

**Design of Dialog-Based Intelligent Tutoring Systems to
Simulate Human-to-Human Tutoring**

Sidney D'Mello¹ and Art Graesser²

¹University of Notre Dame

²University of Memphis

Author Note

This research was supported by the National Science Foundation (NSF) (ITR 0325428, HCC 0834847, DRL 1235958) and Institute of Education Sciences (R305B070349). Any opinions, findings and conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of NSF or IES.

Corresponding author:

Sidney D'Mello

384 Fitzpatrick, University of Notre Dame, Notre Dame, IN 46556, USA

Phone: 901-378-0531, Fax: 574-631-8883, Email: sdmello@nd.edu

Design of Dialog-Based Intelligent Tutoring Systems to Simulate Human-to Human Tutoring

Abstract

This chapter provides an overview of dialog-based intelligent tutoring systems (ITSs), which are learning technologies that help learners develop mastery of difficult subject matter by holding conversations in natural language. The first section discusses some of the basic issues in the design of dialog-based ITSs, while the second section highlights recent advances in this area. The first section begins with an analysis of human-human tutorial dialogs followed with a discussion of the six major components of most dialog-based ITSs: input transformation, speech-act classification, learner modeling, dialog management, output rendering, and domain modeling. These abstract components are concretized within the context of one of the first dialog-based ITSs, AutoTutor. The second section discusses recent advances in the area with an emphasis on systems that model learners' emotional states in addition to their cognitive states. These include a system that automatically adapts its dialogs based on whether the learner is bored, confused, or frustrated, a system with unique mechanisms to monitor and correct learners' disengagement behaviors by tracking eye gaze, and a system that strategically plants confusion in the minds of learners to engender deeper modes of thinking. We conclude the chapter by discussing some of the open issues in dialog-based ITSs, such as identifying benefits of spoken versus typed input, understanding when imperfect natural-language understanding is sufficient, contrasting the importance of the message vs. the medium in influencing learning, and identifying conditions in which dialog-based tutoring are effective.

1. Introduction

There is somewhat of a paradox in the information age. Hardware is light, cheap, and portable while software is efficient, adaptive, and increasingly intelligent. Books are online, breaking news is streamed on our handheld devices, and billions of disparate units of information are integrated, indexed, and easily accessible with a click of a mouse. The computer is now a virtual extension of our human brains. Yet our computer interfaces are still primitive, non-intuitive, difficult to use, and socially challenged. Humans communicate with computers through windows, icons, menus, and pointing devices (i.e. the WIMP paradigm). But humans communicate with each other primarily through speech and a host of non-verbal cues such as facial expressions, paralinguistic features of speech, oculesics, posture, and gesture. Computer systems that are able to recognize and respond to these communication channels will presumably

provide more effective, meaningful, and natural interaction experiences. The ability to communicate with the computer through natural-language dialogs represents a significant advancement towards narrowing the communicative bandwidth between the human and the computer. This was once no more than fanciful science fiction (e.g., HAL 9000), but the recent successes of IBM's Jeopardy-playing AI, Watson, and Apple's Speech Interpretation and Recognition Interface (SIRI), have reignited the imagination about effective, natural-language communicative interfaces.

One class of technologies that would benefit from natural-language interactions is intelligent tutoring. Intelligent tutoring systems (ITSs) mimic one-on-one human tutoring, a proven method for promoting active construction of knowledge beyond textbooks and traditional classroom environments (Graesser, D'Mello, & Cade, 2009; VanLehn, 2011; Woolf, 2009). ITSs have implemented several systematic strategies for increasing learning gains, such as error identification and correction, building on prerequisites, frontier learning (expanding on what the learner already knows), learner modeling (inferring what the learner knows and having that information guide tutoring), and building coherent explanations (Gertner & VanLehn, 2000; Graesser, Conley, & Olney, 2012; Sleeman & Brown, 1982; VanLehn, 2006; Woolf, 2009).

ITSs are effective in promoting learning because they are continually assessing learner knowledge and tailoring the instruction in a manner that is sensitive to individual learners. VanLehn (2011) recently compared the effectiveness of answer-based systems, step-based ITSs, and human tutoring for promoting learning beyond instructional controls that teach equivalent content without tutoring. Answer-based systems (e.g., most Computer Assisted Instruction (CAI), Computer-Based Training (CBT), and Computer Assisted Learning (CAL) systems) typically pose problems and evaluate learners' final answers to those problems; the learner does all of the reasoning. The performance of answer-based systems was disappointing:

0.31 sigma mean effect vs. the human-tutoring mean effect of 0.79 sigma. Step-based ITSs provide learners multiple reasoning opportunities either through tutorial dialog or through prompt-based interactions as learners work through problems. The mean effect for step-based ITSs was an impressive 0.76 sigma (Cohen's *d*) which rivaled the human-tutoring effect of 0.79 sigma.

This chapter discusses a subset of ITSs that implement natural-language dialogs that mimic conversations found in human tutoring. These include AutoTutor (Graesser, Lu, Jackson, Mitchell, Ventura, Olney, & Louwerse, 2004a; VanLehn, Graesser, Jackson, Jordan, Olney, & Rose, 2007), ITSPROKE (Litman, Rose, Forbes-Riley, VanLehn, Bhembe, & Silliman, 2006), why-Atlas (Graesser, VanLehn, Rose, Jordan, & Harter, 2001; VanLehn, Jordan, Rose, Bhembe, Bottner, & A., 2002), CIRCSIM-Tutor (Evens, Chang, Lee, Shim, Woo, Zhang, Michael, & Rovick, 1997), DC-Trains (Pon-Barry, Clark, Schultz, Bratt, & Peters, 2004), My Science Tutor (Ward, Cole, Bolaños, Buchenroth-Martin, Svirsky, Vuuren, Weston, Zheng, & Becker, 2011), Research Methods Tutor (Arnott, Hastings, & Allbritton, 2008), and Mission Rehearsal (Gratch, Rickel, André, Cassell, Petajan, & Badler, 2002). These computer tutors vary in the extent to which they simulate human dialog mechanisms, but all of them attempt to comprehend natural language, formulate adaptive responses, and implement pedagogical strategies to facilitate learning.

The effectiveness of dialog-based ITSs can be understood from the broad perspective of constructivism, which is a theoretical framework adopted by many researchers who are exploring ways to increase learning at deeper levels of comprehension (Biggs, 1996; Bransford, Goldman, & Vye, 1991; Chi, Deleeuw, Chiu, & Lavancher, 1994; Piaget, 1952; Vygotsky, 1978).

According to many of these constructivist approaches, the learner needs to actively construct coherent, explanation-based meanings and knowledge by interacting with the world and other

people. Learning environments should stimulate active construction of knowledge and provide feedback and explanations on these constructions rather than being mere information delivery systems. Dialog-based ITSs that adhere to constructivist principles attempt to get learners to do most of the talking by pumping for information and by providing hints, prompts, forced choices, and other pedagogical scaffolds. The onus of knowledge construction is placed on the learner and involves cognitive processes, such as perception, management of working memory, planning, the production of language and discourse constituents, and the consolidation of subject-matter knowledge. Dialog-based ITSs are thought to be effective because they help scaffold the co-construction of knowledge via mixed-initiative, natural-language dialogs.

The present chapter describes how dialog-based ITSs leverage recent advances in natural-language processing (NLP) to effectively simulate human-human tutorial dialogs. It is beyond the scope of this chapter to delve into the specifics of the large number of successful dialog-based ITSs that have been implemented and tested (Arnott et al., 2008; Evens et al., 1997; Graesser et al., 2001; Gratch et al., 2002; Litman et al., 2006; Pon-Barry et al., 2004; VanLehn et al., 2002; Ward et al., 2011). Instead, we focus on some of the general principles of dialog-based ITSs and anchor our discussions to specific dialog-based ITSs that that we have been developing over the last 15 years.

The remainder of the chapter is organized as follows: We begin with an analysis of human-human tutorial dialogs with an emphasis on curriculum scripts, the five-step tutoring frame, expectation and misconception-tailored dialogs, and conversational-turn management. We then outline six major components that are common to most dialog-based ITSs. These include input transformation, speech-act classification, learner modeling, dialog management, output rendering, and domain modeling. Next, we flesh out these components within the context of AutoTutor, which is one of the first dialog-based ITSs we developed (Graesser, Wiemer-

Hastings, Wiemer-Hastings, & Kreuz, 1999). We also provide a synthesis of several studies that have evaluated the ability of AutoTutor to increase learning by appropriately modeling learner knowledge, dynamically tailoring the interaction to individual learners' knowledge and abilities, and generating natural-language dialogs that are conversationally smooth.

We move beyond the basic AutoTutor system to novel dialog systems that model and respond to emotions and disengagement behaviors in addition to knowledge levels. These include systems that (a) respond to learners' emotional states and synthesize emotions, (b) sense emotions by analyzing textual cues and other discourse features, (c) detect and respond to disengagement behaviors, and (d) induce cognitive disequilibrium in the minds of learners. We conclude by discussing some of the open issues in dialog-based ITSs such as identifying benefits of spoken versus typed input, understanding when imperfect natural-language understanding is sufficient to produce meaningful interactions, contrasting the importance of the content of the dialogs versus the communicative medium for learning, and identifying some of the conditions under which differing forms of dialog-based tutoring have proven to be effective.

2. Structure of Human-Human Tutorial Dialogs

Graesser and Person analyzed dialog patterns of typical human tutors in middle school and in college (Graesser & Person, 1994; Graesser, Person, & Magliano, 1995). Similar analyses have been conducted by other researchers on natural tutoring corpora (Chi, Siler, & Jeong, 2004; D'Mello, Olney, & Person, 2010c; VanLehn, Siler, Murray, Yamauchi, & Baggett, 2003). The following dialog structures are prominent in human-human tutoring and are implemented in many dialog-based ITSs: (a) a curriculum script with didactic content and questions, (b) a 5-step tutoring frame, (c) expectation and misconception tailored (EMT) dialog, and (d) conversational turn management.

2.1. Curriculum Script

The tutor covers a curriculum with didactic content and a set of questions or problems that address the content. Didactic content can be presented in a mini-lecture, ideally at the appropriate time for each individual learner (D'Mello et al., 2010c). The questions/problems require learners to actively apply their knowledge. The curriculum script includes expected answers, misconceptions, hints, prompt questions, and other information.

2.2. 5-Step Tutoring Frame

When a challenging main question (or problem) is selected to work on, the question is answered through an interaction that is structured by a 5-step tutoring frame. The 5 steps are:

1. The tutor presents a main question.
2. The learner gives an initial answer.
3. The tutor gives short feedback on the quality of the learner's initial answer.
4. The tutor and learner collaboratively improve on the answer in a turn-by-turn dialog that may be lengthy (100-200 turns).
5. The tutor evaluates whether the learner understands (e.g., asking "Do you understand?" or testing with a follow-up task).

In the spirit of constructivism, this 5-step tutoring frame involves collaborative discussion, joint action, and encouragement for the learner to construct knowledge rather than merely receiving knowledge.

2.3. Expectation and Misconception Tailored (EMT) Dialog

Human tutors typically have a list of *expectations* (i.e. anticipated good answers or steps in a procedure) and a list of anticipated *misconceptions* (incorrect information) associated with each main question. The expectation content needs to be covered (i.e., articulated over the course of the dialog) in order address the main question that is selected. The tutor guides the learner in

articulating the expectation through a number of dialog moves. They include *pumps* (“What else?”), *hints* (“What about X?”), *prompt questions* to extract specific information from learners (“X is a type of what?”), *assertions* that capture particular expectations (“X is a type of Y”), *answers* to learners’ questions, and *summaries* that synthesize content.

2.4. Conversational Turn Management

Human tutors structure their conversational turns systematically. Nearly every turn of the tutor has three information slots. The first slot is feedback on the quality of the learner’s last turn. This feedback is either positive (e.g., “very good”, “yeah”), neutral (e.g., “uh huh”, “I see”), or negative (e.g., “not quite”, “not really”). The second slot advances the interaction with a prompt for specific information, a hint, an assertion with correct information, a correction of misconceptions, or an answer to the learner’s question. The third slot is a cue for the floor to shift from the tutor as the speaker to the learner.

3. Major Components of Dialog-Based ITSs

The behavior of most ITSs can be characterized by an outer and an inner loop (VanLehn, 2006). The outer loop consists of a series of didactic lessons and challenging problems related to the topic being tutored. For example, a tutor of cell biology might cover cell structures, the cell cycle, mitosis, and cytokinesis. The order of topics can be dynamically selected based on the profile of learner abilities or can be rigidly scripted on the basis of a content analysis that attempts to identify dependencies among topics. The inner loop consists of collaborative tutorial interactions geared towards covering steps or pieces of the content of one topic in the outer loop. It is through this inner loop that the tutor draws out more of the learner’s knowledge with probing questions, helps fill in missing information with explanations, repairs misconceptions, and answers the learner’s questions. The interactive dialog occurs during the co-construction of

knowledge in this inner loop but not during the didactic delivery of information (e.g., reading text, viewing a diagram).

We now turn to the six major components that an ITS needs in order to implement the dialog structure human tutors. These include mechanisms to: (1) transform learner responses, (2) classify responses into speech acts, (3) model learner knowledge, (4) manage the dialog, (5) render the tutor's actions, and (6) represent the domain knowledge needed by the aforementioned five mechanisms. Figure 1 provides a sketch of how these components might interact in a dialog-based ITS. Of course, the exact nature of the ITS will govern how information passes through these components.

3.1. Input Transformation

This component transforms learners' responses into a form that can be interpreted by the ITS. The input transformation component for ITSs that only support typed input is quite minimal and might involve converting the responses to lower case, correcting misspellings, and converting words to root forms (i.e., stemming). ITSs that process spoken input (D'Mello, King, & Graesser, 2010a; Litman et al., 2006) need an additional step of translating learner speech into text via automatic speech recognition.

3.2. Speech Act Classification

In most cases, learners' responses primarily consist of answers to questions posed by the tutor. These responses (called *contributions*) have substantive content potentially relevant to the answer and therefore are analyzed in the subsequent learner modeling phase. However, learners occasionally provide content-free *frozen expressions* that signify particular discourse functions. These might include *conversational acknowledgements* (e.g., "ok"), short responses (e.g., "yes", "no"), *metacognitive statements* (e.g., "I need help", "I don't know"), *metacommunicative statements* (e.g., "please repeat", "could you say that in another way"), and *gripes* (e.g., "ugh").

In addition to these frozen expressions, learners sometimes take initiative by asking questions. The Graesser and Person (1994) taxonomy identifies 16 question categories that occur in educational settings. Examples include *verification questions* that invite yes or no responses (e.g., “is velocity the same as speed?”), *definition questions* (e.g., “What does momentum mean?”), *comparison questions* (e.g., “how is mass similar to weight?”), and *causal antecedent questions* (e.g., “why did the pumpkin land on the runner’s head?”).

Frozen expressions and questions are exceptions to the basic dialog cycle which mainly consists of tutors asking questions, learners providing contributions, and tutors providing feedback. Both frozen expressions and questions need to be handled in separate sub-dialogs that are related to, but break away from, the main dialog. Hence, most dialog-based ITSs need some form of Speech Act Classifier (SAC) in order to differentiate the various types of learner responses. At a coarse-grained level, the SAC needs to distinguish the more common contributions from frozen expressions and questions. At a more fine-grained level, the SAC also needs to classify the different types of frozen expressions and questions.

3.3. Learner Modeling

Learner modeling is one of the most significant components of any ITS because, by definition, an ITS must tailor its feedback and instruction to individual learners (Psocka, Massey, & Mutter, 1988; Sleeman & Brown, 1982; Woolf, 2009). A dialog-based ITS generally needs to model learners at two levels of granularity. Modeling at a *local level* involves comparing learners’ *immediate* contribution to the expected answer (i.e., a specific keyword or phrase). This information is used to provide feedback and make local dialog decisions, such as whether to follow a hint with a prompt or simply assert the information. Learner modeling at the *global level* consists of assessing the learners’ evolution of knowledge of the topic being tutored. This is needed to identify when a particular expectation is covered, which expectations are remaining to

be covered, and which expectation should be covered next. Modeling at both levels might also involve comparing the extent to which learners' responses align with specific misconceptions, so that these might be immediately or eventually corrected. Many ITSs in well-defined domains like mathematics use some form of Bayesian knowledge tracing (Conati, Gertner, & VanLehn, 2002; Corbett & Anderson, 1994) for learner modeling; the task is considerably more difficult for dialog-based ITSs in ill-defined domains. We have addressed the problem of modeling learner knowledge in certain ill-defined domains with semantic processing algorithms as will be subsequently discussed.

3.4. Dialog Management

Effective pedagogy and smooth conversation require dialog management. The dialog manager needs to balance the competing goals of being flexible, opportunistic, goal-oriented, and conversationally smooth. Flexibility is needed in order to accommodate virtually any input of the learner without having the conversation break down. ITSs also need to be opportunistic by capitalizing on learning opportunities as they arise. For example, learners rarely take the initiative by asking questions, yet some types of questions are strongly correlated with deeper levels of cognition per the Bloom's taxonomy of educational objectives in the cognitive domain (Bloom, 1956). Similarly, misconceptions and other impasses provide some of the most promising learning opportunities because they force learners to stop to think (D'Mello, Lehman, Pekrun, & Graesser, in press; VanLehn et al., 2003). An effective dialog manager will take advantage of these learning opportunities by dynamically launching question-answering and misconception-correction sub-dialogs. However, in contrast to being opportunistic, the dialog manager must also ensure that existing dialog plans are completed without too many interruptions, otherwise the dialog will appear fragmented. Hence, a dialog manager needs to decide when it is appropriate to be more goal-oriented versus opportunistic, and it needs

mechanisms to temporarily shelve , but eventually return to, existing plans when sub-dialogs that take advantage of learning opportunities are launched.

Finally, a dialog manager also needs to emulate the pragmatics of human-human tutorial dialogs. A collaborative exchange between a dialog-based ITS and the learner requires a mutual understanding of the turn-taking process. In human-to-human conversations, speakers signal to listeners that they are relinquishing the floor and that it is the listener's turn to say something (Clark, 1996; Sacks, Schegloff, & Jefferson, 1974). However, human-to-computer conversations lack many of the subtle signals inherent to human conversations, so there need to be discourse markers and other cues that guide the learner in the exchange (Freedman, 1996).

3.5. Output Rendering

This component is responsible for communicating the tutor's actions to the learners. In the simplest case, this can involve displaying the content of the tutor's utterance on the screen. An alternate possibility is to use computer-generated animated agents that have facial features synchronized with speech and, in some cases, appropriate gestures. Learning environments with animated conversational agents have become more popular in recent years and it has well documented that such systems improve learning (Baylor & Kim, 2005; Johnson, Rickel, & Lester, 2000; Moreno & Mayer, 2007). With this approach, the computer controls the eyes, eyebrows, mouth, lips, and other parts of the face in a fashion that is appropriately meshed with the content and intent of the speaker. Animated pedagogical agents can be an important feature of dialog-based ITSs because they help ground the conversation between the tutor and learner. The nonverbal facial cues are known to be an important form of backchannel feedback during tutoring (Fox, 1993), as well as in other contexts of conversation (Clark, 1996). Similarly, pitch, pause, duration, amplitude, and intonation contours are among the paralinguistic cues that signal backchannel feedback, affect, and emphasis (Brennan & Williams, 1995). Gestures are

particularly effective at directing attention, simulating chains of cause and effect, and playing other crucial roles in communication (Alibali & Nathan, 2012; Goldin-Meadow, 2003).

3.6. Domain Model

The five components specified above describe an architecture that is common to many dialog-based ITSs. However, the architecture needs various forms of content to produce behavior. Pronunciation and language models are needed to recognize spoken speech. Templates are required for speech-act classification. Semantic models are needed for learner modeling. A curriculum script with didactic descriptions, tutor-posed questions, example problems, figures, and diagrams (along with anticipated good responses to each topic) is also needed. There should also be a glossary of technical terms with definitions in order to answer learner questions (particularly “What does X mean?” questions). Finally, different animations that synchronize speech, facial expressions, and gestures are needed when animated pedagogical agents are used to model the tutor.

4. Fleshing out the Major Components of Dialog-based ITSs in AutoTutor

We have discussed some of the dominant dialog patterns in human-human tutoring and have provided a high-level sketch of how these might be implemented in a computer tutor. We now provide a more concrete sketch of these components by using AutoTutor as an example of a fully functional system. AutoTutor is a dialog-based ITS that simulates a human tutor by holding a conversation with learners in natural language (Graesser, Chipman, Haynes, & Olney, 2005a; Graesser et al., 2004a). AutoTutor has been implemented and tested for the domains of Newtonian physics, various aspects of computer literacy, and research methods. An excerpt of an exchange between AutoTutor and a learner is shown in Table 1.

As with most ITSs, AutoTutor’s behavior can be characterized by an outer and an inner loop (VanLehn, 2006). The outer loop of AutoTutor consists of a series of didactic lessons and

challenging problems or *main questions* (such as *why, how, what-if*). An example main question is “When you turn on the computer, how is the operating system first activated and loaded into RAM?” An ideal answer to a main question requires several sentences of information. After the learner enters an initial response, AutoTutor assists the learner in constructing a more accurate/better answer.

The inner loop of AutoTutor consists of this kind of collaborative interaction for answering a main question (or solving a problem). It is this inner loop that is the distinctive hallmark of AutoTutor. The tutor draws out more of the learner’s knowledge (through hints and prompts), helps fill in missing information, repairs misconceptions, and answers learner questions. The inner loop dialog between AutoTutor and the learner takes approximately 100 turns to answer a single challenging question, approximately the length of a conversation with a human tutor (Graesser et al., 1995).

4.1. Input Transformation

In most versions of AutoTutor, learners type their responses using a keyboard. However, we have developed a version that handles spoken input from the learner with the Dragon Naturally Speaking™ (version 6) speech recognition system (D'Mello et al., 2010a).

4.2. Speech Act Classification

AutoTutor’s Speech-Act Classifier performs a two-step analysis of learners’ responses. Each speech act is assigned to one of 20 speech-act categories: contributions, 16 categories of questions, short responses, meta-cognitive expressions, and meta-communicative expressions. The classifier does a part of speech tagging on the utterance followed by a cascade of finite-state transducers that operate on the tagged text (Olney, Louwerse, Mathews, Marineau, Hite-Mitchell, & Graesser, 2003). Each transducer defines a particular speech-act category. For example, the finite-state transducer to recognize a concept-completion question (e.g. “When is

the Operating System loaded?) would resemble “^(Who|What|When|Where)”. The transducers rely heavily on the output of the part of speech tagger, which in turn, relies on the syntactic integrity of the input utterance.

4.3. Learner Modeling

Learner modeling in AutoTutor requires semantic matching algorithms that compare the learner input with AutoTutor’s expectations and misconceptions. AutoTutor incorporates several semantic evaluation algorithms when performing these matches, but most notably Latent Semantic Analysis (LSA) (Landauer, McNamara, Dennis, & Kintsch, 2007). LSA is a statistical technique that measures the conceptual similarity of two text sources. In this semantic matching algorithm, a vector representing the semantic content of the contribution is created and compared to vectors that represent the semantic content of expectations and misconceptions. The cosine between the two vectors is calculated to produce a match-similarity score from 0 to 1 (negative cosines are rare and are converted to 0 in AutoTutor).

The LSA algorithm in AutoTutor computes the extent to which the information within the learner turns (i.e., an individual turn, a combination of turns, or collective sequence of turns) semantically matches each expectation in the ideal answer. Expectation E_i is considered covered if the content of the learner’s cumulative set of turns meets or exceeds a threshold T in its LSA cosine value (which varies from near 0 to 1). That is, E_i is covered if the cosine match between E_i and the learner input I (including turns 1 through N) is high enough: $\text{cosine}(E_i, I) \geq T$. The threshold has varied between .40 and .75 in various instantiations of AutoTutor.

In the course of the dialog- and learner-modeling, the system periodically identifies a missing expectation and posts the goal of covering the expectation. When expectation E_i is missed (and therefore posted), AutoTutor attempts to get the learner to articulate it by generating hints and prompts affiliated with E_i to help the learner fill in missing words and propositions.

Prompts and hints are selected to maximize an increase in the LSA cosine match score (hereafter called the *match score*) when answered successfully. Stated differently, hints and prompts are selected to maximize pattern completion.

Sometimes the learner expresses misconceptions during the dialog. This happens when the learner input I matches a misconception M with a sufficiently high match score. At that point AutoTutor corrects the misconception and goes on.

The selection of the next E_i to cover follows the principle of the zone of proximal development or what some call frontier learning (Brown, Ellery, & Campione, 1998; Vygotsky, 1986). That is, AutoTutor builds on what the learner has managed to articulate. More formally, AutoTutor selects the next E_i from the set of expectations that (a) has the highest match score and (b) has a subthreshold match-score (i.e., the expectation has not yet been covered). This *subthreshold expectation selection* algorithm assumes that the expectations should not be covered in a prescribed sequential order. However, ordering constraints may also be considered in a *sequential expectation selection* algorithm. Some subject matters have ordering constraints but others do not.

While earlier versions of AutoTutor relied exclusively on LSA for learner modeling, newer versions incorporate an inverse word frequency weighted overlap (IWFOW) algorithm as well. The IWFOW algorithm is a word-matching algorithm in which each word is weighted on a scale from 0.0 to 1.0, relative to its inverse frequency in the English language using a corpus like CELEX (Baayen, Piepenbrock, & Gulikers, 1995). As a consequence, higher frequency words such as closed-class function words (e.g. and, but, a, the, etc.) have comparatively low weights and little effect on the IWFOW match score. Lower frequency words (e.g. RAM, system, speed, etc.) have higher weights and exert more influence on the IWFOW match score. Similar to LSA,

the IFWO algorithm also generates a semantic-match similarity score between 0 and 1 for each expectation.

Both LSA and IFWO have strengths and weaknesses which are extensively discussed in previous publications (Hu, Cai, Wiemer-Hastings, Graesser, & McNamara, 2007). Hence, some versions of AutoTutor use a hybrid match score ($HYBRID = .33 \times LSA + .67 \times IFWO$) to leverage the benefits of both systems while minimizing the effects of their flaws. The most recent systems use a combination of LSA, IFWO and regular expressions to assess semantic matches (Cai, Graesser, Forsyth, Burkett, Millis, Wallace, Halpern, & Butler, 2011). The regular expressions provide a significant added boost in the accuracy of the semantic matches. In contrast, the addition of a syntactic parser has had a non-significant impact on match scores. The impact of syntax is small because a large percentage of student contributions are ungrammatical. Syntax plays an important role in speech act classification, but not in semantic matching algorithms for learner modeling.

4.4. Dialog Management

The dialog-management module in most versions of AutoTutor is an augmented, finite-state transition-network. The nodes in the network refer to knowledge goal-states (e.g., expectation E is under focus and AutoTutor wants to get the learner to articulate it) or dialog states (e.g., the learner just expressed an assertion as his or her first turn in answering the question). The arcs refer to categories of tutor dialog moves (e.g., feedback, pumps, prompts, hints, summaries) or discourse markers that link dialog moves (e.g., “okay,” “moving on,” “furthermore”). A particular arc is traversed when particular conditions are met. For example, a pump arc is traversed when it is the learner’s first turn and the learner’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 dialog. These algorithms and procedures operate on the snapshot of parameters, curriculum content, knowledge goal-states, learner knowledge, dialog states, LSA measures, and so on, which reflect the current conversation constraints and achievements. For example, there are algorithms that select dialog move categories intended to get the learner to fill in missing information in *E* (the expectation under focus). There are several alternative algorithms for achieving this goal.

One of the early algorithms that we adopted relied on fuzzy production rules. If the learner 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 learners. The particular prompt selected from the curriculum script would be tailored to extract the particular missing word through another module that fills posted dialog move categories with particular content. If the learner 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. A more elaborated algorithm to fleshing out *E* uses one or two cycles of hint–prompt–assertion. That is, AutoTutor’s selection of dialog moves over successive turns follows a particular order: first hint, then prompt, then assert in cycle 1 and optionally in cycle 2 another hint, then prompt, then assert. AutoTutor exits the one or two cycles as soon as the learner articulates *E* to satisfaction (i.e., the semantic-match threshold is met) or the cycles are exhausted.

Some versions of AutoTutor have a more complex dialogue-management mechanism that involves more complex planning and a modular architecture (Graesser et al., 2005a). However, it is beyond the scope of this chapter to address such enhancements.

4.5. Output Rendering

AutoTutor communicates through an animated conversational agent with speech, facial expressions, and rudimentary gestures. The various versions of AutoTutor have used several text-to-speech engines and facilities to render facial expressions and body movements. From the standpoint of the present chapter, we have relied on simple solutions to rendering the output. Regarding AutoTutor's spoken utterances, we have primarily used text-to-speech output of available speech synthesizers rather than dynamically computing mark-up codes to make the speech more discourse sensitive. Our attempts to have discourse-sensitive mark-up (for emphasis on words and changes in intonation) have proven futile because of the impenetrable mechanisms of the text-to-speech engines. However, given that some words are not pronounced correctly by the synthesizers, we have routinely composed a library of mark-up codes or have changed the spellings of these words to make them sound more natural. Regarding facial expressions and gestures, we have a library of facial expressions and gestures that are coordinated with specific speech acts or discourse markers; again these are not dynamically composed in a discourse-sensitive manner. For example, the tutor delivers positive feedback with a verbal message (e.g., "good job") and a nod. We also have algorithms for randomly having the agents blink their eyes or make movements to make the demeanor look alive.

A screenshot of one of the more recent versions of the AutoTutor interface is shown in Figure 3. The interface typically has five major windows shown in the figure. Window 1 (top of screen) is the main question that stays on the computer screen throughout the conversation about the question. Window 2 (left middle) is the animated conversational agent that speaks the content of AutoTutor's turns and gestures appropriately. Window 3 (right middle) is either blank or has auxiliary diagrams. Window 4 (right bottom) displays the learners' answers as they type them in. Window 5 (left bottom) displays the dialog history of the learner and the tutor.

4.6. Domain Knowledge

Content in AutoTutor is primarily represented as a set of words, sentences, or paragraphs in a free text format. This makes it easy for a lesson planner to create new topics and content without having to craft the content in structured LISP or Prolog code, as is the case with most ITSs. The system was designed this way so that AutoTutor could be used for virtually any topic except those that require the precision of mathematics, and so lesson planners could develop the content with minimal knowledge of discourse or computer programming.

The curriculum script in AutoTutor organizes the topics and content of the tutorial dialog. 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.

AutoTutor utilizes large text bases for the semantic-processing algorithms. All versions of AutoTutor represent world knowledge as *LSA* spaces, but some versions of AutoTutor and its progeny have incorporated other forms of world-knowledge representation, such as textbooks, glossaries, and conceptual graph structures. Some of this information (e.g., texts and keywords) are also used to build custom language models for the versions that incorporate spoken input.

In addition to the domain-specific content contained in the curriculum script and the text bases, AutoTutor has a repository of different domain-independent data structures. There are *conversation rules* that are represented as production rules, finite-state transition networks, or recursive, augmented state transition networks. There are different templates of speech acts that are used by the speech-act classifier. There are different categories of feedback (e.g., positive, neutral,

negative), each with a set of exemplars that are randomly selected at runtime. Finally, there are discourse markers that help connect clauses in order to form syntactically-correct expressions.

5. Evaluating Dialog-Based ITS

Dialog-based ITSs service multiple goals so they need to be evaluated along multiple dimensions. Perhaps the most important goal is to promote learning gains while simultaneously keeping learners engaged. This goal is serviced by sub-goals such as recognizing speech (when spoken input is enabled), classifying utterances, comprehending contributions to provide appropriate feedback, modeling knowledge, and generating conversationally smooth tutorial dialogs that mirror human-human tutoring dialogs. Using AutoTutor as a case study, we describe how dialog-based ITSs can be evaluated along these different dimensions. We begin with evaluations of the individual components followed by more global evaluations of overall effectiveness.

5.1. Recognizing Spoken Responses

Word error rate (WER) and word recognition rate (WRR) are standard metrics for assessing the reliability of automatic speech-recognition systems. $WER = [S + D + I]/N$, where S , D , and I are the number of substitutions, deletions, and insertions in the automatically recognized text (with errors) when compared to the ideal text (no errors) of N words. $WRR = 1 - WER$. The word-recognition rate (WRR) for the automatic speech-recognition system used in speech-enabled version of AutoTutor (D'Mello et al., 2010a) has ranged from .017 to .887 with a mean of .542 ($SD = .270$). Most of the errors appear to be substitution (52.3%) and insertion errors (39.1%) while deletion errors were comparatively rare (8.8%).

WER and WRR metrics frame automatic speech-recognition as a speech-to-text translation problem. The problem can also be framed as an information-extraction problem if one is simply interested in quantifying how many words were correctly recognized, without regard to

the ordering of words. When word order is ignored, on average 75% ($SD = 15.3\%$) of the words were correctly recognized by our ASR system. Recognition is somewhat higher for content words (81.5%) compared to all words (76.5%).

Taken together, these results suggest that the automatic speech-recognition system would be problematic for a system that requires a syntactically-intact utterance in order to evaluate a learner's response. However, performance is expected to be relatively stable for ITSs that rely on shallow NLU techniques because these algorithms match key words and phrases while ignoring the syntax of the utterance.

5.2. Classifying Learners' Responses

Olney et al. (2003) evaluated how accurately AutoTutor's speech-act classifier could classify approximately 10,000 learner utterances extracted from AutoTutor's log files. The classifier discriminated among contributions, frozen expressions (metacognitive and metacommunicative statements), and the 16 questions from the Graesser and Person (1994) taxonomy. The system achieved a modest average F-measure of .54. This was attributed to the fact that approximately 97% of the learner utterances were contributions and several of the question categories rarely occurred (example questions never occurred). To account for this, a measure that weighted accuracy rates of each category in a manner that reflects occurrence of the categories was computed. The average weighted F-measure was 0.97, which is quite impressive and more reflective of real-world performance.

5.3. Comprehending Learners' Responses

We have analyzed the accuracy of the semantic-matching algorithms by comparing AutoTutor's scores to judgments of subject-matter experts (Graesser, Penumatsa, Ventura, Cai, & Hu, 2007; Graesser, Wiemer-Hastings, Wiemer-Hastings, Harter, & Person, 2000). For example, we have analyzed the complete answers that learners gave as an answer to one of the

challenging physics questions, recorded AutoTutor's match score for each expectation/misconception, and collected ratings from 5 expert physicists as to whether each expectation/misconception was present in the learners' answers. The correlations between these match-evaluation scores and expert ratings have varied between 0.35 and 0.50, depending on the criterion, semantic algorithm, and other details that need not be considered here.

In other studies, graduate students, instead of experts, rate the extent to which learner essays express particular sentence-like expectations. Similarly, the LSA component evaluates the extent to which the expectations are covered. The correlations between computer and these novice humans are significant and impressive, ranging from 0.5 to 0.7, when computing the proportion of expectations covered in an essay. In general, the accuracy of AutoTutor's LSA component is on par with graduate-level research assistants but not quite as good as accomplished experts (Graesser, Hu, & McNamara, 2005b).

Cai et al. (2011) recently conducted an analysis on semantic-match scores with the AutoTutor facilities in Operation ARIES!, a new system we developed, as described in subsequent sections. The semantic-match algorithm included regular expressions in addition to LSA. The performance was quite impressive, showing a correlation between the humans and the computer of 0.66 whereas two humans correlated 0.68. Indeed, the regular-expression component had a substantial improvement in accuracy over LSA alone.

5.4. Modeling Learner Knowledge

If AutoTutor is performing effective user modeling, then the dialog moves it selects should be correlated with the learners' prior knowledge. In one analysis of conceptual physics, we collected pretest scores on a psychometrically-validated test called the Force Concept Inventory (Hestenes, Wells, & Swackhamer, 1992). The learners' physics knowledge had a significant positive correlation with positive feedback moves ($r = .38$) and a negative correlation

with negative feedback ($r = -.37$) (Jackson & Graesser, 2006). Another example applies to the corrections that AutoTutor made when identifying learner errors and misconceptions. The correlation between prior knowledge and corrections was negative ($r = -.24$), and marginally significant.

In addition to feedback and misconception-correction, AutoTutor's major dialog (pumps, hints, prompts, and assertions) should also be sensitive to learners' knowledge of physics. There is a continuum from the learner-provided information to tutor-provided information as we move from pumps, to hints, to prompts, to assertions. The correlations with learner knowledge reflected this continuum perfectly, with values of 0.49, 0.24, -0.19, and -0.40, respectively. Thus, for learners with more knowledge of physics, all AutoTutor needs to do is primarily pump and hint, thereby encouraging or nudging the learner to supply the answer to the question and articulate the expectations. For learners with less knowledge of physics, AutoTutor needs to generate prompts for specific words or to assert the correct information, thereby extracting knowledge piecemeal or telling the learner the correct information.

5.5. Generating Naturalistic Dialogs

Person and Graesser (2002) performed a *bystander Turing test* (described below) to evaluate the naturalness of AutoTutor's dialog moves. They randomly selected 144 tutor moves in the tutorial dialogs between learners and AutoTutor. Six human tutors were asked to fill in what they would say at these 144 points. So, at each of these 144 tutor turns, there was a turn with what the human tutor generated versus what AutoTutor generated. They subsequently tested whether a group of learners could discriminate between dialog moves that were generated by a human versus a computer; half in fact were generated by the human tutors and half were by AutoTutor. It was discovered that the bystander learners were unable to discriminate whether particular dialog moves had been generated by a computer versus a human; the d' discrimination

scores were actually a bit negative (-.08), but not significantly. This rather impressive outcome supports the claim that AutoTutor produces a reasonably good simulation of human tutorial dialog. However, there is one note of qualification in this conclusion. The learners would surely be able to discriminate whether they are talking with a human versus a computer if they had a lengthy conversation with the tutor.

5.6. Learners Perceptions of the System

Learner ratings of AutoTutor have also been collected in order to get their impressions of the tutoring environment. The ratings lean toward the positive side, but there have been no systematic comparisons with human tutors or alternative learning environments. We have compared different versions of AutoTutor, but a provocative finding has made us somewhat skeptical about relying on ratings of learner impressions. Specifically Jackson and Graesser (2007) documented that there was a negative relationship between deep learning and enjoyment: learners least preferred those versions from which they learned most. Learners' metacognition of learning is limited (Graesser, D'Mello, & Person, 2009) so it is perhaps not surprising that their ratings of liking were not positively correlated with learning. Deep learning is challenging and sometimes painful, which may clash with an enjoyable experience for some groups of learners.

5.7. Increasing Learning

The learning gains of AutoTutor have been evaluated in over 20 experiments since its inception in the areas of computer literacy (Graesser et al., 2004a) and Newtonian physics (VanLehn et al., 2007). Training times in these studies varied from 30 minutes to 4 hours and tutorial sessions were sometimes split across multiple days. Measures of learning have varied in scope, depth, difficulty, and open-endedness. They have included: (1) multiple choice questions on shallow knowledge that tap definitions, facts and properties of concepts, (2) multiple choice questions on deep knowledge that tap causal reasoning, justifications of claims, and functional

underpinnings of procedures, (3) essay quality when learners attempt to answer challenging problems, (4) a cloze task that has learners fill in missing words of texts that articulate explanatory reasoning on the subject matter, and (5) performance on problems that require problem solving.

AutoTutor improves learning between 0 and 2.1 sigma, depending on the learning performance measure, the comparison condition, the subject matter, and the version of AutoTutor. The mean effect size of 0.8 sigma (a large effect) is on par with learning effects achieved by several other ITSs and by human tutors (VanLehn, 2011).

Assessments of learning gains obviously depend on the comparison conditions. The learning gains are approximately 0.8 for AutoTutor compared to a do-nothing control or a condition of reading from a textbook on the same topics for an equivalent amount of time. The learning gains are approximately the same for AutoTutor and an expert human tutor who interacts with the learner by computer-mediated communication (as opposed to face-to-face).

The largest learning gains from AutoTutor have been on deep-reasoning measures rather than measures of shallow knowledge (VanLehn et al., 2007). AutoTutor is most effective when there is an intermediate gap between the learner's prior knowledge and the ideal answers of AutoTutor; AutoTutor is not particularly effective in facilitating learning in learners with high domain knowledge, nor when the material is too much over the learner's head.

5.8. Limitations and Points of Failure

We have conducted very detailed analyses of AutoTutor that point to aspects of the dialog and tutoring mechanism that could be improved. One problem lies in errors in interpreting the content of learners' utterances. The pattern matching operations between learner contributions in a turn and AutoTutor's expectation statements (i.e., E1, E2, E3) are not perfect, so some learners get frustrated and conclude that AutoTutor is not listening. This interpretation

problem can be mitigated by improving the depth of the pattern matching algorithms, including some facilities for inferences and entailment (Rus & Graesser, 2007).

A second problem consists of misclassification of the speech acts. The dialog coherence breaks down when some misclassification errors occur, which ends up confusing learners. More efforts are needed to improve the speech act classification accuracy and to manage the dialog to minimize unwanted consequences.

A third problem occurs when the AutoTutor does not generate relevant and informative answers to the learners questions. AutoTutor can handle roughly half of the learner questions, so half of AutoTutor's replies are either incorrect, constitute requests for clarification (*I don't understand your question, so could you rephrase it?*), or pass the burden onto the learner (*That's a good question, so how would you answer it?*). Improvements in the question answering facilities are needed to minimize this third problem. Fortunately, this third problem does not seriously compromise the dialog because the learners rarely ask questions (Graesser & Person, 1994).

6. Beyond the Basic Components and AutoTutor to the Emotional Realm

Our discussion so far has primarily focused on the basic AutoTutor technology that was developed between 1997 and 2005. We have made considerable enhancements to this core system over the last decade. For example, Hu and colleagues (Hu, Cai, Han, Craig, Wang, & Graesser, 2009) developed a minimalistic version of AutoTutor called AutoTutor-Lite. We have also developed a version of AutoTutor, called AutoTutor-3D, that guides learners on using interactive simulations of physics microworlds (Graesser et al., 2005a; Jackson & Graesser, 2006). In addition to these enhancements, AutoTutor has also inspired a number of natural-language learning systems that go beyond the basic components of dialog-based ITSs in a number of significant ways. These systems take the vision of learning by holding a conversation

with a computer to a number of new realms. One system, Operation ARIES!, extends one-on-one dialogs to include one-on-two dialogs, which we call triologs (Millis, Forsyth, Butler, Wallace, Graesser, & Halpern, 2011). There are systems that process learner responses using deeper NLP techniques (DeepTutor), train learners to deploy effective metacognitive and self-regulated learning behaviors (MetaTutor; Azevedo, Witherspoon, Graesser, McNamara, Rus, Cai, Lintean, and Siler (2008)), model expert instead of novice human tutors (Guru Tutor; Olney, D'Mello, Person, Cade, Hays, Williams, Lehman, and Graesser (2012)), and simulate human mentors during multiparty computer-supported learning activities (AutoMentor).

It is beyond the scope of this chapter to describe these next-generation ITSs in sufficient detail. Therefore, we focus on one subset of enhancements that attempt to move the learner models that focus on cognitive states into the realm of emotional states. One system called the Affective AutoTutor automatically adapts its dialogues based on whether the learner is bored, confused, or frustrated (D'Mello & Graesser, in press). Another system, called the Gaze Tutor, has unique mechanisms to monitor and correct learners' disengagement behaviors by tracking their eye gaze in near real time (D'Mello, Olney, Williams, & Hays, 2012). There is also a system that strategically plants seeds of cognitive disequilibrium and confusion in the minds of learners to engender deeper modes of thinking (D'Mello et al., in press). The remainder of this section describes these systems along with a discussion of how textual cues and characteristics of the emerging tutor-student discourse can be used to track emotions.

6.1. Responding to Learners' Emotional States

ITSs have come a long way towards providing individualized instruction to fit the cognitive needs of individual learners. However, ITSs can be more than mere cognitive machines, because emotions can have a major impact on engagement and learning (Calvo & D'Mello, 2011; Pekrun & Stephens, 2012). Affect sensitivity is important for ITSs that aspire to

model human tutors because it has been claimed that accomplished tutors are able to recognize and appropriately respond to learners' emotional states (Lepper & Woolverton, 2002). An affect-sensitive ITS would incorporate assessments of the learners' cognitive and affective states into its pedagogical and motivational strategies in order to keep learners engaged, boost self-confidence, heighten interest, and presumably maximize learning.

We have recently developed a version of AutoTutor (called the Affective Tutor or Supportive Tutor) that automatically detects learners' affective states (specifically boredom, confusion, frustration, and neutral) by monitoring conversational cues and other discourse features (discussed in more detail in the next section), gross body language, and facial features (D'Mello & Graesser, 2010; D'Mello & Graesser, in press). Each channel independently provides its own diagnosis of the learner's affective state. These individual diagnoses are combined with a decision-level fusion algorithm that selects a single affective state and a confidence value of the detection. The algorithm relies on a voting rule enhanced with a few simple heuristics.

The Affective AutoTutor has a set of production rules that map dynamic assessments of learners' cognitive and affective states with tutor actions to address the presence of boredom, confusion, and frustration. There are five parameters in the learner model and five parameters in the tutor model. The parameters in the learner model are: (a) the current affective state detected, (b) the confidence level of that affect classification, (c) the previous state detected, (d) a global measure of learner ability (dynamically updated throughout the session), and (e) the conceptual quality of the learner's immediate response. The Affective AutoTutor incorporates this cognitive-affective assessment of the learner and responds with: (a) feedback for the current answer, (b) an affective statement, (c) the next dialog move, (d) an emotional display on the face of the tutor agent, and (e) an emotional modulation of the voice produced by AutoTutor's text-to-speech engine.

The tutor responds to the sensed emotions with empathetic, encouraging, and motivational dialog-moves and emotional displays. An emotion generator is also needed for the Affective AutoTutor because the system is expected to synthesize emotions as well. Therefore, the agent needs to speak with intonation that is properly integrated with facial expressions that display emotions. For example, an enthusiastic nod accompanied positive feedback after the learner provided a correct response. In contrast, there was a shaking of the head when the learner response was low quality, and a skeptical look when the tutor detected that the learner was hedging. What we see is that even a small set of emotion displays like in these examples given here can go a long way in conveying the tutor's emotions.

We tested the effectiveness of the Affective Tutor in improving learning over the Regular AutoTutor in a study where 84 learners completed two 30-minute training sessions with either tutor (D'Mello, Lehman, Sullins, Daigle, Combs, Vogt, Perkins, & Graesser, 2010b). The results indicated that the Affective tutor helped learning for low-domain knowledge learners during the second 30-minute learning session. The Affective tutor was less effective at promoting learning for high-domain knowledge learners during the first 30-minute session. Importantly, learning gains increased from Session 1 to Session 2 with the Affective tutor whereas they plateaued with the Regular tutor. Learners who interacted with the Affective Tutor also demonstrated higher performance on subsequent transfer tests than those who interacted with the Regular AutoTutor. A follow-up analysis into learners' perceptions of both tutors indicated that their perceptions of how closely the computer tutors resembled human tutors increased across learning sessions, was related to the quality of tutor feedback, and was a powerful predictor of learning (D'Mello & Graesser, 2012). Interestingly, the increase was greater for the Affective tutor.

6.2. Sensing Emotion from Discourse Features

A one-on-one tutoring session with AutoTutor yields a rich trace of contextual information, characteristics of the learner, episodes during the coverage of the topic, and social dynamics between the tutor and learner. These conversational cues cover a broad and deep feature set that includes assessments of deep meaning, world knowledge, and pragmatic aspects of communication. They might also be predictive of learner affect. To test this hypothesis, after completing a learning session with the tutor, learners' emotions were judged by the learners themselves (self-judgments), untrained peers, and trained judges using a retrospective affect judgment protocol (Graesser, McDaniel, Chipman, Witherspoon, D'Mello, & Gholson, 2006). Several conversational features and discourse markers (collectively called dialog features or discourse features) were extracted from AutoTutor's log files and were utilized to infer learner affect. The dialog features were computed for each learner-tutor turn (i.e. learner submits response, tutor provides feedback, tutor presents next question). They included *temporal* features (e.g. time on problem, response time), assessments of *response verbosity* (e.g. number of characters, speech act), assessments of the *conceptual quality* of the learner's response obtained by Latent Semantic Analysis (LSA), *conversational directness* (i.e. how much information the tutor is explicitly providing to the learner), and *tutor feedback* (negative, neutral, positive). The full list of features can be found in D'Mello, Craig, Witherspoon, McDaniel, and Graesser (2008).

Analyses regressing the presence or absence of each emotion on this set of dialog features illuminated a number of interesting patterns. Learners are prone to be confused early on when a problem or difficult question is presented, whereas they tend to become more bored as the tutoring session continues. They provide shorter answers when confused but are more verbose when they are actively engaged in the learning session. Confusion and frustration tend to occur

when the tutor gives negative feedback, whereas positive feedback sometimes triggers delight. The tutor's hints (e.g., "What about X?") have an interesting impact on affect. Hints are indirect speech and are attempts by the computer tutor to lead the learner down a productive line of reasoning, but without being too obvious as to what the correct answer is. The hope is that the learners will actively construct answers and figure out the answers as opposed to the computer simply delivering information. However, the indirectness of hints is sometimes confusing or frustrating to learners, as was confirmed in our analyses. Another discovery is that learners get frustrated when they believe that they expressed something, but the computer fails to understand them and acknowledge their contributions.

We compared the accuracy by which 17 standard classifiers (e.g., Naive Bayes logistic regression, support vector machines) could detect the affective states from the dialog features (D'Mello et al., 2008). Machine learning experiments with 10-fold cross validation indicated that standard classifiers were moderately successful in discriminating the affective states of boredom, confusion, flow/engagement frustration, and neutral, yielding a peak accuracy of 42% with neutral (*chance* = 20%) and 54% without neutral (*chance* = 25%). Individual detections of boredom, confusion, flow, and frustration, when contrasted with neutral, had accuracies of 69%, 68%, 71%, and 78%, respectively (*chance* = 50%). Follow-up classification analyses that assessed the degree to which machine-generated affect labels correlated with affect judgments provided by humans revealed that human-machine agreement was on par with novice judges (self and peer) but quantitatively lower than trained judges. This dialog-based affect detector plays a very prominent role in the Affective AutoTutor system.

6.3. Sensing Emotion from Textual Cues

While the previous analysis focused primarily on discourse patterns and other contextual cues, we recently explored the possibility of predicting learner emotions (boredom,

flow/engagement, confusion, and frustration) by performing a deeper analysis of the text of the tutorial dialogs (D'Mello & Graesser, in press). Transcripts from the tutorial dialogs were analyzed with four methods that included (a) identifying direct expressions of affect, (b) aligning the semantic content of learner responses to affective terms with Latent Semantic Analysis (LSA) (Landauer et al., 2007), (c) identifying psychological and linguistic terms that are predictive of affect with the *Linguistic Inquiry and Word Count* (Pennebaker, Chung, Ireland, Gonzales, & Booth, 2007), and (d) assessing cohesion relationships that might reveal learner affect with *Coh-Matrix* (Graesser, McNamara, Louwerse, & Cai, 2004b).

A detailed description of these four methods is beyond the scope of this chapter, so we proceed with brief descriptions only. The direct-expression models simply use regular expressions to detect affect. This approach is only viable if learners directly express affect in their responses (e.g., “This stuff is so *confusing*”). We used LSA to assess whether affect can be detected by semantically aligning learner responses (e.g., “This problem is a drag”) with affective terms (e.g., “boredom”). The Linguistic Inquiry and Word Count (LIWC) is a validated computer tool that analyzes bodies of text using dictionary-based categorization. LIWC-based affect-detection methods attempt to identify particular words that are expected to reveal the affective content in the text. Features of LIWC that were examined included affect terms (e.g., “sad”, “happy”), cognitive terms (e.g., “cause”, “maybe”), and several linguistic terms (e.g., pronouns, adverbs, prepositions). Finally, Coh-Matrix is a computational facility for automatically analyzing bodies of text on over 600 dimensions. Coh-Matrix provides over 100 measures of various types of cohesion, including co-reference, referential, causal, spatial, temporal, and structural cohesion. Coh-Matrix also has hundreds of measures of linguistic complexity (e.g., syntactic complexity), characteristics of words, and readability scores. The measures that were selected included co-reference cohesion (noun overlap, stem overlap, etc.),

pronoun-referential cohesion, semantic cohesion, connectives, and other measures of word concreteness and readability.

Our results confirmed that learners rarely articulated their emotions to the tutor, thereby rendering the direct expression model inadequate. Nor did their responses semantically align with emotional terms. Instead, a deeper analysis of textual cues via LIWC and Coh-Metrix were needed to infer learner affect. Models constructed by regressing the proportional occurrence of each emotion on textual features derived from these methods yielded large effects ($R^2 = 38\%$). Although learners rarely articulate their emotions to the tutor and their responses mainly consist of words related to the content of the tutoring session (e.g., RAM, operating systems), some of the content words used, function words that connect content words, and the cohesiveness of their responses, ultimately revealed their affective states. The next step of this research is to develop finer-grained text-based affect detectors that operate on the turn level so that AutoTutor might be responsive to learner emotions without the need for expensive posture sensors and facial feature tracking.

6.4. Gaze Tutor: Responding to Learners' Disengagement Behaviors

It is generally acknowledged that engagement is an important precursor to learning, but until recently most ITSs have made little effort to meaningfully engage learners. Although learners might begin a learning session with an ITS with some level of interest and enthusiasm, boredom inevitably creeps in as the session progresses, when the novelty of the system and content fades, and when learners have difficulty comprehending the material (Larson & Richards, 1991; Pekrun, Goetz, Daniels, Stupnisky, & Perry, 2010). When boredom strikes, any further instruction is essentially futile.

Attention to task-related thoughts is one critical precursor of engagement in a learning activity. Therefore, developing interventions that monitor periods of waning attention and

attempt to encourage more productive use of *attentional resources* might be one promising way to increase engagement and promote learning. We tested this claim by developing a dialog-based learning system, called the Gaze Tutor, that used a commercial eye tracker to monitor learners' gaze patterns in order to identify when they were bored, disengaged, mind wandering, or zoning out (D'Mello et al., 2012). The tutor attempted to reengage learners with dialog moves that directed learners to reorient their attention towards the animated pedagogical agent embodying the tutor. These gaze-reorienting dialogs consisted of short direct messages that instructed learners to pay attention to the tutor or to certain parts of the interface.

We evaluated the efficacy of the gaze-reactive tutor in promoting learning, motivation, and engagement in an experiment where 48 learners were tutored on four biology topics with both gaze-reactive and non gaze-reactive (control condition) versions of the tutor. The results indicated that the gaze-sensitive dialogs were successful in dynamically reorienting learners' attentional patterns to the important areas of the interface. The effectiveness of gaze-orientation faded over time but did not entirely diminish. Although gaze-reactivity did not impact self-reported motivation and engagement, posttest scores for deep reasoning questions were higher when learners interacted with the gaze-sensitive tutor. Interestingly, individual differences in scholastic aptitude moderated the impact of gaze-reactivity on learning gains. Gaze-reactivity was associated with a small improvement in overall learning for learners with average aptitude, but learning gains were substantially higher for learners with high aptitude and somewhat lower for their counterparts. Future enhancements of the system include replacing the expensive eye-tracker with web-cameras, more fine-grained modeling of disengagement behaviors, a larger repertoire of context-specific gaze-reactive dialogs, and incorporating individual differences in selecting appropriate gaze-sensitive responses.

6.5. Planting Cognitive Disequilibrium and Confusion

A corollary of theories that highlight the benefits of impasses (VanLehn et al., 2003), cognitive conflict (Limón, 2001), cognitive dissonance (Festinger, 1957), and cognitive disequilibrium (Piaget, 1952) is that events that productively confuse learners are beneficial to learning by providing learning opportunities. This is because learners need to engage in deep cognitive activities in order to resolve their confusion. Confusion resolution requires the learners to stop, think, engage in careful deliberation, problem solve, and revise their existing mental models. It is likely not the confusion itself, but the cognitive activities that accompany confusion resolution that are responsible for learning gains. There is some correlational evidence that is suggestive of a positive relationship between impasses, states of uncertainty, confusion and learning gains (Craig, Graesser, Sullins, & Gholson, 2004; D'Mello & Graesser, 2011; Graesser, Chipman, King, McDaniel, & D'Mello, 2007; VanLehn et al., 2003), but confusion has not been causally linked to positive learning outcomes.

We tested the hypothesis that confusion can impact learning by modifying ARIES (Millis et al., 2011) to systematically induce confusion (D'Mello et al., in press). We did this by manipulating whether or not the tutor agent and the learner agent contradicted each other during the dialog by expressing points that were incorrect. The human learner was asked to intervene after each point of possible contradiction; the agents turned to the human and asked “So what would your decision be, <name of learner>”?). If the human learner experiences uncertainty and is confused, this should be reflected in the incorrectness/uncertainty of the human’s answer. Uncertainty is a likely opportunity to scaffold deep comprehension by forcing learners to stop and think.

The data confirmed that the contradictions and false information had an impact on learners’ confusion. Importantly, levels of confusion moderate the impact of the contradictions

on learning. Specifically, the contradictions had no effect on learning when learners were not confused by the manipulations, whereas performance on multiple-choice posttests and on transfer tests was substantially higher when the contradictions were successful in confusing learners. This suggests that there are some benefits to inducing confusion if learners are *productively* instead of *hopelessly* confused. By productive confusion, we mean that the source of the confusion is closely linked to the content of the learning session, the learner attempts to resolve their confusion, and the learning environment provides necessary scaffolds to facilitate the confusion resolution process.

7. Some Open Issues for Dialog-based ITSs

This section discusses some of the recurring issues that arise in research on dialog-based ITSs. We shed light on four issues that we deem critical, knowing fully well that these four items are only a small subset of the open issues in the field. These items can be framed in terms of the following four questions: (a) Is spoken input more advantageous than typed input? (b) is imperfect natural-language processing good enough for meaningful interactions? (c) do the merits of dialog-based ITSs lie in the medium or the message? and (d) when are natural-language dialogs beneficial for learning?

7.1. Is Spoken Input more Advantageous than Typed Input?

The recent advance of commercially-available spoken interfaces raises the question of whether there are advantages of learners speaking their responses instead of typing them in. A *speech facilitation* hypothesis predicts that spoken input will increase learning, whereas a *text facilitation* hypothesis predicts typed input will be superior. Alternatively, a *modality equivalence hypothesis* claims that learning gains will be equivalent because the advantages and disadvantages offered by both input modalities will cancel each other out. Spoken language is, of course, easier to produce because the expression gap between thought and speech is much less

than the gap between thought and writing. Because of the ease of spoken responses, the volume of content is typically longer when spoken than typed. Given that learning is correlated with the volume of responses by the learners (Chi et al., 2004), following a constructivist framework (Dalgarno, 2001; Moshman, 1982), it would be predicted that spoken responses would yield higher learning gains. On the other hand, typed responses offer some advantages as well. While spoken utterances are evanescent, i.e., they disappear shortly after they are spoken, typed contribution remains on the screen for learners to evaluate and revise. The additional time to reflect on their responses affords added perceptual processing, rereading, and memory encoding, thereby increasing learning gains if learners use these textual representations to process the material more deeply. Therefore, there appears to be a tradeoff between the ease of a learner producing a spoken response and the quality of the learner's typed response.

Previous research that has tested these hypotheses within the context of dialog-based ITSs enabled with automated speech-recognition systems has supported the modality-equivalence hypothesis (D'Mello et al., 2010a; Litman et al., 2006). The one study that compared spoken versus typed input during human-human tutoring sessions offered evidence in support of the speech facilitation hypothesis (Litman et al., 2006). There were many differences between the human-human and human-computer tutorial sessions that could potentially explain the discrepant findings. However, the fact that the automated speech recognition systems used by the dialog-based ITSs had substantial error rates that were detected by learners might represent one of the most significant factors. This concern was addressed in two additional experiments that used a Wizard of Oz procedure involving a human who intercepted the learner's speech and transcribed the utterance before submitting it to the computer tutor (D'Mello, Dowell, & Graesser, 2011). This afforded relatively error-free speech recognition, thereby correcting a confound with the previous experiments. The results of these two experiments also resulted in

null effect, once again supporting the modality equivalence hypothesis. The only exception was that highly motivated learners reported lower cognitive load and demonstrated increased learning when typing compared to speaking. Apart from this exception that warrants replication, the four experiments that compared spoken and typed responses with dialog-based ITSs suggest that the *content* is more important than the *mode* of communication (Graesser, Moreno, Marineau, Adcock, Olney, & Person, 2003), a theme which we return to below.

7.2. Is Imperfect Natural-Language Processing Good Enough?

Critics of natural-language interfaces have sometimes argued that language is too vague, imprecise, and ambiguous for computers to deeply understand. According to these skeptics, decoding language requires vast amounts of world knowledge and socially and culturally specific information that far exceeds the scope of current natural-language processing techniques.

Although it is clear that human-like speech and language comprehension clearly surpasses what can be achieved by a computer, the dialog-based ITSs that have been developed suggest that much can be achieved by imperfect speech recognition and shallow language understanding. Examples of systems that are effective despite ASR and NLU imperfections include Litman's ITSPROKE system (Litman et al., 2006), Mostow et al.'s Reading Tutor (Mostow & Aist, 2001), the Scot system (Pon-Barry et al., 2004; Schultz, Bratt, Clark, Peters, Pon-Barry, & Treeratpituk, 2003), the Tactical Language and Culture Training System (Johnson & Valente, 2008), and AutoTutor (D'Mello et al., 2010a).

Of course imperfect ASR and NLU is unlikely to suffice for all applications, thereby raising the question of when imperfect natural-language processing capabilities are good enough to sustain effective interactions. One way to predict the suitability of natural-language interfaces with imperfect capabilities is to consider the interaction between the expected precision of the system and the degree of common ground (shared knowledge) between the user and the

interface. By expected precision we mean the accuracy of comprehending the user's utterance. Common ground refers to the degree of shared knowledge between the user and the interface (Clark, 1996). According to this two dimensional framework, the simplest cases occur when the expected precision is high. In these situations, the quality of the interaction is not expected to be adversely affected by the lack of common ground between the conversational participants, because any deficits in shared knowledge can be easily corrected through a conversation that both partners can effortlessly comprehend.

The more interesting situations arise when the expected precision is low. If the common ground between the conversational partners is high, the conversation will ultimately break down because one of the participants will be aware of the other's natural language understanding deficits. But what about dialog-based ITSs with low to moderate precision? Our position is that these systems can be functional despite natural-language understanding errors provided two conditions are met: First, while the precision of the system can be low, it should be significantly greater than zero, i.e. the system should be able to recognize some meaning even if only at shallow levels of comprehension (e.g., content words, key phrases). Second, the degree of common ground between the learner and the ITS should also be low. Low domain-knowledge learners rarely answer questions with complete answers or even complete sentences. Instead, they utter a few key phrases, make a few statements that are close to their vernacular discourse and for the most part rarely articulate academically precise statements. Since low domain-knowledge learners share a small degree of common ground with the tutor, they can be expected to be oblivious to moderate natural-language understanding errors. It should be noted that this focus on low domain-knowledge learners does not reduce the impact of ITSs, because it is well known that ITSs are typically more effective for low domain-knowledge learners (McNamara,

2001; McNamara & Kintsch, 1996; VanLehn, Graesser, et al., 2007). Furthermore, the low domain-knowledge learners are the ones in need of one-on-one tutoring.

The above discussion raises the important question of how to design dialog-based ITSs that can compensate for partial failure in the quality of interpreting the learner's responses. A solution that we have explored involves a combination of shallow natural language processing and soft, constraint-satisfaction models. Shallow NLP techniques differ from deep NLP approaches as they do not perform a thorough linguistic analysis in understanding a user's utterance. Instead, shallow NLP approaches focus more on the semantics than on the syntax of an utterance. Shallow NLP techniques can compensate for partial failures in the user's utterance if the failure is handled with soft-constraint satisfaction approaches. According to soft-constraint satisfaction (SCS) models, the performance of an intelligent system should not rely on the integrity of any one level or module, but rather should reflect the confluence of several levels/modules that are statistically combined. For example, natural language understanding involves a multilevel computational analysis including phonemes, morphemes, words, syntax, sentence semantics, discourse, pragmatics, world knowledge, and genre. According to a SCS model, when the fidelity at one level fails, the other levels fill in. When there is an ambiguity at one level, the other levels resolve the ambiguity (i.e. context can be recruited to resolve ambiguity in word meaning).

7.3. Do the Merits of Dialog-based ITSs Lie in the Medium or the Message?

The basic one-on-one tutorial dialogs that model learner knowledge that attempt to eke out what the learner knows, identify and correct misconceptions, and help learners construct mental models, can be augmented in a number of ways. Explanations can be accompanied by animations of dynamical processes (e.g., Moreno & Mayer, 2007). Learners can use simulations to model complex systems and observe how different inputs affect outputs (e.g., Graesser et al.,

2005a). The content of the tutor's utterances can be delivered by animated pedagogical agents with sophisticated facial expressions, speech, and gestures (e.g., Johnson et al., 2000). Learners can speak their responses to the tutor instead of typing them in (e.g., Litman et al., 2006). Progress can be directly conveyed to the learner by graphically presenting the results of the learner model to the student (e.g., Jackson & Graesser, 2007). The interaction can be made more game-like by displaying points, levels, and challenge problems (e.g., Jackson, Dempsey, & McNamara, in press). In other words, the basic one-on-one interaction can be decorated in a number of ways.

To what extent do these *decorations* increase learning gains beyond what can be obtained from text-only tutorial dialogs? Although it is too early to say for many other systems, our experience with AutoTutor indicates that they do not add much. For example, comparisons of the effectiveness of presenting AutoTutor's speech acts in print instead of the animated agent yielded a very small 0.13 sigma effect (Graesser et al., 2003). The interactive AutoTutor-3D version with simulations has a 0.22 effect size over the normal conversational AutoTutor (Graesser et al., 2005a). As described above, comparisons between spoken and typed input has consistently resulted in null effects (D'Mello et al., 2011; D'Mello et al., 2010a; Litman et al., 2006). Similarly, versions that varied different levels of feedback had negligible impacts on learning gains (Jackson & Graesser, 2007).

The fact that these different enhancements have generally yielded negligible to small effects in learning gains, suggests that the *content* is more important than the *medium* of communication. Put simply, when it comes to learning, "*The medium is not the message -- the message is the message.*" We suspect that these enhancements might service alternate goals such as promoting long-term engagement and task persistence, a feat that appears to be difficult to sustain via the tedium of text-based interactions.

7.4. When are Natural Language Dialogs Beneficial for Learning?

This chapter, thus far, has extolled the virtues of dialogs-based ITSs, with an emphasis on the ones we have developed and tested. But it is important to realize that these systems are not the panacea to the problems of learning. As with any complex phenomenon, there are conditions where dialog-based ITSs are *effective* and conditions where they are not. There is still considerable research needed before one can prescribe conditions when dialog-based ITSs are maximally effective, however, some recent research has uncovered some important insights.

VanLehn et al. (2007) conducted seven experiments that compared the effectiveness of dialog-based ITSs to expert human tutors, reading text, and doing nothing. They discovered that the alignment of learner knowledge with the content of the instruction was the best predictor of outcomes. Dialog-based ITSs showed significant advantages over reading text when there was misalignment between prior knowledge and complexity of the content. Specifically, novice learners benefitted from ITSs when the content was at the intermediate level. However, the tutorial dialogs yielded null effects (compared to reading text) when prior knowledge and content complexity aligned (i.e., when novices read materials designed for novices and intermediates read materials designed for intermediates).

The results of our experiments with the Affective AutoTutor and the gaze-sensitive tutor provide additional boundary conditions on the effectiveness of these systems. The Affective AutoTutor was more effective than the regular tutor for low-domain knowledge learners in the second session, but not the first session. Furthermore, learners with more knowledge never benefited from the motivational messages of the Affective AutoTutor. These learners don't need the emotional support, but rather they need to go directly to the content. Therefore, there appears to be a tradeoff to quick support and empathy compared to no affect sensitivity.

Interestingly, a reverse pattern was obtained from the gaze-sensitive tutor. The somewhat more stern, gaze-reorienting messages were more effective in promoting learning for the gifted learners than their counterparts. Similar findings have also been noted for tutors who either adopt politeness principles (Brown & Levinson, 1987) when formulating their dialog moves compared to more direct and less socially-intelligent styles of interaction (McLaren, DeLeeuw, & Mayer, 2011; Wang, Johnson, Mayer, Rizzo, Shaw, & Collins, 2008). Learners who make the most errors appear to benefit from tutors with a degree of social intelligence and polite responses.

These three lines of research are suggestive of an important aptitude-treatment interaction. Struggling learners might benefit from encouraging, motivational, empathetic, and polite interactions when they encounter troublesome impasses, get stuck, and risk disengagement. On the other hand, more direct and assertive interaction styles might be beneficial for the more gifted learners.

8. Concluding Remarks

With much pomp and pomposity, US. Industries Inc. unveiled their Auto Tutor machine at the 1964 World's Fair. Though quite advanced for its time, this was a rather crude machine with buttons and 35mm slides. Despite its rudimentary interaction mechanisms, its main advantage was that it allowed learners to advance at their own pace, thereby benefitting both beginners and advanced students. In some ways, much has changed over the last few decades. We have shown that it is feasible to program digital computers to effectively mirror the nuances of one-on-one human tutorial dialogs. Despite imperfect and imprecise natural-language understanding capabilities, these systems can provide meaningful interactions that yield learning gains that rival the experts. However, in other ways, these systems are still lacking in some fundamental aspects. Most importantly, they are a bit lackluster, conversationally brittle, and somewhat tedious to use, all factors that do not facilitate long-term use and acceptance. Put rather crudely, our systems

work, but the average student does not want to use them over long periods of time. While no one would deny that an increase in learning ability is an important criterion of success, it can be argued that sustaining engagement with the student over long-term interactions is likewise of equal importance. Unfortunately, until recently the challenge of keeping learners engaged over extended interactions has been a somewhat neglected area of research. Therefore, in our view, the next grand challenge of dialog-based ITSs is to implement mechanisms that promote long-term learning by maintaining levels of interest and engagement that persist after the initial novelty of the systems fades. Indeed, the promise of dialog-based ITSs will only be realized once an optimal balance between the dual goals of engagement and learning is achieved.

References

- Alibali, M. W., & Nathan, M. J. (2012). Embodiment in mathematics teaching and learning: A view from students' and teachers' gestures. *Journal of the Learning Sciences, 21*(2), 247-286.
- Arnott, E., Hastings, P., & Allbritton, D. (2008). Research Methods Tutor: Evaluation of a dialogue-based tutoring system in the classroom. *Behavior Research Methods, 40*(3), 694-698.
- Azevedo, R., Witherspoon, A., Graesser, A., McNamara, D., Rus, V., Cai, Z., Lintean, M., & Siler, E. (2008). MetaTutor: An adaptive hypermedia system for training and fostering self-regulated learning about complex science topics. In R. Pirrone, R. Azevedo & G. Biswas (Eds.), *Papers from the AAAI Fall Symposium on Cognitive and Metacognitive Educational Systems* (pp. 14-19). Menlo Park, CA: AAAI Press.
- Baayen, R., Piepenbrock, R., & Gulikers, L. (1995). The CELEX lexical database (CD-ROM). Philadelphia: University of Pennsylvania.
- Baylor, A. L., & Kim, Y. (2005). Simulating instructional roles through pedagogical agents. *International Journal of Artificial Intelligence, 15*(2), 95-115.
- Biggs, J. (1996). Enhancing teaching through constructive alignment. *Higher Education, 32*(3), 347-364.
- Bloom, B. (1956). *Taxonomy of educational objectives: The classification of educational goals. Handbook I: Cognitive domain*. New York: McKay.
- Bransford, J., Goldman, S., & Vye, N. (1991). Making a difference in people's ability to think: Reflections on a decade of work and some hopes for the future. In R. Sternberg & L. Okagaki (Eds.), *Influences on children* (pp. 147-180). Hillsdale, NJ: Erlbaum.

- Brennan, S., & 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(3), 383-398.
- Brown, A., Ellery, S., & Campione, J. (1998). Creating zones of proximal development electronically. In J. Greeno & S. Goldman (Eds.), *Thinking practices in mathematics and science learning* (pp. 341-367). Mahwah, NJ: Lawrence Erlbaum.
- Brown, P., & Levinson, S. (1987). *Politeness: Some universals in language usage*. Cambridge: Cambridge University Press.
- Cai, Z., Graesser, A., Forsyth, C., Burkett, C., Millis, K., Wallace, P., Halpern, D., & Butler, H. (2011). Dialog in ARIES: User input assessment in an intelligent tutoring system. In W. Chen & S. Li (Eds.), *Proceedings of the 3rd IEEE International Conference on Intelligent Computing and Intelligent Systems*. Guangzhou, China: IEEE Press.
- Calvo, R., & D'Mello, S. (2011). *New perspectives on affect and learning technologies*. New York: Springer.
- Chi, M., Deleeuw, N., Chiu, M., & Lavancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive Science*, 18(3), 439-477.
- Chi, M., Siler, S., & Jeong, H. (2004). Can tutors monitor students' understanding accurately? *Cognition and Instruction*, 22(3), 363-387.
- Clark, H. (1996). *Using language*. Cambridge: Cambridge University Press.
- Conati, C., Gertner, A., & VanLehn, K. (2002). Using Bayesian networks to manage uncertainty in student modeling. *User Modeling and User-Adapted Interaction*, 12(4), 371-417.
- Corbett, A. T., & Anderson, J. R. (1994). Knowledge tracing - modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4), 253-278.

- Craig, S., Graesser, A., Sullins, J., & Gholson, J. (2004). Affect and learning: An exploratory look into the role of affect in learning. *Journal of Educational Media*, 29, 241-250. doi: 10.1080/1358165042000283101
- D'Mello, S., Craig, S., Witherspoon, A., McDaniel, B., & Graesser, A. (2008). Automatic detection of learner's affect from conversational cues. *User Modeling and User-Adapted Interaction*, 18(1-2), 45-80.
- D'Mello, S., Dowell, N., & Graesser, A. (2011). Does it really matter whether students' contributions are spoken versus typed in an intelligent tutoring system with natural language? *Journal of Experimental Psychology: Applied*, 17(1), 1-17.
- D'Mello, S., & Graesser, A. (2010). Multimodal semi-automated affect detection from conversational cues, gross body language, and facial features. *User Modeling and User-adapted Interaction*, 20(2), 147-187.
- D'Mello, S., & Graesser, A. (2012). Malleability of students' perceptions of an affect-sensitive tutor and its influence on learning. In G. Youngblood & P. McCarthy (Eds.), *Proceedings of 25th Florida Artificial Intelligence Research Society Conference* (pp. 432-437). Menlo Park, CA: AAAI Press.
- D'Mello, S., King, B., & Graesser, A. (2010a). Towards spoken human-computer tutorial dialogues. *Human-Computer Interaction*, 25(4), 289-323.
- D'Mello, S., Lehman, B., Sullins, J., Daigle, R., Combs, R., Vogt, K., Perkins, L., & Graesser, A. (2010b). A time for emoting: When affect-sensitivity is and isn't effective at promoting deep learning. In J. Kay & V. Aleven (Eds.), *Proceedings of the 10th International Conference on Intelligent Tutoring Systems* (pp. 245-254). Berlin / Heidelberg: Springer.
- D'Mello, S., Lehman, S., Pekrun, R., & Graesser, A. (in press). Confusion can be beneficial for learning. *Learning and Instruction*.

- D'Mello, S., Olney, A., & Person, N. (2010c). Mining collaborative patterns in tutorial dialogues. *Journal of Educational Data Mining*, 2(1), 1-37.
- D'Mello, S., Olney, A., Williams, C., & Hays, P. (2012). Gaze tutor: A gaze-reactive intelligent tutoring system. *International Journal of human-computer studies*, 70(5), 377-398.
- D'Mello, S. K., & Graesser, A. C. (in press). AutoTutor and Affective AutoTutor: Learning by talking with cognitively and emotionally intelligent computers that talk back. *ACM Transactions on Interactive Intelligent Systems*.
- D'Mello, S., & Graesser, A. (2011). The half-life of cognitive-affective states during complex learning. *Cognition & Emotion*, 25(7), 1299-1308.
- D'Mello, S., & Graesser, A. (in press). Language and discourse are powerful signals of student emotions during tutoring. *IEEE Transactions on Learning Technologies*.
- Dalgarno, B. (2001). Interpretations of constructivism and consequences for computer assisted learning. *British Journal of Educational Technology*, 32(2), 183-194.
- Evens, M. W., Chang, R. C., Lee, Y. H., Shim, L. S., Woo, C. W., Zhang, Y., Michael, J. A., & Rovick, A. A. (1997). CIRCSIM-Tutor: An intelligent tutoring system using natural language dialogue *Proceedings of the fifth conference on applied natural language processing* (pp. 13-14). Stroudsburg, PA: Association for Computational Linguistics.
- Festinger, L. (1957). *A theory of cognitive dissonance*. Stanford, CA: Stanford University Press.
- Fox, B. (1993). *The human tutorial dialogue project*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Freedman, R. (1996). *Interaction of discourse planning, instructional planning, and dialogue management in an interactive tutoring system*. (Doctoral Dissertation), Northwestern University, Evanston, IL.

- Gertner, A., & VanLehn, K. (2000). Andes: A coached problem solving environment for physics. In G. Gauthier, C. Frasson & K. VanLehn (Eds.), *Proceedings of the 5th International Conference on Intelligent Tutoring Systems* (pp. 133-142). Berlin / Heidelberg: Springer.
- Goldin-Meadow, S. (2003). *Hearing gesture: How our hands help us think*. Cambridge, MA: Harvard University Press,.
- Graesser, A., Chipman, P., Haynes, B., & Olney, A. (2005a). AutoTutor: An intelligent tutoring system with mixed-initiative dialogue. *IEEE Transactions on Education*, 48(4), 612-618. doi: 10.1109/TE.2005.856149
- Graesser, A., Chipman, P., King, B., McDaniel, B., & D'Mello, S. (2007). Emotions and learning with AutoTutor. In R. Luckin, K. Koedinger & J. Greer (Eds.), *Proceedings of the 13th International Conference on Artificial Intelligence in Education* (pp. 569-571). Amsterdam: IOS Press.
- Graesser, A., D'Mello, S., & Cade, W. (2009). Instruction based on tutoring. In R. E. Mayer & P. A. Alexander (Eds.), *Handbook of research on learning and instruction* (pp. 408-426). Routledge Press.: New York.
- Graesser, A., Lu, S. L., Jackson, G., Mitchell, H., Ventura, M., Olney, A., & Louwerse, M. (2004a). AutoTutor: A tutor with dialogue in natural language. *Behavioral Research Methods, Instruments, and Computers*, 36, 180-193.
- Graesser, A., McDaniel, B., Chipman, P., Witherspoon, A., D'Mello, S., & Gholson, B. (2006). Detection of emotions during learning with AutoTutor. In R. Sun & N. Miyake (Eds.), *Proceedings of the 28th Annual Conference of the Cognitive Science Society* (pp. 285-290). Austin, TX: Cognitive Science Society.

- Graesser, A., McNamara, D., Louwerse, M., & Cai, Z. (2004b). Coh-Metrix: Analysis of text on cohesion and language. *Behavior Research Methods, Instruments, & Computers*, *36*, 193-202.
- Graesser, A., Moreno, K., Marineau, J., Adcock, A., Olney, A., & Person, N. (2003). AutoTutor improves deep learning of computer literacy: Is it the dialog or the talking head? . In U. Hoppe, F. Verdejo & J. Kay (Eds.), *Proceedings of the 11th International Conference on Artificial Intelligence in Education* (pp. 47-54). Amsterdam: IOS Press.
- Graesser, A., Penumatsa, P., Ventura, M., Cai, Z., & Hu, X. (2007). Using LSA in AutoTutor: Learning through mixed-initiative dialogue in natural language. In T. Landauer, D. McNamara, S. Dennis & W. Kintsch (Eds.), *Handbook of Latent Semantic Analysis* (pp. 243-262). Mahwah, NJ: Erlbaum.
- Graesser, A., & Person, N. (1994). Question asking during tutoring. *American Education Research Journal*, *31*(1), 104-137.
- Graesser, A., Person, N., & Magliano, J. (1995). Collaborative dialogue patterns in naturalistic one-to-one tutoring. *Applied Cognitive Psychology*, *9*(6), 495-522.
- Graesser, A., VanLehn, K., Rose, C. P., Jordan, P. W., & Harter, D. (2001). Intelligent tutoring systems with conversational dialogue. *AI Magazine*, *22*(4), 39-51.
- Graesser, A., Wiemer-Hastings, K., Wiemer-Hastings, P., & Kreuz, R. (1999). AutoTutor: A simulation of a human tutor. *Cognitive Systems Research*, *1*(1), 35-51.
- Graesser, A., Wiemer-Hastings, P., Wiemer-Hastings, K., Harter, D., & Person, N. (2000). Using latent semantic analysis to evaluate the contributions of students in AutoTutor. *Interactive Learning Environments*, *8*(2), 129-147.

- Graesser, A. C., Conley, M. W., & Olney, A. M. (2012). Intelligent tutoring systems. In K. R. Harris, S. Graham & T. Urdan (Eds.), *The APA educational psychology handbook* (Vol. 3, pp. 451-473). Washington D.C: American Psychological Association.
- Graesser, A. C., D'Mello, S., & Person, N. (2009). Meta-Knowledge in Tutoring. In D. J. Hacker, J. Dunlosky & A. C. Graesser (Eds.), *Handbook of metacognition in education* (pp. 361-412). Mahwah, NJ: Erlbaum.
- Graesser, A. C., Hu, X., & McNamara, D. S. (2005b). 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, D.C.: American Psychological Association.
- Gratch, J., Rickel, J., André, E., Cassell, J., Petajan, E., & Badler, N. (2002). Creating interactive virtual humans: Some assembly required. *IEEE Intelligent Systems*, 17(4), 54-63.
- Hestenes, D., Wells, M., & Swackhamer, G. (1992). Force concept inventory. *The Physics Teacher*, 30, 141-158.
- Hu, X., Cai, Z., Han, L., Craig, S. D., Wang, T., & Graesser, A. C. (2009). AutoTutor Lite. In V. Dimitrova, R. Mizoguchi, B. du Boulay & A. Graesser (Eds.), *Proceedings of the 14th International Conference on Artificial Intelligence in Education* (pp. 802-802). Amsterdam: IOS Press.
- Hu, X., Cai, Z., Wiemer-Hastings, P., Graesser, A., & McNamara, D. (2007). Strengths, limitations, and extensions of LSA. In T. Landauer, D. McNamara, S. Dennis & W. Kintsch (Eds.), *Handbook of Latent Semantic Analysis* (pp. 401-426). Mahwah, NJ: Lawrence Erlbaum Associates.

- Jackson, G., & Graesser, A. (2006). Applications of human tutorial dialog in AutoTutor: An intelligent tutoring system. *Revista Signos*, 39(60), 31-48.
- Jackson, G. T., Dempsey, K. B., & McNamara, D. S. (in press). Game-based practice in reading strategy tutoring system: Showdown in iSTART-ME. In H. Reinders (Ed.), *Computer games* (pp. 59-68). Bristol, UK: Multilingual Matters.
- Jackson, G. T., & Graesser, A. C. (2007). Content matters: An investigation of feedback categories within an ITS. In R. Luckin, K. Koedinger & J. J. Green (Eds.), *Proceedings of the 13th International Conference on Artificial Intelligence in Education* (pp. 127–134). Amsterdam: IOS Press.
- Johnson, W., Rickel, J., & Lester, J. (2000). Animated pedagogical agents: Face-to-face interaction in interactive learning environments. *International Journal of Artificial Intelligence in Education*, 11, 47-78.
- Johnson, W., & Valente, L. (2008, July). *Tactical language and culture training systems: Using artificial intelligence to teach foreign languages and cultures*. Paper presented at the Proceedings of the 20th National Artificial Intelligence Conference, Menlo Park, CA.
- Landauer, T., McNamara, D., Dennis, S., & Kintsch, W. (Eds.). (2007). *Handbook of Latent Semantic Analysis*. Mahwah, NJ: Erlbaum.
- Larson, R. W., & Richards, M. H. (1991). Boredom in the middle school years - Blaming schools versus blaming students. *American Journal of Education*, 99(4), 418-443.
- Lepper, M., & Woolverton, M. (2002). The wisdom of practice: Lessons learned from the study of highly effective tutors. In J. Aronson (Ed.), *Improving academic achievement: Impact of psychological factors on education* (pp. 135-158). Orlando, FL: Academic Press.

- Limón, M. (2001). On the cognitive conflict as an instructional strategy for conceptual change: a critical appraisal. *Learning and Instruction, 11*(4-5), 357-380. doi: 10.1016/s0959-4752(00)00037-2
- Litman, D., Rose, C., Forbes-Riley, K., VanLehn, K., Bhembe, D., & Silliman, S. (2006). Spoken versus typed human and computer dialogue tutoring. *International Journal of Artificial Intelligence In Education, 16*(2), 145-170.
- McLaren, B. M., DeLeeuw, K. E., & Mayer, R. E. (2011). A politeness effect in learning with web-based intelligent tutors. *International Journal of Human-Computer Studies, 69*(1-2), 70-79. doi: 10.1016/j.ijhcs.2010.09.001
- Millis, K., Forsyth, C., Butler, H., Wallace, P., Graesser, A., & Halpern, D. (2011). Operation ARIES! A serious game for teaching scientific inquiry. In M. Ma, A. Oikonomou & J. Lakhmi (Eds.), *Serious games and edutainment applications* (pp. 169-196). London, UK: Springer-Verlag.
- Moreno, R., & Mayer, R. (2007). Interactive multimodal learning environments. *Educational Psychology Review, 19*(3), 309-326. doi: 10.1007/s10648-007-9047-2
- Moshman, D. (1982). Exogenous, endogenous, and dialectical constructivism. *Developmental Review, 2*(4), 371-384.
- Mostow, J., & Aist, G. (2001). Evaluating tutors that listen: An overview of project LISTEN. In K. Forbus & P. Feltovich (Eds.), *Smart machines in education: The coming revolution in educational technology* (pp. 169-234). Cambridge, MA: MIT Press.
- Olney, A., D'Mello, A., Person, N., Cade, W., Hays, P., Williams, C., Lehman, B., & Graesser, A. (2012). Guru: A computer tutor that models expert human tutors. In S. Cerri, W. Clancey, G. Papadourakis & K. Panourgia (Eds.), *Proceedings of the 11th International*

- Conference on Intelligent Tutoring Systems* (pp. 256-261). Berlin/Heidelberg: Springer-Verlag.
- Olney, A., Louwerse, M., Mathews, E., Marineau, J., Hite-Mitchell, H., & Graesser, A. (2003). *Utterance classification in AutoTutor*. Paper presented at the Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, Edmonton, Canada.
- Pekrun, R., Goetz, T., Daniels, L., Stupnisky, R. H., & Perry, R. (2010). Boredom in achievement settings: Exploring control–value antecedents and performance outcomes of a neglected emotion. *Journal of Educational Psychology, 102*(3), 531-549. doi: 10.1037/a0019243
- Pekrun, R., & Stephens, E. J. (2012). Academic emotions. In K. Harris, S. Graham, T. Urdan, S. Graham, J. Royer & M. Zeidner (Eds.), *APA educational psychology handbook, Vol 2: Individual differences and cultural and contextual factors* (pp. 3-31). Washington, DC: American Psychological Association.
- Pennebaker, J., Chung, C., Ireland, M., Gonzales, A., & Booth, R. (2007). The development and psychometric properties of LIWC2007. Austin, Texas: The University of Texas at Austin and The University of Auckland, New Zealand.
- Person, N., & Graesser, A. (2002). Human or computer? AutoTutor in a Bystander Turing Test. In S. Cerri, G. G. & P. F. (Eds.), *Proceedings of the 6th International Conference on Intelligent Tutoring Systems* (pp. 821-830). Berlin / Heidelberg: Springer.
- Piaget, J. (1952). *The origins of intelligence*. New York: International University Press.
- Pon-Barry, H., Clark, B., Schultz, K., Bratt, E. O., & Peters, S. (2004). Advantages of spoken language interaction in dialogue-based intelligent tutoring systems. In J. Lester, R. Vicari

- & F. Paraguacu (Eds.), *Proceedings of Seventh International Conference on Intelligent Tutoring Systems* (pp. 390-400). Berlin / Heidelberg: Springer.
- Potka, J., Massey, D., & Mutter, S. (1988). *Intelligent tutoring systems: Lessons learned*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Rus, V., & Graesser, A. (2007). Lexico-syntactic subsumption for textual entailment. In N. Nicolov, K. Bontcheva, G. Angelova & R. Mitkov (Eds.), *Recent Advances in Natural-language Processing IV: Selected Papers from RANLP 2005* (pp. 187-196). Amsterdam: John Benjamins Publishing Company.
- Sacks, H., Schegloff, E. A., & Jefferson, G. (1974). A simplest systematics for the organization of turn-taking for conversation. *Language*, 50(4), 696-735.
- Schultz, K., Bratt, E., Clark, B., Peters, S., Pon-Barry, H., & Treeratpituk, P. (2003). *A scalable, reusable spoke conversational tutor: Scot*. Paper presented at the Workshop on Tutorial Dialogue Systems: With a View toward the Classroom. In conjunction with the 11th International Conference on Artificial Intelligence in Education, Sydney, Australia.
- Sleeman, D., & Brown, J. (Eds.). (1982). *Intelligent tutoring systems*. New York: Academic Press.
- VanLehn, K. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education*, 16(3), 227-265.
- VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4), 197-221.
- VanLehn, K., Graesser, A., Jackson, G., Jordan, P., Olney, A., & Rose, C. P. (2007). When are tutorial dialogues more effective than reading? *Cognitive Science*, 31(1), 3-62.
- VanLehn, K., Jordan, P., Rose, C., Bhembe, D., Bottner, M., & A., G. (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 System* (pp. 158-167). Berlin: Springer-Verlag.
- VanLehn, K., Siler, S., Murray, C., Yamauchi, T., & Baggett, W. (2003). Why do only some events cause learning during human tutoring? *Cognition and Instruction, 21*(3), 209-249. doi: 10.1207/S1532690XCI2103_01
- Vygotsky, L. (1978). *Mind in society: The development of higher psychological processes*. Cambridge, MA: Harvard University Press.
- Vygotsky, L. (1986). *Thought and Language*. Cambridge, MA: MIT Press.
- Wang, N., Johnson, W. L., Mayer, R. E., Rizzo, P., Shaw, E., & Collins, H. (2008). The politeness effect: Pedagogical agents and learning outcomes. *International Journal of Human-Computer Studies, 66*(2), 98-112. doi: 10.1016/j.ijhcs.2007.09.003
- Ward, W., Cole, R., Bolaños, D., Buchenroth-Martin, C., Svirsky, E., Vuuren, S. V., Weston, T., Zheng, J., & Becker, L. (2011). My science tutor: A conversational multimedia virtual tutor for elementary school science. *ACM Transactions on Speech and Language Processing, 7*(4), 18.
- Woolf, B. (2009). *Building intelligent interactive tutors*. Burlington, MA: Morgan Kaufmann Publishers.

Table 1. Excerpt of tutorial dialog between a learner and AutoTutor

Speaker	Utterance
AutoTutor	When you turn on the computer, how is the operating system first activated and loaded into RAM? <i>[this is the main question]</i>
Learner	When the computer is turned on a file is automatically booted up.
AutoTutor	Anything else? <i>[this is a pump for more information]</i>
Learner	This file contains the operating system.
AutoTutor	What about storage? <i>[this is a hint]</i>
Learner	The operating system is then read into RAM and activated.
AutoTutor	Right. The CPU executes instructions in ROM, which helps the operating system to start. <i>[after giving positive feedback, there is an assertion]</i>
Learner	After being read from ROM the OS is activated.
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. <i>[after positive feedback comes a summary]</i>

Figure 1. Sketch of six major components of a dialog-based ITS and how they might interact

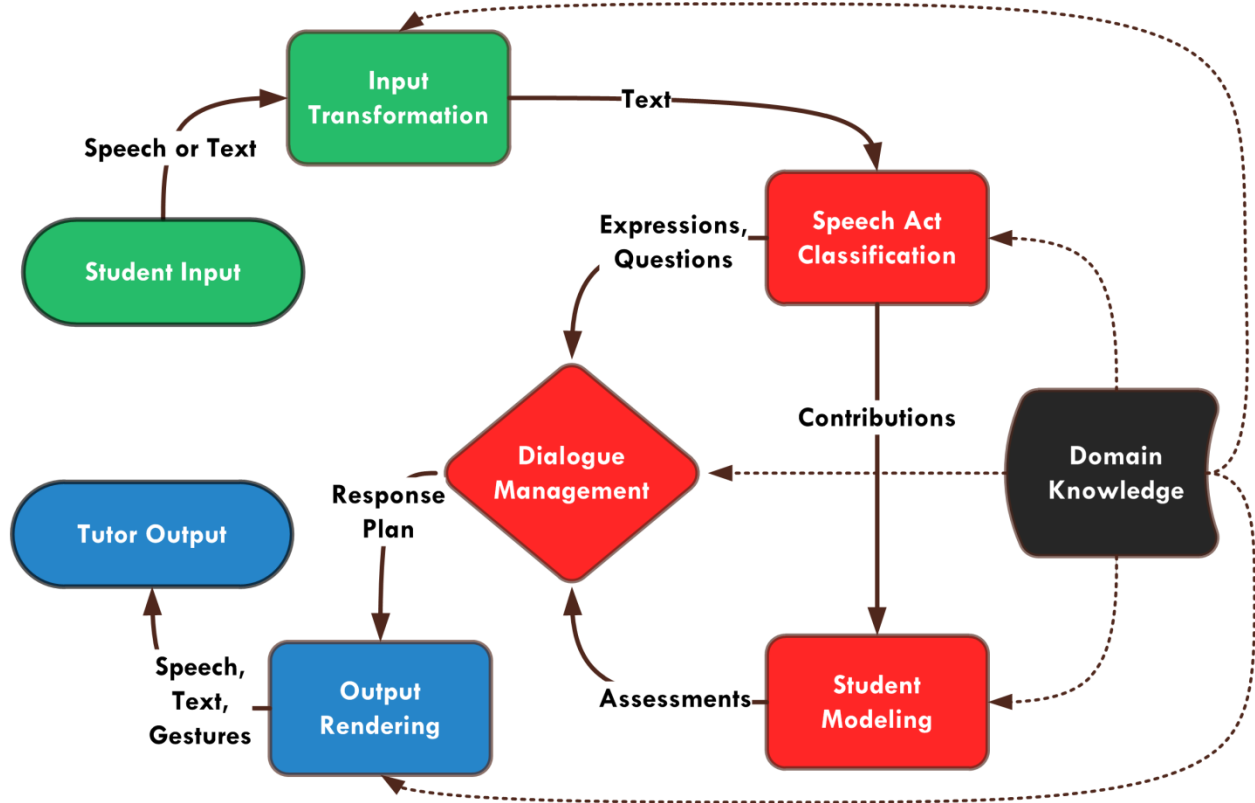


Figure 2. Sample Dialog Advancer Network

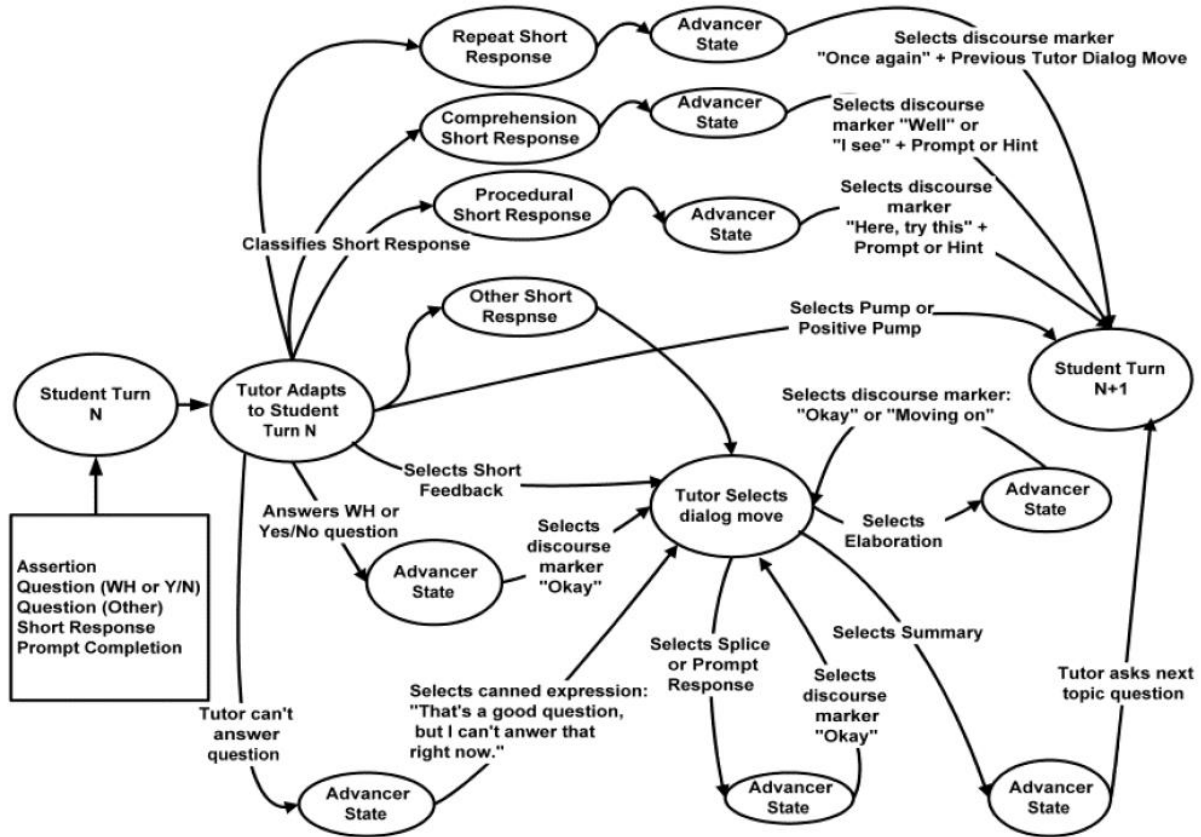


Figure 3. Screenshot of AutoTutor interface

The screenshot displays the AutoTutor interface. At the top, a blue title bar reads "AutoTutor" with a menu bar containing "File", "Edit", "Session", "Plugins", and "Help". Below the menu bar, a question is posed: "How does the operating system interact with the word processing program when you create a document?".

On the left side, a 3D-rendered female avatar with short dark hair and a blue blazer looks towards the user. On the right side, a diagram of a computer system is shown. It includes a "HARD DISK" at the top, a "CPU" at the bottom, and a "RAM" section on the right. Inside the RAM section, there is a "WORD PROCESSING PROGRAM" and a "DOCUMENT". Arrows indicate the flow of data between the hard disk, the OS, the word processing program, and the document.

At the bottom of the interface, there is a "Log of previous responses:" section on the left and an "Enter your response here:" section on the right. The log contains the following text:

Student: the operating system allows you to save new information on a document

Tutor: I see, ok.
Tutor: Can you elaborate a bit on that?

Student: yes, the operating system creates space to save the document so that it is not lost when you open another program

The response area on the right contains the text: "yes, the operating system creates space to save the document so not lost when you open another program".