Question processing involves parsing, memory retrieval, question categorization, initiation of appropriate answer-retrieval heuristics, answer formulation, and output. Computational and psychological models have traditionally treated these processes as separate, sequential, independent, and in pursuit of a single answer type at a time. Here this view is challenged and the implications of a theory in which question processes operate simultaneously on multiple question interpretations are explored. A highly interactive model is described in which an expectation-driven parser generates multiple question candidates, including partially-specified candidates. Question candidates act as constraints for a matcher which activates memory items. An answer retrieval process examines question candidates and the active portions of memory in an attempt to generate answer candidates. Answer candidates are examined by an output process that derives the final answer. These processes run simultaneously and interact. Three experiments on human question answering are also described which provide evidence that working memory load during question reading is affected by processes related to answer retrieval.

1. INTRODUCTION

The processing of questions requires parsing, matching of antecedent concepts in memory, answer candidate retrieval, answer selection, and output. Researchers in this area have concentrated on the processes involved in determining and applying retrieval heuristics for various question types or the pragmatic constraints on answers (Dyer, 1983; Graesser, 1981; Graesser & Clark, 1985; Lehnert, 1977, 1978; Lehnert, Dyer, Johnson, Yang, & Harley, 1983; Singer, 1982, 1986). Largely as a simplifying assumption, these researchers have proceeded as if question parsing is an independent
process that comes before any question-related retrieval or answer formulation processes. In this article an alternative view of human question answering is proposed in which many processes operate simultaneously with parsing. In the first section, question answering theory is briefly reviewed. In the second section, a new, parallel architecture for question understanding and answer retrieval is described. In the third section, experimental evidence is presented in favor of a parallel view of question understanding and answering. In the conclusion, connectionist and hybrid architectures are contrasted with the model presented here and implications of a parallel view for existing question-answering theories are discussed.

1.1 Comprehension and Retrieval
Question comprehension involves two main components. One component is the location of memory elements that are specified by the question presupposition, or what Clark & Clark (1977) referred to as the given information in the question. For example, the question “Why did Pete drive to the store?” presupposes that Pete drove to the store. A cooperative question asker will only ask this question of someone who can reasonably be expected to share the presupposition. Researchers assume that the first step in question answering is extraction of the question presupposition. Lehnert (1977, 1978) referred to the extracted presupposition as the question concept. The question concept is matched against items in the question answerer’s memory to find the reference concept. Once found, the reference concept serves as a source node for the more directed retrieval that comes in response to the question type.

The second component of question answering is determined by the question word and differs depending on what is being asked. Comprehension of the question word leads to application of rules specific to the question type. “Why,” for example, signals a causal antecedent or goal question and appropriate rules for searching memory for such information are applied. “When,” on the other hand, signals a time question and involves the activation and application of a different set of rules. Lehnert (1977, 1978) referred to this stage of question processing as question categorization.

As an example of the main components of question processing, consider Q1-Q3 below.

Q1. Why did Jim go to the store?
Q2. When did Jim go to the store?
Q3. Why did the tree fall down?

Q1 and Q2 share the same presupposition (that Jim drove to the store), and therefore will require activation of the same source node in memory. According to most question answering researchers, question processing will be essentially the same for Q1 and Q2 in the initial comprehension step that
involves parsing, generation of the question concept, and activation of the reference concept in memory. After this step the question words, *Why* and *When*, will trigger divergent retrieval processes as search begins for reasons or causal antecedents in the case of Q1 and times in the case of Q2.

There is an interaction between the question word and the presupposition of a question as a comparison of questions Q1 and Q3 reveals. Because the presupposition of Q1 is the action of an animate actor, the why-question may be answered by a goal or motivating state (e.g., "He wanted to buy bread" or "He was hungry"). Because the presupposition of Q3, that a tree fell down, is an event involving a nonanimate object, the why-question is more likely to be answered by a prior, causally related event or state (e.g., "A car hit it" or "It had been dead for 20 years"). Thus retrieval rules must utilize information gained from both the question word and characteristics of the question presupposition.

Question answering becomes even more interesting in situations where the answer information is not explicitly present in memory or when many connecting inferences are required. For example, even though it is an unmotivated event, "Because a tree fell on Jim’s kitchen" is a reasonable answer to Q1 if one assumes that Jim had a plan that was blocked by this event and decided to achieve his goal another way. Similarly, even though it is a psychological state, "Because the woodsman always hated it" is a reasonable answer to Q3 if one makes the connecting inference that the woodsman cut the tree down. In such situations knowledge-based inferences and strategic problem solving may be required. Retrieval heuristics associated with particular combinations of question words and presupposition types, and the generation of explanatory inferences in order to generate reasonable answers, are topics that have been explored in great detail (Lehnert, 1978; Graesser & Franklin, 1990).

Singer (1982, 1984a, 1984b, 1986) has examined question answering in some depth, proposing a series of models for question comprehension and memory retrieval. Singer’s VAIL model assumes that there is an initial “sentence encoding” stage that corresponds closely to encoding processes described in prior psycholinguistic research (Clark & Haviland, 1977; Kintsch, 1974). This stage is followed by processes that extract the given information in the question and determine whether the question is a verification (yes/no) or other (wh-) type of question. Input to the retrieval system is assumed to be a parsed representation of the question, and reaction time predictions under various question conditions were made based on processes hypothesized to follow parsing. Singer’s reaction time data were consistent with his predictions, yet subjects in these studies were presented with whole questions and reaction times were measured from the onset of question presentation to the beginning of answer generation. Thus question comprehension times and answer formulation times were combined in Singer’s dependent
measures such that there is no way to judge if the processes Singer hypothe-
sized are affecting answer retrieval time, as Singer suggests, parsing time, as
a parallel model might suggest, or both.

Graesser and his colleagues have been concerned for some time with the
interaction of question processes with different types of knowledge structures
and with the selection of an answer from a set of candidates (Graesser &
Clark, 1985; Graesser & Franklin, 1990; Graesser & Goodman, 1984; Graesser
& Murachver, 1984). Graesser has stressed description of the symbolic pro-
cedures that are applied in different knowledge contexts. He points out that
question-answering researchers have assumed that retrieval goes on in a
single, coherent knowledge representation. In Graesser & Murachver (1984)
and the QUEST model of Graesser & Franklin (1990) this work has shifted
toward an understanding of how diverse knowledge sources might be util-
ized in answering questions.

Graesser's models have also assumed an initial question parsing step that
yields a question concept. Relevant generic and episodic knowledge structures
are identified from the question concept and other contextual factors, and
retrieval takes place within these knowledge structures. As in other models,
question categorization operates to access rules relevant to a particular
question type. Rules associated with the question type are then applied within
the activated knowledge structures and may identify many candidate answer
concepts. A convergence mechanism narrows the candidate concepts within
the activated knowledge structures and applies constraints to further deter-
mine "good" answers. Finally, pragmatic considerations are brought into
play to determine an appropriate answer.

Graesser's model explains subjects' ratings of answer quality and many
memory phenomena revealed by question answering. The QUEST model
has interactive components, and Graesser, Lang, & Roberts (1991) have
recently studied the time course of question answering in the QUEST frame-
work. However, the model is still discussed as if it operates on conceptual
input derived after parsing. Graesser's mechanisms would be good candi-
dates for examination in the context of parallel comprehension and answer-
ing models.

1.2 Simultaneous Question Understanding and Answering

As we have seen, models of question processing have usually been presented
as a series of stages. Each process required for question processing—pres-
supposition extraction, source node activation, rule activation, rule applica-
tion, search, answer pruning, and inference—has been considered separately.
It is interesting that models of question answering have not explored highly
interactive mechanisms for question parsing and answer retrieval since this
has been a lively debate in the literature on word and sentence comprehension
for some time (Altmann & Steedman, 1988; Anderson, 1976, 1983; Anderson
& Bower, 1973; Collins & Loftus, 1975; Gorrell, 1989; Just & Carpenter,
1980, 1987; Kintsch, 1988; Marslen-Wilson, 1989; Marslen-Wilson & Tyler, 1980; McClelland & Kawamoto, 1989; McClelland & Rumelhart, 1981; Miikkulainen & Dyer, 1991; Perfetti, 1990; Rumelhart & McClelland, 1981, 1982; St. John & McClelland, 1990; Taraban & McClelland, 1990; Waltz & Pollack, 1985). In fact, debate in the sentence processing literature now centers less on whether structural and semantic processing interact, and more on how strong the interaction is (Altmann, Garnham & Dennis, 1992; Altmann & Steedman, 1988) or how autonomous the two processes are (Ferreira & Clifton, 1986; Perfetti, 1990; Taraban & McClelland, 1990). Just and Carpenter's "immediacy hypothesis" (1987) proposes that syntactic and semantic knowledge is constantly being applied to the input during sentence comprehension. In their view, multiple interpretations of a sentence's meaning are generated during reading and each new word adds constraints to the interpretation of a sentence. Kintsch's (1988) "construction-integration" model of sentence processing proposes that, in an initial "construction" phase of interpretation, a semantic network is generated to represent the concepts conveyed in a sentence. The construction phase is followed by an "integration" phase during which nodes are assigned weights according to constraints derived from the context.

In light of the fact that models of sentence comprehension routinely employ interacting syntactic and semantic interpreters, it is time to update question answering models by adding an interacting answer retrieval component. Data and models are present in the literature that suggest parallelism in simple query or query-like situations. Anderson (1974) provided evidence for spreading activation in a semantic network during sentence verification, a task closely related to verification question answering. Shastri (1988, 1989) presented a model in which fact queries, including recognition and property inheritance questions, are answered using a weighted semantic network with parallel, spreading activation. A query is represented in Shastri's model as a node network that affects activation levels in a semantic network in a manner that is specific to each query's content. "Response nodes" are activated by application of the query to the knowledge base, and the response node with the strongest weight becomes the answer to the question. Interestingly, even in this context Shastri describes question answering as a "three step process" involving

1. query handling to set up activation in the memory network,
2. spreading activation, and
3. competition among response nodes.

Miikkulainen and Dyer (1991) have incorporated a question answering module into a connectionist comprehension model that operates on a distributed representation, but the model deals only with yes/no questions.

Most work to date on parallelism in question answering has dealt with recognition, likelihood, or verification queries in the context of connectionist
models. These queries are relatively easily interpreted and matched to facts in the knowledge base. In the context of a modular, nonconnectionist model designed to deal with a wider range of question types, a strict serial view of question answering has only recently been challenged by a parallel view (Robertson, Ullman, & Mehta, 1992; Robertson & Weber, 1990; Robertson, Weber, Ullman, & Mehta, 1993). In the following sections we elaborate the model, provide a walkthrough, and bring together data from several experiments that support its basic structure.

2. A NEW QUESTION-ANSWERING ARCHITECTURE

In this section an architecture is proposed which will support question comprehension in a framework that allows simultaneous question parsing, activation and application of retrieval heuristics, answer formulation and output. The model is referred to as "TSUNAMI," for "Theory of Simultaneous UNDERstanding Answering and Memory Interaction," and it differs dramatically from existing proposals for symbol-based question-answering architectures (i.e., Dyer, Graesser, Lehnert, Singer). At this point the model is offered as a broad architecture for supporting highly interactive application of the mechanisms already identified by question answering researchers. It remains to be seen how the nature of these mechanisms will change, and what new mechanisms might be necessary, when implemented in the TSUNAMI framework.

2.1 Structure of the Model

The TSUNAMI model is depicted in Figure 1. In the Figure, processes are indicated by ovals and memories are indicated by squares. There are several separate memory components in the model, however there is no strong claim that they are separate from each other. Rather, the different memory stores are really claims about different types of data with which the various processes are concerned. Arrows pointing to a memory from a process indicate that the process generates data of that type and can modify data of that type. Arrows pointing to a process from a memory indicate that the process uses information of that type. Processes and memories may interact in different ways and on different problems (e.g., the parser may be building two parsing structures at once, or a retrieval process may be utilizing several structures in episodic memory at the same time). There are no direct linkages between processes, rather, in the spirit of "blackboard" models (Erman, Hayes-Roth, Lesser, & Reddy, 1980; Hayes-Roth, 1985), the behavior of each process depends on the state of the memory or memories with which it

---

1 A "tsunami" is a tidal wave that is caused by an earthquake under the ocean.
interacts. Also, like blackboard models, items stored in a particular memory may be inspected and altered by several processes operating on it at once. Thus, there is no formal "flow of control" to the model. Instead, the processing behavior is a function of the input, the states of the memories, and the speed of the processors.

Figure 1 distinguishes between semantic and episodic knowledge and is predicated on a situation in which questions are asked about a particular event (e.g., a story or a personal memory) that would be represented in episodic memory. Two special purpose working memory components are also
isolated. One memory stores question candidates, or possible interpretations of the question. The other stores answer candidates, or potential answers to the question candidates. The candidates are represented as propositions and may be partially specified. The question candidate memory and answer candidate memory are the only knowledge structures that take output from processes in the model. The influences of processes on these memories might be to add propositions, update proposition contents, or delete propositions. While in the model we take the position that the question and answer buffers are separate memories, this is not a strong architectural claim. In fact, they may all be part of a single working memory with data elements tagged in some way that discriminates questions from answers. Either view is consistent with the model as presented and with the experimental data in section 3.

The parser takes input from the question and utilizes grammatical, case, and pragmatic information in semantic memory to begin producing various propositional representations which are then stored in the question candidate buffer. The position taken by the TSUNAMI model is much like that of Just and Carpenter (1987) in that each word provides information about possible interpretations of the sentence. Multiple arrows from the parser into the question candidate buffer suggest that the parser can produce many candidates in response to the input at any given time. As words come into the parser, prior candidate structures may be updated or disconfirmed and deleted by the parser.

As soon as there is any information in the question candidate buffer, a matcher begins comparing question candidate structures with activated information in episodic memory. The goal of the matcher is to find the question presupposition which will ultimately form the source node from which retrieval begins. The matching process occurs in parallel for several candidates as indicated by multiple arrows from both the question buffer and episodic memory into the matcher. It is reasonable to conceptualize the matcher in a manner similar to the connectionist fact retrieval mechanisms described by Shastri (1988, 1989) or Miikkulainen and Dyer (1991). Evidence presented in section 3.2 suggests that the speed of the matcher is influenced by the number of candidates and the number of matches.

When the matcher finds a proposition in episodic memory that corresponds to a question candidate, then this is identified as a likely source node for answer retrieval processes. It is proposed that a presupposition match in episodic memory raises the activation level of this and related information and makes it more available to future analysis by any other processes that utilize episodic memory. This influence of the matcher on episodic memory is indicated by an arrow from the matcher into episodic memory.

The presupposition matching process may also influence the contents of the question buffer. If there are multiple question candidates, then the iden-
tification of a matching presupposition for one will lessen the influence of the others in further processing. Partial propositions in the question candidate memory can be matched to complete propositions in memory, thereby updating the question candidate list to include expectations. Influences of the matcher on question candidates are indicated by an arrow from the matcher into the question candidate memory.

As soon as there is information in the question candidate buffer that might suggest a question or question type, the answer retrieval process can begin. This mechanism examines question candidates, attempts to determine question categories appropriate to the various candidates, and begins applying retrieval heuristics in episodic memory. The answer retrieval mechanism utilizes question answering rules stored in semantic memory and other, relevant general knowledge. This process operates simultaneously with the parser and the matcher, but is dependent on the contents of the question candidate memory and the activation states of nodes in episodic memory.

The answer retrieval mechanism operates to highlight relevant portions of episodic memory. If one of the question candidates is a goal-orientation question, for example, then the answer retrieval mechanism may activate goal hierarchies in the episodic trace. On the other hand, if a causal-antecedent question candidate is present in the question buffer, then causal sequences may become more active. Both types of questions may be present in the buffer, especially early in question parsing, in which case the retrieval mechanism will activate both types of knowledge structures and initiate appropriate retrieval heuristics in each. The heuristics utilized by the answer retrieval process are those already identified by question-answering researchers for various question types, however, their implementation may need to be augmented to deal with differential weights of information in episodic memory, likelihoods assigned to different question candidates, constraints imposed by incoming information, and potential interactions among different heuristics.

The outputs of the answer retrieval process are candidate answers to questions that the system finds consistent with the input and memory and any given time. Potential answers produced by the answer retrieval process are stored in the answer candidate buffer as propositions. These propositions may also be partially specified, and are subject to subsequent modification and deletion by the operation of the answer retrieval process. For example, if a question candidate that spawned a retrieval process is later disconfirmed, then the answer candidate built by that process will be deleted by the answer retrieval process.

The answer candidates are examined by an output generation process which also utilizes grammatical, case, pragmatic, and relevant general knowledge in semantic memory. This process can influence the answer candidate set. For example, if pragmatic concerns dictate that an answer is inappropriate, then the output generation process will delete it from the answer
TABLE 1
An Example Story and Associated Propositional Analysis

<table>
<thead>
<tr>
<th>John woke up early.</th>
<th>P1. (ACT wake-up, John)</th>
</tr>
</thead>
<tbody>
<tr>
<td>John was hungry.</td>
<td>P2. (TIME early, P1)</td>
</tr>
<tr>
<td>John wanted some food.</td>
<td>P3 (STATE hungry, John)</td>
</tr>
<tr>
<td>John drove to the store.</td>
<td>P4. (ACT eat, John, food)</td>
</tr>
<tr>
<td></td>
<td>P5. (REL reason, P4, P3)</td>
</tr>
<tr>
<td>Mary saw John at the store.</td>
<td>P6. (ACT drive, John, store)</td>
</tr>
<tr>
<td></td>
<td>P7. (REL reason, P6, P4)</td>
</tr>
<tr>
<td>Mary wanted to say hi to John.</td>
<td>P8. (ACT see, Mary, John)</td>
</tr>
<tr>
<td>She waved at him.</td>
<td>P9. (STATE location, P8, store)</td>
</tr>
<tr>
<td>John bought oatmeal.</td>
<td>P10. (ACT greet, Mary, John)</td>
</tr>
<tr>
<td>John dropped the oatmeal outside</td>
<td>P11. (ACT wave-at, Mary, John)</td>
</tr>
<tr>
<td></td>
<td>P12. (REL reason, P11, P10)</td>
</tr>
<tr>
<td>It spilled in the trunk.</td>
<td>P13. (ACT buy, John, oatmeal)</td>
</tr>
<tr>
<td></td>
<td>P14. (REL reason, P13, P4)</td>
</tr>
<tr>
<td>John put it back in the container.</td>
<td>P15. (EVENT drop, John, oatmeal)</td>
</tr>
<tr>
<td>Later he cooked it,</td>
<td>P16. (STATE location, P15, outside)</td>
</tr>
<tr>
<td>and ate it.</td>
<td>P17. (STATE slick, parking lot)</td>
</tr>
<tr>
<td></td>
<td>P18. (REL cause, P15, P17)</td>
</tr>
<tr>
<td></td>
<td>P19. (EVENT spill, oatmeal, trunk)</td>
</tr>
<tr>
<td></td>
<td>P20. (REL cause, P19, P15)</td>
</tr>
<tr>
<td></td>
<td>P21. (ACT put, John, oatmeal, container)</td>
</tr>
<tr>
<td></td>
<td>P22. (ACT cook, John oatmeal)</td>
</tr>
<tr>
<td></td>
<td>P23. (REL reason, P22, P4)</td>
</tr>
<tr>
<td></td>
<td>P24. (ACT eat, John, oatmeal)</td>
</tr>
<tr>
<td></td>
<td>P25. (REL reason, P24, P4)</td>
</tr>
</tbody>
</table>

buffer. Finally, when one candidate remains in the answer buffer and all of the question has been input to the parser, the final answer is formulated. It is reasonable to assume that the output generation mechanism will not commit to a final interpretation until all of the input has been processed since a final phrase on a question can change its focus, and hence the appropriate answer, tremendously.

2.2 Walkthrough

In this section a traditional approach to question processing is contrasted with TSUNAMI in an example walkthrough. The comparison is meant to illustrate general points and so many details are necessarily glossed. The walkthrough, however, should give a clear picture of how drastically different previous serial models and the TSUNAMI model are. For the walkthrough the story in the left-hand portion of Table 1 will be used.

Following Kintsch and other researchers in text comprehension, I claim that the list of 25 propositions (designated as P1-P25) in the right-hand portion of Table 1 captures something of the meaning of the explicit details in the text. In this list, each proposition has a type indicator in capital letters in
the first position. Type indicators are important for question answering since the determination of question categories often depends on the type of proposition being queried and since search is often directed along nodes of a particular type. In this example there are node types for states (STATE), goal-directed actions (ACT), and parts of causal sequences (EVENT). The node type REL indicates a proposition linking two other propositions by virtue of some relation. The predicate of each proposition is contained in the second argument. As in most propositional analyses of stories the predicates basically correspond to verbs and modifiers. The remaining slots of the propositions contain arguments to the predicates, for example actors and objects. The predicate of an REL node is the name of a relation. In the example there are reason predicates for intentional relationships between goals and actions/subgoals, and cause predicates between causally linked events. The final two arguments to REL nodes are the identifications of the linked nodes.

I make no strong claims about the representational system or notation, but use it only for explication in this section. Note that the proposition list in Table 1 contains implied linking propositions about goal relations (P5, P7, P14, P23, and P25 for John, and P12 for Mary) and causal relations (P20). Considerable research on story comprehension has suggested that such inferences are constructed during reading (Long, Golding, & Graesser, 1992; Seifert, Robertson, & Black, 1985) and that they provide connectivity among propositions in the internal representation which is useful in recall and reasoning about character actions (Graesser & Clark, 1985; Kintsch, 1974). Also, intentional actions and non-intentional events are explicitly distinguished by the notation, as Graesser (1981, 1985) might have it, but I leave the possibility open that this distinction might be made in a different manner.

2.2.1 Traditional Serial Models
In the context of this story, consider the question “Why did John drive to the store?,” and the derivation of a reasonable answer, “To get some food.” In serial models of question answering, the presupposition of the question would first be derived from a complete parse of the input sentence which we might represent as the proposition Q4:

Q4. (ACT drive, John, store)

A question categorization mechanism would next determine that the “Why” word suggests either a causal antecedent question category or a goal orientation question category. Examination by this mechanism of the question presupposition reveals the facts that “drive” is a motivated action and that “John” is an animate actor, either of which would dictate that the question
category is goal orientation. After question categorization, further analysis of the pragmatics of the situation or language conventions helps to focus the query and might even alter the question categorization. For example, if the question answerer knows that John lives next door to the store, then the question is about why he drove rather than walking. If the question answerer knows that Mary always does the shopping, then the question is about why John was the one who went to the store. In this case, context offers no alternative foci so the emphasis is assumed to be on John’s goal.

The question concept and question category are passed to a process that will locate the question concept in memory and apply appropriate retrieval heuristics. Thus, the next step is to match Q4 against the propositional representation in Table 1, thereby activating P6 as a source node for retrieval. Retrieval rules specify that propositions related to a source node via a reason predicate will be possible answers to a goal orientation question. Further, additional candidates can be generated by following a sequence of reason predicates up a goal hierarchy. In our example, retrieval rules would attempt to find a proposition matching the following retrieval pattern:

\[(\text{REL reason}, \ P6, \ ?P)\],

where \(?P\) is a variable. The pattern describes an explicit reason relation between P6 and another, as yet unknown proposition. If found, the proposition bound to \(?P\) would be an answer candidate. P7 in Table 1 matches the question pattern, binding \(?P\) to P4 and therefore leading the retrieval mechanism to P4:

\[
P7. \ (\text{REL reason, P6, P4}), \text{ matches the retrieval pattern.}
\]

\[
P4. \ (\text{ACT eat, John, food}), \text{ is an answer candidate and a new source node.}
\]

Thus, “John wanted to eat some food” is a potential answer. The retrieval mechanism can continue to follow the goal hierarchy, searching for a reason predicate involving P4. The retrieval pattern for such a search is as follows:

\[
(\text{REL reason, P4, ?P}).
\]

P5 matches this specification and leads to P3:

\[
P5. \ (\text{REL reason, P4, P3}), \text{ matches the retrieval pattern.}
\]

\[
P3. \ (\text{STATE hungry, John}), \text{ is an answer candidate and a new source node.}
\]

Thus, “John was hungry” is another potential answer.

There are no further reason links (i.e., there is no match to \((\text{REL reason, P3, ?P})\)), and so two answer candidates are activated. Many considerations about the appropriateness, or “goodness,” of the answers may now be applied in order to choose one (Graesser & Clark, 1985; Graesser & Franklin, 1990). When one answer is finally determined, an output mechanism formulates the linguistic response.
Walkthrough of TSUNAMI's Processing of the Question
"Why Did John Drive to the Store?" in the Context of the Story in Table 1

<table>
<thead>
<tr>
<th>Time</th>
<th>Input</th>
<th>Question Candidates</th>
<th>Memory Nodes</th>
<th>Answer Candidates</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>Why</td>
<td>C111.(REL reason, ?Pa, ?Pb) P5</td>
<td>P7, P14, P23, P25</td>
<td>P3 P4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C211.(REL cause, ?Pc, ?Pd) P18</td>
<td>P20</td>
<td>P17 P15</td>
</tr>
<tr>
<td>t2</td>
<td>did</td>
<td>(no change)</td>
<td>(no change)</td>
<td>(no change)</td>
</tr>
<tr>
<td>t3</td>
<td>John</td>
<td>C112.(REL reason, C312, ?Pb) P5-P4</td>
<td>P7-P6, P14-P13, P23-P22, P25-P24</td>
<td>P3 P4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C312.(ACT ?a, John, ?object1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>C212.(REL cause, C412, ?Pd) P18-P15</td>
<td>P17</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>C412.(EVENT ?e, John, ?object2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>t4</td>
<td>drive</td>
<td>C114.(REL reason, C314, ?Pb) P7-P6</td>
<td>P4, P3</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>C314.(ACT drive, John, ?object1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>t5</td>
<td></td>
<td>C115.(REL reason, C315, C515) P7-P6</td>
<td>P4</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>C315.(ACT drive, John, store)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>C515.(ACT eat, John, food) P5-P4</td>
<td>P3</td>
<td></td>
</tr>
</tbody>
</table>

Note that in analysis of answer retrieval only a few of the concepts in the story were activated. Specifically, P6, the source node, and P3, P4, P5, and P7. A question answering system that waits until a complete representation of the question is obtained before retrieval begins maximally constrains the search process.

2.2.2 The TSUNAMI Model
Now let us consider how TSUNAMI would handle the same question, "Why did John drive to the store?" In many ways the decision rules and retrieval heuristics are similar, but retrieval begins immediately based on partial understandings of the question as it is parsed. Because we assume that retrieval and parsing occur together, the walkthrough will be considered in a word-by-word manner. In the TSUNAMI example, we will need to specify the states of the question and answer buffers, the activity in story memory, and the operation of some processes as each word is input. Table 2 shows this activity at various stages in parsing. Table 2 is organized by time slices, indicated as t1-t5, which show the state of the question answering process after each word is processed. The six columns of Table 2 show the time slice, the input word, the contents of the question buffer, the active memory nodes,
and the contents of the answer buffer. Examination of Table 1 and Table 2 together will be helpful in following the discussion.

It is proposed that the parser may place patterns in the question buffer which are consistent with likely interpretations of the input at any point. This view is consistent with the "immediacy hypothesis" of Just and Carpenter (1980, 1987) and contrasts with approaches that allow semantic processes to operate only when the syntactic processor is not occupied, for example, at clause boundaries. By allowing each word to add constraints to the question interpretation, the retrieval process can narrow its search earlier. The cost is to initiate fruitless searches early on, but we provide evidence in section 3.1 that this cost is outweighed by the headstart of the retrieval process that ultimately "wins."

The matcher and answer retrieval processes will begin to use information in the question buffer as soon as it appears. First the question word "Why" is input to the parser, as the input column of Table 2 shows at time t1. Already, there are two possible patterns that might be placed in the question buffer. These question candidates are designated as C1 and C2. Since the form of the question candidate propositions will change, C1 and C2 are further specified with the subscript t1 to indicate the time:

\[
C_{1t1}. \ (\text{REL reason, } ?Pa, ?Pb) \\
C_{2t1}. \ (\text{REL cause, } ?Pc, ?Pd)
\]

The node designations appear at time t1 in the question candidates column of Table 2. Both question candidates are propositions that specify a relationship between two nodes. C1t1 specifies a reason relation while C2t1 specifies a causal relation. These patterns correspond respectively to the goal orientation and causal antecedent interpretations of "Why" questions. The symbols ?Pa, ?Pb, ?Pc, and ?Pd stand for the propositions that are connected by the reason and cause relations. The symbols are preceded by question marks to indicate that they are variables which have not yet been specified by the input or matched to any memory items.

In general, in response to a question word, all of the question-type patterns that are highly associated with the question word will be generated by the parser and used by the matcher and retrieval process. Nonetheless, it is possible for the matcher to begin a comparison with memory, making all possible matches to the variables. C1t1 and C2t1 match all reason and cause predicates that are active in memory. Such a comparison reveals that C1t1 matches six propositions in memory, namely P5, P7, P12, P14, P23, and P25. C2t1 matches two propositions, namely P18 and P20. If P18 and P20 were not present in the story (e.g., if the oatmeal dropping incident were eliminated) then it would be possible to rule out the causal antecedent interpretation of the question in this case before seeing any more of the question! Once ruled out, the matcher would remove C2t1 from memory and this pattern would play no more role in processing.
Operation of the matcher has resulted in increased activation of eight nodes among the 25 nodes which form the story representation in episodic memory. This increased activation makes those nodes more likely to be used by retrieval processes, much like the activation level of nodes in Anderson's ACT* model (Anderson, 1983) affect the speed of production rule matching. It is possible for the answer retrieval mechanism to begin examining these nodes at this stage. In the representational scheme in Table 1, the superordinate member of a pair of nodes related by a reason link can be found in the last slot of a reason proposition. Thus, the answer retrieval mechanism will tag all matches to ?Pb among the active memory nodes and treat them as potential answers. Matches to ?Pb for each of the six activated reason nodes are as follows: the P5 node matches ?Pb to P3 ("John was hungry"), the nodes P7, P14, P23, and P25 all match ?Pb to P4 ("John wanted some food"), and the node P12 matches ?Pb to P10 ("Mary wanted to say hi to John"). The matched nodes appear in the answer candidate column of Table 2 at time t1. There is an interesting empirical issue here concerning whether P4 should be a "stronger" answer candidate since it is pointed to by several propositions. A priming study could be designed to examine such an issue.

Causal antecedents can be found in the last slot of propositions with cause predicates. These match to ?Pd in C2(t). The answer retrieval mechanism will tag all matches to ?Pd among the active memory nodes and treat them as potential answers. Matches to ?Pd for the two activated cause nodes are as follows: the P18 node matches ?Pd to P17 ("The parking lot was slick"), and the P20 node matches ?Pd to P15 ("John dropped the oatmeal"). These nodes also appear in the answer candidate column of Table 2.

From this example we see that it possible to identify two potential question types and generate five unique answer candidates even at the first word of the question in this example. The ability of the question answering system to move this far on the first word would depend on available resources. It seems unreasonable to assume that this much information could be held in memory at once or that this much processing could go on in the brief time it takes to read a familiar word like "why." However, the system should be expected to move as far as possible at each step (Just & Carpenter, 1987). The values of various memory and processing parameters will be left to future empirical investigation.

At t2 "did" is input to the matcher. While the parser performs a syntactic analysis of this word, it plays no role in updating the question or answer buffers since it contributes no new constraints on the interpretation of the question. In the time it takes to read the word, however, processes that were set in motion during reading of the first word but which were not finished by the time the second word appeared could be continuing.

At t3 "John" is input to the parser. John could be a subject, as in "Why did John hit a tree?," or an object, as in "Why did John get hit by a tree?" If we make the assumption that the parser follows the path of least resistance,
building the simplest structural representation of the sentence (Frazier, 1987), then "John" is treated as the likely subject. In this case, the proposition being queried is likely to be an action with John as the actor or an event with John as the subject. To represent these concepts the parser adds two new candidates to the question buffer at time t3, C3t3 and C4t3:

\[
\begin{align*}
C_{3t3} & : (\text{ACT ?a, John, ?object1}) \\
C_{4t3} & : (\text{EVENT ?e, John, ?object2})
\end{align*}
\]

and updates C1t3 and C2t3 to refer to the new propositions:

\[
\begin{align*}
C_{1t3} & : (\text{REL reason, C3t3, ?Pb}) \\
C_{2t3} & : (\text{REL cause, C4t3, ?Pd})
\end{align*}
\]

The updated contents of the question buffer are indicated in the question candidate column in Table 2 at time t3.

The matcher responds immediately by activating all pairs of propositions in episodic memory that can be matched to the new constraints of the question buffer. The new constraint on the actor slot of C3t3 results in the elimination of P12 as an item activated by the matcher and P10 ("Mary wanted to say hi to John") as an answer candidate since the actor in P10 is Mary. For the same reason, P19 fails to match C4t3 and is eliminated, resulting in the removal of P15 ("John dropped the oatmeal") from the answer candidate buffer. Remaining memory items that match the constraints of the question buffer at time t3 are the action pairs P5-P4, P7-P6, P14-P13, P23-P22, and P25-P24. Also remaining is the event pair P18-P15. Answer candidates that remain at time t3 and P3 ("John was hungry"), P4 ("John wanted some food"), and P17 ("The parking lot was slick").

At time t4, "drive" enters the parser and C3t4 is updated in the question buffer to reflect identification of the action:

\[
\begin{align*}
C_{1t4} & : (\text{REL reason, C3t4, ?Pb}) \\
C_{2t4} & : (\text{ACT drive, John, ?object1})
\end{align*}
\]

Up to this point there has still been a reasonable expectation of the event question "Why did John drop the oatmeal?,” represented by C4t3 and C2t3. However, this interpretation cannot be reconciled with the verb "drive," and so it is dropped from the question buffer. This deletion results in deactivation of the P18-P15 node pair in memory and the removal of P17 ("The parking lot was slick") from the answer buffer.

At time t4 a comparison of the contents of the question buffer with story memory shows only one way that the constraints of the question buffer can be satisfied. Specifically, C3t4 matches P6 ("John drove to the store") and C1t4 matches its parent node, P7. The matcher can now reduce the activation level of all the other pairs of propositions and give all resources to P6 and P7. The match of C3t4 to P6 unambiguously binds the missing object
slot (?object1) to "store." Also the match of Clt4 to P7 unambiguously binds to ?Pb slot of Clt4 to P4.

Since ?Pb in Clt4 is bound to P4, this is added to the question buffer at t5 as C5t5 and Clt5 is updated to reflect the embedded relationship. These changes, and the matched ?object1 slot in C3t4, are reflected in the question candidate column of Table 2 at time t5:

C1t5. (REL reason, C3t5, C5t5)
C3t5. (Act drive, John, store)
C5t5. (ACT eat, John, foot)

The answer retrieval mechanism responds to the current configuration of the question buffer by eliminating answer candidates that can not be derived from P6 by reason links. P4 remains as an answer candidate because it is related to P6 by virtue of the direct match in P7. P3 remains as answer candidate because it is related to P6 indirectly through P4.

The parser continues to receive input from the question, but unless future interpretations contradict the expectation generated by the matcher there will be no more influence on the question buffer. If the question buffer does not change, then the matcher will not operate beyond this point and this frees resources for other processes.

As soon as propositions appear in the answer candidate buffer their appropriateness can be evaluated by the output generation mechanism. As with the other processes, the operation of this mechanism is affected by the number and strength of answer candidates, the resources shared with other processes, the complexity of the pragmatics of the interaction, and the complexity of the answer candidates' contents. The number of answer candidates varied from five at time t1, to two at time t5. If resources were available, the output generation mechanism could be evaluating all five answer candidates at time t1, applying "goodness of answer" criteria to all of them. It is reasonable to assume, however, that this mechanism might be delayed until a small number of candidates are being considered.

The two answer candidates identified by time t5 remain viable candidates after the complete question has been read and, indeed, either is a reasonable answer to the question. Perhaps the question answerer would decide that P3 is better than P4 according to some obviousness criterion since "buying food" is a more obvious, default goal than being hungry. Such a criterion could be applied as early as t4 in our example since the final answer candidates are present at this time and other processes and memory resources have begun to require less capacity. The final answer could be ready, then, just after reading the verb in our example.

The TSUNAMI model is expectation-driven at every step. The parser is predicting possible interpretations of the input and the answer retrieval mechanism is predicting possible answer categories. The matcher will often
find several nodes in episodic memory that are consistent with partially specified question candidates, and thus produce "guesses" at question completions. The output generation mechanism may be ready to give several answers at any given time. Because any process can operate at any time, control is exerted through the contents of the memories and buffers. This is a radical departure from previous question-answering models in which control is exerted through the serial nature of the architectures.

The walkthrough should be considered as an example of possible TSUNAMI behavior. The actual behavior of the model will vary depending on many factors, for example the speed of input to the parser, the form of the question being input, the intentions of the question answerer, the memory load from the specific contents of the episodic trace being examined. In the next section we manipulate some of these factors and make differential predictions about the behavior of a model like TSUNAMI in contrast to serial models of question processing.

3. EXPERIMENTS

In this section several experiments are discussed which provide evidence that answer-related processes are operating during question comprehension. In all of the experiments subjects read and answered visually presented questions about short stories. The experiments utilize several convergent measures of mental workload, or the demand placed on cognitive resources by on-going processes and the extent of active memory. There is an assumption in the TSUNAMI model (and most cognitive models) that the various processes compete for resources so that the more there is for any one process to do, the slower all processes will operate. This demand should be apparent in slowed reading times or slowed reaction times to secondary tasks like lexical decision.

In the experiments to follow, increased workload is shown during parsing when the question type is known vs. unknown, when the answer may be uniquely identified early vs. later, and when the subjects’ intent is to provide an answer vs. a paraphrase. In addition, in most cases the added workload during parsing is accompanied by a savings in answering time at the end of the question. This suggests that the workload is in fact associated with answer retrieval. Finally, in all of the experiments the workload effects differ with question type, again suggesting that they are related to question-specific retrieval processes.

3.1 Position of the Query Component of Questions

In a series of experiments reported in detail in Robertson & Weber (1990) and Robertson, Weber, Ullman, & Mehta (1993) the position of the question component relative to the presupposition of a question was manipulated.
and effects on reading and answering times were measured. Consider the following questions Q5 and Q6 in which the question word appears either at the beginning or end of the main part of the question.

Q5. For what reason did Pete go to the store?
Q6. Pete went to the store for what reason?

When answering Q5, the reader has early knowledge of the question type and so, if the human question answering system is designed for it, answer retrieval heuristics might be activated and even applied during reading of the presupposition. In Q6, however, the question type is unknown until the last word. Thus Q5 forces a staged process in which source node activation precedes directed retrieval.

If comprehension and retrieval processes are staged during normal question answering, then there should be no difference in the reading and answer retrieval times between Q5 and Q6 since both would require source node activation followed by search rule application and retrieval. If retrieval begins early in Q5, however, then reading time for the words might increase relative to Q6 since retrieval and comprehension resources are competing. Also, the time to generate an answer after reading the last word of Q5 might be faster relative to Q6 since retrieval has a head start in the former case.

Two experiments were conducted in this paradigm. In each experiment, subjects read a sequence of short stories presented on a computer screen. Each story was followed by a question which was presented word-by-word at the subjects' normal reading rate in response to self-paced keypresses that revealed each new word. Subjects answered verification questions, goal questions, and time questions.

In the first experiment, the questions were presented as a single question word and an assertion, and the order of these two question elements was varied. In the question-first (Q-first) condition, subjects saw questions of the form “Question-word? + Presupposition,” and in the question-last (Q-last) condition subjects saw questions of the form “Presupposition + Question-word,” where the question words were either “True?,” “Why?,” or “When?,” and the presupposition was an active sentence, that is, Noun1 + Verb + Preposition + Determiner + Noun2. For example, a subject might have seen “Why? John went to the store.” or “John went to the store. Why?” In the second experiment, the question words were replaced by phrases. In the question-phrase-first (QP-first) condition subjects saw “Isn’t it true that . . . ,” “What was the reason that . . . ,” and “At what time did . . . ,” before the question presuppositions. In the question-phrase-last (QP-last) condition subjects saw “. . . isn’t it true?,” “. . . for what reason?,” and “. . . at what time?” after the question presuppositions. Subjects read the questions one word at a time at their own pace by pressing a key to see each new word. Their reading times were measured.
Example Stimuli for the Three Question Types and Two Question-Position Conditions in the Query-Position Experiments. Reading Times Were Collected for the Words in Bold Type. Answering Times Were Calculated from the Words in Underlined Bold Type.

<table>
<thead>
<tr>
<th>Example</th>
<th>Verification Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q-first</td>
<td><strong>True? Pete drove to the store?</strong></td>
</tr>
<tr>
<td>QP-first</td>
<td><strong>Isn't it <strong>true</strong> that Pete drove to the store?</strong></td>
</tr>
<tr>
<td>Q-last</td>
<td><strong>Pete drove to the store, True?</strong></td>
</tr>
<tr>
<td>QP-last</td>
<td><strong>Pete drove to the store isn't it true?</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
<th>Reason Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q-first</td>
<td><strong>Why? Pete drove to the store?</strong></td>
</tr>
<tr>
<td>QP-first</td>
<td><strong>What was the reason that Pete drove to the store?</strong></td>
</tr>
<tr>
<td>Q-last</td>
<td><strong>Pete drove to the store, Why?</strong></td>
</tr>
<tr>
<td>QP-last</td>
<td><strong>Pete drove to the store for what reason?</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
<th>Time Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q-first</td>
<td><strong>When? Pete drove to the store?</strong></td>
</tr>
<tr>
<td>QP-first</td>
<td><strong>What was the <strong>time</strong> that Pete drove to the store?</strong></td>
</tr>
<tr>
<td>Q-last</td>
<td><strong>Pete drove to the store, When?</strong></td>
</tr>
<tr>
<td>QP-last</td>
<td><strong>Pete drove to the store at what <strong>time</strong>?</strong></td>
</tr>
</tbody>
</table>

Dependent measures in all of the experiments were word-by-word reading times for the presuppositions and "answer time." Answer time was measured as the time to read the question word or parts of the question phrase plus time to read the last word of the presupposition. Table 3 shows the source of reading and answering time dependent measures. Note that the answering time always includes time to read the same two words plus time before pressing the response key "when the answer comes to mind."

Figures 2 and 3 show the mean word-by-word question presupposition reading times for all three question types in the two experiments respectively. In all cases the overall time to read was slower when the question word or question phrase came first. The overall mean times combined across words in the conditions where the question type was known and not known were 379ms vs. 359ms, \( F(1,23) = 9.90, p < .01 \) in the first experiment and 500ms vs. 390ms, \( F(1,44) = 44.89, p < .001 \) in the second experiment. The cost of knowing the question type was measured by dividing the difference between the conditions, an estimate of the absolute cost per word, by the mean time taken when the question type was not known, an estimate of absolute reading time per word that does not include the cost. The costs were 6% and 28% per word in the two experiments respectively.
Figure 2. Mean word-by-word reading times for three question types when the question word appeared first or last (adapted from Robertson, Weber, Ullman, & Mehta, 1993).

Figure 3. Mean word-by-word reading times for three question types when the question phrase appeared first or last (adapted from Robertson, Weber, Ullman, & Mehta, 1993).
Table 4 shows the mean answer retrieval times for verification, reason, and time questions in the two experiments. As predicted by the parallel model, answer retrieval was faster when the question word or phrase came first. Knowledge of the question type gave subjects a 507ms retrieval advantage, $F(1,23) = 10.90, p < .01$ in Experiment 1 and a 528ms retrieval advantage, $F(1,41) = 6.73, p < .05$ in Experiment 2. Despite large differences in absolute answering times across the two experiments, the savings in time enabled by knowledge of the question type was measured by dividing the difference between the two conditions, an estimate of absolute savings, by the mean time taken when the question type was not known, an estimate of absolute time that does not include the savings. The savings were astonishingly consistent in the two experiments, 17% and 16% respectively.

In the first experiment, the comprehension cost for the presupposition was small relative to the savings at retrieval time. In the second experiment, the cost seems out of proportion to the savings. However, because the cost is extracted on relatively short reading times whereas the savings is achieved on the longer retrieval times, in the end there was an absolute savings in total time to read and comprehend a question in all cases when the question type was known. The overall costs over the four words of the presuppositions were 80ms and 444ms in the two experiments, whereas the savings were 507ms and 528ms. These yield cost/savings ratios of .16 and .84 respectively, all impressive advantages for simultaneous question comprehension and answer retrieval.
These experiments provided strong support for the view that processes related to answer retrieval operate during question comprehension. These processes extract a cost on reading time (sometimes high) but offer a savings at retrieval time. The TSUNAMI model is consistent with the data in the above experiments. When the question words or phrases came last, the answer retrieval component of the model could not operate during reading. Only the matcher and the parser operated during this time. Also, nothing would have been stored in the answer candidate buffer during reading and no use would have been made of question answering rules. In contrast, when the question word or phrase came first, the answer retrieval processor could operate. This would increase the contents of the question candidate buffer and the answer candidate buffer. With data in the answer candidate buffer the output preparation processor would begin to operate. Assuming limited resources, the simultaneous operation of these processes would take a toll on the parser and increase reading times. In addition, since the processes of the answer retrieval mechanism depend on the contents of the question candidate buffer, questions requiring more complex retrieval will increase reading times more. This effect was demonstrated in Experiment 2, where verification questions were read faster than reason questions, which in turn were read faster than time questions.

Regarding answering times, the conditions in which the question words or phrases were last (Q/QP-last) allowed the question candidate buffer to contain a single item by the time the question component was reached. At this time, although the answer retrieval mechanism had only one input from the question buffer it would begin search in episodic memory from scratch and generate several answer candidates. These candidates were examined at the same time by the output preparation processor. Thus in these conditions, when the last word was read the answer and output processes operated simultaneously and on a considerable amount of data. In contrast, when the question words or phrases came first, many answer candidates had already been generated and eliminated by the time the parser reached the last word of the question. Thus, at the end of a question in these conditions there was less left to be done by the answer and output processors and fewer data items to process. This explains the faster answering times in these conditions.

3.2 Number of Answers

3.2.1 Introduction

Question answering is a process that requires considerable interaction with knowledge structures relevant to the question. Graesser & Murachver (1984) & Graesser & Franklin (1990) have pointed out that one step in the question answering process is the identification of relevant knowledge sources. In the
TSUNAMI framework the identification of knowledge sources occurs during comprehension. As the walkthrough in section 2.2.2 demonstrated, increasingly specific portions of a knowledge structure are activated as more and more of a question is being read. If this is true, then changes in the content of knowledge structures will affect question processing. If these effects are apparent during reading of the question, then question-related knowledge structure activation will be shown to be occurring during parsing.

The course of question processing under different knowledge structure contexts may be studied by varying the number of potential answers there are to a question at different points during comprehension. As an example, consider the following story in which John drives to the store three times for three different purposes while Mary drives to the store once:

- John drove to the store to buy bread on Monday.
- John drove to the store to buy milk on Tuesday.
- John drove to the store to buy cheese on Wednesday.
- Mary drove to the store to buy eggs on Thursday.

In the context of the above story, consider the following questions:

- Q7. Why did John drive to the store on Tuesday?
- Q8. Why did Mary drive to the store on Thursday?

In answering Q7 it is impossible to identify the exact source node in memory that corresponds to the question presupposition until the end of the question. In Q8, however, it is possible to identify the unique source node in memory as early at the subject noun, "Mary." In a question answering architecture with parallelism, like TSUNAMI, answer retrieval heuristics should begin earlier when reading Q8 than Q7. In strict serial models, however, source node activation and application of retrieval heuristics are always delayed until after question parsing, and thus no effects of the uniqueness of the source node concept should be observed during reading of a question.

It is again assumed in this experiment that retrieval processes compete with parsing processes for cognitive resources. If this is true, then the operation of retrieval processes should be evident in increased reading times. Thus the prediction in the above example is that Q8 would be read more slowly than Q7. The slowing of reading, however, should be accompanied by a speedier answering time since the retrieval process was finding an answer. It is important to show that the reading cost is accompanied by an answering savings in order to claim that the effects are due to retrieval.

Note that the reading speed prediction is at odds with the "fan effect" literature (Anderson, 1974, 1976, 1983; Anderson & Bower, 1973). When complex answer retrieval is not considered, for example if Q7 and Q8 were simple verification questions, then Q7 should be read more slowly than Q8 because there are more nodes related to Q7 than Q8. Spreading activation in
the episodic network derived from the four example sentences would be
greater when the concepts in Q7 were activated, and this additional resource
allocation would slow reading of Q7. Thus, the prediction in this study is
counter to the fan effect and may even have to work against a fan effect to
be noticeable.

3.2.2 Method

Subjects. Twenty-eight Rutgers University undergraduates participated
for credit in Introductory Psychology courses. Each subject served in all
conditions. Subjects spent about 40–50 minutes in the experiment.

Materials. Sixteen stories like the previous John and Mary story were
constructed for the experiment. Each story consisted of four actions per-
formed at four different times for four different purposes. In each story
there were two characters. When the stories were presented, one character
was associated with three actions while the second was associated with a
single action. For each story, one action was chosen as the query action,
about which a question would be asked. The actor associated with the query
action was varied across subjects so that for some subjects the query action
was performed by the unique character and for other subjects the query
action was performed by the character who did several things.

Design and Procedure. Each subject read the sixteen stories and answered
two questions about each one. The entire text of each story was presented
on a computer screen and subjects spent as long as they liked reading it.
When they were finished they pressed a response key. At this time a prompt
(plus sign), marking where the question would begin, appeared on the
screen. Each subsequent keypress revealed a word of the question. The
words appeared side-by-side in their normal positions, but only one word
was visible at a time (each previous word disappeared). Reason questions
were in the form “Why did NOUN1 VERB PREP1 DET NOUN2 PREP2
TIME?”, for example “Why did John drive to the store on Tuesday?”
Time questions were of the form “When did NOUN1 VERB PREP1 DET
NOUN2 AUX VERB2 NOUN3?”, for example “When did John drive to
the store to buy cheese?” Subjects were instructed that on the last word of
the questions they should press the response key “as soon as an answer
comes to mind.” The reading times for each word were recorded although
analyses were performed only on the words that the two question types had
in common, specifically NOUN1 VERB PREP1 DET NOUN2.

Subjects were first asked the reason or time question about each story.
Each subject was asked reason questions about eight stories and time ques-
tions about eight stories. For each subject, half of the questions were about
the action performed uniquely by one actor (unique action) and half were about one of the three actions performed by the other actor (non-unique action). The question type condition (reason/time) and action uniqueness condition (unique/non-unique) were completely crossed and the presentation order of various combinations was random. Across subjects, the stories were rotated through all conditions.

Since it was possible to answer the time and reason questions without paying attention to the actors in the story, the reason/time question was followed by a "who" question about each story. The antecedent for the who question was chosen randomly between the unique action and one of the nonunique actions. The who question guaranteed that subjects would pay close attention to the actors. Reading times were not collected for these questions.

### 3.2.3 Results and Discussion

Figure 4 shows the word-by-word reading times for the reason-questions and time-questions in the single-proposition and multiple-proposition conditions. As predicted, the time to read the questions was greater in the unique action condition relative to the nonunique action condition (495ms/word vs. 464ms/word), $F(1,27)=7.50, p<.05$. The cost of finding a unique source node and beginning retrieval was calculated by dividing the difference between the conditions, an estimate of the absolute cost per word, by the mean reading time per word in the nonunique action condition, an estimate of reading time which does not include the cost. The cost in reading time was 7% per word, in line with the costs found in the prior experiment. Although the time to read each word varied, $F(4,108)=6.13, p<.05$, the action uniqueness effect did not interact with words. There was no effect of the question type and no other interactions.

Table 5 shows the answer times for the reason-questions and time-questions in the unique and nonunique action conditions. The answer time was much faster in the unique action condition relative to the nonunique action condition (2449ms vs. 3554ms), $F(1,27)=19.11, p<.001$. The savings of finding a unique source node and beginning retrieval was calculated by dividing the difference between the conditions, an estimate of the absolute savings in answer time, by the mean answer time in the nonunique action condition, an estimate of answer time which does not include the savings. By identifying a source node early, subjects achieved a savings of 31% in their answering time.

In contrast to other experiments in our lab, reason questions were answered more slowly overall than time questions (3256ms vs. 2746ms), $F(1,27)=9.39, p<.05$. There was no interaction.

Subjects saved an average of 1105ms in their answering times when they were able to identify a unique source node early. The average cost to their reading times was 31ms/word as measured across five words. Assuming this
Figure 4. Mean word-by-word reading times for two question types when there was a unique and non-unique answer to the questions.
cost was extracted on all eight words in the questions, which is an overestimate of the cost, the overall cost to begin retrieval during parsing was 248ms. Thus, a liberal estimate of the cost/benefit ratio of parallel retrieval and parsing in this experiment is 22%.

A critical prediction of the TSUNAMI model, or any other parallel model, is that the content of episodic memory will influence the course of question comprehension. In this experiment we demonstrated that when early convergence on a source node is possible, reading times increase and answering times decrease. This effect was achieved despite the possibility that fan effects were working against the hypothesis. These results provide strong evidence that retrieval begins during reading.

It is interesting to consider the implications for more complex retrieval contexts such as those treated by Graesser & Murachver (1984) and Graesser & Clark (1985). According to these researchers, it is important for the question answerer to determine relevant knowledge sources in order to constrain retrieval. Different question types seem to be associated with specific types of knowledge structures, for example goal orientation questions are associated with goal hierarchies, causal antecedent questions are associated with causal chains, time questions are associated with temporal knowledge (if it is encoded in a special knowledge structure), etc. The TSUNAMI architecture allows for the early identification of these components of a knowledge structure and hypothesizes that candidate source nodes within these structures are gradually narrowed as each word of the question adds constraints to its interpretation. Once a small enough set of source nodes is identified, retrieval can begin.

### 3.3 Comprehension Instructions

#### 3.3.1 Introduction

The TSUNAMI model has a special answer retrieval process that operates separately, but in parallel, with the language parser. Presumably this process operates only when subjects know that they will be answering a question. In the above experiments evidence for parallel processes in question
answering was gathered by showing workload effects on the parser. However, it is important to show that these workload effects are special to question answering, and at the same time to show that subjects have control over the operation of the answer retrieval processor.

The connection of the workload effects to question processing has been demonstrated so far by showing workload effects that are linked to the question types. In the query-position experiment (section 3.1), time, reason, and verification questions were answered at different speeds, and in the second of those experiments they were also read at different speeds. In the experiment manipulating number of answers (section 3.2) the reason and time questions were also answered at different speeds. It is difficult to imagine an explanation for these effects that doesn't involve question-specific processing. However, in this experiment subjects' intentions as they read questions were manipulated. Specifically, sometimes subjects were asked to read and answer goal and time questions about stories, and sometimes subjects were asked to read and paraphrase goal and time questions about stories. In the latter case, the subjects did not have to answer the questions. In another condition subjects were asked to read and paraphrase goal and time questions which had no story context and could not be answered.

The well-known reading time effect related to the question type serves as a test of parallel processes in this experiment. Subjects should show a reading time difference between goal- and time-questions only when they intend to answer the questions. This result would be consistent with the view that subjects have control over the answer retrieval process and with the view that effects specific to the question answering process are influencing parsing times. If subjects in the paraphrase condition show a question-type effect, then the answer retrieval process may be automatic and not under the voluntary control of subjects.

3.3.2 Method

Subjects. Twenty-six Rutgers University undergraduates participated in this study for credit in Introductory Psychology courses. Subjects spent about 40–50 minutes in the experiment.

Materials. Forty-eight short (5–7 line) stories were written. In each story a character went to some location, and this action was queried in subsequent questions. All of the stories contained an explicit goal and time for the queried action. A reason question and a time question were prepared for each story. The reason questions read “Why did <ACTOR> go to the <LOCATION>?,” whereas the time questions read “When did <ACTOR> go to the <LOCATION>?”
**Design and Procedure.** There were three instruction conditions in the experiment: no story paraphrase, story paraphrase, and story answer. In the no story paraphrase condition subjects read questions (self-paced one word at a time) and were told to come up with a paraphrase. When they had reached the last word of the question they were to press the response key "when the meaning of the question was understood." They then wrote down their paraphrase. After this they pressed the response key and saw a computer generated question and were asked to judge if it "meant the same thing" as the question. The latter task was intended to reinforce the paraphrase instruction. Subjects worked through eight why-questions and eight when-questions randomly intermixed in a block. In the story paraphrase condition subjects read a paragraph-long story which was then followed by a question. The question was presented in the same manner as the no story paraphrase condition and subjects were instructed to come up with a paraphrase in the same way. Each question was followed by a "means the same thing" judgement and there were again eight why- and eight when-questions randomly intermixed in a block. Finally, in the story answer condition the subjects received stories followed by questions as in the story paraphrase condition, but this time for instruction was to come up with answers to the questions and press the response key "when an answer comes to mind." Similarly, they were asked to judge if the second question "has the same answer" as the first. Stories were randomly assigned to conditions and rotated through conditions across subjects. Instruction block orders were counterbalanced.

3.3.3 Results and Discussion
Reading times were summed over all words in the questions except the final word and then divided by the number of syllables in the words. This provides a per-syllable measure that controls for differing word lengths. Reading time for the last word includes time for memory retrieval plus answer/paraphrase formulation, and is therefore not included in the analysis designed to assess on-line processes.

Table 6 shows the mean reading times per syllable for reason and time questions read under the three instruction conditions. Overall, time questions were read more slowly than reason questions, $F(1,25) = 7.44$, $p < .05$, and the instruction manipulation also affected reading times, $F(2,50) = 3.33$, $p < .05$. A significant interaction between these factors, however, supports the hypothesis that the question-type effect is present only under the answering instruction $F(2,50) = 3.24$, $p < .05$. Time questions were read at a 39ms/syllable slower rate than reason questions under the answering condition, but the difference was only 11ms/syllable and 3ms/syllable in the no-story- and story-paraphrase conditions respectively.

There was a difference in the answering time for time questions vs. reason questions (1039ms vs. 1173 ms) which was not present in either paraphrase condition (1055ms for time-questions vs. 1001ms for reason-ques-
Table 6
Mean Reading Times (ms/syllable) for Reason and Time Questions
In the Three Comprehension Instruction Conditions

<table>
<thead>
<tr>
<th>Question Type</th>
<th>Comprehension Instruction</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No Story Paraphrase</td>
<td></td>
</tr>
<tr>
<td>Reason</td>
<td>390</td>
<td>395</td>
</tr>
<tr>
<td>Time</td>
<td>401</td>
<td>395</td>
</tr>
<tr>
<td>Mean</td>
<td>395</td>
<td>372</td>
</tr>
<tr>
<td></td>
<td>Story Paraphrase</td>
<td></td>
</tr>
<tr>
<td>Reason</td>
<td>367</td>
<td>368</td>
</tr>
<tr>
<td>Time</td>
<td>370</td>
<td>368</td>
</tr>
<tr>
<td>Mean</td>
<td>368</td>
<td>372</td>
</tr>
<tr>
<td></td>
<td>Story Answer</td>
<td></td>
</tr>
<tr>
<td>Reason</td>
<td>332</td>
<td>351</td>
</tr>
<tr>
<td>Time</td>
<td>371</td>
<td>351</td>
</tr>
<tr>
<td>Mean</td>
<td>351</td>
<td>372</td>
</tr>
</tbody>
</table>

The results can be interpreted in support of the hypothesis that question-related retrieval processes are being activated during reading. Such processes are not related to comprehension alone since comprehension is required in both the paraphrase and answering conditions.

Two possible types of processes might explain the result. Since when-questions have more potential answers than why-questions (Graesser & Murachver, 1984), there may be more retrieval routes active during normal processing of a when-question than a why-question. With more retrieval information active in short term memory, normal reading is slowed (MacDonald, Just, & Carpenter, 1992). Alternatively, there could be more rules or more complex sets of rules readied for answering a time question than a reason question. On this view retrieval does not begin, but the more cumbersome rule set burdens processing of the words in when-questions relative to why-questions.

3.4 Summary of Experimental Results

Three experiments were described which provide evidence for different aspects of the TSUNAMI framework. In the first, knowledge of the question type increased reading time for questions but decreased answering time. The results support the view that answer retrieval begins during parsing. In the second experiment the state of episodic memory affected question processing. When a single source node could be identified in episodic memory as the question presupposition, question reading time increased and answer retrieval time decreased when compared to cases in which the source node could not be uniquely identified until late in the question. This effect appeared early in the question strings suggesting that retrieval processes are not waiting for a full sentence to be input before beginning. In both experiments, the cost in time for combining parsing and retrieval was well worth the savings at answering time.
A third experiment examined whether retrieval effects on parsing times were specific to question answering. A discrepancy in reading times between reason questions and time questions was present only when subjects were instructed to answer the questions and not when they intended only to paraphrase the questions. The results reinforce the hypothesis that answer-retrieval processes affect question parsing times. The results also suggest that the retrieval components of question processing are not automatic.

4. CONCLUSION

The literature on question answering has always assumed serial operation of the many processes necessary to understand a question and find and generate an answer. Recent connectionist models of question answering have begun to challenge this view with parallel architectures (Miikkulainen & Dyer, 1991; Shastri, 1988, 1989; St. John & McClelland, 1990), but these models handle simple question types that require little more than a match/no-match heuristic to generate answers. In some cases (Shastri, 1988) they even preserve a serial flavor by incorporating separate and ordered network processes for query formulation, knowledge base access, and response generation. They also typically operate on a highly pre-encoded representation of the query—implying at least that considerable parsing has been done before retrieval begins.

The experiments were not designed to distinguish connectionist from symbol-based models of question answering, and they cannot do so. The results were obtained in the context of question tasks that current connectionist models do not perform. In particular, connectionist models do not exist for answering complex goal and time questions in the context of narratives, and connectionist models do not mix parsing with question answering. Modular, symbol-based models of these complex tasks have been proposed, however, and so this article focuses on augmenting the architecture underlying these models so that parsing and retrieval can be intermixed.

The experimental results are not inconsistent with connectionist models as far as they go. In fact, the matcher in TSUNAMI might best be implemented as a connectionist network with local nodes much like Shastri (1988, 1989) proposes. Since the goal of the matcher is to locate the question presupposition in memory, the verification or recognition process in connectionist question-answering models maps well to this aspect of TSUNAMI. Some part of the increased reading times observed when question parsing and retrieval were operating in parallel might be due to simple presupposition matching, however the bulk of the time cost is surely due to the application of retrieval heuristics for complex reason and time searches. Matching is only the start of a highly interactive set of processes whose nature should be the topic of future research. Some researchers have proposed hybrid
models of sentence processing that combine aspects of connectionism with symbol-based mechanisms (Kintsch, 1988; Mannes & Doane, 1991; Moisl, 1992), and this is a fascinating direction for question answering models as well.

The TSUNAMI model is a proposed architecture that supports parallelism in question comprehension and answer retrieval in the tradition of modular, symbol-based processing. Viewing question comprehension and answer retrieval as parallel processes has consequences for many aspects of current models in this tradition. The nature and implementation of question categorization and answer retrieval heuristics might be significantly different when considered in the context of a parallel architecture. Many new problems arise once this view is adopted. For example, the issue of interaction among parsing and answering components becomes important. The problem of determining which components of knowledge structures to search becomes more complex as the constraints are apparent earlier in parsing and must be applied in a dynamic manner. If partially specified question candidates can start retrieval processes, a problem arises concerning how they might be tracked and stopped if the question candidate is later disconfirmed. In future research new models of question processing will certainly arise as a parallel framework is adopted.

5. REFERENCES


