Behavior Research Methods, Instruments, & Computers
2004, 36 (2), 180-192

AutoTutor: A tutor with dialogue in natural language

ARTHUR C. GRAESSER, SHULAN LU, GEORGE TANNER JACKSON, HEATHER HITE MITCHELL, MATHEW VENTURA, ANDREW OLNEY, and MAX M. LOUWERSE
University of Memphis, Memphis, Tennessee

AutoTutor is a learning environment that tutors students by holding a conversation in natural language. AutoTutor has been developed for Newtonian qualitative physics and computer literacy. Its design was inspired by explanation-based constructivist theories of learning, intelligent tutoring systems that adaptively respond to student knowledge, and empirical research on dialogue patterns in tutorial discourse. AutoTutor presents challenging problems (formulated as questions) from a curriculum script and then engages in mixed initiative dialogue that guides the student in building an answer. It provides the student with positive, neutral, or negative feedback on the student's typed responses, pumps the student for more information, prompts the student to fill in missing words, gives hints, fills in missing information with assertions, identifies and corrects erroneous ideas, answers the student's questions, and summarizes answers. AutoTutor has produced learning gains of approximately .70 sigma for deep levels of comprehension.

The vision of having a computer communicate with humans in natural language was entertained shortly after the computer was invented. Weizenbaum's (1966) ELIZA program was the first conversation system that was reasonably successful, popular, and widely used. ELIZA simulated a Rogerian client-centered psychotherapist, as illustrated in the conversation below.

Patient: You are like my father in some ways.
ELIZA: How am I like your father?
Patient: You are very aggressive but I think you don't want me to notice that.
ELIZA: What makes you think I am aggressive?
Patient: You always argue with me.
ELIZA: Why do you think I argue with you?

Like a typical Rogerian therapist, ELIZA tried to get the patient to do the talking by asking the patient questions about the patient's verbal contributions to the exchange. Conversation partners are expected to answer questions, so a therapist's question readily turns the spotlight onto the patient. ELIZA turned the patient's assertions into a therapist's questions by simple syntactic transformational rules. That is, ELIZA detected keywords and word combinations that triggered rules, which in turn generated ELIZA's responses. The only intelligence in ELIZA was the stimulus–response knowledge captured in production rules that operate on keywords and that perform syntactic transformations. What was so remarkable about ELIZA is that 100–200 simple production rules could very often create an illusion of comprehension, even though ELIZA had no depth. Computer scientists entertained themselves by adding a progressively larger set of rules to handle all sorts of contingencies. One could imagine that an ELIZA with 20,000 well-selected rules might very well simulate a responsive, intelligent, compassionate therapist. However, no one ever tried.

Efforts to build conversational systems continued in the 1970s and early 1980s. PARRY attempted to simulate a paranoid agent (Colby, Weber, & Hilf, 1971). SCHOLAR tutored students on South American geography by asking and answering questions (Collins, Warnock, & Passafiume, 1975). Moonrocks (Woods, 1977) and Elinor (Norman & Rumelhart, 1975) syntactically parsed questions and answered users' queries. Schank and his colleagues built computer models of natural language understanding and rudimentary dialogue about scripted activities (Lehnert & Ringle, 1982; Schank & Riesbeck, 1982). SHRDLU manipulated simple objects in a blocks world in response to a user's command (Winograd, 1972).

Unfortunately, by the mid-1980s most researchers in cognitive science and artificial intelligence were convinced that the prospect of building a good conversation system was well beyond the horizon. The chief chal-

The Tutoring Research Group is an interdisciplinary research team comprised of approximately 35 researchers with backgrounds in psychology, computer science, physics, and education (for more information, visit http://www.autotutor.org). The research on AutoTutor was supported by National Science Foundation (NSF) Grants SBR 9720314, REC 0106965, REC 0126265, and ITR 0325428 and by Grant N00014-00-1-0600 of the Department of Defense Multidisciplinary University Research Initiative administered by the Office of Naval Research (ONR). Any opinions, findings, and conclusions or recommendations expressed in this article are those of the authors and do not necessarily reflect the views of the Department of Defense, the ONR, or the NSF. Kurt VanLehn, Carolyn Rosé, Pam Jordan, and others at the University of Pittsburgh collaborated with us in preparing AutoTutor materials on conceptual physics. Correspondence concerning this article should be addressed to A. Graesser, Department of Psychology, 202 Psychology Building, University of Memphis, Memphis, TN 38152-3230 (e-mail: a-graesser@memphis.edu).

lenges were (1) the inherent complexities of natural language processing, (2) the unconstrained, open-ended nature of world knowledge, and (3) the lack of research on lengthy threads of connected discourse. In retrospect, this extreme pessimism about discourse and natural language technologies was arguably premature. A sufficient number of technical advances has been made in the last decade for researchers to revisit the vision of building dialogue systems.

The primary technical breakthroughs came from the fields of computational linguistics, information retrieval, cognitive science, artificial intelligence, and discourse processes. For example, the field of computational linguistics has produced an impressive array of lexicons, syntactic parsers, semantic interpretation modules, and dialogue analyzers that are capable of rapidly extracting information from naturalistic text for information retrieval, machine translation, and speech recognition (Allen, 1995; DARPA, 1995; Harabagiu, Maiorano, & Pasca, 2002; Jurafsky & Martin, 2000; Manning & Schütze, 1999; Voorhees, 2001). These advancements in computational linguistics represent world knowledge symbolically, statistically, or through a hybrid of these two foundations. For instance, Lenat's (1995) CYC system represents a large volume of mundane world knowledge in symbolic forms that can be integrated with a diverse set of processing architectures. The world knowledge contained in an encyclopedia can be represented statistically in high-dimensional spaces, such as latent semantic analysis (LSA; Foltz, Gilliam, & Kendall, 2000; Landauer, Foltz, & Laham, 1998) and Hyperspace Analogue to Language (HAL; Burgess, Livesay, & Lund, 1998). An LSA space provides the backbone for statistical metrics on whether two text excerpts are conceptually similar; the reliability of these similarity metrics has been found to be equivalent to that of human judgments. The representation and processing of connected discourse is much less mysterious after two decades of research in discourse processing (Graesser, Gernsbacher, & Goldman, 2003) and relevant interdisciplinary research (Louwerse & van Peer, 2002). There are now generic computational modules for building dialogue facilities that track and manage the beliefs, knowledge, intentions, goals, and attention states of agents in two-party dialogues (Graesser, VanLehn, Rosé, Jordan, & Harter, 2001; Gratch et al., 2002; Moore & Wiemer-Hastings, 2003; Rich & Sidner, 1998; Rickel, Lesh, Rich, Sidner, & Gertner, 2002).

WHEN ARE NATURAL LANGUAGE DIALOGUE FACILITIES FEASIBLE?

Natural language dialogue (NLD) facilities are expected to do a reasonable job in some conversational contexts but not in others. Success depends on the subject matter, the knowledge of the learner, the expected depth of comprehension, and the expected sophistication of the dialogue strategies. We do not believe that current NLD facilities will be impressive when the subject matter requires mathematical or analytical precision, when the knowledge level of the learner is high, and when the user would like to converse with a humorous, witty, or illuminating partner. For example, an NLD facility would not be well suited to an eCommerce application that manages precise budgets that a user carefully tracks. An NLD facility would not be good for an application in which the dialogue system must simulate a good spouse, parent, comedian, or confidant. An NLD facility is more feasible in applications that involve imprecise verbal content, a low to medium level of user knowledge about a topic, and earnest literal replies.

We are convinced that tutoring environments are feasible NLD applications when the subject matter is verbal and qualitative. NLD tutors have been attempted for mathematics, with limited success (Heffernan & Koedinger, 1998), whereas those in qualitative domains have shown more promise (Graesser, VanLehn, et al., 2001; Rickel et al., 2002; VanLehn, Jordan, et al., 2002). Tutorial NLD is feasible when the shared knowledge (i.e., common ground) between the tutor and the learner is low or moderate rather than high. If the common ground is high, then both dialogue participants (i.e., the computer tutor and the learner) will be expecting a more precise degree of mutual understanding and, therefore, will run a higher risk of failing to meet the other's expectations.

It is noteworthy that human tutors are not able to monitor the knowledge of students at a fine-grained level, because much of what students express is vague, underspecified, ambiguous, fragmentary, and error ridden (Fox, 1993; Graesser & Person, 1994; Graesser, Person, & Magliano, 1995; Shah, Evens, Michael, & Rovick, 2002). There are potential costs if a tutor attempts to do so. For example, it is often more worthwhile for the tutor to help build new correct knowledge than to become bogged down in dissecting and correcting each of the learner's knowledge deficits. Tutors do have an approximate sense of what a student knows, and this appears to be sufficient to provide productive dialogue moves that lead to significant learning gains in the student (Chi, Siler, Jeong, Yamauchi, & Hausmann, 2001; Cohen, Kulik, & Kulik, 1982; Graesser et al., 1995). These considerations motivated the design of AutoTutor (Graesser, Person, Harter, & the Tutoring Research Group (TRG), 2001; Graesser, VanLehn, et al., 2001; Graesser, K. Wiemer-Hastings, P. Wiemer-Hastings, Kreuz, & the TRG, 1999), which will be described in the next section. The central assumption, in a nutshell, is that dialogue can be useful when it advances the dialogue and learning agenda, even when the tutor does not fully understand a student.

Tutorial NLD appears to be a more feasible technology to the extent that the tutoring strategies, unlike strategies that are highly sophisticated, follow what most human tutors do. Most human tutors anticipate particular correct answers (called *expectations*) and particular misunderstandings (*misconceptions*) when they ask the learner questions and trace the learner's reasoning. As the learner articulates the answer or solves the problem, this content

is constantly being compared with the expectations and anticipated misconceptions. The tutor responds adaptively and appropriately when particular expectations or misconceptions are expressed. This tutoring mechanism is *expectation- and misconception-tailored* (EMT) *dialogue* (Graesser, Hu, & McNamara, in press). The EMT dialogue moves of most human tutors are not particularly sophisticated from the standpoint of ideal tutoring strategies that have been proposed in the fields of education and artificial intelligence (Graesser et al., 1995). Graesser and colleagues (Graesser & Person, 1994; Graesser et al., 1995) videotaped over 100 h of naturalistic tutoring, transcribed the data, classified the speech act utterances into discourse categories, and analyzed the rate of particular discourse patterns. These analyses revealed that human tutors rarely implement intelligent pedagogical techniques such as bona fide Socratic tutoring strategies, modeling–scaffolding–fading, reciprocal teaching, frontier learning, building on prerequisites, cascade learning, and diagnosis/remediation of deep misconceptions (Collins, Brown, & Newman, 1989; Palincsar & Brown, 1984; Sleeman & Brown, 1982). Instead, tutors tend to coach students in constructing explanations according to the EMT dialogue patterns. Fortunately, the EMT dialogue strategy is substantially easier to implement computationally than are sophisticated tutoring strategies.

Researchers have developed approximately half a dozen intelligent tutoring systems with dialogue in natural language. AutoTutor and Why/AutoTutor (Graesser et al., in press; Graesser, Person, et al., 2001; Graesser et al., 1999) were developed for introductory computer literacy and Newtonian physics. These systems help college students generate cognitive explanations and patterns of knowledge-based reasoning when solving particular problems. Why/Atlas (VanLehn, Jordan, et al., 2002) also has students learn about conceptual physics with a coach that helps build explanations of conceptual physics problems. CIRCSIM Tutor (Hume, Michael, Rovick, & Evens, 1996; Shah et al., 2002) helps medical students learn about the circulatory system by using strategies of an accomplished tutor with a medical degree. PACO (Rickel et al., 2002) assists learners in interacting with mechanical equipment and completing tasks by interacting in natural language.

Two generalizations can be made from the tutorial NLD systems that have been created to date. The first generalization addresses dialogue management. Finite-state machines for dialogue management (which will be described later) have served as an architecture that can produce working systems (such as AutoTutor, Why/AutoTutor, and Why/Atlas). However, no full-fledged dialogue planners have been included in working systems that perform well enough to be satisfactorily evaluated (as in CIRCSIM Tutor and PACO). The *Mission Rehearsal System* (Gratch et al., 2002) comes closest to being a full-fledged dialogue planner, but the depth and sophistication of such planning are extremely limited. Dialogue planning is very difficult because it requires the precise recognition of knowledge states (goals, intentions, beliefs, knowledge) and a closed system of formal reasoning. Unfortunately, the dialogue contributions of most learners are too vague and underspecified to afford precise recognition of knowledge states. The second generalization addresses the representation of world knowledge. An LSA-based statistical representation of world knowledge allows the researcher to have some world knowledge component up and running very quickly (measured in hours or days), whereas a symbolic representation of world knowledge takes years or decades to develop. AutoTutor (and Why/AutoTutor) routinely incorporates LSA in its knowledge representation, so a new subject matter can be quickly developed.

AUTOTUTOR

AutoTutor is an NLD tutor developed by Graesser and colleagues at the University of Memphis (Graesser, Person, et al., 2001; Graesser, VanLehn, et al., 2001; Graesser et al., 1999; Song, Hu, Olney, Graesser, & the TRG, in press). AutoTutor poses questions or problems that require approximately a paragraph of information to answer. An example question in conceptual physics is "Suppose a boy is in a free-falling elevator and he holds his keys motionless right in front of his face and then lets go. What will happen to the keys? Explain why." Another example question is "When a car without headrests on the seats is struck from behind, the passengers often suffer neck injuries. Why do passengers get neck injuries in this situation?" It is possible to accommodate questions with answers that are longer or shorter; the paragraph span is simply the length of the answers that have been implemented in AutoTutor thus far, in an attempt to handle open-ended questions that invite answers based on qualitative reasoning. Although an ideal answer is approximately three to seven sentences in length, the initial answers to these questions by learners are typically only one word to two sentences in length. This is where tutorial dialogue is particularly helpful. AutoTutor engages the learner in a dialogue that assists the learner in the evolution of an improved answer that draws out more of the learner's knowledge that is relevant to the answer. The dialogue between AutoTutor and the learner typically lasts 30–100 *turns* (i.e., the learner expresses something, then the tutor does, then the learner, and so on). There is a *mixed-initiative* dialogue to the extent that each dialogue partner can ask questions and start new topics of discussion.

Figure 1 shows the interface of AutoTutor. The major question (involving, in this example, a boy dropping keys in a falling elevator) is selected and presented in the top right window. This major question remains at the top of the Web page until it is finished being answered during a multiturn dialogue between the student and AutoTutor. The student uses the bottom right window to type in his or her contribution for each turn, and the content of both tutor and student turns is reflected in the bottom left window. The animated conversational agent resides in the upper left area. The agent uses either an AT&T, SpeechWorks,

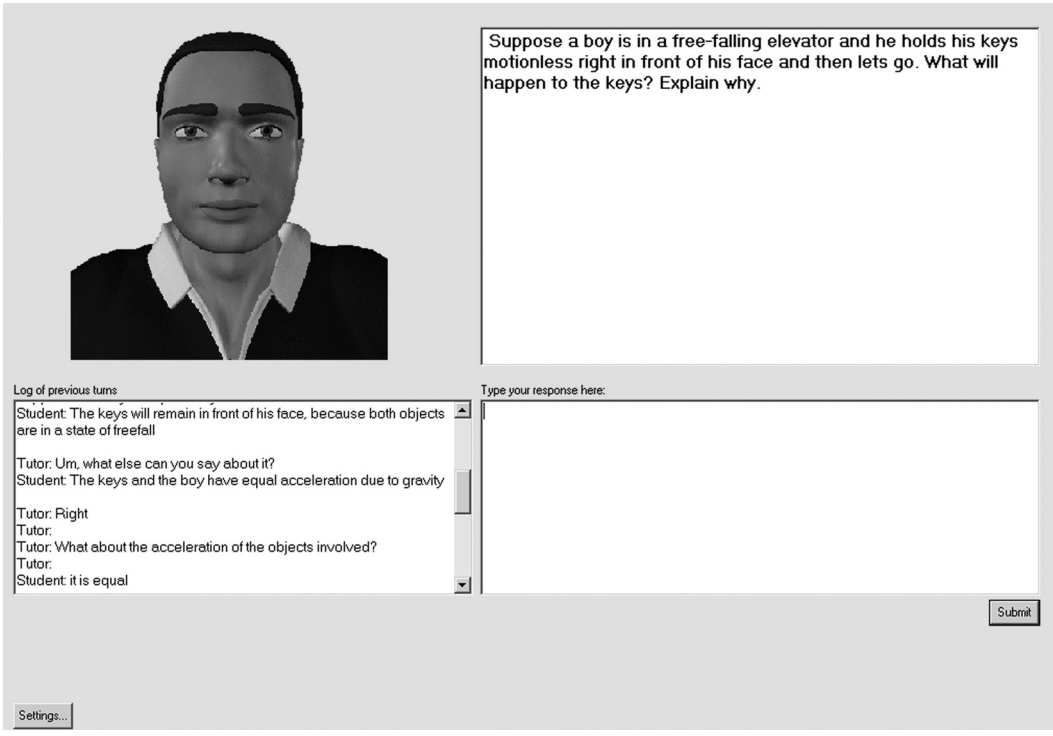

Figure 1. A computer screen of AutoTutor for the subject matter of conceptual physics.

or Microsoft Agent speech engine (dependent on licensing agreements) to say the content of AutoTutor's turns during the process of answering the presented question. Figure 2 shows a somewhat different interface that is used when tutoring computer literacy. This interface has a display area for diagrams, but no dialogue history window.

The design of AutoTutor was inspired by three bodies of research: theoretical, empirical, and applied. These include explanation-based constructivist theories of learning (Aleven & Koedinger, 2002; Chi, de Leeuw, Chiu, & LaVancher, 1994; VanLehn, Jones, & Chi, 1992), intelligent tutoring systems that adaptively respond to student knowledge (Anderson, Corbett, Koedinger, & Pelletier, 1995; VanLehn, Lynch, et al., 2002), and empirical research that has documented the collaborative constructive activities that routinely occur during human tutoring (Chi et al., 2001; Fox, 1993; Graesser et al., 1995; Moore, 1995; Shah et al., 2002). According to the explanation-based constructivist theories of learning, learning is more effective and deeper when the learner must actively generate explanations, justifications, and functional procedures than when he or she is merely given information to read. Regarding adaptive intelligent tutoring systems, the tutors give immediate feedback on the learner's actions and guide the learner on what to do next in a fashion that is sensitive to what the system believes the learner knows. Regarding the empirical research on tutorial dialogue, the patterns of discourse uncovered in naturalistic tutoring are imported into the dialogue management facilities of AutoTutor.

Covering Expectations, Correcting Misconceptions, and Answering Questions with Dialogue Moves

AutoTutor produces several categories of *dialogue moves* that facilitate covering the information that is anticipated by AutoTutor's *curriculum script*. The curriculum script includes the questions, problems, expectations, misconceptions, and most relevant subject matter content. AutoTutor delivers its dialogue moves by an animated *conversational agent* (synthesized speech, facial expressions, gestures), whereas learners enter their answers by keyboard. AutoTutor provides positive, neutral, and negative *feedback* to the learner, *pumps* the learner for more information (e.g., with the question "What else?"), *prompts* the learner to fill in missing words, gives *hints*, fills in missing information with *assertions*, identifies and *corrects* bad answers, *answers* learners' questions, and *summarizes* answers. As the learner expresses information over many turns, the information in the three to seven sentences of an answer is eventually covered, and the question is answered. During the process of supplying the ideal answer, the learner periodically articulates misconceptions and false assertions. If these misconceptions have been anticipated and incorporated into the program, AutoTutor provides the learner with information to correct the misconceptions. Therefore, as the learner expresses information over the turns, this information is compared with expectations and misconceptions, and AutoTutor formulates its dialogue moves in a fashion that is sensitive to the learner's input. That is,

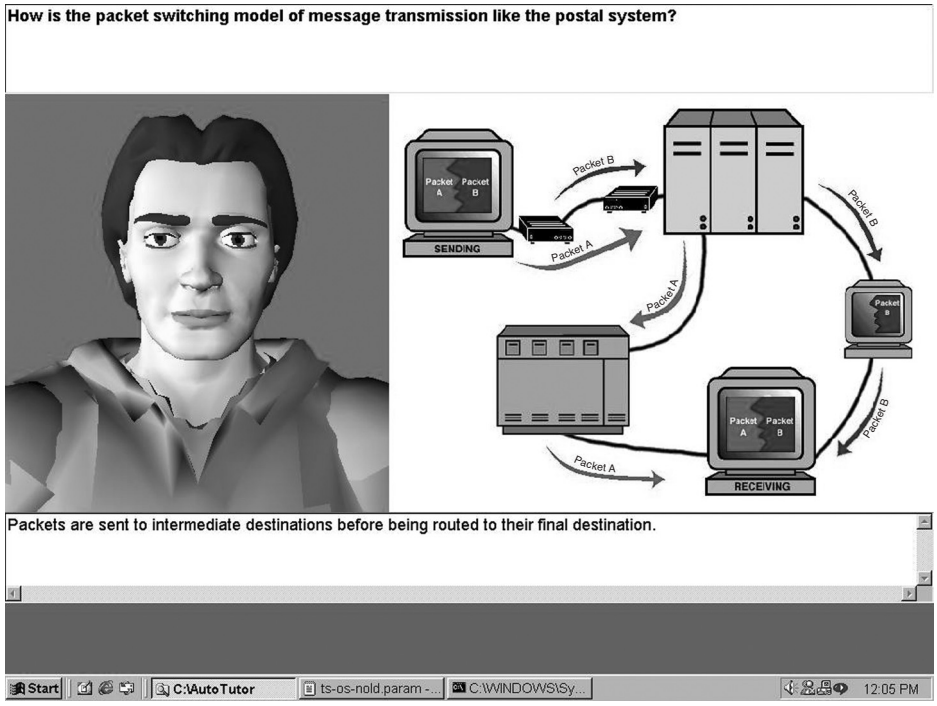

Figure 2. A computer screen of AutoTutor for the subject matter of introductory computer literacy.

AutoTutor implements the EMT dialogue, which is known to be common in human tutoring.

The conversation below illustrates some of the features of AutoTutor's dialogue. This was an actual conversation with a student taken from one of the AutoTutor log files. Discourse tags have been added in capitals, whereas other descriptive information is added in italics.

AutoTutor: When a car without headrests on the seats is struck from behind, the passengers often suffer neck injuries. Why do passengers get neck injuries in this situation? (MAIN QUESTION)
Student: The passenger experiences a backward acceleration, while the car moves forward.
AutoTutor: And can you add to that? (PUMP)
Student: This is because the passenger experiences a force opposite in direction to the force on the car.
AutoTutor: Um, yes, correct! (POSITIVE FEEDBACK) When the neck provides a large force, what within the neck would get damaged? (HINT)
Student: The spine.
AutoTutor: Umm (NEUTRAL NEGATIVE FEEDBACK), I think you can get this. Damage within the neck will occur in the ligaments and the . . .? [*spoken with rising intonation and accompanied with a prompting hand gesture*] (PROMPT):
Student: Muscles.
AutoTutor: Brilliant! (POSITIVE FEEDBACK) Ligaments and the muscles. (PROMPT RESPONSE) Let's keep going. (DIALOGUE ADVANCER) Additionally, when there is no headrest present in the car, what accelerates the head of the passenger? (HINT)
Student: The impact from behind.
AutoTutor: I am sure you can get this. The head of the passenger will be accelerated by the passenger's . . . ? [*spoken with rising intonation and accompanied with a prompting hand gesture*] (PROMPT)
Student: Car.
AutoTutor: Wrong! (NEGATIVE FEEDBACK) By the passenger's neck. (PROMPT RESPONSE):
[*moving ahead 22 turns in the dialogue*]
AutoTutor: Do you perhaps have any general conceptual physics questions? (QUESTION PROMPT)
Student: No.
AutoTutor: How would you answer my first question, having gone over what we've just talked about? (FINAL ANSWER PROMPT)
Student: The passengers get neck injuries because … [*81 additional words*].
AutoTutor: Great! (POSITIVE FEEDBACK) To recap, when a car is struck from behind, the force of impact will cause the car to experience a large forward acceleration. The car seat is attached to the car and will also have the same large acceleration. When the car is hit from behind, the seat will exert the same acceleration to the torso of the body. In order for the head to go along with the torso, it will need to be accelerated at the same rate as the torso. Since the head has no acceleration being applied to it by a headrest, the neck gets stretched, damaging its muscles and ligaments in an attempt to produce the required large acceleration. (SUMMARY)

In addition to asking questions, AutoTutor attempts to handle questions posed by the learner. However, somewhat surprisingly, students rarely ask questions in classrooms, human tutoring sessions, and AutoTutor sessions

(Graesser & Olde, 2003; Graesser & Person, 1994). The rate of learner questions is one question per 6–7 h in a classroom environment and one question per 2 min in tutoring. Although it is pedagogically disappointing that learners ask so few questions, the good news is that this aspect of human tutor interaction makes it easier to build a dialogue-based intelligent tutoring system such as AutoTutor. It is not computationally feasible to interpret any arbitrary learner input from scratch and to construct a mental space that adequately captures what the learner has in mind. Instead, the best that AutoTutor can do is to compare learner input with expectations through pattern-matching operations. Therefore, what human tutors and learners do is compatible with what currently can be handled computationally within AutoTutor.

Latent Semantic Analysis

AutoTutor uses LSA for its conceptual pattern-matching algorithm when evaluating whether student input matches the expectations and anticipated misconceptions. LSA is a high-dimensional statistical technique that, among other things, measures the conceptual similarity of any two pieces of text, such as words, sentences, paragraphs, or lengthier documents (Foltz et al., 2000; E. Kintsch, Steinhart, Stahl, & the LSA Research Group, 2000; W. Kintsch, 1998; Landauer & Dumais, 1997; Landauer et al., 1998). A cosine is calculated between the LSA vector associated with expectation *E* (or misconception *M*) and the vector associated with learner input *I*. *E* (or *M*) is scored as covered if the match between *E* or *M* and the learner's text input *I* meets some threshold, which has varied between .40 and .85 in previous instantiations of AutoTutor. As the threshold parameter increases, the learner needs to be more precise in articulating information and thereby cover the expectations.

Suppose that there are four key expectations embedded within an ideal answer. AutoTutor expects all four to be covered in a complete answer and will direct the dialogue in a fashion that finesses the students to articulate these expectations (through prompts and hints). AutoTutor stays on topic by completing the subdialogue that covers *E* before starting a subdialogue on another expectation. For example, suppose an answer requires the expectation: *The force of impact will cause the car to experience a large forward acceleration.* The following family of prompts is available to encourage the student to articulate particular content words in the expectation:

1. The impact will cause the car to experience a forward _____?
2. The impact will cause the car to experience a large acceleration in what direction? _____
3. The impact will cause the car to experience a forward acceleration with a magnitude that is very _____?
4. The car will experience a large forward acceleration after the force of _____?
5. The car will experience a large forward acceleration from the impact's _____?
6. What experiences a large forward acceleration? _____

The particular prompts that are selected are those that fill in missing information if answered successfully. Thus, the dialogue management component adaptively selects hints and prompts in an attempt to achieve pattern completion. The expectation is covered when enough of the ideas underlying the content words in the expectation are expressed by the student so that the LSA threshold is met or exceeded.

AutoTutor considers everything the student expresses during Conversation Turns 1–*n* to evaluate whether expectation *E* is covered. If the student has failed to articulate one of the six content words (*force*, *impact*, *car*, *large*, *forward*, *acceleration*), AutoTutor selects the corresponding prompts 5, 4, 6, 3, 2, and 1, respectively. Therefore, if the student has made assertions X, Y, and Z at a particular point in the dialogue, then all possible combinations of X, Y, and Z would be considered in the matches: X, Y, Z, XY, XZ, YZ, and XYZ. The degree of match for each comparison between *E* and *I* is computed as cosine (vector ***E***, vector ***I***). The maximum cosine match score among all seven combinations of sentences is one method used to assess whether *E* is covered. If the match meets or exceeds threshold T, then *E* is covered. If the match is less than T, then AutoTutor selects the prompt (or hint) that has the best chance of improving the match if the learner provides the correct answer to the prompt. Only explicit statements by the learner (not by AutoTutor) are considered when determining whether expectations are covered. As such, this approach is compatible with constructivist learning theories that emphasize the importance of the learner's generating the answer. We are currently fine-tuning the LSA-based pattern matches between learner input and AutoTutor's expected input (Hu et al., 2003; Olde, Franceschetti, Karnavat, Graesser, & the TRG, 2002).

LSA does a moderately impressive job of determining whether the information in learner essays matches particular expectations associated with an ideal answer. For example, we have instructed experts in physics or computer literacy to make judgments concerning whether particular expectations were covered within learner essays on conceptual physics problems. Using either stringent or lenient criteria, these experts computed a coverage score based on the proportion of expectations that were believed to be present in the learner essays. LSA was used to compute the proportion of expectations covered, using varying thresholds of cosine values on whether information in the learner essay matched each expectation. Correlations between the LSA scores and the judges' coverage scores have been found to be approximately .50 for both conceptual physics (Olde et al., 2002) and computer literacy (Graesser, P. Wiemer-Hastings, et al., 2000). Correlations generally increase as the length of the text increases, reaching as high as .73 in research conducted at other labs (see Foltz et al., 2000). LSA metrics have also done a reasonable job tracking the coverage of expectations and the identification of misconceptions dur-

ing the dynamic, turn-by-turn dialogue of AutoTutor (Graesser et al., 2000).

The conversation is finished for the main question or problem when all expectations are covered. In the meantime, if the student articulates information that matches any misconception, the misconception is corrected as a subdialogue and then the conversation returns to finish coverage of the expectations. Again, the process of covering all expectations and correcting misconceptions that arise normally requires a dialogue of 30–100 turns (i.e., 15–50 student turns).

A Sketch of AutoTutor's Computational Architecture

The computational architectures of AutoTutor have been discussed extensively in previous publications (Graesser, Person et al., 2001; Graesser, VanLehn, et al., 2001; Graesser, K. Wiemer-Hastings, et al., 1999; Song et al., in press), so this article will provide only a brief sketch of the components. The original AutoTutor was written in Java and resided on a Pentium-based server platform to be delivered over the Internet. The most recent version has a more modular architecture that will not be discussed in this article. The software residing on the server has a set of permanent databases that are not updated throughout the course of tutoring. These permanent components include the five below.

Curriculum script repository. Each script contains the content associated with a question or problem. For each, there is (1) the ideal answer; (2) a set of expectations; (3) families of potential hints, correct hint responses, prompts, correct prompt responses, and assertions associated with each expectation; (4) a set of misconceptions and corrections for each misconception; (5) a set of key words and functional synonyms; (6) a summary; and (7) markup language for the speech generator and gesture generator for components in (1) through (6) that require actions by the animated agents. Subject-matter experts can easily create the content of the curriculum script with an authoring tool called the *AutoTutor Script Authoring Tool* (ASAT; Susarla et al., 2003).

Computational linguistics modules. There are lexicons, syntactic parsers, and other computational linguistics modules that are used to extract and classify information from learner input. It is beyond the scope of this article to describe these modules further.

Corpus of documents. This is a textbook, articles on the subject matter, or other content that the question-answering facility accesses for paragraphs that answer questions.

Glossary. There is a glossary of technical terms and their definitions. Whenever a learner asks a definitional question ("What does X mean?"), the glossary is consulted and the definition is produced for the entry in the glossary.

LSA space. LSA vectors are stored for words, curriculum script content, and the documents in the corpus.

In addition to the five static data modules enumerated above, AutoTutor has a set of processing modules and dynamic storage units that maintain qualitative content and quantitative parameters. Storage registers are frequently updated as the tutoring process proceeds. For example, AutoTutor keeps track of student ability (as evaluated by LSA from student assertions), student initiative (such as the incidence of student questions), student verbosity (number of words per student turn), and the incremental progress in having a question answered as the dialogue history grows, turn by turn. The dialogue management module of AutoTutor flexibly adapts to the student by virtue of these parameters, so it is extremely unlikely that two conversations with AutoTutor are ever the same.

Dialogue Management. The dialogue management module is an *augmented finite state transition network* (for details, see Allen, 1995; Jurafsky & Martin, 2000). The *nodes* in the network refer to knowledge goal states (e.g., expectation *E* is under focus and AutoTutor wants to get the student to articulate it) or dialogue states (e.g., the student just expressed an assertion as his or her first turn in answering the question). The *arcs* refer to categories of tutor dialogue moves (e.g., feedback, pumps, prompts, hints, summaries) or discourse markers that link dialogue moves (e.g., "okay," "moving on," "furthermore"; Louwerse & Mitchell, 2003). A particular arc is traversed when particular conditions are met. For example, a pump arc is traversed when it is the student's first turn and the student's assertion has a low LSA match value.

Arc traversal is normally contingent on outputs of computational algorithms and procedures that are sensitive to the dynamic evolution of the dialogue. These algorithms and procedures operate on the snapshot of parameters, curriculum content, knowledge goal states, student knowledge, dialogue states, LSA measures, and so on, which reflect the current conversation constraints and achievements. For example, there are algorithms that select dialogue move categories intended to get the student to fill in missing information in *E* (the expectation under focus). There are several alternative algorithms for achieving this goal. Consider one of the early algorithms that we adopted, which relied on *fuzzy production rules*. If the student had almost finished articulating *E* but lacked a critical noun or verb, then a prompt category would be selected because the function of prompts is to extract single words from students. The particular prompt selected from the curriculum script would be tailored to extract the particular missing word through another module that fills posted dialogue move categories with particular content. If the student is classified as having high ability and has failed to articulate most of the words in *E*, then a hint category might be selected. Fuzzy production rules made these selections.

An alternative algorithm to fleshing out *E* uses two cycles of hint–prompt–assertion. That is, AutoTutor's selection of dialogue moves over successive turns follows a particular order: first hint, then prompt, then assert,

then hint, then prompt, then assert. AutoTutor exits the two cycles as soon as the student articulates *E* to satisfaction (i.e., the LSA threshold is met).

Other processing modules in AutoTutor execute various important functions: linguistic information extraction, speech act classification, question answering, evaluation of student assertions, selection of the next expectation to be covered, and speech production with the animated conversational agent. It is beyond the scope of this paper to describe all of these modules, but two will be described briefly.

Speech act classifier. AutoTutor needs to determine the intent or conversational function of a learner's contribution in order to respond to the student flexibly. AutoTutor obviously should respond very differently when the learner makes an assertion than when the learner asks a question. A learner who asks "Could you repeat that?" would probably not like to have "yes" or "no" as a response, but would like to have the previous utterance repeated. The classifier system has 20 categories. These categories include assertions, metacommunicative expressions ("Could you repeat that?" "I can't hear you"), metacognitive expressions ("I don't know," "I'm lost"), short responses ("Oh," "Yes"), and the 16 question categories identified by Graesser and Person (1994). The classifier uses a combination of syntactic templates and key words. Syntactic tagging is provided by the Apple Pie parser (Sekine & Grishman, 1995) together with cascaded finite state transducers (see Jurafsky & Martin, 2000). The finite state transducers consist of a transducer of key words (e.g., "difference" and "comparison" in the comparison question category) and syntactic templates. Extensive testing of the classifier showed that the accuracy of the classifier ranged from 65% to 97%, depending on the corpus, and was indistinguishable from the reliability of human judges (Louwerse, Graesser, Olney, & the TRG, 2002; Olney et al., 2003).

Selecting the expectation to cover next. After one expectation is finished being covered, AutoTutor moves on to cover another expectation that has a subthreshold LSA score. There are different pedagogical principles that guide this selection of which expectation to post next on the goal stack. One principle is called the *zone of proximal development* or *frontier learning*. AutoTutor selects the particular *E* that is the smallest increment above what the student has articulated. That is, it modestly extends what the student has already articulated. Algorithmically, this is simply the expectation with the highest LSA coverage score among those expectations that have not yet been covered. A second principle is called the *coherence* principle. In an attempt to provide a coherent thread of conversation, AutoTutor selects the (uncovered) expectation that has the highest LSA similarity to the expectation that was most recently covered. A third pedagogical principle is to select a *central pivotal* expectation that has the highest likelihood of pulling in the content of the other expectations. The most central expectation is the one with the highest mean LSA similarity to the remaining uncovered expectations. We normally weight these three principles in tests of AutoTutor, but it is possible to alter these weights by simply changing three parameters. Changes in these parameters, as well as in others (e.g., the LSA threshold *T*), end up generating very different conversations.

EVALUATIONS OF AUTOTUTOR

Different types of performance evaluation can be made in an assessment of the success of AutoTutor. One type is technical and will not be addressed in depth in this article. This type evaluates whether particular computational modules of AutoTutor are producing output that is valid and satisfies the intended specifications. For example, we previously reported data on the accuracy of LSA and the speech act classifier in comparison with the accuracy of human judges. A second type of evaluation assesses the quality of the dialogue moves produced by AutoTutor. That is, it addresses the extent to which AutoTutor's dialogue moves are coherent, relevant, and smooth. A third type of evaluation assesses whether AutoTutor is successful in producing learning gains. A fourth assesses the extent to which learners like interacting with AutoTutor. In this section, we present what we know so far about the second and third types of evaluation.

Expert judges have evaluated AutoTutor with respect to conversational smoothness and the pedagogical quality of its dialogue moves (Person, Graesser, Kreuz, Pomeroy, & the TRG, 2001). The experts' mean ratings were positive (i.e., smooth rather than awkward conversation, good rather than bad pedagogical quality), but there is room for improvement in the naturalness and pedagogical effectiveness of the dialogue. In more recent studies, a *bystander Turing test* has been performed on the naturalness of AutoTutor's dialogue moves (Person, Graesser, & the TRG, 2002). In these studies, we randomly selected 144 tutor moves in the tutorial dialogues between students and AutoTutor. Six human tutors (from the computer literacy tutor pool at the University of Memphis) were asked to fill in what they would say at these 144 points. Thus, at each of these 144 tutor turns, the corpus contained what the human tutors generated and what AutoTutor generated. A group of computer literacy students was asked to discriminate between dialogue moves generated by a human versus those generated by a computer; half, in fact, were by humans and half were by computer. It was found that the bystander students were unable to discriminate whether particular dialogue moves had been generated by a computer or by a human; the d' discrimination scores were near zero.

The results of the bystander Turing test presented above are a rather impressive outcome that is compatible with the claim that AutoTutor is a good simulation of human tutors. AutoTutor manages to have productive and reasonably smooth conversations without achieving a complete and deep understanding of what the student expresses. There is an alternative interpretation, however,

which is just as interesting. Perhaps tutorial dialogue is not highly constrained, so the tutor has great latitude in what can be said without disrupting the conversation. In essence, there is a large landscape of options regarding what the tutor can say at most points in the dialogue. If this is the case, then it truly is feasible to develop tutoring technologies around NLD. These conversations are flexible and resilient, not fragile.

AutoTutor has been evaluated on learning gains in several experiments on the topics of computer literacy (Graesser, Moreno, et al., 2003; Person, Graesser, Bautista, Mathews, & the TRG, 2001) and conceptual physics (Graesser, Jackson, et al., 2003; VanLehn & Graesser, 2002). In most of the studies, a pretest is administered, followed by a tutoring treatment, followed by a posttest. In some experiments, there is a posttest-only design and no pretest. In the tutoring treatments, AutoTutor's scores are compared with scores of different types of comparison conditions. The comparison conditions vary from experiment to experiment because colleagues have had different views on what a suitable control condition would be. AutoTutor posttest scores have been compared with (1) pretest scores (*pretest*); (2) read-nothing scores (*read nothing*); (3) scores after relevant chapters from the course textbook are read (*textbook*); (4) same as (3) except that content is included only if it is directly relevant to the content during training by AutoTutor (*textbook reduced*); and (5) scores after the student reads text prepared by the experimenters that succinctly describes the content covered in the curriculum script of AutoTutor (*script content*). The dependent measures were different for computer literacy and physics, so the two sets of studies will be discussed separately.

Table 1 presents data on three experiments on computer literacy. The data in Table 1 constitute a reanalysis of studies reported in two published conference proceedings (Graesser, Moreno, et al., 2003; Person, Graesser, Bautista, et al., 2001). The numbers of students run in Experiments 1, 2, and 3 were 36, 24, and 81, respectively. The students learned about hardware, operating systems, and the Internet by collaboratively answering 12 questions with AutoTutor or being assigned to the read-nothing or the textbook condition. Person, Graesser, Bautista, et al. used a repeated measures design, counterbalancing training condition (AutoTutor, textbook, and read nothing) against the three topic areas (hardware, operating systems, and Internet). The students were given three types of tests. The shallow test consisted of multiple-choice questions that were randomly selected from a test bank associated with the textbook chapters in the computer literacy course. All of these questions were classified as shallow questions according to Bloom's (1956) taxonomy. Experts on computer literacy constructed the deep questions associated with the textbook chapters; these were multiple-choice questions that tapped causal mental models and deep reasoning. It should be noted that these shallow and deep multiple-choice questions were prepared by individuals who were not aware of the content of AutoTutor. They were written to cover content in the textbook on computer literacy. In contrast, the cloze task was tailored to AutoTutor training. The ideal answers to the questions during training were presented on

Table 1
Results of AutoTutor Experiments on Computer Literacy

	Test									
	Shallow				Deep				Cloze	
	Pretest		Posttest		Pretest		Posttest		Posttest	
Experiments	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*
Experiment 1[a]										
AutoTutor		–	.597	.22		–	.547	.27	.383	.15
Textbook		–	.606	.21		–	.515	.22	.331	.15
Read nothing		–	.565	.26		–	.476	.25	.295	.14
Effect size$_t$				−0.04				0.15		0.35
Effect size$_n$				0.12				0.28		0.63
Experiment 2[a]										
AutoTutor		–	.577	.18		–	.580	.26	.358	.19
Textbook		–	.553	.23		–	.452	.28	.305	.17
Read nothing		–	.541	.23		–	.425	.32	.256	.14
Effect size$_t$				0.10				0.46		0.31
Effect size$_n$				0.16				0.48		0.73
Experiment 3[b]										
AutoTutor	.541	.26	.520	.26	.383	.15	.496	.16	.322	.19
Textbook reduced	.561	.23	.539	.24	.388	.16	.443	.17	.254	.15
Read nothing	.523	.21	.515	.25	.379	.16	.360	.14	.241	.16
Effect size$_t$				−0.08				0.31		0.45
Effect size$_n$				0.02				0.97		0.51
Effect size$_p$				−0.08				0.75		–

Note—Effect size$_t$ = effect size using the textbook comparison group; effect size$_n$ = effect size using the read-nothing comparison group; effect size$_p$ = effect size using the pretest. [a]Reanalysis of data reported in Person, Graesser, Bautista, Mathews, and the Tutoring Research Group (2001). [b]Reanalysis of data reported in Graesser, Moreno, et al. (2003).

the cloze test, with four content words deleted for each answer. The student's task was to fill in the missing content words. The proportion of questions answered correctly served as the metric for the shallow, deep, and cloze tasks. The Graesser, Moreno, et al. study had the same design except that a pretest was administered, there was an expanded set of deep multiple-choice questions, and a textbook-reduced comparison condition was used instead of the textbook condition.

In Table 1, the means, standard deviations, and effect sizes in standard deviation units are reported. The effect sizes revealed that AutoTutor did not facilitate learning on the shallow multiple-choice test questions that had been prepared by the writers of the test bank for the textbook. All of these effect sizes were low or negative (mean effect size was .05 for the seven values in Table 1). Shallow knowledge is not the sort of knowledge that AutoTutor was designed to deal with, so this result is not surprising. AutoTutor was built to facilitate deep reasoning, and this was apparent in the effect sizes for the deep multiple-choice questions. All seven of these effect sizes in Table 1 were positive, with a mean of .49. Similarly, all six of the effect sizes for the cloze tests were positive, with a mean of .50. These effect sizes were generally larger when the comparison condition was read nothing ($M = .43$ for nine effect sizes in Table 1) than when the comparison condition was the textbook or textbook-reduced condition ($M = .22$). These means are in the same ball park as human tutoring, which has shown an effect size of .42 in comparison with classroom controls in the meta-analysis of Cohen et al. (1982). The control conditions in Cohen et al.'s meta-analyses are analogous to the read-nothing condition in the present experiments, since the participants in the present study all had classroom experience with computer literacy.

Table 2 shows results of an experiment on conceptual physics that was reported in a published conference proceeding (Graesser, Jackson, et al., 2003). The participants were given a pretest, completed training, and were given a posttest. The conditions were AutoTutor, textbook, and read nothing. The two tests tapped deeper comprehension and consisted of either multiple-choice questions or conceptual physics problems that required essay answers (which were graded by PhDs in physics). All six of the effect sizes in Table 2 were positive ($M = .71$). Additional experiments are described by VanLehn and Graesser (2002), who administered the same tests with a variety of control conditions. When all of the conceptual physics studies to date as well as the multiple-choice and essay tests are taken into account, the mean effect sizes of AutoTutor have varied when contrasted with particular comparison conditions: read nothing (.67), pretest (.82), textbook (.82), script content (.07), and human tutoring in computer-mediated conversation (.08).

There are a number of noteworthy outcomes of the analyses presented above. First, AutoTutor is effective in promoting learning gains at deep levels of comprehension in comparison with the typical ecologically valid situation in which students (all too often) read nothing, start out at pretest, or read the textbook for an amount of time equivalent to that involved in using AutoTutor. We estimate the effect size as .70 when considering these comparison conditions, the deeper tests of comprehension, and both computer literacy and physics. Second, it is surprising that reading the textbook is not much different than reading nothing. It appears that a tutor is needed to encourage the learner to focus on deeper levels of comprehension. Third, AutoTutor is as effective as a human tutor who communicates with the student over terminals in computer-mediated conversation. The human tutors had doctoral degrees in physics and extensive experience within a learning setting, yet they did not outperform AutoTutor. Such a result clearly must be replicated before we can consider it to be a well-established finding. However, the early news is quite provocative. Fourth, the impact of AutoTutor on learning gains is considerably reduced when a comparison is made with reading of text that is carefully tailored to exactly match the content covered by AutoTutor. That is, textbook-reduced and script-content controls yielded an AutoTutor effect size of only .22 when the subject matters of computer literacy and physics were combined. Of course, in the real world texts are rarely crafted to copy tutoring content, so the status of this control is uncertain in the arena of practice. But it does suggest that the impact of AutoTutor is

Table 2
Results of AutoTutor Experiment[a] on Conceptual Physics

	Multiple Choice				Physics Essays			
	Pretest		Posttest		Pretest		Posttest	
	M	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*
AutoTutor	.597	.17	.725	.15	.423	.30	.575	.26
Textbook	.566	.13	.586	.11	.478	.25	.483	.25
Read nothing	.633	.17	.632	.15	.445	.28	.396	.25
Effect size$_t$			1.26				.37	
Effect size$_n$			.62				.72	
Effect size$_p$			.75				.51	

Note—Effect size$_t$ = effect size using the textbook comparison group; effect size$_n$ = effect size using the read-nothing comparison group; effect size$_p$ = effect size using the pretest. [a]Reanalysis of data reported in Graesser, Jackson, et al. (2003).

dramatically reduced or disappears when there are comparison conditions that present content with information equivalent to that of AutoTutor.

What it is about AutoTutor that facilitates learning remains an open question. Is it the dialogue content or the animated agent that accounts for the learning gains? What role do motivation and emotions play, over and above the cognitive components? We suspect that the animated conversational agent will fascinate some students and possibly enhance motivation. Learning environments have only recently had animated conversational agents with facial features synchronized with speech and, in some cases, appropriate gestures (Cassell & Thorisson, 1999; Johnson, Rickel, & Lester, 2000; Massaro & Cohen, 1995). Many students will be fascinated with an agent that controls the eyes, eyebrows, mouth, lips, teeth, tongue, cheekbones, and other parts of the face in a fashion that is meshed appropriately with the language and emotions of the speaker (Picard, 1997). The agents provide an anthropomorphic human–computer interface that simulates a conversation with a human. This will be exciting to some, frightening to a few, annoying to others, and so on. There is some evidence that these agents tend to have a positive impact on learning or on the learner's perceptions of the learning experience in comparison with speech alone or text controls (Atkinson, 2002; Moreno, Mayer, Spires, & Lester, 2001; Whittaker, 2003). However, additional research is needed to determine the precise conditions, agent features, and levels of representation that are associated with learning gains. According to Graesser, Moreno, et al. (2003), it is the dialogue content, and not the speech or animated facial display, that influences learning, whereas the animated agent can have an influential role (positive, neutral, or negative) in motivation. One rather provocative result is that there is a near-zero correlation between learning gains and how much the students like the conversational agents (Moreno, Klettke, Nibbaragandla, Graesser, & the TRG, 2002). Therefore, it is important to distinguish liking from learning in this area of research. Although the jury may still be out on exactly what it is about AutoTutor that leads to learning gains, the fact is that students learn from the intelligent tutoring system and some enjoy having conversations with AutoTutor in natural language.

REFERENCES

Aleven, V., & Koedinger, K. R. (2002). An effective metacognitive strategy: Learning by doing and explaining with a computer-based cognitive tutor. *Cognitive Science*, **26**, 147-179.

Allen, J. (1995). *Natural language understanding*. Redwood City, CA: Benjamin/Cummings.

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *Journal of the Learning Sciences*, **4**, 167-207.

Atkinson, R. K. (2002). Optimizing learning from examples using animated pedagogical agents. *Journal of Educational Psychology*, **94**, 416-427.

Bloom, B. S. (Ed.) (1956). *Taxonomy of educational objectives: The classification of educational goals. Handbook I: Cognitive domain.* New York: Longmans, Green.

Burgess, C., Livesay, K., & Lund, K. (1998). Explorations in context space: Words, sentences, and discourse. *Discourse Processes*, **25**, 211-257.

Cassell, J., & Thorisson, K. (1999). The power of a nod and a glance: Envelope vs. emotional feedback in animated conversational agents. *Applied Artificial Intelligence*, **13**, 519-538.

Chi, M. T. H., de Leeuw, N., Chiu, M., & LaVancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive Science*, **18**, 439-477.

Chi, M. T. H., Siler, S. A., Jeong, H., Yamauchi, T., & Hausmann, R. G. (2001). Learning from human tutoring. *Cognitive Science*, **25**, 471-533.

Cohen, P. A., Kulik, J. A., & Kulik, C. C. (1982). Educational outcomes of tutoring: A meta-analysis of findings. *American Educational Research Journal*, **19**, 237-248.

Colby, K. M., Weber, S., & Hilf, F. D. (1971). Artificial paranoia. *Artificial Intelligence*, **2**, 1-25.

Collins, A., Brown, J. S., & Newman, S. E. (1989). Cognitive apprenticeship: Teaching the craft of reading, writing, and mathematics. In L. B. Resnick (Ed.), *Knowing, learning, and instruction: Essays in honor of Robert Glaser* (pp. 453-494). Hillsdale, NJ: Erlbaum.

Collins, A., Warnock, E. H., & Passafiume, J. J. (1975). Analysis and synthesis of tutorial dialogues. In G. H. Bower (Ed.), *The psychology of learning and motivation: Advances in research and theory* (Vol. 8, pp. 49-87). New York: Academic Press.

DARPA (1995). *Proceedings of the Sixth Message Understanding Conference* (MUC-6). San Francisco: Morgan Kaufman.

Foltz, P. W., Gilliam, S., & Kendall, S. (2000). Supporting content-based feedback in on-line writing evaluation with LSA. *Interactive Learning Environments*, **8**, 111-128.

Fox, B. (1993). *The human tutorial dialogue project*. Hillsdale, NJ: Erlbaum.

Graesser, A. C., Gernsbacher, M. A., & Goldman, S. (Eds.) (2003). *Handbook of discourse processes*. Mahwah, NJ: Erlbaum.

Graesser, A. C., Hu, X., & McNamara, D. S. (in press). Computerized learning environments that incorporate research in discourse psychology, cognitive science, and computational linguistics. In A. F. Healy (Ed.), *Experimental cognitive psychology and its applications: Festschrift in honor of Lyle Bourne, Walter Kintsch, and Thomas Landauer*. Washington, DC: American Psychological Association.

Graesser, A. C., Jackson, G. T., Mathews, E. C., Mitchell, H. H., Olney, A., Ventura, M., & Chipman, P. (2003). Why/AutoTutor: A test of learning gains from a physics tutor with natural language dialog. In R. Alterman & D. Hirsh (Eds.), *Proceedings of the Twenty-Fifth Annual Conference of the Cognitive Science Society* (pp. 1-6). Mahwah, NJ: Erlbaum.

Graesser, A. C., Moreno, K., Marineau, J., Adcock, A., Olney, A., Person, N., & the Tutoring Research Group (2003). AutoTutor improves deep learning of computer literacy: Is it the dialogue or the talking head? In U. Hoppe, F. Verdejo, & J. Kay (Eds.), *Proceedings of artificial intelligence in education* (pp. 47-54). Amsterdam: IOS Press.

Graesser, A. C., & Olde, B. A. (2003). How does one know whether a person understands a device? The quality of the questions the person asks when the device breaks down. *Journal of Educational Psychology*, **95**, 524-536.

Graesser, A. C., & Person, N. K. (1994). Question asking during tutoring. *American Educational Research Journal*, **31**, 104-137.

Graesser, A. C., Person, N. K., Harter, D., & the Tutoring Research Group (2001). Teaching tactics and dialogue in AutoTutor. *International Journal of Artificial Intelligence in Education*, **12**, 257-279.

Graesser, A. C., Person, N. K., & Magliano, J. P. (1995). Collaborative dialogue patterns in naturalistic one-on-one tutoring. *Applied Cognitive Psychology*, **9**, 359-387.

Graesser, A. C., VanLehn, K., Rosé, C. P., Jordan, P. W., & Harter, D. (2001). Intelligent tutoring systems with conversational dialogue. *AI Magazine*, **22**(4), 39-52.

Graesser, A. C., Wiemer-Hastings, K., Wiemer-Hastings, P., Kreuz, R., & the Tutoring Research Group (1999). AutoTutor: A simulation of a human tutor. *Journal of Cognitive Systems Research*, **1**, 35-51.

Graesser, A. C., Wiemer-Hastings, P., Wiemer-Hastings, K., Har-

ter, D., Person, N. K., & the Tutoring Research Group (2000). Using latent semantic analysis to evaluate the contributions of students in AutoTutor. *Interactive Learning Environments*, **8**, 129-148.

Gratch, J., Rickel, J., André, E., Cassell, J., Petajan, E., & Badler, N. (2002). Creating interactive virtual humans: Some assembly required. *IEEE Intelligent Systems*, **17**, 54-63.

Harabagiu, S. M., Maiorano, S. J., & Pasca, M. A. (2002). Open-domain question answering techniques. *Natural Language Engineering*, **1**, 1-38.

Heffernan, N. T., & Koedinger, K. R. (1998). A developmental model for algebra symbolization: The results of a difficulty factor assessment. In M. A. Gernsbacher & S. J. Derry (Eds.), *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society* (pp. 484-489). Mahwah, NJ: Erlbaum.

Hu, X., Cai, Z., Graesser, A. C., Louwerse, M. M., Penumatsa, P., Olney, A., & the Tutoring Research Group (2003). An improved LSA algorithm to evaluate student contributions in tutoring dialogue. In G. Gottlob & T. Walsh (Eds.), *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence* (pp. 1489-1491). San Francisco: Morgan Kaufmann.

Hume, G. D., Michael, J. A., Rovick, A., & Evens, M. W. (1996). Hinting as a tactic in one-on-one tutoring. *Journal of the Learning Sciences*, **5**, 23-47.

Johnson, W. L., Rickel, J. W., & Lester, J. C. (2000). Animated pedagogical agents: Face-to-face interaction in interactive learning environments. *International Journal of Artificial Intelligence in Education*, **11**, 47-78.

Jurafsky, D., & Martin, J. H. (2000). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Upper Saddle River, NJ: Prentice Hall.

Kintsch, E., Steinhart, D., Stahl, G., & the LSA Research Group (2000). Developing summarization skills through the use of LSA-based feedback. *Interactive Learning Environments*, **8**, 87-109.

Kintsch, W. (1998). *Comprehension: A paradigm for cognition*. New York: Cambridge University Press.

Landauer, T. K., & Dumais, S. T. (1997). An answer to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, **104**, 211-240.

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes*, **25**, 259-284.

Lehnert, W. G., & Ringle, M. H. (Eds.) (1982). Strategies for natural language processing. Hillsdale, NJ: Erlbaum.

Lenat, D. B. (1995). CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, **38**, 33-38.

Louwerse, M. M., Graesser, A. C., Olney, A., & the Tutoring Research Group (2002). Good computational manners: Mixed-initiative dialogue in conversational agents. In C. Miller (Ed.), *Etiquette for human–computer work: Papers from the 2002 fall symposium, Technical Report FS-02-02* (pp. 71-76). North Falmouth, MA: AAAI Press.

Louwerse, M. M., & Mitchell, H. H. (2003). Towards a taxonomy of a set of discourse markers in dialog: A theoretical and computational linguistic account. *Discourse Processes*, **35**, 199-239.

Louwerse, M. M., & van Peer, W. (Eds.) (2002). *Thematics: Interdisciplinary studies*. Philadelphia: John Benjamins.

Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. Cambridge, MA: MIT Press.

Massaro, D. W., & Cohen, M. M. (1995). Perceiving talking faces. *Current Directions in Psychological Science*, **4**, 104-109.

Moore, J. D. (1995). *Participating in explanatory dialogues*. Cambridge, MA: MIT Press.

Moore, J. D., & Wiemer-Hastings, P. (2003). Discourse in computational linguistics and artificial intelligence. In A. C. Graesser, M. A. Gernsbacher, & S. R. Goldman (Eds.), *Handbook of discourse processes* (pp. 439-486). Mahwah, NJ: Erlbaum.

Moreno, K. N., Klettke, B., Nibbaragandla, K., Graesser, A. C., & the Tutoring Research Group (2002). Perceived characteristics and pedagogical efficacy of animated conversational agents. In S. A. Cerri, G. Gouarderes, & F. Paraguacu (Eds.), *Proceedings of the Sixth International Conference on Intelligent Tutoring Systems* (pp. 963-971). Berlin: Springer-Verlag.

Moreno, R., Mayer, R. E., Spires, H. A., & Lester, J. C. (2001). The case for social agency in computer-based teaching: Do students learn more deeply when they interact with animated pedagogical agents? *Cognition & Instruction*, **19**, 177-213.

Norman, D. A., & Rumelhart, D. E. (1975). *Explorations in cognition*. San Francisco: Freeman.

Olde, B. A., Franceschetti, D. R., Karnavat, A., Graesser, A. C., & the Tutoring Research Group (2002). The right stuff: Do you need to sanitize your corpus when using latent semantic analysis? In W. D. Gray & C. D. Schunn (Eds.), *Proceedings of the Twenty-Fourth Annual Meeting of the Cognitive Science Society* (pp. 708-713). Mahwah, NJ: Erlbaum.

Olney, A., Louwerse, M. M., Mathews, E. C., Marineau, J., Mitchell, H. H., & Graesser, A. C. (2003). Utterance classification in AutoTutor. In J. Burstein & C. Leacock (Eds.), *Building educational applications using natural language processing: Proceedings of the Human Language Technology, North American chapter of the Association for Computational Linguistics Conference 2003 Workshop* (pp. 1-8). Philadelphia: Association for Computational Linguistics.

Palincsar, A. S., & Brown, A. (1984). Reciprocal teaching of comprehension-fostering and comprehension-monitoring activities. *Cognition & Instruction*, **1**, 117-175.

Person, N. K., Graesser, A. C., Bautista, L., Mathews, E. C., & the Tutoring Research Group (2001). Evaluating student learning gains in two versions of AutoTutor. In J. D. Moore, C. L. Redfield, & W. L. Johnson (Eds.), *Artificial intelligence in education: AI-ED in the wired and wireless future* (pp. 286-293). Amsterdam: IOS Press.

Person, N. K., Graesser, A. C., Kreuz, R. J., Pomeroy, V., & the Tutoring Research Group (2001). Simulating human tutor dialogue moves in AutoTutor. *International Journal of Artificial Intelligence in Education*, **12**, 23-39.

Person, N. K., Graesser, A. C., & the Tutoring Research Group (2002). Human or computer? AutoTutor in a bystander Turing test. In S. A. Cerri, G. Gouarderes, & F. Paraguacu (Eds.), *Proceedings of the Sixth International Conference on Intelligent Tutoring Systems* (pp. 821-830). Berlin: Springer-Verlag.

Picard, R. W. (1997). *Affective computing*. Cambridge, MA: MIT Press.

Rich, C., & Sidner, C. L. (1998). COLLAGEN: A collaborative manager for software interface agents. *User Modeling & User-Adapted Interaction*, **8**, 315-350.

Rickel, J., Lesh, N., Rich, C., Sidner, C. L., & Gertner, A. S. (2002). Collaborative discourse theory as a foundation for tutorial dialogue. In S. A. Cerri, G. Gouarderes, & F. Paraguacu (Eds.), *Proceedings of the Sixth International Conference on Intelligent Tutoring Systems* (pp. 542-551). Berlin: Springer-Verlag.

Schank, R. C., & Riesbeck, C. K. (Eds.) (1982). *Inside computer understanding: Five Programs Plus Miniatures*. Hillsdale, NJ: Erlbaum.

Sekine, S., & Grishman, R. (1995). A corpus-based probabilistic grammar with only two non-terminals. In H. Bunt (Ed.), *Fourth international workshop on parsing technology* (pp. 216-223). Prague: Association for Computational Linguistics.

Shah, F., Evens, M. W., Michael, J., & Rovick, A. (2002). Classifying student initiatives and tutor responses in human keyboard-to-keyboard tutoring sessions. *Discourse Processes*, **33**, 23-52.

Sleeman, D., & Brown, J. S. (Eds.) (1982). *Intelligent tutoring systems*. New York: Academic Press.

Song, K., Hu, X., Olney, A., Graesser, A. C., & the Tutoring Research Group (in press). A framework of synthesizing tutoring conversation capability with Web based distance education courseware. *Computers & Education*.

Susarla, S., Adcock, A., Van Eck, R., Moreno, K., Graesser, A. C., & the Tutoring Research Group (2003). Development and evaluation of a lesson authoring tool for AutoTutor. In V. Aleven, U. Hoppe, J. Kay, R. Mizoguchi, H. Pain, F. Verdejo, & K. Yacef (Eds.), *AIED2003 supplemental proceedings* (pp. 378-387). Sydney: University of Sydney School of Information Technologies.

VanLehn, K., & Graesser, A. C. (2002). *Why2 report: Evaluation of Why/Atlas, Why/AutoTutor, and accomplished human tutors on learning gains for qualitative physics problems and explanations*. Unpublished report prepared by the University of Pittsburgh CIRCLE group and the University of Memphis Tutoring Research Group.

VanLehn, K., Jones, R. M., & Chi, M. T. H. (1992). A model of the self-explanation effect. *Journal of the Learning Sciences*, **2**, 1-60.

VanLehn, K., Jordan, P., Rosé, C. P., Bhembe, D., Bottner, M., Gaydos, A., Makatchev, M., Pappuswamy, U., Ringenberg, M., Roque, A., Siler, S., & Srivastava, R. (2002). The architecture of Why2-Atlas: A coach for qualitative physics essay writing. In S. A. Cerri, G. Gouarderes, & F. Paraguacu (Eds.), *Proceedings of the Sixth International Conference on Intelligent Tutoring Systems* (pp. 158-167). Berlin: Springer-Verlag.

VanLehn, K., Lynch, C., Taylor, L., Weinstein, A., Shelby, R., Schulze, K., Treacy, D., & Wintersgill, M. (2002). Minimally invasive tutoring of complex physics problem solving. In S. A. Cerri, G. Gouarderes, & F. Paraguacu (Eds.), *Proceedings of the Sixth International Conference on Intelligent Tutoring Systems* (pp. 367-376). Berlin: Springer-Verlag.

Voorhees, E. M. (2001). The TREC question answering track. *Natural Language Engineering*, **7**, 361-378.

Weizenbaum, J. (1966). ELIZA—A computer program for the study of natural language communication between man and machine. *Communications of the ACM*, **9**, 36-45.

Whittaker, S. (2003). Theories and methods in mediated communication. In A. C. Graesser, M. A. Gernsbacher, & S. R. Goldman (Eds.), *Handbook of discourse processes* (pp. 243-286). Mahwah, NJ: Erlbaum.

Winograd, T. (1972). *Understanding natural language*. New York: Academic Press.

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.

(Manuscript received January 25, 2004;
accepted for publication March 14, 2004.)