

12

Answering Questions About Information in Databases

Arthur C. Graesser
Paul J. Byrne
Michael L. Behrens
Memphis State University

When individuals answer questions, they tap multiple sources of information that embody world knowledge. We assume that each information source is a structured database containing content *nodes* that are connected by relational *arcs*. Structured databases are incorporated in various types of information systems, such as intelligent tutoring systems, expert systems, and decision support systems. We also assume that there are a variety of question-answering strategies that produce good answers by accessing particular information sources and systematically selecting nodes within each information source. A psychological theory of question answering would (at minimum) document the question-answering strategies that are associated with different categories of questions.

Consider a scenario in which an experienced computer user is teaching a novice how to use a computer system. Suppose that the novice asks the following question: "Why do you save your program every 15 minutes?" One strategy of answering this question consists of a logical trace of facts and rules that justifies the action of saving the program. This strategy is precisely the one that many rule-based expert systems execute when "why" questions are answered (Buchanan & Shortliffe, 1984). This logic-based strategy could produce Answer 1 as follows.

Answer 1. There is some likelihood that the computer will crash (1). The amount of wasted programming work should be minimized (2). If the computer system crashes, you will lose all work since the last time you saved the program (3). If you save your program frequently, you will minimize the amount of wasted programming work (4). Therefore, you should save your program frequently (5).

The conclusion sentence (5) is derived logically from the facts (Sentences 1 and 2) and rules (Sentences 3 and 4).

Unfortunately, a logical trace of generic facts and rules does not provide a good answer to a "why" question (Lang, Graesser, & Hemphill, 1990; Schank, 1986; Young, 1989). A logical trace can provide useful information for other types of tasks, such as debugging program logic. However, it does not provide a coherent explanation of why computer work needs to be saved frequently. A more satisfying explanation would provide information about the programmer's goal structure and planning (Baudet & Denhiere, 1989; Clancey, 1982; Ram, 1990), as illustrated in Answer 2.

Answer 2. A program needs to be saved in order to store the most recent version of your program in permanent memory. It needs to be stored frequently in order to minimize losing your work if the computer crashes.

Answers 1 and 2 are products of question-answering strategies that access general facts, rules, and goal structures. Instead of accessing general knowledge, some question-answering strategies access particular cases stored in memory. This case-based strategy is illustrated in Answer 3.

Answer 3. One day two years ago I spent eight hours working on a program that sorts addresses on a mailing list. Then the computer crashed because of a power failure. All eight hours of my work were wasted—I was so angry. I decided at that point that I would never face that again. Now I save my program every 15 minutes.

One way of justifying an action is to tell a story that illustrates the virtues of performing the action (Dyer, 1983; Schank, 1986). For example, action A is justified if such an action previously solved a problem, circumvented a bottleneck, or prevented an unfortunate event. Cases have a high likelihood of being stored in memory if they present interesting dilemmas, novel solutions, and challenging problems (Hammond, 1986; Kolodner, 1984; Schank, 1986). The answerer is reminded of these specific cases when the person justifies why an action is performed in a new situation.

There is yet another strategy of justifying why an action is performed. This additional strategy simply identifies an expert who recommended the action, as illustrated in Answer 4.

Answer 4. When my computer instructor taught me how to use the computer, he told me to save the program every 15 minutes.

The expert could be a knowledgeable person, a book, a manual, an institution, or some other highly regarded source of knowledge.

The preceding question and answers illustrate the diverse information sources and question-answering strategies that may operate when a person answers a

"why" question. As cognitive scientists, our goal is to specify the knowledge structures that are activated and cognitive procedures that are executed when people answer various categories of questions. We are entirely convinced that a psychological theory of question answering would be highly beneficial to designers of query-based interfaces for information systems. That is, computers should formulate human-like answers in expert systems, decision support systems, and intelligent tutoring systems. The expert system enterprise might have benefitted greatly if knowledge engineers had known that the logic-based answers to "why" questions (e.g., Answer 1) were comparatively unnatural for the human mind. Had there been a psychological theory of question answering 10 years ago, the question-answering facilities of modern information systems might not have been so inept.

The adequacy of a question-answering (Q/A) system can be evaluated on a number of dimensions. The Q/A system should have a high *hit rate*. That is, the answer should include most, if not all, of the information in the database that is relevant to the question. The Q/A system should have a low incidence of *false alarms* (i.e., irrelevant answers) and *errors* (i.e., incorrect answers). Sometimes several answers are produced in response to a question, that is, a cogent explanation in answer to a "why" question. A complex response should be *coherent* in the sense that the ideas hang together conceptually. A Q/A system should minimize the *response time* to answer a question. It is very difficult to design a Q/A system that satisfies all of these criteria.

The format of the database is a critical consideration in the design of Q/A systems (see Lang, Graesser, Dumais, & Kilman, chap. 8 this volume). Unstructured text is the easiest database format to implement because it is simply a copy of the information content under consideration (e.g., a copy of an encyclopedia article). Unfortunately, it is extremely difficult (if not impossible) to access relevant information and to compose cogent answers to most questions when the database is unstructured. For this reason, unstructured databases are essentially never used in intelligent tutoring systems, expert systems, and intelligent decision support systems. At the other end of the continuum, structured databases impose organization on the information. That is, the information is segmented into theoretical units (e.g., words, propositions, statements, rules), sets of units are grouped into "packages" of information (i.e., higher order units), and units are connected by relations. It is possible to access relevant information and to formulate coherent answers when the database is structured. However, substantial effort must be devoted to organizing the knowledge in a special way that makes it easy to retrieve during question answering.

This chapter presents highlights of a model of human question answering, named QUEST. QUEST simulates the psychological processes and answers of adults when they answer questions. QUEST handles many question categories, but the major focus has been on *open-class* questions, such as "why," "how," "when," "where," and "what-if" questions. Open-class questions invite replies

with elaborate verbal descriptions, such as Answers 1-4 shown earlier. In contrast, appropriate answers to *closed-class* questions are restricted to a limited number of alternatives that are usually short. For example, answers to verification questions are "yes," "no," "maybe," and "I don't know."

Previous studies have tested whether QUEST (and its predecessor models) can account for answers that adults give to open-class questions. In a typical study, adults first read a passage and then answer a series of open-class questions. Alternatively, questions are answered in the context of a generic concept (e.g., kitchen, computer). These studies have evaluated whether the answers produced by QUEST are the same as the answers produced by adult subjects. That is, there should be a high overlap between (a) QUEST's distribution of answers to a particular question and (b) the distribution of answers generated by people. We gain more confidence in QUEST to the extent that this overlap approaches 100%. QUEST has been tested in the context of narrative passages (Golding, Graesser, & Millis, 1990; Graesser & Clark, 1985; Graesser, Lang, & Roberts, 1991; Graesser & Murachver, 1985; Graesser, Robertson, & Anderson, 1981; Graesser, Robertson, Lovelace, & Swinehart, 1980), expository texts on scientific mechanisms (Graesser & Hemphill, 1991; Graesser, Hemphill, & Brainerd, 1989), scripts and other generic concepts (Graesser, 1978; Graesser & Clark, 1985), and everyday social interactions (Graesser, Roberts, & Hackett-Renner, 1990). Therefore, QUEST holds some promise in accounting for empirical data.

This chapter summarizes the components of QUEST rather than providing an exhaustive account of the technical details and empirical evidence. We refer to previous reports for a more complete description of QUEST (Graesser, 1990; Graesser & Franklin, 1990; Graesser, Gordon, & Brainerd, in press; Graesser, Lang, & Roberts, 1991) and for coverage of the empirical support (Graesser, 1990; see Graesser references).

OVERVIEW OF THE QUEST MODEL OF HUMAN QUESTION ANSWERING

It is convenient to segregate QUEST into four major components (see Fig. 12.1). The first component translates the question into a logical form and assigns it to one of several question categories. The second identifies the information sources that are relevant to the question. The information sources are represented as conceptual graph structures that contain goal/plan hierarchies, causal networks, taxonomic hierarchies, and descriptive structures. In the third component, convergence mechanisms compute the subset of nodes within each information source that serves as relevant answers to the particular question. The fourth component considers pragmatic features of the communicative interaction, such as the goals and the common ground of the speech participants. Although we

COMPONENTS OF QUEST

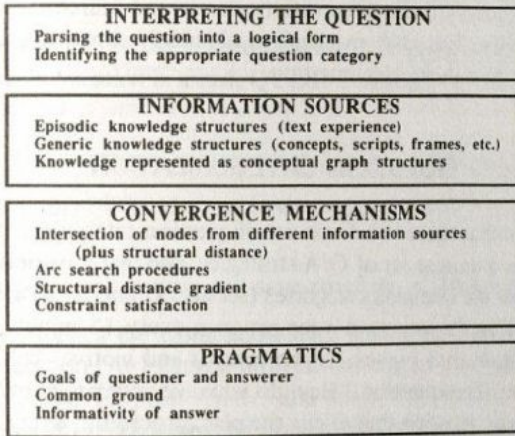


FIG. 12.1. Components of the QUEST model of human question answering.

segregate question answering into these four components, we acknowledge that an adequate model of question answering would integrate these components in a highly interactive fashion (Dyer, 1983; Lehnert, Dyer, Johnson, Young, & Harley, 1983; Robertson, Black, & Lehnert, 1985).

QUEST was not developed to account for the linguistic features of question answering. QUEST does not explain the process of parsing the question syntactically and the process of articulating replies linguistically. Instead, QUEST was developed to account for the conceptual content of answers.

We should acknowledge some of the models and research that assisted us in developing the QUEST model. Cognitive psychologists have investigated the process of answering closed-class questions at considerable depth (Glucksberg & McCloskey, 1981; Reder, 1987; Singer, 1986, 1990). Some of the mechanisms that are part of answering closed-class questions are also relevant to the answering of open-class questions. Progress in investigating the answering of open-class questions has been comparatively slow in psychology, but there have been some major contributions (Collins, Warnock, Aiello, & Miller, 1975; Graesser & Black, 1985; Norman, 1973; Norman & Rumelhart, 1975; Piaget, 1952; Shanon, 1983; Trabasso, van den Broek, & Liu, 1988).

The fields of artificial intelligence and computational linguistics have furnished detailed models of question answering that account for the content of answers and the world knowledge that supplies the content (Allen, 1983; Bruce, 1982; Dahlgren, 1988; Dyer, 1983; Kaplan, 1983; Lehnert, 1978; Lehnert et al., 1983; McKeown, 1985; Ram, 1990; Schank, 1986; Schank & Abelson, 1977; Souther, Acker, Lester, & Porter, 1989; Woods, 1977). In most of these models,

texts and world knowledge are organized in the form of structured databases. The Q/A procedures access these information sources and search through the structures systematically. The Q/A mechanism ultimately constructs an informative coherent answer to a particular question.

QUESTION CATEGORIZATION

QUEST assumes that there is a finite set of question categories, that each question category has a unique set of Q/A strategies, and that a particular question is assigned to one of the question categories (see also Lehnert, 1978). For example, the question "Why do you save your program every 15 minutes?" is a *goal-orientation* question that probes for the reasons and motives behind an agent's intentional action. The question "How do you save your program?" is an *instrumental/procedural* question that elicits the plan, procedure, or style of executing an action. The chapter by Graesser, Person, and Huber (chap. 9 in this volume, Table 9.1) presents a set of question categories that would be included in QUEST's catalogue of questions.

In order to complete question categorization successfully, it is necessary to determine the *question focus*, the element or expression that constitutes the target of the question. In most cases there are many nodes in a question, so there needs to be a way to select the question focus. In the question "How do you save a program?", the question focus is the *statement node* "you save program." A statement node contains one predicate (e.g., save) and one or more arguments (you, program). The question focus may be an argument of a statement node rather than the statement as a whole. In the question "Who saved the program?" the focus is the agent slot (*X*) of the statement node "X saved program." In a long-winded question there are many statement nodes, only one of which would be selected as the question focus. For example, consider the following question and its three statement nodes:

How do you save a program when there is a blackout and the computer crashes?

S-Node 1: You save program.

S-Node 2: A blackout occurs.

S-Node 3: The computer crashes.

Statement Node 1 would be the question focus in this example, whereas Nodes 2 and 3 furnish context. The focusing mechanism is a rather complex process that considers semantic and pragmatic constraints. QUEST assumes that focusing is successfully completed but does not currently explain the operation of the focusing mechanism.

INFORMATION SOURCES

An information source is a structured database that furnishes answers to a question. Whenever a question is asked, QUEST computes an expression with three slots:

QUESTION (<Q-category>, <question focus>, <information source>)

The expression for "Why do you save your program?" is presented in the following:

QUESTION (goal-orientation, you save program, <information source>)

The third slot supplies the world knowledge structures that are tapped for answers to the question. At least one information source must be accessed before the question can be fully interpreted and answered. Without an information source, it is difficult, if not impossible, to understand the question and to identify the question focus.

When most questions are answered, several information sources are relevant to the question. There are *episodic knowledge structures* (EKSs) that correspond to particular episodes, cases, or exemplars that a person experienced in the past. For example, Answer 3 referred to a particular episode of a person losing 8 hours of work when the computer crashed. In addition to a vast inventory of EKSs, there are thousands of *generic knowledge structures* (GKSs) in long-term memory. A GKS is a more abstract representation that summarizes the typical properties of the concept it represents. Two GKSs relevant to the example question are "computer program" and "saving."

The content of a GKS is undoubtedly derived from the family of EKSs that are associated with the GKS (Kolodner, 1984; Schank, 1986; Smith & Medin, 1981). For example, the generic concept (GKS) that a person has for "computer program" is a product of thousands of EKSs that are created from particular experiences with particular programs. QUEST does not make any informative or controversial claims about the relationship between a GKS and its associated family of EKSs. Presumably, there is a family of EKSs associated with the GKS, such that each EKS has some features that are common with the GKS and some features that distinguish the particular EKS from others in the family. For example, there may be 1000 EKSs involving computer programs, but only one of these EKSs involves a computer program that controls a sprinkler system in a house (e.g., EKS-118). EKS-118 would be indexed with unique, distinctive features (see Kolodner, 1984, 1989; Schank, 1986) that set it apart from other EKSs.

Many of the information sources for a question are accessed by the content words in the question, such as nouns, main verbs, and adjectives. Those information sources that are accessed by content words are termed *word-activated*

information sources. In contrast, *pattern-activated* information sources are activated by the features that accumulate from (a) the context of the question, (b) the goals of the speech participants, and (c) the combinations of content words that are mentioned in the discourse.

The information sources for a particular question consist of a family of GKSs and EKSs that are relevant to the question. Each information source is a structured database with dozens, even hundreds of nodes. Therefore, working memory is crowded with a wealth of information during question answering. If there were five information sources, with 100 nodes per source, then 500 nodes would be activated in working memory. Obviously, most of these nodes would not be produced as answers to the question. Only a small set of answers (fewer than 10) is normally produced when an adult answers an open-ended question. QUEST's *convergence mechanisms* begin with the 500 nodes in the node space and prune the space to approximately 10 good answers. These convergence mechanisms are discussed later in this chapter.

REPRESENTATION OF INFORMATION SOURCES

Each information source is a package of highly structured nodes. In the representational system adopted by QUEST, an information source is a database that is organized in the form of a *conceptual graph structure*. A conceptual graph structure consists of a set of nodes that are categorized and that are interrelated by labeled, directed arcs. There are five basic node categories, which are specified in the following:

1. *Concept*. An entity that is normally expressed as a noun (e.g., computer, tree, time).
2. *State*. An ongoing characteristic that remains unchanged within the time frame under consideration (e.g., the computer has an on/off switch, George knows how to run the program).
3. *Event*. A state change within the time frame under consideration (e.g., George presses the wrong key, the computer crashes).
4. *Goal*. An event or state that an agent desires (e.g., George wants to save the program).
5. *Style*. The speed, intensity, force, or qualitative manner in which an event unfolds (e.g., an event occurs quickly, in circles).

An *action* is an amalgamation of a goal node and an event, state, or style specification that achieves the goal; the agent must have done something that causally led up to the successful outcome.

There are several arc categories in QUEST's representational system. Table

12.1 presents a list of 14 arc categories that have been adopted by QUEST. Table 12.1 defines each arc category and presents examples. One or more composition rules are associated with each arc category. A composition rule declares which node categories can be connected by the particular arc. For example, a Reason arc connects only goal nodes whereas a Consequence arc never connects goal nodes. Except for And/Or arcs, all arcs are directed, such that the end node is connected to the head of the arc and the source node is connected to the tail:

(source node)—arc→ (end node)

A more complete specification of the arc categories is provided in other reports (Graesser & Clark, 1985; Graesser & Franklin, 1990; Graesser, Gordon, & Brainerd, in press).

We have devoted much of our past research efforts at understanding four types of knowledge structures:

1. *Causal networks* contain event chains, along with states that enable the events to occur; there is a high density of Consequence and Manner arcs.
2. *Goal hierarchies* convey the goals, plans, and intentional actions of agents, along with events/states in the world that initiate the goal hierarchies; Reason, Manner, Initiate, and Outcome arcs are very prevalent in goal hierarchies.
3. *Taxonomic hierarchies* express (a) what classes of entities are nested within other classes and (b) those properties that are both distinctive and characteristic of each concept; there is a high density of Isa and Property arcs.
4. *Spatial region hierarchies* express what regions are embedded within other regions and how regions are directionally related (e.g., X North-of Y, X Left-of Y, X On-top-of Y.).

We acknowledge that a particular information source is an amalgamation of these and other types of knowledge structures. The purpose of segregating these types of substructures is to identify the systematic characteristics of both the structures and the Q/A procedures that operate on the structures.

Figure 12.2 shows an example information source that contains a goal hierarchy and a causal network. Nodes 1–7 consist of a hierarchical structure of goals that are connected by Reason arcs. The most superordinate goal is Node 1 (“George uses program whenever George wants to”), whereas the most subordinate goal is Node 7 (“George makes up file name”). Whenever a goal is achieved, a goal node is linked to an event or state with an Outcome arc, and the event/state signifies a positive outcome. Goals 4, 5, and 7 are achieved because there are positive outcomes in Nodes 10, 11, and 12, respectively. There are no outcomes associated with Nodes 1–3 because these goals are not achieved. Goal

TABLE 12.1
Definitions and Composition Rules for Fourteen Categories of Arc

Arc Category	Definition and Temporal Constraints	Composition Rules	Example
Consequence (C)	A causes or enables B A precedes B in Time	(event / state / style)-C-> (event / state / style)	(Event: George presses the wrong key)-C-> (Event: The screen goes blank)
Implies (Im)	A implies B A and B overlap in time	(event / state / style)-Im-> (event / state / style)	(Event: The screen goes blank)-Im-> (Event: The computer crashed)
Reason (R)	B is a reason or motive for A B is a superordinate goal of A A is achieved before B is achieved	(goal)-R->(goal)	(Goal: George types "save")-R-> (Goal: George saves program)
Outcome (O)	B specifies whether or not the goal A is achieved A precedes B in time	(goal)-O->(event / state / style)	(Goal: George types "save")-O-> (Event: "Save" is typed)
Initiate (I)	A initiates or triggers the goal in B A precedes B in time	(event / state / style)-I->(goal)	(State: George knows how to save program)-I-> (Goal: George saves program)
Manner (M)	B specifies the manner in which A occurs A and B overlap in time	(goal)-M->(goal / style) (style)-M->(style) (event)-M->(event / style)	(Event: George pressed a key)-M-> (Style: The key was pressed hurriedly)

Before / After / During: Temporal relationship	A has a temporal relationship with B	(goal)-Before->(goal) (event / state)-Before-> (event / state)	(Goal: George types "save")-Before-> Goal: George types space)
And / Or	Both A and B exist / occur Either A or B exists / occurs	(concept)-Or->(concept) (goal)-Or->(goal) (event / state)-Or->(event / state)	(State: George had a deadline)-And-> (Event: The computer crashed)
Is a (isa)	A is a kind / type / instance of B	(concept)-isa->(concept)	(Concept: computer)-isa-> (concept: electronic device)
Has As Part (HAP)	A has as a part B	(concept)-HAP->(concept)	(Concept: computer)-HAP-> (Concept: keyboard)
Property (P)	A has a property P	(concept)-P->(state / event / goal)	(Concept: Macintosh)-P-> (State: computer has iconic interface)
Referential Pointer (Ref)	An argument of A refers to a concept B	(state / event / goal)-Ref-> (concept)	Noun argument of (Event: The computer crashed)-Ref->(Concept: computer)
Contains (Cont)	A contains B	(concept)-Cont->(concept)	(Concept: computer case)-Cont-> (Concept: disk drive)
Spatial relationship Left-of / Right-of Top-of / Bottom-of East-of / West-of North-of / South-of Touch / Connect	A has a spatial relationship with B	(concept)-East-of->(concept)	(Concept: computer case)- Top-of->(Concept: table)

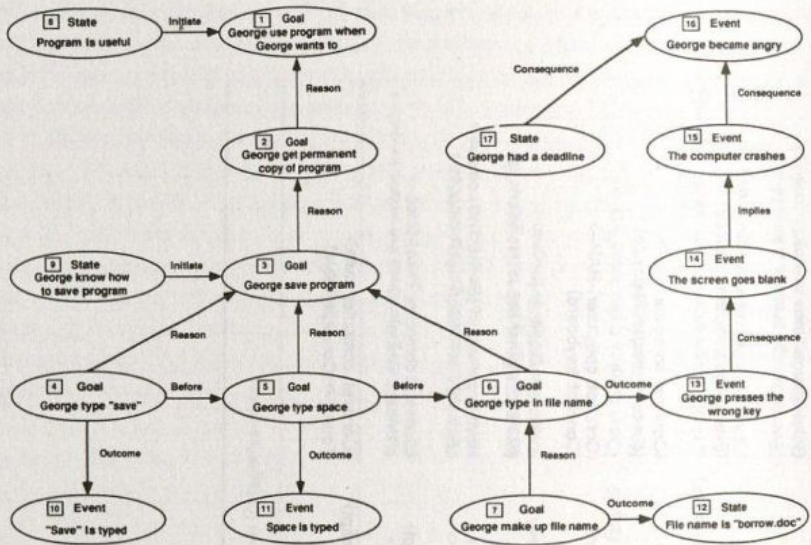


FIG. 12.2. An example structure that contains a goal hierarchy and a causal network.

6 ("George types in file name") has an outcome node, but the outcome is negative (Node 13, "George presses wrong key"). In addition to the goal hierarchy and outcome nodes, there are states and events that initiate the goal hierarchy. For example, the goal hierarchy (Nodes 1–7) is initiated by States 8 and 9 via the Initiate arc. Finally, Fig. 12.2 contains a causal network (Nodes 13–17). The event chain begins with George pressing the wrong key and ends with George becoming angry.

Figure 12.3 shows a taxonomic structure for computers and tools. There is a hierarchy of concept nodes that are connected by *Isa* arcs (Nodes 1–10). The most superordinate concept is *tool* whereas *Macintosh*, *IBM/XT*, and *Cray* are the most subordinate concepts. Most of the concept nodes have properties (via Property arcs). A property is not only very typical of the concept to which it is connected; it is also distinctive in the sense that it is typical of concept C, but atypical of the sibling nodes of C. For example, a microcomputer is inexpensive (Node 14) but inexpensiveness is not a property of supercomputers and Connection Machines.

Figures 12.2 and 3 furnish an impression of conceptual graph structures, their node categories, and their arc categories. We emphasize that these structures are not composed haphazardly according to idiosyncratic intuitions of the investigator. There are specific constraints that must be satisfied when one node is connected to another node by a particular arc category (see Table 12.1). These are captured by each arc's definition, temporal constraints, and rules of composition.

important to note that two nodes may match by virtue of argument substitution (e.g., *X* is substituted for George).

The intersecting node identifier isolates all nodes in working memory that overlap. These intersecting nodes have a privileged status. There is evidence that these intersecting nodes have a higher likelihood of being produced as answers than nonintersecting nodes (Golding et al., 1990; Graesser & Clark, 1985; Graesser, Lang, & Roberts, 1991).

Arc Search Procedure

Each question category has its own arc search procedure that operates on the information sources relevant to a question and that produces candidate answers. When an information source is accessed, the arc search procedure is applied to each of the "entry nodes" in the information source. An entry node usually consists of a node in the structure that matches the question focus in the question. Suppose that the question is "Why does a person save a program?" and the information source is Fig. 12.2. The entry node would be Node 3 in Fig. 12.2 because the question focus (i.e., person saves program) matches Node 3 ("George saves program") by virtue of argument substitution. In some cases, the entry node is one of the intersecting nodes that was extracted by the intersecting node identifier. For example, Nodes 1 and 8 may be intersecting nodes that are stored both in the EKS for Fig. 12.2 and the GKS for computer program.

Once an entry node is located in an information source, the arc search procedure executes a breadth first search from the entry node by traversing legal arcs that radiate from the entry node. For each question category, there is a particular set of arc categories and arc directions that is legal. For example, the arc search procedure for a goal orientation question is specified in the following:

Arc search procedure for goal orientation question: Generate superordinate goals (via paths of forward Reason arcs and backward Manner arcs) and goal initiators (connected to the entry node and superordinate nodes via backward Initiate arcs).

Consider, for example, the goal-orientation question "Why does a person save a program?" If this question were interpreted in the context of the structure in Fig. 12.2, the entry node would be Node 3 and the following nodes would be legal candidate answers:

- "In order to get a permanent copy of the program" (Node 2).
- "In order to use the program whenever the user wants to" (Node 1).
- "Because the program is useful" (Node 8).
- "Because the person knows how to save the program" (Node 9).

All of the other nodes in Fig. 12.2 are illegal. Given that there are 16 possible answers, but only 4 are legal answers, the arc search procedure has reduced the node space to 25% of the possible nodes.

Instrumental/procedural questions have a completely different arc search procedure.

Arc search procedure for instrumental/procedural question: Generate subordinate goals and style specifications (via paths of backward Reason and forward Manner arcs).

When this arc search procedure is applied to the question "How does a person save a program?" in the context of Fig. 12.2, the following legal answers are generated.

- "You type save" (Node 4).
- "You type a space" (Node 5).
- "You make up a file name" (Node 7).
- "You type in the file name" (Node 6).

Therefore, answers to instrumental/procedural questions include subordinate goals in the goal hierarchy whereas answers to goal-orientation questions include superordinate goals.

The arc search procedures for causal antecedent and causal consequence questions are normally applied to events and states.

Arc search procedure for causal antecedent question: Generate nodes on paths with the following arcs: backwards Consequence, forward and backward Implies, backward Initiate, and backward Outcome.

Arc search procedure for causal consequence question: This procedure is the inverse of that of causal antecedent questions: forward Consequence, forward and backward Implies, forward Initiate, and forward Outcome.

The causal antecedent question "Why did the screen go blank?" would generate the following legal answers when applied to the Fig. 12.2 structure:

- "Because the computer crashed" (Node 15).
- "Because the person pressed the wrong key" (Node 13).
- "The person wanted to type the file name" (Node 6).

The causal consequence question "What were the consequences of the screen going blank?" would generate the following legal answers:

"The computer crashed" (Node 15).

"George became angry" (Node 16).

The arc search procedure for the causal consequence question is probably incomplete, as it has been formulated so far. Good answers to such questions also should include goal failures (i.e., "George was not able to save his program," negated Node 3) and failed obligations (e.g., "George missed his deadline," negated Node 17).

Why, how, and consequence questions are very natural for goal hierarchies and causal networks. A different set of questions is natural for taxonomic structures. Definition questions are frequently asked about the concept nodes in taxonomic hierarchies: "What does X mean?" and "What is an X?". The arc search procedure for these definition questions consists of a genus-differentiae frame, which is adopted in most dictionary definitions.

Arc search procedure for definition questions: Generate the immediate superordinate node of concept X (via the forward Isa arc) and the properties directly linked to X (via the forward Property arc).

The question and answer is articulated as follows:

"What is an X?"

"An X is a <superordinate node>that<property 1> . . . <property n>"

QUEST would generate the following answer to the question "What is a computer?" when the database is Fig. 12.3: "A computer is an electronic device that manipulates symbols." Whereas definition questions produce concepts that are superordinate in the Isa hierarchy, subordinate concepts are produced when answering the question "What is an example of X?" Legal answers to the question "What is an example of a computer?" would be a microcomputer, a supercomputer, and a Connection Machine.

Comparison questions are frequently asked about concepts in taxonomic hierarchies, that is, "How is X similar to Y?" and "How is X different from Y?". The answers include properties that either compare or contrast the two concepts (X and Y). When these properties are computed, QUEST considers the properties that are inherited from its superordinate concepts as well as properties directly linked to the concept. For example, the concept computer has Node 13 directly linked to it whereas Nodes 11 and 12 are inherited properties:

"A computer manipulates symbols" (Node 13).

"A computer uses electricity" (substituting computer for electronic device in Node 12).

"People want to use computers" (substituting computer for tool in Node 11).

The arc search procedure for comparison questions selects the appropriate properties from the total set of properties of X and Y (both directly linked and inherited). However, the procedure for selecting these properties will not be specified in this chapter (see Graesser, Gordon, & Brainerd, in press).

The arc search procedures in QUEST are compatible with some theories of question answering in artificial intelligence that have emphasized the importance of knowledge organization and of restricting search by pursuing nodes that are linked by particular conceptual relations (Lehnert, 1978; Lehnert et al., 1983; Schank & Abelson, 1977; Souther et al., 1989). There also is extensive empirical evidence that the arc search procedures robustly predict what answers adults produce to questions (Golding et al., 1990; Graesser, 1990; Graesser & Hemphill, 1991; Graesser et al., 1989; Graesser, Lang, & Roberts, 1991; Graesser & Murachver, 1985; Graesser et al., 1981).

Structural Distance

When an information source is tapped for answers to a question, there is a structural distance score associated with each candidate answer. Structural distance consists of the number of arcs between the entry node and the candidate answer node. For example, in Fig. 12.2 the distance between Nodes 3 and 13 is two whereas the distance between Nodes 3 and 17 is six.

There is some evidence that the likelihood of a node being produced as an answer decreases as a function of its distance from the entry node in the information source (Golding et al., 1990; Graesser & Clark, 1985; Graesser & Hemphill, 1991; Graesser et al., 1989; Graesser, Lang, & Roberts, 1991). The decrease appears to be exponential. If d is structural distance and t is the likelihood of traversing a single arc, then the likelihood is t^d of producing a node as an answer (given that it is a legal answer that passed the arc search procedure). The value of t was 0.67 in the study by Graesser et al. (1989). If the candidate answer were four arcs away from the entry node, then the likelihood of it being produced as an answer would be $0.67^4 = 0.21$.

The predicted effects of structural distance on answer production are consistent with models of question answering other than QUEST (Shanon, 1983; Winston, 1984). For example, Winston described a Q/A model that answers "why" and "how" questions in the context of goal hierarchies and problem spaces. His Q/A algorithm specifies that good answers are only one arc away from the question focus. QUEST assumes there is an exponential gradient rather than assuming the single arc rule. Theories of marker passing and spreading activation also would predict the exponential distance gradient (Anderson, 1983).

Constraint Satisfaction

The semantic content of the answer should not be incompatible with the content of the question focus. Constraint satisfaction discards those candidate answers in

the node space that are incompatible with the question focus. Stated differently, the question focus has semantic and conceptual constraints that are propagated among the nodes in the information sources, ultimately pruning out the incompatible nodes.

There are several ways in which a candidate answer could be incompatible with the question focus (Graesser & Clark, 1985). Suppose that the question focus is "X saves program." A node is pruned if it directly contradicts the question focus (e.g., "X lost the program"). A node is pruned if it is in the wrong time frame (e.g., "X was born"). Incompatibilities in planning may occur when a plan conveyed in the candidate answer cannot be executed when the action in the question focus is executed (e.g., "X went to sleep"). An answer may be pruned if it is not causally related to the question focus. A candidate answer is not produced if it has incompatibilities on one or more of these dimensions.

Once again, the above four components of convergence are capable of narrowing the node space from hundreds of nodes in working memory to a handful of good answers to a question. Graesser, Gordon, and Brainerd (in press) analyzed convergence mechanisms in a study that involved answering "why," "how," and consequence questions in the context of simple short narrative passages (150 words). A convergence score was defined as the proportion of nodes in the node space that constituted good answers to a question after application of the following convergence mechanisms: arc search procedure, structural distance, and constraint satisfaction. The arc search procedures reduced the node space to 10% of the original nodes. The structural distance component reduced the node space further; 38% of the above 10% of the nodes passed the structural distance component, yielding 3.8%. Constraint satisfaction reduced the node space to 55% of the remaining 3.8% of the nodes. Therefore, the overall constraint satisfaction score was 2.1%. If the information source were a text structure with 200 nodes, then convergence mechanisms would have produced four good answers to the question. This estimate is quantitatively close to available studies on human question answering (Graesser & Clark, 1985; Graesser & Murachver, 1985).

PRAGMATICS

The pragmatic components of QUEST address the social and communicative functions of answering a question. One key component considers the goals of the questioner and answerer. From the perspective of the questioner, a question may be asked to solve a problem, to assess how much the answerer knows, to persuade, to control a conversation, and so on. From the perspective of the answerer, the answer may be formulated to inform the questioner, to let the questioner know the answerer knows something, to entertain the questioner, and so on. A complete model of question answering would consider the goals of the

speech participants in the discourse context and would tailor answers to achieve these goals (Allen, 1983; Bruce, 1982; Francik & Clark, 1985; Kaplan, 1983).

One important goal to assess is whether the questioner genuinely seeks the information invited by the question. Some questions are not genuine information-seeking questions: Indirect requests (e.g., "Could you delete this file?"), greetings ("How are you doing?"), gripes ("Why does this always happen to me?"), and rhetorical questions. Van der Meij (1987) has identified the preconditions that must be met before an utterance constitutes a genuine information-seeking question.

1. The questioner does not know the information asked for with the question.
2. The questioner believes that the presuppositions of the question are true.
3. The questioner believes that an answer exists.
4. The questioner wants to know the answer.
5. The questioner can assess whether a reply constitutes an answer.
6. The questioner poses the question only if the benefits exceed the costs. The benefits of knowing the answer must exceed the costs of asking the question. The benefits of knowing the answer must exceed the benefits of not knowing the answer.
7. The questioner believes the answerer knows the answer.
8. The questioner believes that the answerer will not give an answer in absence of the question.
9. The questioner believes that the answerer will supply an answer.

To the extent that these nine preconditions are violated, a speech act would not be a genuine information-seeking question.

A second pragmatic component is the common ground (i.e., shared knowledge) between the questioner and answerer (Clark & Marshall, 1981; Clark & Schaefer, 1989; Miyake & Norman, 1979; Shanon, 1983; Sleeman & Brown, 1982). The answerer first estimates the common ground between speech participants and then selects an answer that moderately extends the boundaries of the common ground. That is, the answer should be somewhat more informative, elaborate, or detailed than the common ground, but should not be (a) entirely within the sphere of the common ground or (b) substantially more detailed than the common ground.

In principle, QUEST could keep track of the common ground between the questioner and answerer. QUEST would identify those information sources that each participant has stored in memory, and what nodes are stored in each information source. The fringe or boundary of knowledge also can be computed in a straightforward manner. A fringe answer would be one or two arcs away from a node in the common ground.

Common ground could have some counterintuitive effects on answer production. If the common ground is high and the answerer wants to be informative by supplying information that the questioner does not already know, then the answerer would avoid nodes that are in multiple information sources. Surprisingly, this would yield a negative correlation between answer production and the number of information sources that supplied the answer. Common ground also might have an interesting prediction regarding the impact of structural distance on answer production. There might be a preference for distant nodes because proximate nodes would be easy to infer. Perhaps the net effect would yield a curvilinear relationship, with answers at intermediate distances being better than answers at close and at far distances from the entry node in the information source.

The impact of pragmatic components on question answering should not be taken lightly. Indeed, there are ample reasons for being skeptical about the external validity of QUEST's convergence mechanisms when questions are embedded in conversations. The context and constraints of a conversation can potentially modify the literal meaning of a question and thereby radically alter appropriate replies. Suppose, for example, that a customer visits a used car lot, points to a car, and asks the salesperson "Why is this 1985 Chevy so expensive?" Some replies are presented in the following:

1. The engine has only 2,000 miles on it.
2. This 1983 Buick is in good condition.
3. Why don't you look at this 1983 Buick?
4. What price range are you considering?

Reply 1 would be produced by QUEST because it specifies a causal antecedent to the Chevy's being expensive. Replies 2, 3, and 4 would be reasonable replies to the question in the conversation, but these replies would not be generated by QUEST. Reply 2 does not address the customer's question directly; the salesperson inferred that the customer could not afford the Chevy so the salesperson recommended a less expensive car. Reply 3 is syntactically a question, but functionally an indirect request. Neither of these speech act categories are produced as answers by QUEST; assertions are the only appropriate replies to questions in the QUEST model. Reply 4 is a question, both syntactically and functionally, and therefore is beyond the scope of QUEST.

Answers to questions in naturalistic conversations are constrained by the goals and common ground of the speech participants. To the extent that these components are increasingly complex, questions are less likely to be information-seeking questions that can be accommodated by QUEST's convergence mechanisms. QUEST will fail when questions are functionally commands, indirect requests, gripes, sarcastic comments, conversation monitors, or rhetorical de-

vices. These pragmatic considerations illustrate some limitations of QUEST but do not imply the model is useless. QUEST would hold up quite well in contexts where pragmatics does not play a major role. QUEST would be quite useful if it could account for many, although not all, the answers in naturalistic conversations. Graesser, Roberts, and Hackett-Renner (1990) reported that approximately 80% of the answers in naturalistic conversations were compatible with QUEST's arc search procedures. Accounting for 80% of the answers would be considered by some colleagues as either encouraging or impressive results.

It is easy to declare that a good answer must address the goals of the questioner and answerer. It is difficult to write a computer program that simulates the computational mechanisms that accomplish this requirement. We are currently exploring how the speech participants' goals impact on (a) the selection of good answers to a question and (b) the selection of the most relevant question-answering strategies. We are also investigating how the answerer pieces together the nodes from diverse information sources in an effort to formulate a coherent answer. A good long-winded answer is both coherent and informative. Unfortunately, we know next to nothing about the mechanisms that generate coherent, informative answers.

In closing, we believe that the QUEST model furnishes a promising start at developing an adequate theory of human question answering. However, we would not at all be surprised if we undertake extensive remodeling in the future.

ACKNOWLEDGMENTS

This research was funded by grants awarded to the first author by the Office of Naval Research (N00014-88-K-0110 and N00014-90-J-1492).

REFERENCES

- Allen, J. F. (1983). Recognizing intentions from natural language utterances. In M. Brady & R. C. Berwick (Eds.), *Computational models of discourse* (pp. 107-166). Cambridge, MA: MIT Press.
- Anderson, J. R. (1983). *The architecture of cognition*. Cambridge, MA: Harvard University Press.
- Baudet, S., & Denhiere, G. (1989). Representation and acquisition of knowledge of functional systems. In *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society* (pp. 546-552). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Bruce, B. C. (1982). Natural communication between person and computer. In W. G. Lehnert & M. H. Ringle (Eds.), *Strategies for natural language processing* (pp. 55-88). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Buchanan, B. G., & Shortliffe, E. H. (1984). *Rule-based expert systems: The MYCIN experiments of the Stanford heuristic programming project*. Reading, MA: Addison-Wesley.
- Clancey, W. J. (1982). Tutoring roles for guiding a case method dialogue. In D. Sleeman & J. S. Brown (Eds.), *Intelligent tutoring systems* (pp. 201-225). New York: Academic Press.

- Clark, H. H., & Marshall, C. R. (1981). Definite reference and mutual knowledge. In A. Joshi, B. Webber, & I. Sag (Eds.), *Elements of discourse understanding* (pp. 10-63). Cambridge: Cambridge University Press.
- Clark, H. H., & Schaefer, E. F. (1989). Contributing to discourse. *Cognitive Science*, 13, 259-294.
- Collins, A., Warnock, E. H., Aiello, N., & Miller, M. L. (1975). Reasoning from incomplete knowledge. In D. G. Bobrow & A. Collins (Eds.), *Representation and understanding* (pp. 383-415). New York: Academic Press.
- Dahlgren, K. (1988). *Naive semantics for natural language understanding*. Boston: Kluwer Academic Publishers.
- Dyer, M. G. (1983). *In-depth understanding: A computer model of integrated processing for narrative comprehension*. Cambridge, MA: MIT Press.
- Francik, E. P., & Clark, H. H. (1985). How to make requests that overcome obstacles to compliance. *Journal of Memory and Language*, 24, 560-568.
- Glucksberg, S., & McCloskey, M. (1981). Decisions about ignorance: Knowing that you don't know. *Journal of Experimental Psychology: Human Learning and Memory*, 7, 311-326.
- Golding, J. M., Graesser, A. C., & Millis, K. K. (1990). What makes a good answer to a question?: Testing a psychological model of question answering in the context of narrative text. *Discourse Processes*, 13, 279-303.
- Graesser, A. C. (1978). How to catch a fish: The memory and representation of common procedures. *Discourse Processes*, 1, 72-89.
- Graesser, A. C. (1990). *Investigations of human question answering* (Technical Report #1 for Office of Naval Research contract, Memphis State University, Memphis, TN).
- Graesser, A. C., & Black, J. B. (Eds.). (1985). *The psychology of questions*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Graesser, A. C., & Clark, L. F. (1985). *Structures and procedures of implicit knowledge*. Norwood, NJ: Ablex.
- Graesser, A. C., & Franklin, S. P. (1990). QUEST: A cognitive model of question answering. *Discourse Processes*, 13, 279-303.
- Graesser, A. C., Gordon, S., & Brainerd, L. E. (in press). QUEST: A model of question answering. *Computers & Mathematics with Applications*.
- Graesser, A. C., & Hemphill, D. D. (1991). Question answering in the context of scientific mechanisms. *Journal of Memory and Language*, 31, 186-209.
- Graesser, A. C., Hemphill, D. D., & Brainerd, L. E. (1989). Question answering in the context of causal mechanisms. In *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society* (pp. 621-626). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Graesser, A. C., Lang, K. L., & Roberts, R. M. (1991). Question answering in the context of stories. *Journal of Experimental Psychology: General*, 120, 254-277.
- Graesser, A. C., & Murachver, T. (1985). Symbolic procedures of question answering. In A. C. Graesser & J. B. Black (Eds.), *The psychology of questions* (pp. 15-88). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Graesser, A. C., Roberts, R. M., & Hackett-Renner, C. (1990). Question answering in the context of telephone surveys, business interactions, and interviews. *Discourse Processes*, 13, 327-348.
- Graesser, A. C., Robertson, S. P., & Anderson, P. A. (1981). Incorporating inferences in narrative representations: A study of how and why. *Cognitive Psychology*, 13, 1-26.
- Graesser, A. C., Robertson, S. P., Lovelace, E. R., & Swinehart, D. M. (1980). Answers to why-questions expose the organization of story plot and predict recall of actions. *Journal of Verbal Learning and Verbal Behavior*, 19, 110-119.
- Hammond, K. J. (1986). *Care-based planning: An integrated theory of planning learning and memory*. Ph.D. thesis, Yale University, New Haven, CT.
- Kaplan, S. J. (1983). Cooperative response from a portable natural language system. In M. Brady &

- R. C. Berwick (Eds.), *Computational models of discourse* (pp. 167–208). Cambridge, MA: MIT Press.
- Kolodner, J. (1984). *Retrieval and organization strategies in conceptual memory: A computer model*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Kolodner, J. (1989). Selecting the best case for a best-case reasoner. In *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society* (pp. 155–162). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Lang, K. L., Graesser, A. C., & Hemphill, D. D. (1990). The role of questioning in knowledge engineering and the interface of expert systems. *Poetics*, 19, 143–166.
- Lehnert, W. G. (1978). *The process of question answering*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Lehnert, W. G., Dyer, M. G., Johnson, P. N., Young, C. J., & Harley, S. (1983). BORIS—An experiment in in-depth understanding of narratives. *Artificial Intelligence*, 20, 15–62.
- McKeown, K. R. (1985). Discourse strategies for generating natural language text. *Artificial Intelligence*, 27, 1–41.
- Miyake, N., & Norman, D. A. (1979). To ask a question, one must know enough to know what is not known. *Journal of Verbal Learning and Verbal Behavior*, 18, 357–354.
- Norman, D. A. (1973). Memory, knowledge and the answering of questions. In R. Solso (Ed.), *Contemporary issues in cognitive psychology* (pp. 135–165). Washington, DC: Winston.
- Norman, D. A., & Rumelhart, D. E. (1975). *Explorations in cognition*. San Francisco: Freeman.
- Piaget, J. (1952). *The origins of intelligence in the child*. London: Routledge & Kegan Paul. (Original work published 1936).
- Ram, A. (1990). Knowledge goals: A theory of intelligence. In *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society* (pp. 206–214). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Reder, L. M. (1987). Strategy selection in question answering. *Cognitive Psychology*, 19, 90–138.
- Robertson, S. P., Black, J. B., & Lehnert, W. G. (1985). Misleading question effects as evidence for integrated question understanding and memory search. In A. C. Graesser & J. B. Black (Eds.), *The psychology of questions* (pp. 191–218). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Schank, R. C. (1986). *Explanation patterns: Understanding mechanically and creatively*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Schank, R. C., & Abelson, R. (1977). *Scripts, plans, goals, and understanding*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Shanon, B. (1983). Answers to where-questions. *Discourse Processes*, 6, 319–352.
- Singer, M. (1986). Answering yes-no questions about causes: Question acts and question categories. *Memory & Cognition*, 14, 55–63.
- Singer, M. (1990). Answering questions about discourse. *Discourse Processes*, 13, 261–277.
- Sleeman, D. H., & Brown, J. S. (Eds.). (1982). *Intelligent tutoring systems*. New York: Academic Press.
- Smith, E. E., & Medin, D. L. (1981). *Categories and concepts*. Cambridge, MA: Harvard University Press.
- Souther, A., Acker, L., Lester, J., & Porter, B. (1989). Using view types to generate explanations in intelligent tutoring systems. In *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society* (pp. 123–130). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Trabasso, T., van den Broek, P. W., & Liu, L. (1988). A model for generating questions that assess and promote comprehension. *Questioning Exchange*, 2, 25–38.
- van der Meij, H. (1987). Assumptions of information-seeking questions. *Questioning Exchange*, 1, 111–117.
- Winston, P. H. (1984). *Artificial Intelligence* (2nd ed.). Reading, MA: Addison-Wesley.

- Woods, W. A. (1977). Lunar rocks in natural English: Explorations in natural language question answering. In A. Zampoli (Ed.), *Linguistic structures processing* (pp. 201-222). New York: Elsevier North-Holland.
- Young, R. (1989). Human interface aspects of expert systems. In L. A. Murray, & J. T. E. Richardson (Ed.), *Intelligent systems in a human context* (pp. 101-111). Oxford: Oxford University Press.