ELSEVIER

Journal of Cognitive Systems Research 1 (1999) 35–51

Cognitive Systems RESEARCH

www.elsevier.com/locate/cogsys

AutoTutor: A simulation of a human tutor

Action editor: Gregg C. Oden

Arthur C. Graesser*, Katja Wiemer-Hastings, Peter Wiemer-Hastings, Roger Kreuz, Tutoring Research Group, University of Memphis

Department of Psychology, University of Memphis, Campus Box 526400, Memphis, TN 38152-6400, USA

Received 6 August 1999; accepted 9 August 1999

Abstract

AutoTutor is a computer tutor that simulates the discourse patterns and pedagogical strategies of a typical human tutor. AutoTutor is designed to assist college students in learning the fundamentals of hardware, operating systems, and the Internet in an introductory computer literacy course. Most tutors in school systems are not highly trained in tutoring techniques and have only a modest expertise on the tutoring topic, but they are surprisingly effective in producing learning gains in students. We have dissected the discourse and pedagogical strategies these unskilled tutors exhibit by analyzing approximately 100 hours of naturalistic tutoring sessions. These mechanisms are implemented in AutoTutor. AutoTutor presents questions and problems from a curriculum script, attempts to comprehend learner contributions that are entered by keyboard, formulates dialog moves that are sensitive to the learner's contributions (such as short feedback, pumps, prompts, elaborations, corrections, and hints), and delivers the dialog moves with a talking head. AutoTutor has seven modules: a curriculum script, language extraction, speech act classification, latent semantic analysis, topic selection, dialog move generation, and a talking head.

Keywords: Computer literacy; Latent semantic analysis; Question answering; Tutorial dialog; Tutoring

1. AutoTutor: A simulation of a human tutor

The Tutoring Research Group at the University of Memphis has been developing a computer tutor, called AutoTutor, that simulates a typical human tutor (Graesser et al., 1998; Wiemer-Hastings et al., 1998). AutoTutor attempts to comprehend student contributions and to simulate dialog moves of human tutors. AutoTutor is currently simulating the dialog moves of normal (unskilled) tutors, but eventually we hope to incorporate more sophisticated tutoring strategies. AutoTutor is currently being developed for college students who take an introductory course in computer literacy. These students learn the fundamentals of computer hardware, the operating system, and the Internet.

AutoTutor never would have been developed without the contributions from cognitive science during the last 30 years. AutoTutor incorporates several theories and computational architectures that have been inherited from cognitive science. It is extremely challenging to build a complex artifact like AutoTutor, particularly in an academic environ-

*Corresponding author. Fax: + 1-901-678-2579.

E-mail address: a-graesser@memphis.edu (A.C. Graesser)

1389-0417/99/$ – see front matter
PII: S1389-0417(99)00005-4

ment with a comparatively limited budget. It requires the integration of expertise from many different fields: computer science, cognitive and social psychology, discourse processes, computational linguistics, education, learning technology, and even art. Many researchers are reluctant to stretch their identities to facilitate progress in an interdisciplinary team. However, those with the creative, flexible intellectual styles soon learn that the product of an interdisciplinary effort is superior to whatever could be accomplished within one discipline. Fortunately, there have been enough cognitive scientists aboard the research team building AutoTutor that substantial progress has been made.

A brief snapshot of AutoTutor in action should concretize the nature of AutoTutor. AutoTutor has a talking head that serves as a conversation partner with the learner. The talking head delivers AutoTutor's dialog moves with synthesized speech, intonation, facial expressions, and gestures. AutoTutor prints the major question or problem that is being worked on at the top of the screen. These questions are generated systematically from a curriculum script, a module that will be discussed later. AutoTutor's questions and problems are not the fill-in-the-blank, true/false, or multiple-choice questions that are so popular in the US educational system. Instead, the questions and problems invite lengthy explanations and deep reasoning (e.g., answers to why, how, what-if questions). The goal is to encourage students to articulate lengthier answers that exhibit deep reasoning, rather than to recite short snippets of shallow knowledge. There is a multi-turn tutorial dialog between AutoTutor and the learner during the course of answering a question (or solving a problem). When considering both the learner and AutoTutor, it typically takes 10 to 30 turns during the tutorial dialog that occurs when a single question from the curriculum script is answered. The learner types in his/her contributions during the exchange by keyboard. For some topics, there are graphical displays and animation, with components that AutoTutor points to. We want AutoTutor to be a good conversation partner that comprehends, speaks, points, and displays emotions, all in a coordinated fashion. Other questions and problems do not have a graphical display and animation, as is the case in normal tutoring.

The exchange below is one of the shorter tutorial dialogs between a college student and AutoTutor. This student was an above-average student in the computer literacy class.

1. AUTOTUTOR: When you turn on the computer, how is the operating system first activated and loaded into RAM?
2. STUDENT: When the computer is turned on a file is automatically booted up.
3. AUTOTUTOR: Anything else?
4. STUDENT: This file contains the operating system.
5. AUTOTUTOR: What else?
6. STUDENT: The operating system is then read into RAM and activated.
7. AUTOTUTOR: Right. The CPU executes instructions in ROM, which help the operating system to start.
8. STUDENT: After being read to ROM the OS is activated.
9. AUTOTUTOR: Right. Let's review what we've just gone over. The operating system must first be stored on the hard disk. When you turn on the computer, the CPU executes instructions stored in ROM. These instructions help boot the operating system. The operating system is then loaded from the hard disk into RAM.

AutoTutor first asks a how-question in turn 1. Then the student provides answer information in turns 2, 4, and 6. These answers are encouraged by AutoTutor's *pumps* in turns 3 and 5 (i.e., 'Anything else?' and 'What else?'). AutoTutor delivers these pumps in an effort to encourage active learning and question answering on the part of the student. Thus, instead of being an *information delivery system* that bombards the student with a large volume of information, AutoTutor is a *discourse prosthesis* that attempts to get the student to do the talking and explore what the student knows. However, it would not be good for the student to flounder unproductively for a long time. AutoTutor sometimes needs to bring the student back on track by supplying cues and clues that lead to the evolution of a complete answer to the question. The student had forgotten the role of ROM in launching the operating system, so AutoTutor brings up ROM in turn 7. The student builds on this suggestion in turn 8. At that point, the important pieces of a good complete answer have been covered, so AutoTutor summarizes the answer in turn 9. AutoTutor periodically gives positive immediate feedback after the student contributions (i.e., 'Right'). This feedback not only is motivating, but creates the impression that AutoTutor is listening to what the student is communicating. As we will discuss shortly, these characteristics of a tutorial exchange are quite similar to discourse patterns in normal tutoring between humans.

Now that the general features of AutoTutor have been described, we plan on accomplishing two major objectives in the remainder of this article. In the next section we will describe what normal tutors do when students in a school system are tutored. We have videotaped, transcribed, and analyzed nearly 100 hours of naturalistic tutoring sessions during the last decade. After analyzing this rich corpus, we discovered what tutors do versus do not do during most tutoring sessions. Our discoveries were enlightening and often counterintuitive. Most people have a misleading, comic-book impression of what goes on in a tutoring session. The subsequent section describes the mechanisms of AutoTutor that simulate the normal human tutor. AutoTutor has seven modules: a curriculum script, language extraction, speech act classification, latent semantic analysis, topic selection, dialog move generation, and a talking head. It is beyond the scope of this article to describe these modules in detail, but our coverage will be sufficient for conveying a functional understanding of the computational architectures and for presenting highlights on the performance of each module.

2. What do typical human tutors do?

The obvious question that most of our colleagues have asked is why we bother simulating only a normal, unskilled tutor. Why don't we hunt for the rare, accomplished, expert tutors and simulate them? We have two answers to this question. The first is that normal unskilled tutors are surprisingly effective in producing learning gains. The second is that we learn a lot about normal discourse mechanisms by analyzing the normal unskilled tutor.

It is well documented that human tutoring is a very effective method of instruction. Human-to-human tutoring enhances learning by 0.4 to 2.3 standard deviation units compared to classroom controls and other suitable controls (Cohen et al., 1982; Bloom, 1984). Human tutors are extremely effective even though over 90% percent of the tutors in actual school systems are untrained in tutoring skills and have moderate domain knowledge. They are peer tutors, cross-age tutors, or paraprofessionals, but rarely accomplished professionals. Cohen et al.'s meta-analyses of dozens of tutoring studies revealed that the amount of tutoring experience did not significantly predict learning gains (Cohen et al., 1982) . However, this result must be tempered by the fact that there is a serious burnout rate in the tutoring enterprise and that the vast majority of the tutors in the sample had less than 10 hours of tutoring experience. Nevertheless, the notion that untrained tutors are extremely effective in promoting learning is quite counterintuitive.

The provocative result that unskilled tutors are so effective motivated us to investigate what real tutors do. We analyzed videotapes of approximately 100 hours of untrained tutors in naturalistic tutoring sessions (Graesser & Person, 1994; Graesser et al., 1995; Person & Graesser, 1999; Person et al., 1994; Person et al., 1995). In one corpus of tutorial dialog, undergraduate psychology majors were tutored by graduate students on difficult topics in research methods and statistics. In another corpus, seventh-graders were tutored on challenging topics in algebra by high-school students. These videotaped sessions were transcribed and dissected in fine detail. We analyzed paralinguistic activities in addition to the verbal transcript. For example, tutors typically give short immediate feedback immediately after a student's turn. These speech acts are extremely important because they convey to the student the quality of the student's discourse contributions and most students worry how well they are doing. This feedback is normally either positive ('That's right,' 'Yeah'), neutral ('Okay,' 'Uh-huh'), or negative ('Not quite,' 'No'), although mixed feedback and a finer gradation of feedback is often delivered by the tutor. We recorded the facial expressions, pauses, duration, pitch, intensity, and intonation contours of the short immediate feedback in an effort to unpack subtle features of these important speech acts. Other paralinguistic activities consisted of drawing on a white board, referring to elements in these drawings, pointing to a textbook, and so on. It is important to point out that the detailed observational methodology is business as usual in the fields of discourse processes, sociology, and anthropology, but not in the mainstream practices of experimental psychology and computer science. The mark of a cognitive scientist is to adopt the methodology that is needed to dissect the phenomenon being explored, as opposed to selecting a phenomenon which fits the constraints of a familiar methodology.

Our discourse analyses convinced us that there is something about interactive discourse that is responsible for learning gains. What are these discourse patterns? One way to begin the anatomy of naturalistic tutoring is to describe what was conspicuously missing. Then we will turn to a discussion of what normal tutors do.

2.1. Active learning, curiosity, and question asking

The students were not active, self-regulated learners who are aware of their knowledge deficits and who took command of the tutorial agenda. Instead, it was the tutor who set the agenda and introduced most of the topics, questions, and problems. In the sample of tutoring sessions that we examined, the tutor set 100% of the agenda in a tutoring session, introduced 93% of the topics, presented 82% of the examples, and asked 80% of the questions. A typical student asks only 6–8 genuine information-seeking questions per hour. Researchers have identified the particular conditions that trigger questions, such as anomalies, contradictions, obstacles to goals, glaring gaps in knowledge, and other forms of cognitive disequilibrium (Berlyne, 1960; Dillon, 1988; Festinger, 1957; Graesser & McMahen, 1993; Schank, 1986). However, the incidence of these conditions is surprisingly rare in tutoring. Instead, the tutor invokes a curriculum script of topics, problems, questions, and examples in a tutor-driven fashion. Stated differently, the matter of control in the mixed-initiative dialog is heavily slanted toward the tutor.

2.2. Enhanced shared knowledge

One conceivable advantage of tutoring is an enhanced ‘meeting of the minds’ between student and tutor. That is, the tutor infers the idiosyncratic knowledge, bugs, and misconceptions of the student, and the student’s knowledge drifts to the tutor’s knowledge base. Designers of intelligent tutoring systems have implemented ‘student modeling,’ which is an attempt to infer the knowledge states of a student on the basis of the student’s questions, answers to questions, and solutions to problems (Anderson et al., 1995; Ohlsson, 1986). Similarly, discourse theorists have frequently emphasized the importance of establishing shared meanings for successful communication (Clark, 1996; Roschelle, 1992; Schober, 1995). There is a radically different perspective on the matter of common ground and student modeling, however. ITS researchers have sometimes cast doubt on the possibility, the need, and the pedagogical utility of detailed student modeling (Newman, 1989). It is computationally difficult, if not impossible, to induce student knowledge. The educational payoffs of methodically unraveling the bugs and misconceptions of a student might be very small compared to the tutor’s modeling good skills and knowledge. Learning can perhaps proceed quite effectively when the tutor presents an environment, a set of tasks, and feedback to the learner, without carefully tracking the knowledge of the student.

Our detailed analyses of actual tutoring sessions revealed that there is a very slow convergence towards shared meanings during tutoring (Graesser et al., 1995; Person et al., 1994). The gap in knowledge between the tutor and student is frequently so wide that the two parties in the conversation frequently misunderstand each other and give each other incorrect feedback. For example, tutors normally give positive feedback (‘Yeah,’ ‘Uh-huh’) rather than negative feedback to student contributions that are vague, incoherent, or error-ridden. When tutors ask the students comprehension gauging questions (‘Do you understand?’), the lost students normally say yes, or nod their heads. In fact, it is the good student who is most likely to say, ‘*No*, I don’t follow’ (see also Chi et al., 1989). These data suggest that the feedback mechanisms between tutor and learner have low fidelity or are even misleading. More detailed analyses of the tutorial dialog convinced us further that detailed student modeling is very rare in naturalistic tutoring and that the tutor has only an approximate global metric of the student’s knowledge.

The large gulf that frequently exists between the knowledge of tutors and students also made us more confident that it is feasible to build a computer tutor that mimics human tutors. Human tutors do not normally achieve a deep and complete understanding of a student, particularly when the student contributions are fragmentary, ungrammatical, incoherent, underspecified, and vague. Instead, misunderstandings frequently occur as the tutor scrambles to piece together a minimal understanding of the student's knowledge and to manage the discourse. Thus, AutoTutor might manage quite well with a shallow understanding of the student's knowledge and a strategic selection of dialog moves. This conclusion is compatible with tests of the ANIMATE tutor developed by Nathan et al. (1992). ANIMATE produced impressive learning gains on algebra word problems, but did not construct a detailed map of what the student knows. Moreover, a key feature of effective learning lies in assisting students in actively constructing subjective explanations and elaborations of the material (Chi et al., 1994; Chi et al., 1989; McDaniel & Donnelly, 1996; Webb et al., 1995). The tutor's dialog moves in a collaborative exchange might provide sufficient scaffolding for a student to build such explanations, without the computer fully understanding what the student knows.

2.3. Sophisticated tutoring techniques

Our anatomy of normal tutoring sessions revealed that normal unskilled tutors do not use most of the ideal tutoring strategies that have been identified in education and the intelligent tutoring system enterprise. These strategies include the Socratic method (Collins, 1985), modeling–scaffolding–fading (Collins et al., 1989), reciprocal training (Palinscar & Brown, 1984), anchored situated learning (Bransford et al., 1991; Greeno et al., 1993), error diagnosis and correction (Van Lehn, 1990; Lesgold et al., 1992), frontier learning, building on prerequisites (Gagné, 1977), and sophisticated motivational techniques (Lepper et al., 1991). Detailed discourse analyses have been performed on small samples of accomplished tutors in an attempt to identify sophisticated tutoring strategies (Fox, 1993; Hume et al., 1996; McArthur et al., 1990; Merrill et al. 1992; Moore, 1995; Putnam, 1987). However, we discovered that these sophisticated tutoring strategies were practically nonexistent in the unskilled tutoring sessions that we videotaped and analyzed (Graesser et al., 1995; Person & Graesser, 1999). Tutors clearly need to be trained how to use the sophisticated tutoring skills because they do not routinely emerge in naturalistic tutoring with untrained tutors.

Our plan is to have later versions of AutoTutor incorporate some of the ideal tutoring strategies to the extent that it is computationally feasible to implement them. There will be a hybrid of unskilled strategies and ideal strategies. But our initial goal was to simulate the dialog moves of the normal unskilled tutor, which are known to be very effective (Cohen et al., 1982).

2.4. Curriculum scripts with examples and deep questions

A curriculum script is a loosely ordered but well-defined set of skills, concepts, example problems, and question–answer units (McArthur et al., 1990; Putnam, 1987). Most tutors follow a script-like macrostructure, but briefly deviate from the structure when the student manifests difficulties, misconceptions, and errors. In essence, there are vestiges of the classroom lecturer in most tutoring sessions; the tutor sets the agenda, the curricula, the topics, and the problems to work on. Unlike the classroom, however, the tutor listens to the individual student and produces dialog moves that encourage lengthier student contributions and stretches of reasoning.

The content of the curriculum script is deeper and more detailed than the normal classroom. There are more deep reasoning questions (e.g., why, how, what-if, what-if-not), more problems to solve, and more examples. There is independent evidence, outside of the arena of tutoring per se, that learning gains are enhanced by integrating didactic declarative knowledge with example problems and cases (Forbus et al., 1995; Hammond et al., 1991; Sweller, 1988) and with questions that promote deep reasoning (Edelson, 1996; King, 1994). Therefore, one explanation of the advantages of tutoring over classroom environments is that tutoring sessions have curriculum scripts with a higher density of cases, example problems, and questions that promote deep reasoning.

2.5. Collaborative construction of explanations

We discovered that a salient feature of naturalistic tutoring consists of generating dialog moves that assist learners in the active construction of explanations, elaborations, and mental models of the material. Other researchers have similarly proposed that the process of a learner actively constructing an explanation is critical for learning, and usually has a greater impact than merely presenting information to learners (Chi et al., 1989; Chi et al., 1994; Moore, 1995; Pressley et al., 1992; Webb et al., 1995). There are two faces to this claim, namely the role of explanations and the role of collaboration. At this point we are uncertain which of these two is most prominent because the two of them go hand-in-hand in naturalistic tutoring. Human tutors assist this construction of knowledge by delivering collaborative discourse moves that encourage learners to build explanations and that fill in the holes of missing information. In turn, the student actively builds 'self-explanations' of the material.

Graesser et al. (1995) documented some of the dialog moves that tutors generate to nurture the collaborative building of explanations. We have already mentioned the short feedback that occurs immediately after the student's turn. There is positive, neutral, and negative feedback, as illustrated below.

(1) *Positive immediate feedback*. 'That's right,' 'Yeah.'

(2) *Neutral immediate feedback*. 'Okay,' 'Uh-huh.'

(3) *Negative immediate feedback*. 'Not quite,' 'No.'

After this immediate feedback, the tutor normally issues a more substantive dialogue move. Presented below are the dialog moves that we have identified in naturalistic tutoring and implemented in AutoTutor.

(4) *Pumping*. The tutor pumps the student for more information during the early stages of answering a particular question (or solving a problem). The pump consists of positive feedback (e.g., 'Right,' 'Yeah,' dramatic head-nod), neutral backchannel feedback ('Uh-huh,' 'Okay,' subtle head-nod), or explicit requests for more information ('Tell me more,' 'What else?'). The tutor pumps for one or two cycles of turns before the tutor contributes information. Pumping serves the functions of exposing knowledge of the student and of encouraging students to construct content by themselves.

(5) *Prompting*. The tutor supplies the student with a discourse context and prompts him/her to fill in a missing word, phrase, or sentence. An example in the context of computer literacy would be 'The primary memories of the CPU are ROM and ___.' The prompt is delivered with intonation, facial expressions, and gestures that signal the learner to fill in the missing word or phrase. The one or two words that precede the missing word are drawn out, with a complex, bending intonation contour. Then there is a pause that gives the floor to the learner and invites the learner to fill in the missing information. Sometimes the facial expression or hand gestures have an encouraging stance, as if to say, 'Give me the next word.' Prompting is a scaffolding device for students who are reluctant to supply information. Students are expected to supply more content and more difficult content as they progress in learning the domain knowledge.

(6) *Hinting*. When the student is having problems answering a question or solving a problem, the tutor gives hints by presenting a fact, asking a leading question, or reframing the problem. A hint may be a memory cue or a critical problem-solving clue. Hints are frequently indirect speech acts, so they run the risk of being missed by an insensitive student.

(7) *Elaborating*. This is an assertion by the tutor that fills in missing information that the tutor regards as important, but otherwise might be missed by the student. In essence, information is simply transmitted from the tutor to the learner, as opposed to having the learner generate the information.

(8) *Splicing/correcting*. The tutor jumps in and splices correct information as soon as the student produces a contribution that is obviously error-ridden. The tutor needs to be able to recognize errors, bugs, and slips in order to do this. Deep misconceptions in the student are more difficult to detect and are not handled by splicing.

(9) *Summarizing*. Unskilled tutors normally give a summary that recaps an answer to a question or solution to a problem. This summary serves the function of succinctly codifying a lengthy, multi-turn, collaborative exchange when a question is answered or problem is solved. A skilled tutor might

encourage the student to construct the summary instead of the tutor supplying one. This would promote a more active construction of knowledge on the part of the student, an activity that is known to facilitate learning.

(10) *Requestioning*. The tutor re-asks the original main question when the thread of the tutorial dialog is moving off course or on a tangent.

We discovered that a five-step dialog frame was a pervasive discourse pattern in naturalistic tutoring (Graesser & Person, 1994; Graesser et al., 1995). The five steps in this frame are presented below, along with an example.

Step 1: Tutor asks question: 'Now, what is a factorial design?'
Step 2: Learner answers question. 'The design has two variables.'
Step 3: Tutor gives short immediate feedback on the quality of the answer: 'Uh-huh.'
Step 4: Tutor and learner collaboratively improve the quality of the answer.
Tutor: 'So there are two or more independent variables and one [pause].'
Learner: 'Dependent variable.'
Step 5: Tutor assesses learner's understanding of the answer. 'Do you see that?'
Learner: 'Uh-huh.'

This five-step dialog frame in tutoring is a significant augmentation over the three-step dialog frame in classrooms. Mehan (1979) and others have documented the pervasive IRE pattern in classrooms, which stands for Initiation (a question or claim by the teacher), Response (by the student), and Evaluation (of the student response by the teacher). We suspect that the advantage of tutoring over classroom settings might be the extra two steps in the five-step frame, but particularly step 4. Step 4 normally is a lengthy multi-turn dialog that builds the explanation that answers the question or solves the problem.

At this point it is time to shift gears and turn to our computer simulation of the unskilled human tutor. It is beyond the scope of this article to present all the details of the mechanisms of AutoTutor (Graesser et al., 1998; Wiemer-Hastings et al., 1998). Our goal is to provide a functional specification of AutoTutor's components and to selectively report some results that test how well the components perform.

3. The seven modules of AutoTutor

Educators and advocates of intelligent tutoring systems (ITS) have frequently envisioned having computers tutor students on skills and domain knowledge. The computer tutor would be fully automated in the best of worlds. Unfortunately, however, language and discourse have constituted serious barriers in these efforts. As a consequence, language and discourse facilities have been either nonexistent or extremely limited in the most impressive and successful intelligent tutoring systems available, such as Anderson's tutors for geometry, algebra, and computer languages (Anderson et al., 1995), Van Lehn's tutor for basic mathematics (Van Lehn, 1990), and Lesgold's tutor for diagnosing and repairing electronic equipment (Lesgold et al., 1992). There have been some attempts to augment ITSs with language and dialogue facilities (Holland et al., 1995; Moore, 1995). But such attempts have been limited by (1) the inherent difficulty of getting computers to 'comprehend' the language of users, including utterances that are not well formed syntactically and semantically, (2) the lack of research on human tutorial dialog and on patterns of normal discourse, and (3) the difficulty of getting computers to effectively use a large body of open-ended, fragmentary, and unstructured world knowledge. However, recent advances have provided approximate solutions to minimizing these barriers, so a computer tutor with a natural language and dialog facility is much more feasible. Some of these advances are briefly identified below.

Natural language. In the arena of natural language, the first major barrier, there have been serious advances during the 1990s. The 'message understanding' initiative, funded by DARPA, has evaluated the performance of natural language extraction systems developed in artificial intelligence and computational linguistics (DARPA, 1995; Jacobs, 1992; Lehnert, 1997). There has been noticeable progress in automating many components of language analysis that lie within the span of a sentence and short discourse segments, such as identifying the correct sense of words with multiple senses, parsing sentence syntax for sentences that are short or moderate in length, extracting important information, and constructing semantic representations (e.g., propositions, conceptual graphs). Traditional symbolic parsers have made substantial progress in handling the large volume of lexical, syntactic, and semantic knowledge that is required to understand real-world texts in a restricted semantic domain, such as news articles on terrorism (Grishman et al., 1992). Nontraditional parsers are more streamlined or adopt alternative computational architectures, such as finite state automata (Hobbs et al., 1992), heuristic fuzzy parsing algorithms, probabilistic parsers (Charniak, 1993; Jurafsky, 1996), and neural network architectures (Miikkulainen, 1996).

Tutorial dialog. The previous section reviewed the research on tutorial dialog that has accumulated during the last decade. This is in marked contrast to twenty years ago when there was virtually no research on discourse patterns in tutoring and other registers of conversation.

World knowledge. The fact that world knowledge is inextricably bound to natural language comprehension and discourse is widely acknowledged in psycholinguistics, cognitive science and discourse processing (Gernsbacher, 1994; Graesser et al., 1994; Kintsch, 1998; Schank & Reisbeck, 1982), but researchers in computational linguistics and artificial intelligence have not had a satisfactory approach to handling the deep abyss of world knowledge. The traditional approach to representing world knowledge in artificial intelligence has been structured representations, such as semantic networks and conceptual graphs (Graesser & Clark, 1985; Lehmann, 1992; Lenat, 1995). However, world knowledge is frequently open-ended, imprecise, vague, and incomplete, so simple algorithms and computational procedures cannot handle the role of world knowledge in understanding language and in tutoring. In fact, these pervasive characteristics of world knowledge motivated Collins' work on the SCHOLAR tutor (Collins, 1985), an ITS that generated plausible inferences and tutor contributions on the basis of fragments of imprecise and incomplete world knowledge.

A new statistical technique for representing world knowledge, called latent semantic analysis, has recently been introduced to the cognitive sciences. We have used latent semantic analyses (LSA) as the backbone for representing the knowledge associated with computer literacy, as will be discussed shortly. LSA has recently been proposed as a statistical representation of a large body of world knowledge (Landauer & Dumais, 1997; Landauer et al., 1998). LSA capitalizes on the fact that particular words appear in particular texts (called 'documents'); the co-occurrence of words in documents reflects the constraints that exist in world knowledge. The input to LSA is a co-occurrence matrix that specifies the number of times that word W_i occurs in document D_j. These frequencies are adjusted with a logarithm transformation that also corrects for the base rates of words appearing across documents; a word is a distinctive index for a document to the extent that its occurrence in the document is above the base rate for that word across documents. A standard statistical method, called singular value decomposition, reduces the large $W \times D$ co-occurrence matrix to K dimensions (typically, 100 to 300 dimensions). Each word, sentence, or text ends up being a weighted vector on the K dimensions. The 'match' (i.e., similarity in meaning, conceptual relatedness) between two words, sentences, or texts is computed as a geometric cosine (or dot product) between the two vectors, with values ranging from 0 to 1. The match between two language strings can be high even though there are few if any words in common between the two strings. LSA goes well beyond simple string matches because the meaning of a language string is partly determined by the company (other words) that each word keeps.

The empirical success of LSA has been promising and sometimes remarkable. Landauer and Dumais (1997) created an LSA representation with 300 dimensions from 4.6 million words that appeared in 30,473 articles in Grolier's Academic American Encyclopedia. They submitted to the LSA representation the synonym portion of the TOEFL test, a test developed by the Educational Testing Service to assess how well non-native English speakers have mastered the words in the English language. The test has a four-alternative, forced choice format, so there is a 25% chance of answering the questions correctly. The LSA model selected the alternative that had the highest match with a comparison word. The LSA model answered 64.4% of the questions correctly, which is essentially equivalent to the 64.5% performance for college students from non-English speaking countries. LSA has had remarkable success in capturing the world knowledge that is needed to grade essays of students (Foltz, 1996) and in matching texts to students of varying abilities to optimize learning (Wolfe et al., 1998). There are now LSA-based graders of essays that assign grades to essays with the validity and reliability of human experts in composition. In our own research on computer literacy, LSA has been quite successful in evaluating the quality of college students' answers to deep reasoning questions and to the contributions of learners during the tutorial interactions with AutoTutor (Graesser et al., in press; Wiemer-Hastings et al., 1999), as will be discussed later.

AutoTutor has seven modules which are briefly described in this section. The modules include a curriculum script, language extraction, speech act classification, latent semantic analysis, topic selection, dialog move generation, and a talking head. AutoTutor is a working system, but we have not yet incorporated it into the curriculum of any courses. AutoTutor runs on a Pentium computer in an NT operating system. Most of the code is written in the Java language.

3.1. Curriculum script

AutoTutor has a curriculum script that organizes the topics and content of the tutorial dialog. The script includes didactic descriptions, tutor-posed questions, example problems, figures, and diagrams (along with anticipated good responses to each topic). Each topic in the curriculum script is represented either as a structured set of propositions or as a free text format. Symbolic computational modeling requires the former whereas the LSA can accept the latter. Each of the topics in the curriculum script is scaled on difficulty (easy, medium, difficult) and has one of four different formats, as discussed below. Associated with each topic in the curriculum script is a set of good student responses (i.e., answers to questions, solutions to problems) and bad student responses that have errors, bugs and misconceptions. Human tutors normally anticipate what student responses to expect and try to steer the student in that direction; this is called *model tracing* in the ITS literature (Anderson et al., 1995). They also anticipate traps and common wrong student responses. AutoTutor starts out with a small number of good and bad student responses to each topic, which are provided by the designers of the original curriculum script. However, the corpus of good and bad responses ends up growing with tutoring experience. That is, whenever a good response or a bad student response is identified by the system in response to topic T, it adds the response to the good or bad response list for that topic (in free text format). Therefore, a corpus of good and bad responses grows with tutoring experience, as in the case of human tutors.

AutoTutor's curriculum script for computer literacy has three macrotopics: hardware, the operating system, and the Internet. These three macrotopics follow an order that parallels our computer literacy course at the University of Memphis and the associated textbook (Beekman, 1997). There are twelve topics within each macrotopic. Within each set of twelve topics, three levels of difficulty are crossed with four topic formats.

The three levels of difficulty (easy, medium, difficult) have a systematic relationship with some taxonomies of cognitive difficulty and question difficulty (Bloom, 1956; Graesser & Person, 1994). Easy questions tap lists of components and properties of components (e.g., What are the components or properties of X?). Medium questions tap causal networks, plan-driven procedures, and logical justifications (e.g., Why or how does X occur?, What if X occurs?, What are the consequences of X?). Difficult questions required the comparison, synthesis, or integration of disparate ideas, as in the case of analogical reasoning or the application of knowledge to a real world problem.

The four topic formats are: (1) Question + Answer, (2) Didactic information + Question + Answer, (3) Graphic display + Question + Answer, and (4) Problem + Solution. Each topic format includes a main, focal question that is presented to the learner. The Question + Answer format simply asks a question, with no didactic content leading up to it (e.g., Why do computers need operating systems?). The Didactic information + Question + Answer format starts out with didactic content and then asks a question related to that content. The Graphic display + Question + Answer format presents pictorial information and then asks a question that refers to the pictorial information. The Problem + Solution format presents a short problem scenario that learner is asked to solve; a question directly captures what the focal problem is.

The Answer or Solution content that is associated with each topic includes a number of data slots that specify anticipated responses. The content that students type into the keyboard will ultimately be matched to the content within these slots, and LSA is used in these pattern matching operations. For example, presented below is the question in our previous example tutorial dialog, along with a set of good answers (i.e., aspects of a complete answer) that would be expected by AutoTutor.

QUESTION: When you turn on the computer, how is the operating system first activated and loaded into RAM?
GOOD ANSWER 1: The operating system is first stored on the hard disk.
GOOD ANSWER 2: The CPU executes instructions stored in ROM when you turn on the computer.
GOOD ANSWER 3: The instructions from ROM help boot the operating system.
GOOD ANSWER 4: The operating system is loaded from the hard disk into RAM.

The set of N good answers or aspects, $\{A_1, A_2, \ldots A_N\}$, is the definition of the *ideal complete answer*. All of these aspects need to be covered in the tutorial dialog after the focal question in the topic is asked. The learner enters into the keyboard a string of information during each turn. This input is matched to each aspect and all possible combinations of aspects. When considering all 36 questions, each topic had three to nine aspects that comprised the ideal complete answer. Given that there are at most nine aspects, the upper bound of number of combinations is $[2^9-1] = 511$.

There are additional data slots associated with each topic, over and above the ideal complete answer. There is a list of anticipated bad answers, corresponding to misconceptions and bugs that need correction. When the learner's input has a high enough LSA match to one of the bad answers (B_i), AutoTutor produces a dialog move that corrects the misconception or bug (i.e., a correction is spliced in). Thus, there is a corrective splice associated with each anticipated bad answer. Associated with each good answer aspect (A_i), there are different articulation formats that correspond to three different dialog move categories: elaborations, hints, and prompts. For example, the hint format for A_2 would be 'What about ROM?' whereas the prompt format would be 'When you turn on the computer, the CPU executes instructions that are stored in ____.' The hints and prompts are designed to get the learner to contribute information, whereas the information is merely delivered by AutoTutor in the case of dialog moves that are elaborations, e.g., 'The CPU executes instructions stored in ROM when you turn on the computer.' There are other data slots within each topic, but these need not be elaborated here.

3.2. Language extraction

Language modules analyze the words in the messages that the learner types into the keyboard during a particular conversational turn. There is a large lexicon with approximately 10,000 words. Each lexical entry specifies its alternative syntactic classes; these are called 'tags' in the computational linguistics community. There are 17 alternative syntactic tags, which include such classes as noun, pronoun, verb, auxiliary verb, adjective, adverb, preposition, conjunction, article, interjection, question-words, digits, and operators. Many words may be categorized into more than one syntactic class; these end up being resolved by context. For example, the word 'access' can be a noun, adjective, or verb, but it is an adjective in the sentence 'Random access devices provide faster retrieval times.' The average number of tags per word was 1.70 in the corpus of tutoring protocols analyzed by Olde et al. (1999). Each word also has a frequency of usage in the English language, broken down by word tag, which has been estimated by Francis and Kucera (1982).

Each word that the learner enters is matched to the appropriate entry in the lexicon in order to fetch the alternative syntactic classes and word frequency values. AutoTutor is capable of segmenting the input into a sequence of words and punctuation marks with

99% + accuracy, of assigning alternative syntactic tags to words with 97% accuracy, and of assigning the correct syntactic class to a word (based on context) with 92% accuracy (Olde et al., 1999). A neural network assigns the correct syntactic class to word W, taking into consideration the syntactic classes of the preceding word (W − 1) and subsequent word (W + 1). More specifically, there is a feedforward neural network with 51 input units, 16 hidden units, and 17 output units (corresponding to the 17 part-of-speech tags). The 51 input units consisted of the 17 POS categories associated with the 3 word positions. The value of each input node varies from 0 (not a tag for the word) to 1 (always the tag for the word); the value is weighted according to frequency of usage when the word had more than one tag associated with it. The 92% accuracy of our neural network tagger is somewhat better than most alternative POS taggers. Allen (1995) states, for example, that most POS taggers can achieve an accuracy rating of 90% in well-formed printed text. Our neural network does better than that gold standard, even for the scruffy naturalistic discourse in our corpus.

3.3. Speech act classification

AutoTutor first segments the string of words and punctuation marks within a learner's turn into speech act units. Our current system relies on punctuation to perform this segmentation. That is, there is a juncture after each period, question mark, and exclamation point. An improved system would take into consideration commas, syntax, and semantics when performing this segmentation, but AutoTutor currently relies on!.?.

A neural network is used to classify each speech act into one of five speech act categories: Assertion, WH question, Yes/No question, Directive, and Short Response. AutoTutor can currently classify 89% of the speech acts correctly. The neural network has the following banks of input nodes when classifying speech act S: (1) the tutor's previous speech act category, (2) the student's previous speech act category, S − 1 (if any occurred after the tutor turn), (3) the word tags of the first three words of speech act S, and (4) the final punctuation mark (if any). The first three words of the speech act are very diagnostic in classifying the speech act. For example, WH questions normally begin with a question-word in word position 1 (e.g., what, why, how, where), whereas Yes/No questions normally begin with modal auxiliary verbs (e.g., is, are, was, were). The previous speech act by the tutor and learner capture most of the discourse context that is needed to compute the category of speech act S (Graesser et al., 1997).

The content of the learner's Assertions is used to assess the quality of learner contributions. The short feedback that AutoTutor delivers after a conversational turn (i.e., positive, negative, neutral) is sensitive to AutoTutor's evaluation of the quality of the set of Assertions within a conversational turn. Suppose there is a high LSA match between the set of Assertions within turn T and one or more of the good answer aspects associated with the topic, $\{A_1, A_2, \ldots A_N\}$. AutoTutor would then present positive immediate feedback at the beginning of its next turn. In contrast, if there is a high LSA match between the set of Assertions within turn T and one of the anticipated bad answers, B_i, AutoTutor would present some sort of negative short feedback at the beginning of its next turn, and splice in the correction. When the Assertions of turn T have modest or low matches with the good answer aspects or the bad answers, then various forms and degrees of neutral short feedback are delivered by AutoTutor.

AutoTutor has a different strategy for dealing with the remaining speech act categories: WH question, Yes/No question, Directive, and Short Response. AutoTutor attempts to answer the questions asked by the learner. Consider the case of Yes/No questions, such as 'Isn't ROM primary memory?' The proposition being queried ('ROM is primary memory') is matched, via LSA, to the good answers in the entire curriculum script, and also to the bad answers. If there is a high LSA match to a good aspect, then AutoTutor answers yes. The answer is no if there is a high match to a bad answer. Otherwise, the answer of AutoTutor is indecisive ('maybe,' 'sometimes,' 'perhaps'). In the case of WH questions, LSA is used to match the queried proposition to anticipated questions. For example, in the case of definition questions ('What does X mean?'), X is matched to the entries in a glossary and AutoTutor produces the definition if there is a high match. These methods of handling questions are needed for a smooth mixed-initiative dialog between AutoTutor and the learner. The strategies of handling Directives and Short Responses of learners are currently not developed in AutoTutor.

3.4. Latent semantic analysis (LSA)

The knowledge about computer literacy is represented by LSA (Landauer & Dumais, 1997; Landauer et al., 1998). As discussed earlier, LSA is a statistical technique that compresses a large corpus texts into a space of 100 to 500 dimensions. The *K*-dimensional space is used when evaluating the relevance or similarity between any two bags of words, *X* and *Y*. The relevance or similarity value varies from 0 to 1; in most applications of LSA, a geometric cosine is used to evaluate the match between the *K*-dimensional vector for one bag of words and the vector for the other bag of words. From the present standpoint, one bag of words is the set of Assertions within turn T. The other bag of words is the content of the curriculum script associated with a particular topic, i.e., good answer aspects and the bad answers.

The LSA space for the domain of computer literacy was derived from a corpus of texts that included the curriculum script, two books on computer literacy, and approximately 30 articles that focus on hardware, operating systems, and the Internet. An LSA analysis requires the preparation of a document by word ($D \times W$) co-occurrence matrix. Each cell in the matrix specifies the number of occurrences of word W_i in Document D_j. The computation of the *K* dimensions is then derived from a statistical procedure called singular value decomposition. In order to prepare the $D \times W$ matrix, the researcher needs to define what constitutes a document unit. A single document was defined as (1) a good answer aspect in the case of the curriculum script and (2) a paragraph in the case of the textbooks and 30 articles. When we performed the LSA on the 2.3 MB corpus of documents, the solution that we adopted had 200 dimensions.

The LSA space did an impressive job in evaluating the quality of Assertions in the domain of computer literacy (Graesser et al., in press; Wiemer-Hastings et al., 1999). Wiemer-Hastings et al. (1999) analyzed how well the LSA space on computer literacy could accurately evaluate a sample of 192 answers to the questions in the curriculum script. College students enrolled in the computer literacy course answered the questions in the curriculum script by typing their answers into a Web site facility. The data were collected after the college students had read the relevant chapters in the book and had received a lecture on each macrotopic (i.e., hardware, operating system, Internet). Four trained experts also rated the 192 answers to the questions. Two of these raters were intermediate experts. They were graduate students in cognitive science who had read the Beekman text on computer literacy and had three or more years of active experience with computer technologies. The other two raters were accomplished experts. They had a graduate degree in computer science and also read the Beekman text. The results of correlational analyses revealed that the LSA did an excellent job evaluating the quality of student answers. The correlation between LSA's answer quality scores and the mean quality scores of the four experts was 0.49. This 0.49 correlation is indistinguishable from the 0.51 correlation between the ratings of the two intermediate experts, but significantly lower than the 0.78 correlation between two accomplished experts. It appears, therefore, that the LSA space of AutoTutor exhibits the performance of an intermediate expert, but not an accomplished expert. Once again, however, most human tutors have intermediate rather than accomplished expertise, so the LSA space does an excellent job simulating the normal human tutor.

3.5. Topic selection

Topics are selected by fuzzy production rules that are sensitive to the structure of the curriculum and to the learner's ability. The learner's knowledge about the topic is induced from the exchanges during previous topics, as reflected in the quality of the Assertions. The topics within the hardware macrotopic are selected before the operating system topics, which in turn are covered before the Internet. Within each macrotopic, the difficulty level of the topic is matched to the student ability according to the zone of proximal development (Rogoff, 1990; Vygotsky, 1978; Wolfe et al., 1998). That is, the good student receives difficult topics whereas the underachieving student receives easy topics.

Student ability is based on the learner's Assertions in all of the previous learner turns in the tutorial session. An *Assertion quality score* is computed for the set of Assertions in any given learner turn, T_i, which we denote as $Q(T_i)$. Part of the computation of the quality score is based on its resonance with the ideal good answer of the current topic. This consists of the highest LSA cosine match between T_i and any particular good answer aspect (A_j) associated with the current topic, or any combination of aspects in the ideal good answer, $\{A_1, A_2, \ldots A_N\}$. However, the Assertion quality score also needs to be penalized for matches to bad answers. Therefore, a difference score provides a better metric for the Assertion quality score: {max[cosine(T_i, aspect combination)] − max[cosine(T_i, bad answer B_j)]}. The computation of *student ability* is the mean of the Assertion quality scores for all of the previous N learner turns in the tutorial session: $[\Sigma\ Q(T_i)]/N$.

3.6. Dialog move generator

After the learner types in the content of his turn, AutoTutor needs to generate dialog moves. Sometimes AutoTutor needs to answer a question asked by the learner, but this is not the most typical activity because student questions are not particularly frequent speech acts in tutoring (Graesser & Person, 1994). More often, AutoTutor responds to and builds on the Assertions of the learner as the two parties collaboratively answer questions or solve problems. AutoTutor serves as a *discourse prosthesis*, drawing out what the learner knows and scaffolding the learner to an enhanced level of mastery. AutoTutor follows the first four steps of the five-step dialogue frame that was discussed earlier.

Most of AutoTutor's turns (after the initial focal question for the topic) consist of a two-part frame: short immediate feedback followed by a more substantive dialog move. AutoTutor delivers short immediate feedback about the quality of the learner's Assertions in the preceding turn (i.e., positive, neutral, versus negative feedback). Then AutoTutor gives a lengthier substantive contribution that prompts the learner for more information, that adds information, or that corrects a student error. The more substantive contributions include pumps, prompts, hints, elaborations, splices, requestions, and summaries.

The *categories* of the substantive dialog moves during AutoTutor's turns are determined by a set of fuzzy production rules. These rules are tuned to (a) the quality of the student's Assertions in the preceding turn, as computed by LSA, (b) global parameters that refer to the ability, verbosity, and initiative of the student, and (c) the extent to which the good answer aspects of the topic had been covered. The computation of the global student parameters is straightforward. Student ability is the mean quality of the Assertions within all of the previous turns in the session, as we have already defined. Student verbosity is the mean number of words per student turn. Student initiative is the proportion of previous student speech acts that are questions or directives.

Given the values of the local and global parameters, AutoTutor selects the substantive dialog moves by a set of approximately 20 fuzzy production rules. Thus, AutoTutor adopts a traditional production rule architecture (Anderson et al., 1995; Just & Carpenter, 1992; Laird & Rosenbloom, 1996), except that the condition elements are evaluated by fuzzy matches (Kosko, 1992). For example, two production rules for producing a hint are presented below.

IF [(student ability = MEDIUM or HIGH) & ($Q(T_i)$ = LOW)]
    THEN [select HINT]
IF [(student ability = LOW) & (student verbosity = HIGH) & ($Q(T_i)$ = LOW)]
    THEN [select HINT]

The hint in the first production rule steers the good student on course when the student is off in left field. The hint in the second production rule steers a verbose poor student on the right track.

The *content* of the hint is determined by an algorithm that captures the zone of proximal development. That is, the hint tries to drag out information (specified in the curriculum script) that slightly extends the boundaries of what the student knows or what has been covered in the topic. Stated differently, AutoTutor selects the next good answer aspect to focus on. So how, more specifically, does AutoTutor select the next aspect? AutoTutor keeps track of the extent to which each aspect (A_i) has been covered as the dialog evolves for a topic. The coverage metric varies from 0 to 1 and gets updated as each Assertion is produced by the tutor or learner. LSA is used to compute the extent to which the various Assertions cover the particular aspects associated with a topic. If some threshold (t) is met or exceeded, then the aspect A_i is considered covered. Our analyses revealed that a threshold value between 0.5 and 0.7 was a reasonable value for considering an aspect covered (Wiemer-Hastings et al., 1999). AutoTutor selects, as the next aspect to cover, that aspect that has the highest subthreshold coverage score. Therefore, AutoTutor builds on the fringes of what is known. A topic is finished when all of the aspects have coverage values that meet or exceed the threshold t. This zone of proximal development is only one mechanism for determining the next good answer to cover. Other considerations involve temporal order (X precedes Y in time) and logical prerequisite (X is a prerequisite for Y).

How well does AutoTutor do in generating dialog moves? We conducted a study that evaluated the dialog moves produced by AutoTutor. We typed into AutoTutor fragments of answers that had been generated by college students in the computer literacy class; they had answered the questions and problems in the curriculum script. We had expert judges evaluate AutoTutor's dialog moves in a multi-turn exchange. The evaluations were on two six-point scales: conversational smoothness (i.e., how well does the dialog move fit in the flow of conversation?) and pedagogical quality. Ratings of 4, 5, and 6 corresponded to good and ratings of 1, 2, and 3 corresponded to bad. Table 1 shows the mean ratings for the different categories of dialog moves.

Table 1
Ratings of AutoTutor's Dialog Moves

Category	*Smoothness*	*Pedagogy*
Positive feedback	5.43	4.99
Neutral feedback	4.83	3.81
Negative feedback	3.32	2.00
Pump	5.17	4.67
Prompt	4.58	3.63
Hint	5.16	3.93
Elaboration	4.86	3.79
Splice/Correction	5.75	5.75
Summary	5.28	4.71
Overall	4.97	4.25

Overall, AutoTutor did a good job generating moves that fit in with the flow of the conversation and that are pedagogically sound. Mean ratings of 3.5 or higher are on the 'good' side of the continuum. However, there is room to improve on both of these dimensions. Apparently, it is difficult to generate negative feedback. There are problems with prompts, compared to other categories of dialog moves. We are currently in the process of tuning the dialog moves to provide better context-sensitive feedback. Ideally, these dialog moves will provide an effective, context-sensitive, discourse prosthesis that facilitates the process of learners actively constructing self-explanations.

3.7. Talking head with gestures

Researchers have recently developed computer-generated animated talking heads that have facial features synchronized with speech (Cohen & Massaro, 1994; Pelachaud et al., 1996). Ideally, the computer would control 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. A talking head is an important enhancement to AutoTutor because it concretely grounds the conversation between the tutor and learner. A talking head also offers a separate channel of cues for providing mixed feedback to the learner. When a learner's contribution is incorrect or vague, for example, the speech is often positive and polite whereas the face has a puzzled expression; this conflicting message that satisfies both pedagogical and politeness constraints would be preferable to a threatening speech message that says 'That's wrong,' or 'I'm having trouble understanding you.' The nonverbal facial cues are known to be an important form of backchannel feedback during tutoring (Fox, 1993; Graesser et al., 1995; Person et al., 1994), as well as other contexts of conversation (Clark, 1996). Similarly, pitch, pause, duration, amplitude, and intonation contours are among the variety of intonation cues that signal backchannel feedback, affect, and emphasis (Brennan & Williams, 1995; Dressler & Kreuz, in press). Some of these intonation parameters have been implemented successfully in synthesized and digitized speech (Cowley & Jones, 1992). These intonation parameters are important to tutoring because they qualify the backchannel feedback ('Uh-huh,' 'Okay') and substantive contributions of the tutor (Fox, 1993; Graesser et al., 1995). For example, the tutor frequently pauses after a vague student contribution or pounces in quickly after an obvious error-ridden student contribution.

Most of the tutor's dialog moves are delivered by a talking head that synchronizes synthesized speech, facial expressions, and gestures. Microsoft Agent is currently being used as the talking head with synthesized speech. The facial expressions and intonation in the immediate short feedback are sensitive to the quality of the Assertions in the student's most recent turn. The parameters of the facial expressions and intonation are generated by fuzzy production rules (McCauley et al., 1998).

4. Final comments

It is important to reiterate, once again, that the development of AutoTutor was an interdisciplinary effort that incorporated several different computational architectures and methodologies developed in the cognitive sciences. The primary fields included psychology, discourse processes, artificial intelligence, computational linguistics, and statistics. The major computational architectures and models included structured script representations, neural networks, fuzzy production systems, and latent semantic analysis. The repertoire of methodologies included naturalistic observation, corpus analysis, computer simulation, correlational statistics, and true experiments. AutoTutor never would have been developed by an isolated researcher who settled within the provinces of a single field, model, or methodology.

Acknowledgements

This research was funded by the National Science Foundation (SBR 9720314) in a grant awarded to the first author of this manuscript. The following members of the Tutoring Research Group at the University of Memphis conducted research on this project: Ashraf Anwar, Myles Bogner, Tim Brogdon, Patrick Chipman, Scotty Craig, Rachel DiPaolo, Stan Franklin, Max Garzon, Barry Gholson, Art Graesser, Doug Hacker, Peggy Halde, Derek Harter, Jim Hoeffner, Xiangen Hu, Jeff Janover, Bianca Klettke, Roger Kreuz, Kristen Link, William Marks, Johanna Marineau, Lee McCaulley, Brent Olde, Natalie Person, Victoria Pomeroy, Mat Weeks, Shannon Whitten, Katja Wiemer-Hastings, Peter Wiemer-Hastings, Shonijie Yang, Holly Yetman, and Zhaohua Zhang.
Requests for reprints should be sent to Art Graesser, Department of Psychology, Campus Box 526400, University of Memphis, Memphis, TN, 38152-6400, USA, a-graesser@memphis.edu.

References

Allen, J. (1995). Natural language understanding, Benjamin/Cummings, Redwook City, CA.

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

Beekman, G. (1997). Computer confluence, Benjamin/Cummings, New York.

Berlyne, D. E. (1960). Conflict, arousal, and curiosity, McGraw-Hill, New York.

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

Bloom, B. S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher 13*, 4–16.

Bransford, J. D., Goldman, S. R., & Vye, N. J. (1991). Making a difference in people's ability to think: Reflections on a decade of work and some hopes for the future. In: Sternberg, R. J., & Okagaki, L. (Eds.), Influences on children, Erlbaum, Hillsdale, NJ, pp. 147–180.

Brennan, S. E., & Williams, M. (1995). The feeling of another's knowing: Prosody and filled pauses as cues to listeners about the metacognitive states of speakers. *Journal of Memory and Language 34*, 383–398.

Charniak, E. (1993). Statistical language analysis, Cambridge University Press, Cambridge, MA.

Chi, M. T., Bassok, M., Lewis, M. W., Reimann, P., & Glaser, R. (1989). Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science 13*, 145–182.

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

Clark, H. H. (1996). Using language, Cambridge University Press, Cambridge, UK.

Cohen, M. M., & Massaro, D. W. (1994). Development and experimentation with synthetic visible speech. *Behavior Research Methods, Instruments, and Computers 26*, 260–265.

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.

Collins, A. (1985). Teaching reasoning skills. In: Chipman, S. F., Segal, J. W., & Glaser, R. (Eds.), Thinking and learning skills, Vol. 2, Erlbaum, Hillsdale, NJ, pp. 579–586.

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

Cowley, C. K., & Jones, D. M. (1992). Synthesized or digitized? A guide to the use of computer speech. *Applied Ergonomics 23*, 172–176.

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

Dillon, T. J. (1988). Questioning and teaching: A manual of practice, Teachers College Press, New York.

Dressler, R.A., & Kreuz, R.J. (in press). Transcribing oral discourse: A survey and a model system. *Discourse Processes*.

Edelson, D. C. (1996). Learning from cases and questions: The Socratic case-based teaching architecture. *Journal of the Learning Sciences 5*, 357–409.

Festinger, T. J. (1957). A theory of cognitive dissonance, Row and Peterson, Evanston, IL.

Foltz, P. W. (1996). Latent semantic analysis for text-based research. *Behavior Research Methods, Instruments, and Computers 28*, 197–202.

Forbus, K., Gentner, D., & Law, K. (1995). MAC/FAC: A model of similarity-based retrieval. *Cognitive Science 19*, 141–205.

Fox, B. (1993). The human tutorial dialog project, Erlbaum, Hillsdale, NJ.

Francis, W. N., & Kucera, N. (1982). Frequency analysis of English usage, Houghton-Mifflin, Hillsdale, NJ.

Gagné, R. M. (1977). The conditions of learning, 3rd ed., Holdt, Rinehart, & Winston, New York.

Gernsbacher, M.A. (Ed.) (1994). *Handbook of Psycholinguistics* (pp. 517–538). San Diego, CA: Academic Press.

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

Graesser, A. C., Franklin, S., Wiemer-Hastings, P., & the, Tutoring Research Group (1998). In: Simulating smooth tutorial dialog with pedagogical value. Proceedings of the American Association for Artificial Intelligence, AAAI Press, Menlo Park, CA, pp. 163–167.

Graesser, A. C., & McMahen, C. L. (1993). Anomalous information triggers questions when adults solve problems and comprehend stories. *Journal of Educational Psychology 85*, 136–151.

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

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

Graesser, A. C., Singer, M., & Trabasso, T. (1994). Constructing inferences during narrative text comprehension. *Psychological Review 101*, 371–395.

Graesser, A. C., Swamer, S., & Hu, X. (1997). Quantitative discourse psychology. *Discourse Processes 23*, 229–264.

Graesser, A.C., Wiemer-Hastings, P., Wiemer-Hastings, K., Harter, D., Person, N., & the Tutoring Research Group (in press). Using latent semantic analysis to evaluate the contributions of students in AutoTutor. *Interactive Learning Environments*.

Greeno, J. G., Smith, D. R., & Moore, U. L. (1993). Transfer of situated learning. In: Detterman, D. K., & Sternberg, R. J. (Eds.), Transfer on trial: Intelligence, cognition, and instruction, Ablex, Norwood, NJ, pp. 99–167.

Grishman, R., Macleod, C., & Sterling, J. (1992). In: Description of the PROTEUS system as used for MUC-4. Proceedings of the Fourth Message Understanding Conference (MUC-4), Morgan Kaufmann, San Mateo, CA, pp. 233–241.

Hammond, K. J., Seifert, C. M., & Gray, K. C. (1991). Functionality in analogical transfer: A hard match is good to find. *Journal of the Learning Sciences 1*, 111–152.

Hobbs, J., Appelt, D., Tyson, M., Bear, J., & Israel, D. (1992). In: SRI International: Description of the FASTUS System used for MUC-4. Proceedings of the Fourth Message Understanding Conference (MUC-4), Morgan Kaufmann, San Mateo, CA, pp. 268–275.

Holland, V.M., Kaplan, J.D., & Sams, M.R. (Eds.) (1995). *Intelligent language tutors*. Mahwah, NJ: Erlbaum.

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.

Jacobs, P.S. (Ed.) (1992). *Text-based intelligent systems: Current research and practice in information extraction and retrieval*. Hillsdale, NJ: Erlbaum.

Jurafsky, D. (1996). A probabilistic model of lexical semantic access and disambiguation. *Cognitive Science 20*, 177–194.

Just, M. A., & Carpenter, P. A. (1992). A capacity theory of comprehension: Individual differences in working memory. *Psychological Review 99*, 122–149.

King, A. (1994). Guiding knowledge construction in the classroom: Effects of teaching children how to question and how to explain. *American Educational Research Journal 31*, 338–368.

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

Kosko, B. (1992). Neural networks and fuzzy systems, Prentice Hall, New York.

Laird, J., & Rosenbloom, P. (1996). The evolution of the Soar cognitive architecture. In: Steier, D. M., & Mitchell, T. M. (Eds.), Mind Matters: A tribute to Allen Newell, Erlbaum, Mahwah, NJ, pp. 1–50.

Landauer, T. K., & Dumais, S. T. (1997). A solution 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.

Lehmann, F. (Ed.) (1992). *Semantic networks in artificial intelligence*. New York: Pergamon.

Lehnert, W. (1997). Information extraction: What have we learned? *Discourse Processes 23*, 441–470.

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

Lepper, M. R., Woolverton, M., Mumme, D. L., & Gurtner, J. L. (1991). Motivational techniques of expert human tutors: Lessons for the design of computer-based tutors. In: Lajoie, S. P., & Derry, S. J. (Eds.), Computers as cognitive tools, Erbaum, Hillsdale, NJ, pp. 75–105.

Lesgold, A., Lajoie, S., Bunzo, M., & Eggan, G. (1992). SHERLOCK: A coached practice environment for an electronics troubleshooting job. In: Larkin, J. H., & Chabay, R. W. (Eds.), Computer-assisted instruction and intelligent tutoring systems, Erlbaum, Hillsdale, NJ, pp. 201–238.

McArthur, D., Stasz, C., & Zmuidzinas, M. (1990). Tutoring techniques in algebra. *Cognition and Instruction 7*, 197–244.

McDaniel, M. A., & Donnelly, C. M. (1996). Learning with analogy and elaborative interrogation. *Journal of Educational Psychology 88*, 508–519.

McCauley, L., Gholson, B., Hu, X., Graesser, A.C., & the Tutoring Research Group (1998). Delivering smooth tutorial dialog using a talking head. *Proceedings of the Workshop on Embodied Conversation Characters* (pp. 31–38). Tahoe City, CA: AAAI and ACM.

Mehan, H. (1979). Learning lessons: Social organization in the classroom, Harvard University Press, Cambridge, MA.

Merrill, D. C., Reiser, B. J., Ranney, M., & Trafton, J. G. (1992). Effective tutoring techniques: A comparison of human tutors and intelligent tutoring systems. *Journal of the Learning Sciences 2*, 277–305.

Miikkulainen, R. (1996). Subsymbolic case-role analysis of sentences with embedded clauses. *Cognitive Science 20*, 47–74.

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

Nathan, M. J., Kintsch, W., & Young, E. (1992). A theory of word algebra problem comprehension and its implications for the design of learning environments. *Cognition & Instruction 9*, 329–389.

Newman, D. (1989). Is a student model necessary? Apprenticeship as a model for ITS. In: Bierman, D., Breuker, J., & Sandberg, J. (Eds.), Artificial intelligence and education, IOS Press, Amsterdam.

Ohlsson, S. (1986). Some principles of intelligent tutoring. *Instructional Science 14*, 293–326.

Olde, B.A., Hoeffner, J., Chipman, P., Graesser, A.C., & the Tutoring Research Group (1999). A connectionist model for part of speech tagging. *Proceedings of the American Association for Artificial Intelligence* (pp. 172–176). Menlo Park, CA: AAAI Press.

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

Pelachaud, C., Badler, N. I., & Steedman, M. (1996). Generating facial expressions for speech. *Cognitive Science 20*, 1–46.

Person, N. K, & Graesser, A. C. (1999). Evolution of discourse in cross-age tutoring. In: O'Donnell, A. M., & King, A. (Eds.), Cognitive perspectives on peer learning, Erlbaum, Mahwah, NJ, pp. 69–86.

Person, N. K., Graesser, A. C., Magliano, J. P., & Kreuz, R. J. (1994). Inferring what the student knows in one-to-one tutoring: The role of student questions and answers. *Learning and Individual Differences 6*, 205–229.

Person, N. K., Kreuz, R. J., Zwaan, R., & Graesser, A. C. (1995). Pragmatics and pedagogy: Conversational rules and politeness strategies may inhibit effective tutoring. *Cognition and Instruction 13*, 161–188.

Pressley, M., Wood, E., Woloshyn, V. E., Martin, V., King, A., & Menk, D. (1992). Encouraging mindful use of prior knowledge: Attempting to construct explanatory answers facilitates learning. *Educational Psychologist 27*, 91–110.

Putnam, R. T. (1987). Structuring and adjusting content for students: A study of live and simulated tutoring of addition. *American Educational Research Journal 24*, 13–48.

Rogoff, B. (1990). Apprenticeship in thinking, Oxford University Press, New York.

Roschelle, J. (1992). Learning by collaboration: Convergent conceptual change. *Journal of the Learning Sciences 2*, 235–276.

Schank, R. C. (1986). Explanation patterns: Understanding mechanically and creatively, Erlbaum, Hillsdale, NJ.

Schank, R., & Reisbeck, C. (Eds.) (1982). Inside computer understanding. Hillsdale, NJ: Erlbaum.

Schober, M. F. (1995). Speakers, addressees, and frames of reference: Whose effort is minimized in conversations about locations? *Discourse Processes 20*, 219–247.

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science 12*, 257–285.

Van Lehn, K. (1990). Mind bugs: The origins of procedural misconceptions, MIT Press, Cambridge, MA.

Vygotsky, L. S. (1978). Mind in society, Harvard University Press, Cambridge, MA.

Webb, N. M., Troper, J. D., & Fall, R. (1995). Constructive activity and learning in collaborative small groups. *Journal of Educational Psychology 87*, 406–423.

Wiemer-Hastings, P., Graesser, A.C., Harter, D., & the Tutoring Research Group (1998). The foundations and architecture of AutoTutor. *Proceedings of the 4th International Conference on Intelligent Tutoring Systems* (pp. 334–343). Berlin: Springer.

Wiemer-Hastings, P., Wiemer-Hastings, K., & Graesser, A. (1999). In: Improving an intelligent tutor's comprehension of students with Latent Semantic Analysis. Artificial Intelligence in Education, IOS Press, Amsterdam, pp. 535–542.

Wolfe, M. B. W., Schreiner, M. E., Rehder, B., Laham, D., Foltz, P. W., Kintsch, W., & Landauer, T. K. (1998). Learning from text: Matching readers and texts by latent semantic analysis. *Discourse Processes 25*, 309–336.