and ambiguous vehicle is explained by the fact that natural language is anchored in the real world. Real objects and real events are referred to using words that have meaning representations that are close enough to the truth, enough of the time, to make reference possible (Boyd, 1986). The stability of the extensional world is also what makes NS representations portable to other languages. A car is a car in the U.S. and Japan. People do not have radically different naive views of them in modern industrial societies.

NS differs from approaches that employ exhaustive decompositions into primitive concepts. In these decompositional approaches, a term such as *lawyer* has a

meaning representation like *adult* and *human* and *counselor*. All and only those objects that have those features are lawyers. Thus, the meaning representation of a word forms a set of conditions which all members of the set of objects which the word names share. This is essentially the approach in many knowledge representation schemes such as KL-ONE (Brachman & Schmolze, 1985) and AI approaches to natural language (Schank, 1982). In contrast, NS representations are not limited to a set of primitives, but contain many English words as feature values. Furthermore, the feature values correspond to naive theories about the nature of the world, which may or may not be correct. A meaning representation in NS consists of those properties of things that most believe to be true. The result is richer representations with more content, which are not treated as logical formulas (which are true of all objects named by a word). For example, in the representation of "lawyer" a feature "function, defends clients in court" appears. Although this property is typical of lawyers, it is not a logical necessity that if someone is a lawyer he or he must defend clients in court. Some lawyers do not. The result is that NS representations of word meaning contain more information than is found in alternative approaches.

NS representations are of two types, ontological and generic. Ontological classes reflect naive beliefs concerning the structure of the actual world and the significant "joints" in that structure, such as animate versus inanimate and real versus abstract. A piece of this knowledge is illustrated in Fig. 2. This piece of the ontology shows that "lawyer" is a social role. In many AI and psychological theories this is called a taxonomy or semantic net.

Generic knowledge contains features of objects and implications of events, the knowledge that a generation of psycholinguistic studies has shown to be the cognitive structure of nouns and verbs (Graesser & Clark, 1985; Rosch & Mervis, 1975). Generic knowledge is represented as two lists, one of inherent and one of typical features. For the knowledge base, these get translated into indi-

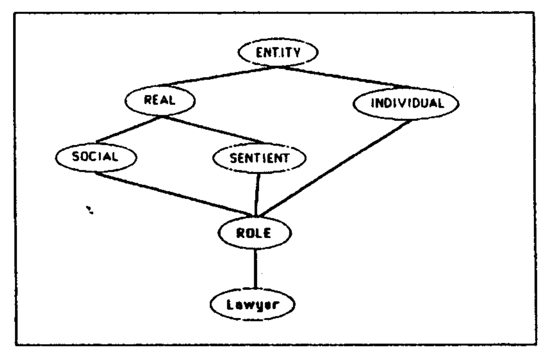

FIG. 10.2. A piece of the ontology.

vidual ground clauses in Prolog. Each feature is typed. For nouns there are 59 types: size, shape, color, function, and so on. Part of the representation for "lawyer" looks as follows:

lawyer	
typically -	internal_trait(articulate), function(defend(noun,Y) & client(Y)), income(high), function(negotiate)
inherently -	education(school(X) & law(X)), function(represent(noun,Y) & client(Y))

For verbs there are very few types: cause, goal, enable, consequence, what happened next, where, when, how, implies, and selection restriction. Alternative senses of ambiguous words have different entries, indexed by sense numbers.

Generic knowledge for nouns and verbs are collected using a computer program that asks a native speaker of American English intuitively to fill values for feature types. The feature types for a noun are narrowed to those that have been established empirically to be found under a given node in the ontology. For example, under the Social Role node of the ontology, several feature types are applicable. A knowledge engineer is prompted for these, as illustrated in (2). The values are some of those for "lawyer."

(2)

Feature type	Feature value
Function	negotiate
Function	represent(X,Y) & client(Y)
Education	school(X) & law(X)
Income	high
Internal trait	articulate

NS can be encoded easily for computational purposes because it is limited to a relatively small number of types of features (though feature values are unlimited), and nonprofessionals can be taught to encode it. Such entries are translated by computer program into a Prolog knowledge base, so that each feature for a word has a format like (3) in Interpretext.

(3) lawyer(income,high,typical).

ARCHITECTURE AND OPERATION

Interpretext reads text or queries as input and processes both of these with the same modules until the very last component (the reasoner) is executed. Figure 10.3 shows the flow. Taking text Table 10.1a as an example input, the parser first produces a labeled bracketing as in Table 10.1b for the first sentence. The

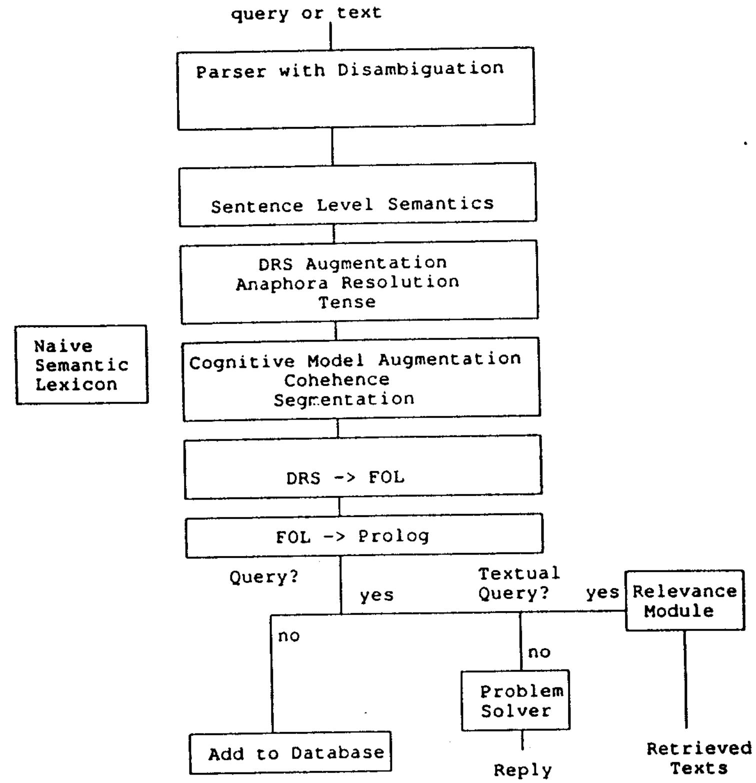

FIG. 10.3. Flow chart of interpretext.

disambiguation step of the parsing disambiguates the parse tree structurally (which is not necessary for this sentence). It also selects from among the available senses for each noun and verb. In the first sentence of Table 10.1a, the legal sense of "charge" is chosen, rather than the monetary or physical senses. Then the DRT module (Sentence Level Semantics) builds a DRS from the parse tree as in Table 10.1c. At this point, DRS augmentation and cognitive model augmentation are irrelevant because they are discourse-level modules and not enough sentences have been received. Next, the DRS to first-order logic translator produce Table 10.1f, and the first-order logic to Prolog translator produces Table 10.1g. At this point, the reasoning module is not called because not all of the input has been processed. Instead, the interpretation process begins anew with parsing for the second sentence in the incoming text. It is processed through all of the levels in exactly the same way as the first sentence up until anaphor resolution in the DRS augmentation module.

For the second sentence, discourse-level processing is necessary. DRS augmentation first invokes anaphor resolution, which uses both syntactic rules (such as number agreement) and world knowledge (such as the relatedness of charging and citing), to infer that the antecedent of *they1* is *the U.N.* Next, it invokes temporal reasoning to produce tense equations reflecting both that the charging (*e1*) happened the same day as the text was written, and that the citing (*e2*) happened during a time interval included in the time of *e1*. The Reasoning module is invoked and inspects whether this is text or query. Finding that it is text, it invokes the data base updating module, which adds Table 10.1g to the Prolog workspace. This is available to answer questions such as "What did the U.N. do?"; "What did Guatemala do?"; "When did the U.N. charge Guatemala?" For such queries, again the input is parsed and translated as for the first sentence of Table 10.1a. When the translation reaches the first-order logic to Prolog module, it is converted into a Prolog command to query the data base, and the answer is produced by the Problem Solver module. If the query is a text retrieval query, the Relevance module is invoked. In this case, it determines precisely that Table 10.1a is about the U.N.'s act of charging Guatemala with illegal attack, rather than about heart attacks, monetary fees, or book citations, all of which would be possibilities for key-word-based systems. At each stage of the analysis, the system is forced to choose among possible structures and interpretations in accordance with preference strategies (i.e., it selects which one is most plausible or typical).

LEVELS OF INTERPRETATION

Turning to a defense of deep natural language understanding, a number of major figures in the field, most notably Fodor (1987), have claimed that a rich, open-ended approach such as NS is logically infeasible. Happily, the prototyped Interpretext demonstrates the model's feasibility. Others argue that any sort of linguistic analysis requires too much reliance on inadequate theories and hand-encoding of representations, so that the only answer is to treat language as a massive pattern like visual or audio input (Streeter, 1990). In this section, I demonstrate the value and feasibility of multilevel linguistic analysis of text by examples from several levels of analysis. I explore parsing, structural disambiguation, word sense disambiguation, synonymy, anaphor resolution, and reasoning for relevance in order to illustrate their importance in achieving precise answers to questions.

Parsing

Parsing is needed to translate the surface string of words into the argument structure of the query or text. In queries such as (4) and (5), a system needs the syntactic fact that *Guatemala* is object rather than subject of the sentence. It can

thereby look for instances of Guatemala being invaded, rather than of Guatemala invading some other country.

(4) "Who invaded Guatemala?"
(5) "Did the U.S. invade Guatemala?"

Similarly, in trying to answer a query about murder of government officials, a system with no parsing can be fooled into selecting any of the following texts as relevant:

(6) "The CIA murdered officials of the government."
(7) "The government murdered officials."
(8) "The government has made murder an official policy."
(9) "The government tried the official for murder."
(10) "Government officials charged the death squads with murder."

In contrast, a system with parsing converges only on (6), because *officials* is the object of *murder*, and *officials* is modified by *government*.

Thousands of English words have syntactic roles of both noun and verb, and have different meanings in these roles. One product of parsing is that it assigns each word in the string to a part of speech. When a query is parsed, the correct thematic role for each word must be assigned. Thus *charge* is a verb in (11), yet a noun in (12). Only (11) is relevant to a query such as (13).

(11) "Guatemala charges that the U.N. interfered."
(12) "The charges against Guatemala were dropped."
(13) "What did Guatemala charge?"

Structural Disambiguation

Natural language structure is highly ambiguous, so the parse output is also ambiguous (or, alternatively, for each sentence many parses must be produced). Structural ambiguities arise in sentences with prepositional phrases (PPs) after the object of the verb because the first PP after the object can modify either the object, the verb, or the sentence constituent. Such sentences have at least three readings due to structure alone. Consider this structure in (14). The three readings are illustrated with different words after the preposition.

(14) "Guerrillas attacked the outpost with (a) a sentry
(b) grenades.
(c) fury.

In (14a), the PP *with a sentry* modifies *the outpost;* the outpost is the one with a sentry. In (14b), *with grenades* modifies *attack;* the attacking had grenades as instrument. In (14c), *with fury* modifies the whole sentence; the guerrilla attack was done in a furious manner. These structural alternatives are illustrated in Fig. 10.4. If the structural ambiguity is not resolved, a system cannot accurately answer questions. For example, the answer to "Which outpost did the guerrillas attack?" is found in (14a) and not (14b) or (14c), whereas the answer to "What did the guerrillas use to attack?" is in (14b) and not (14a) or (14c). In sentences with two PPs after the object of the verb, the number of possible structures rises to seven. Computational linguists are fond of generating examples that illustrate exponential growth in the number of readings for sentences in which various structural ambiguities interact. Apparently simple sentences can have hundreds of potential parse trees. A computational system that tries to interpret text without human intervention must apply "preference strategies" to select among the trees. Preference in this phrase alludes to the fact that lacking a human reasoning capacity, the program must guess and try to come as close as possible to the selection of the correct reading with whatever information and algorithms it has at its disposal.

We have developed a computational method for PP disambiguation using preposition-specific rules, syntax, and naive semantics (Dahlgren & McDowell, 1986). The algorithm was found to be 97% accurate in a corpus of 12,000 words of geography and newspaper text. To illustrate its function, consider example (14b). In this case, global rules are tried and fail, such as a rule that assigns S-attachment to PPs with temporal nouns, and VP-attachment to motion verbs with certain prepositions. Then the preposition-specific "with" rule is tried and it attempts to prove VP-attachment by checking whether in the NS lexicon the head of the PP is a typical instrument of the verb *attack*. In this case, it is, so the parse is reformed with the PP attached to the VP. In (14a), the "with" rule checks in the NS lexicon to see whether a typical part of an outpost is a sentry, and finds that it is, so it assigns the "with" phase as a modifier of the object NP *the outpost*. In example (14c), none of the specific cases in the "with" rule applies, so the default is taken. The default assigns the "with" phrase as a sentence-

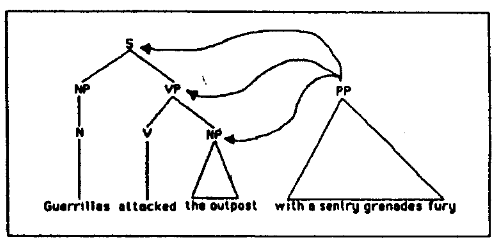

FIG. 10.4. Possible attachment sites for a prepositional phrase.

modifier. The algorithm also records the case role, by replacing the preposition with *instrument* for (14b) and *manner* for (14c).

In sentences with both PPs and ambiguous words, the algorithm disambiguates words in relation to the structure it selects. In (14b) the verb *attack* has several different senses. Two of them are "attack physically" and "attack verbally." The physical sense is the one that has *grenades* as instrument, not the verbal sense. Thus, the algorithm selects the physical reading for *attack* and puts the word-sense index on it as part of the processing. When the word-sense disambiguation module processes the sentence, it ignores previously disambiguated words.

Another type of structural ambiguity arises with conjunction. Conjoined noun phrases and verb phrases allow for a variety of interpretations. Consider (15) versus (16). In (15), the most probable interpretation is one in which there is one event of driving to Guatemala, and John and Mary do it together. In (16), such a collective interpretation is plausible, but equally plausible is a distributive interpretation in which there are two separate events of visiting Guatemala, one by John and another by Mary. Verb phrase conjunctions also introduce ambiguity, as in (17). A plausible interpretation has one wife for each man, but an allowable interpretation has only one woman that both love.

(15) "John and Mary drove to Guatemala."
(16) "John and Mary visited Guatemala."
(17) "John loves his wife, and Bill does, too."

Similarly, conjunctions have multiple interpretations in queries. In (18), there may be only one, or there may be two, individuals to identify.

(18) "Who criticized the dancing and praised the singing?"

Therefore, conjunction introduces the problem of determining how many individuals are referred to by a sentence. To resolve these ambiguities, we can assume a tendency to infer collective interpretations, because the conjunction construction implies the formation of sets. Furthermore, NS filtering can be used to discover cases where the distributive interpretation is more plausible. In (15), a joint event of driving somewhere is a typical situation, so that the collective interpretation is preferred. In (19), people typically read books by themselves, so the distributive interpretation is more plausible.

(19) "John and Mary read *Lord of the Flies*."

Conjoined verb phrases also introduce ambiguity, as in (20).

(20) "John ate fish and Mary had veal."

As with conjoined noun phrases, conjoined verb phrases similarly imply a single broader event. In (20) the broader event is a meal that John and Mary had ogether. We are working on the development of a preference strategy for conunction disambiguation using structural probability and NS filtering.

Word-Sense Disambiguation

Ambiguity of word senses is a critical problem in query interpretation and data base searching. Irrelevant replies stemming from ambiguity in the text or query are the source of most of the precision problem in text retrieval. Gentner (1981) reported that the 20 most frequent nouns in English have an average of 7.3 senses, and the 20 most frequent verbs have an average of 12.4 senses. Dahlgren (1987) noticed that the word *hand* has 16 senses, 10 of which are uses in any type of text. The verb *work* has 8 intransitive senses.

Word-sense ambiguity is a problem for question-answering and text retrieval systems, even when vocabulary has been restricted to a specific domain. Suppose that (21) is a target text, and there are three queries to it (22)–(24).

(21) "Garcia moved that the people evacuate to the countryside."
(22) "Did Garcia move the action be taken?"
(23) "Did Garcia move the people to action."
(24) "Did Garcia move the people to the countryside?"

Each of the queries has a different reading for transitive *move:* the parliamentary, the mental, and the physical. Without disambiguation, a query system could easily be fooled into erroneous affirmative responses to both (23) and (24).

The Interpretext word-sense disambiguation method employs three levels of knowledge: frequency, syntax, and naive semantics. The algorithm was developed by studying the syntactic and semantic environments of word senses for 16 words in live text. The system has a list of sense numbers for ambiguous words. Words with no such list are unambiguous.

First, the algorithm tries fixed and frequent phrases that have only one reading, and occur often enough to be recorded: *by hand, for once, in hand, leveraged buyout, human rights,* and so on. If an ambiguous word does not occur in any such phrase, syntactic tests are tried. Some syntactic environments reduce or eliminate ambiguity (e.g., in the case of noun senses being sensitive to the presence or absence of determiners). For example, the electoral sense of *office* is the only sense possible when there is no determiner ("for office," "in office"). Some noun senses are sensitive to the presence of quantifiers. "Many interests" cannot have the financial reading, but may have the sentient or abstract readings. Other noun senses are selected by particular prepositions. "In the office" or "at the office" must have the place reading, not the sentient (institutional) or electoral readings.

For verbs, various syntactic constructions help disambiguate the word, such as certain modals and complement constructions. This is illustrated with several senses of the verb *charge* in (25).

(25)	(a)	physical	to attack
	(b)	physical	to replenish (as electricity)
	(c)	social,financial	to impose a tax or fine
	(d)	social,financial	to debit
	(e)	social	to entrust
	(f)	social, mental	to accuse
	(g)	social, legal	to indict
	(h)	social, mental	to direct

"Someone charges that . . ." can only have the mental reading (25f) for *charge*, as opposed to physical or financial readings. Prepositions may help select senses of verbs. In combination with prepositions, these senses are partially or fully selected. "Charge with" has legal (25g) or financial readings (25c,d), whereas "charge toward" has the physical reading (25a). Some adverbs help select senses of verbs. "Charge heavily" has the electrical (25b), financial (25c,d), or legal readings (25g); "charge vehemently" the verbal reading (25f); "charge continually" the electrical reading (25b); and "charge abruptly" the physical reading (25a).

The third layer of reasoning for word-sense disambiguation is the application of NS knowledge. This knowledge suffices to disambiguate almost all of the cases (98%) that were not disambiguated by either (a) fixed and frequent phrases or (b) syntactic tests (Dahlgren, 1987). In (26), the algorithm is able to select reading (25f). The algorithm uses knowledge that in the accusation reading (25f), typically the head noun of a PP is an event (e.g., *interference*), whereas for the other senses, typical head nouns of PPs are different. For example, typically a "with" phrase after the physical reading (25a) is a weapon. On the other hand, for (27) the algorithm selects the financial sense (25d) because the PP contains a financial word. In (28), sense (25d) is selected over (25c) because typical subjects of that reading are financial institutions, and typical objects are assets. The two financial senses are distinguished by subject selectional restrictions in the verb generic knowledge. Typical subjects of sense (25c) are governments or government officials, whereas typical subjects of sense (25d) are individual persons, merchant roles, or financial institutions.

(26) "The U.S. charged Guatemala with interference."
(27) "The U.S. charged Guatemala with interest."
(28) "The bank charged Guatemala interest."

Synonymy

Precision of question-answering systems requires mechanisms for recognizing synonymy. Some systems for text retrieval address this through concept clustering. Sets of vocabulary words that are found in texts on similar topics are identified. These sets of synonyms are treated as though they were the same word in the retrieval task. However, this statistical method ignores word-sense differences. Returning to our verb *charge*, only sense (25b) is quasi-synonymous with *electrify*, *energize*, and *power up*. Only in sense (25a) is *charge* quasi-synonymous with *attack* and *set upon*. When query (29) is applied to textual data base (30), the relevance of (30) is recognizable only in relation to disambiguated *charge*, because only in that reading is it synonymous with *accuse*.

(29) "Who charged Guatemala with violations?"
(30) "The U.S. accused Guatemala of abrogations of human rights."

Anaphor Resolution

Pronouns, demonstratives, or definite noun phrases often need to be equated with some entity mentioned previously in the text. Texts with such anaphoric expressions are uninterpretable without anaphor resolution. In (31) the pronoun *they* must be correctly resolved before (32) can be answered.

(31) "Guatemala criticized the U.S., and they responded by cutting off aid."
(32) "What did the U.S. do to Guatemala?"

Pronouns refer to events as well as individuals. In (33), the antecedent of *it* is the event of dropping a bomb on the village. An algorithm must make this assignment despite the presence of three other potential antecedents in the sentence which match "it" in gender and number (Guatemala, bomb, village). The correct selection is required for answering the question in (34).

(33) "Guatemala dropped a bomb on the village. It shocked Latin America."
(34) "What shocked Latin America?"

Pronouns are important for series of queries. Otherwise the user is forced to repeat names and full NP's, which is experienced as awkward and redundant. This is illustrated in the query session in (35).

(35) "What did the President do yesterday?"
—"He made an announcement."
"Who is the ambassador to Guatemala?"
—"White."
"Did he make the announcement with him?"
—"Yes."
"Where did he make it?"
—"On television."

For Interpretext we have built an anaphor resolution method that combines agreement criteria, syntactic and structural factors, and NS filters to resolve anaphors (Dahlgren, Lord & McDowell, 1990; Lord & Dahlgren, 1989). The input to the anaphor resolution algorithm is the DRS, as described in the Architecture section. The DRS directly and recursively exhibits which entities in the prior discourse are available as antecedents. The details of this formalism and the theory behind it are far beyond the scope of this chapter. Suffice it to say that the anaphor resolution module applies rules to narrow an incoming list of available antecedent using the factors previously listed until it converges on a best guess for the antecedent of an anaphor. The familiar agreement criteria are number, gender, and animacy. With naive semantics, the algorithm has knowledge that certain animals can corefer with *he*, *she*, or *it*. There are two syntactic tests. One makes sure that nonreflexive pronouns in most syntactic positions do not find their antecedent in the same clause. For example, in "The Guatemalans criticized them," the antecedent of *them* cannot be *Guatemalans*. The second syntactic test guarantees that if a pronoun is in the subject position, antecedents that are in subject positions are preferred. NS filters test antecedents for contextual plausibility. In (36), both the President and his colleague satisfy agreement tests for *he*.

(36) "The Ambassador stopped to see the President, who was sitting at his desk. He rose and greeted him."

But NS knowledge of *sit* has an implicational feature that "what happens next" is typically to *rise* or *stand up*. Structure prefers the Ambassador over the President. However, naive semantic filters indicate that the President rose, after sitting, so President is selected as antecedent of *he*.

Relevance

Precision has eluded existing text-retrieval systems. They return from 35% to 80% irrelevant references (Blair & Maron, 1985). In the text-retrieval task, the objective is to find all and only those texts in a target data base or a stream of incoming messages that are *relevant* to a query. Existing key-word-based systems such as Verity and enhancement from Thinking Machines have achieved success in ordering a file of texts (or a series of messages) in a relevance ranking. A file of 100 texts is ranked from 1 to 100 for relevance to the query. In a process called *relevance feedback* the user selects one of the high-ranking texts as most relevant to the query, and the system processes entire text as though it were a query to order more accurately the remaining texts. The second ranked list tends to more accurately reflect the user's interests.

A full cognitive model makes possible a significant improvement in precision because it builds a more precise representation at each level of linguistic analysis. Interpretext is intended to process queries stated in English rather than key words, and to return the short, accurate list of all and only the articles that are relevant without a feedback step. Another interesting possibility would be to use the sophisticated and computation-heavy reasoning of Interpretext to select from a rank-ordered list produced by a key-word-based system. Returning to our state department official in the introduction, suppose the task is to match a profile that describes the official's domain of interest, as in (37).

(37) "I want to read about acts of terrorism in Latin America perpetrated either by governments or guerrillas, but not by armies."

Interpretext has the immediate advantage over competitors because it can read English queries or profiles. Thus, the relevance task becomes one of comparing two Prolog data bases, the query and the target texts or messages. There are three elements of relevance reasoning that Interpretext can handle in matching target texts to this query:

1. Concept Clustering;
2. Ontological Similarity; and
3. Topic Extraction.

Each of these is sketched in the following. In each case, the NS lexicon and cognitive model make possible very precise forms of relevance reasoning.

Concept Clustering

A number of systems have shown that valuable data structure in text retrieval algorithms is the concept cluster or synonymy group. Used alone, the concept cluster has the same problems of precision and recall as other key-word-based reasoning. However, it can be a valuable filter for higher-level relevance reasoning. Interpretext's Word-Sense Disambiguation module makes more precise because it factors out irrelevant word senses. Furthermore, NS representations yield concept clusters for free. Very often the associated concepts are mentioned in the NS generic information. For example, the words *borrow, credit, debt1* (the financial sense), *owe, bank2* (the financial sense), *debtor, loan1* (the financial sense), *principal2* (the financial sense), *lend, pay, service3* (the financial sense), and *deposit* all have the others as feature values in the NS generic lexicon. In particular, function features of nouns mention *loan* and *money;* "how" and

"why" features of verbs mention *money, owe,* and *have.* Thus the "financial" concept cluster is fully represented without additional work or analysis, and simply falls out the cognitive approach to word meaning. Similarly, NS knowledge associated with proper names aids concept clustering. In Guatemala we find Latin America and Third World. Clearly, with such relationships encoded, the user does not need to provide Interpretext with list of Latin America terms, as in our introductory scenario.

Ontological Similarity

Ontological similarity refers to relationships in the ontology that makes near-nodes in the lower reaches of the ontology similar to each other. For example, the lower portion under the node "automobile" branches into a subtree with "sedan," "sports car," "four-wheel drive," and so on. Any of the nodes under "automobile" counts as ontologically similar to the others because they are sisters in the ontology. If a text-retrieval system user is interested in texts about automobiles, any of the words *sedan, sports car,* or *four-wheel drive* should count as in the domain of interest of the user. The Interpretext ontology permits the use of one of the ontologically similar words in the query, and all of the others will count equally well in determining the relevance of texts to the query. This is done automatically. The user need not recall all of the similar words of English to form a boolean expression mentioning all of the words that count as automobiles in the texts of interest.

Topic Extraction

In topic extraction, detailed and exhaustive linguistic analysis pays off. The cognitive model of the target text contains several types of information that are important indicators of topic: argument structure, discourse entities and events, and coherence relations. The topic of a text is the event that the text is about, at least for those genres of text that have a topic. We have analyzed a corpus of 20 newspaper articles, a novel, and a number of other texts for topic. The topic of segments of text is identifiable as the event that is most often referred to (i.e., the event for which the discourse entity symbol [e.g., e1] occurs most often). In the DRS, the anaphor resolution process identifies all instances of the same event with each other. For example, in (38) the words *charge, move,* and *it* all corefer to the event of U.N.'s charge.

(38) "The U.N.'s charge came as a shock to the Guatemalan government. The move stunned the President. It causes a flurry of denials."

We have found that the event most often mentioned in any predicate in the cognitive model is the topic event. This is related to the fact that the cognitive model contains coherence relations that connect each of the clauses to some other clause. This assumption is consistent with some psychological studies that have tested the notion that text contains a causal chain in which the most salient event is that which is most related to all others in the text (Black & Bower, 1980; Graesser & Clark, 1985; Kintsch, 1988). Although a counting mechanism seems simplistic, it works because of the psychologically justified layers of analysis that produce the reference markers to be counted. Precisely because the theory and the program attempt to model human linguistic reasoning at all levels, the resulting representation easily performs the task of extracting the topic or gist.

SUMMARY AND CONCLUSIONS

In this chapter I have described a multilevel, deep natural language analysis approach to queries. I have shown that many of the ambiguities that plagued earlier approaches to natural language understanding can be overcome when a shallow layer of world knowledge is encoded and used in disambiguation algorithms. The power of an approach that incorporates the findings of modern linguistics and artificial intelligence was illustrated with examples from several of the levels of analysis. The approach appears to be feasible because it has been implemented as a computational system. All of the components of this analysis are presently prototyped and running in Prolog, with the exception of the temporal equations and the relevance reasoning for queries.

REFERENCES

Asher, N. (1987). A typology for attitude verbs and their anaphoric properties. *Linguistics and Philosophy, 10*, 125–198.

Barwise, J., & Perry, J. (1983). *Situations and attitudes*. Cambridge, MA: MIT Press.

Black, J. B. & Bower, G. H. (1980). Story understanding as problem-solving. *Poetics, 9*, 223–250.

Blair, D. D., & Maron, M. E. (1985). An evaluation of retrieval effectiveness for a full-text document retrieval system. *Communications of the Association of Computer Machinery, 28*, 289–299.

Boyd, R. (1986). Natural kinds, homeostasis and the limits of essentialism. Unpublished manuscript.

Brachman, R. J., & Schmolze, J. G. (1985). An overview of the KL-One knowledge representation system. *Cognitive Science, 9*, 171–216.

Dahlgren, K., & McDowell, J. P. (1986). Using commonsense knowledge to disambiguate prepositional phrase modifiers. *Proceedings American Associations of Artificial Intelligence*. pp. 589–593.

Dahlgren, K. (1985). The cognitive structure of social categories. *Cognitive Science, 9*, 379–398.

Dahlgren, K. (1987). Using commonsense knowledge to disambiguate word senses. In V. Dahl & P. Saint-Dizier (Eds.), *Natural language understanding and logic programming*, II. Amsterdam: North-Holland.

Dahlgren, K. (1988). *Naive semantics for natural language understanding*. Norwell, MA: Kluwer Academic Press.

Dahlgren, K. (1989). Coherence relation assignment. *Proceedings Cognitive Science Society* pp. 588–596.

Dahlgren, K., McDowell, J. P., & Stabler, E. P., Jr. (1989). Knowledge representation for commonsense reasoning with text. *Computational Linguistics, 15*, 149–170.

Dahlgren, K., Lord, C., & McDowell, J. P. (1990). *Lexical Knowledge for Accurate Anaphora Resolution*. Unpublished manuscript.

Fodor, J. (1987). Modules, frames, fridgeons, sleeping dogs, and the music of spheres. In J. Garfield, (Ed.), *Modularity in knowledge representation and natural language understanding* (pp. 1–15). Cambridge, MA: MIT Press.

Gentner, D. (1981). *Some interesting differences between nouns and verbs. Cognition and Brain Theory, 4*, 161–178.

Graesser, A., & Clark, L. (1985). *Structures and procedures of implicit knowledge*. Norwood, NJ: Ablex.

Johnson-Laird, P. N. (1983). *Mental models*. Cambridge, England: Cambridge University Press.

Kamp, H. (1981). *A theory of truth and semantic representation*. In J. Groenendijk, T. Janssen, & M. Stokhof (Eds.), *Formal methods in the study of language*. Mathematisch Centrum, Amsterdam.

Lord, C., & Dahlgren, K. (1989). *Tracking participants and events in newspaper articles*. Unpublished manuscript.

Morrow, D. G., Greenspan, S. L., & Bower, G. H. (1987). Accessibility and situation models in narrative comprehension. *J. Memory and Language, 26*, 165–187.

Partee, B. H. (1984). Nominal and temporal anaphora. *Linguistics and Philosophy, 7*, 243–286.

Rosch, E., & Mervis, C. (1975). Family resemblances. *Cognitive Psychology, 7*, 573–605.

Schank, R. (1982). Reminding and memory organization: An introduction to MOPs. In W. Lehnert & M. Ringle (Eds.), *Strategies for natural language processing* (pp. 120–156). Hillsdale, NJ: Lawrence Erlbaum Associates.

Shieber, S., van Noord, G., Moore, R., & Pereira, F. C. N. (1989). A semantic-head-driven generation algorithm for unification-based formalisms. *Proceedings Association of Computational Linguistics*. 7–17.

Streeter, L. (1990). Learning a lesson from the law of large numbers. *Proceedings American Association Artificial Intelligence Symposium on Text-Based Intelligent Systems*. 91–92.