Chapter 10

AutoTutor

Arthur C. Graesser
The University of Memphis, USA

Sidney D’Mello
The University of Memphis, USA

Xiangen Hu
The University of Memphis, USA

Zhiqiang Cai
The University of Memphis, USA

Andrew Olney
The University of Memphis, USA

Brent Morgan
The University of Memphis, USA

ABSTRACT

AutoTutor is an intelligent tutoring system that helps students learn science, technology, and other technical subject matters by holding conversations with the student in natural language. AutoTutor’s dialogues are organized around difficult questions and problems that require reasoning and explanations in the answers. The major components of AutoTutor include an animated conversational agent, dialogue management, speech act classification, a curriculum script, semantic evaluation of student contributions, and electronic documents (e.g., textbook and glossary). This chapter describes the computational components of AutoTutor, the similarity of these components to human tutors, and some challenges in handling smooth dialogue. We describe some ways that AutoTutor has been evaluated with respect to learning gains, conversation quality, and learner impressions. AutoTutor is sufficiently modular that the content and dialogue mechanisms can be modified with authoring tools. AutoTutor has spawned a number of other agent-based learning environments, such as AutoTutor-lite, Operation Aries!, and Guru.

INTRODUCTION AND BACKGROUND

Intelligent Tutoring Systems (ITS) are computerized learning environments that incorporate computational models in the cognitive sciences, learning sciences, artificial intelligence, computational linguistics, and other fields that develop intelligent systems (Sleeman & Brown, 1982; Woolf, 2009). In a process called student modelling, the ITS tracks the psychological states of learners, such as subject matter knowledge, cognitive skills, strategies, motivation, and emotions. An ITS adaptively responds with activities that are sensitive to these psychological states, the history of the student-tutor interaction, and the instructional agenda. An ITS is very different from more rigid, insensitive, and inflexible learning environments such as reading a book or listening to a lecture.

DOI: 10.4018/978-1-60960-741-8.ch010
ITS environments were originally developed for mathematically well-formed subject matters. Impressive systems have been developed and tested for algebra, geometry, and programming languages (the Cognitive Tutors: Anderson et al., 1995; Koedinger et al., 1997; Ritter et al., 2007, ALEKS: Doignon & Falmagne, 1999), for physics (Andes, Atlas: VanLehn et al., 2002), for electronics (SHERLOCK: Lesgold, Lajoie, Bunzo, & Eggen, 1992), and for information technology (KERMIT: Mitrovic, Martin, & Suraweera, 2007). More recently the ITS enterprise has evolved to handle conversational interaction in natural language on verbal topics that require conceptual reasoning. This chapter focuses on AutoTutor (Graesser, Lu et al., 2004), but other systems have been developed with similar goals: ITSPOKE (Litman et al., 2006), Spoken Conversational Computer (Pon-Barry, Clark, Schultz, Bratt, Peters, & Haley, 2005), Tactical Language and Culture Training System (Johnson & Valente, 2008), Why-Atlas (VanLehn et al., 2007), and iSTART (McNamara, Levinstein, & Boonthum, 2004). These systems automatically analyze language and discourse by incorporating recent advances in computational linguistics (Jurafsky & Martin, 2008) and statistical representations of world knowledge (Landauer, McNamara, Dennis, & Kintsch, 2007).

Most ITSs fit within VanLehn’s (2006) analyses of the outer loop and the inner loop when characterizing the scaffolding of solutions to problems, answers to questions, or completion of complex tasks. The outer loop involves the selection of topics and problems to cover, assessments of the student’s topic knowledge and general cognitive abilities, and global aspects of the tutorial interaction. The inner loop consists of covering individual steps within a problem at a micro-level. Adaptivity and intelligence are necessary at both the outer loop and the inner loop in a bona fide ITS.

This chapter describes the computational components of AutoTutor and some of the challenges faced when simulating smooth and pedagogically effective dialogue. AutoTutor’s architecture incorporates dialogue mechanisms of human tutors in addition to ideal tutoring strategies. We describe evaluations of AutoTutor with respect to learning gains, conversation quality, and learner impressions. The modular architecture of AutoTutor allows developers to develop new content and dialogue strategies with authoring tools. We end the chapter by identifying some of AutoTutor’s progeny that also have conversational agents, such as AutoTutor-lite, Guru, and Operation Aries!.

**AUTOTUTOR MECHANISMS**

AutoTutor simulates a tutor by holding a conversation in natural language (Graesser, Chipman, Haynes, & Olney, 2005; Graesser, Jeon, & Dufty, 2008; Graesser, Graesser, Lu et al., 2004; Graesser, Person, & Harter, 2001). Students type in their contributions through a keyboard in most applications. However, we have developed a version that handles spoken input from the student through the Dragon Naturally Speaking™ (version 6) speech recognition system (D’Mello, King, Chipman, & Graesser, in press). AutoTutor communicates through an animated conversational agent with speech, facial expressions, and some rudimentary gestures.

Figure 1 shows a screen shot of AutoTutor on the topic of computer literacy. Most versions of AutoTutor have the three major areas shown in Figure 1. Area 1 (top of screen) is the main question (or problem) that stays on the computer screen throughout the conversation that collaboratively constructs an answer to the question. Area 2 (left middle) is the animated conversational agent that speaks the content of AutoTutor’s turns. Area 3 (right middle) is either blank or has auxiliary diagrams on the subject matter. When the students type in their contributions, there is an area at the bottom that displays what the student types in. In versions with speech recognition, there are two buttons on the keyboard that the learner presses.