Achieving Domain Independence in AutoTutor

Eric C. Mathews, G. Tanner Jackson, Andrew Olney, Patrick Chipman & Arthur C. Graesser

Institute for Intelligent Systems, University of Memphis
Memphis, TN 38152
{emathews, gtjacksn, aolney, pchipman, a-graesser}@memphis.edu

ABSTRACT

AutoTutor is a computer tutor system that simulates the discourse patterns and pedagogical strategies of human tutors. AutoTutor engages the learner in a natural language conversation on a topic, presenting questions, posing problems, answering questions, providing information and other tutoring dialog moves. AutoTutor is presently designed to assist college students in learning the fundamentals of two domains (computer literacy and conceptual physics). The system architecture allows it to maintain domain independence.

Keywords: Interactive systems, artificial intelligence, intelligent tutoring systems

INTRODUCTION

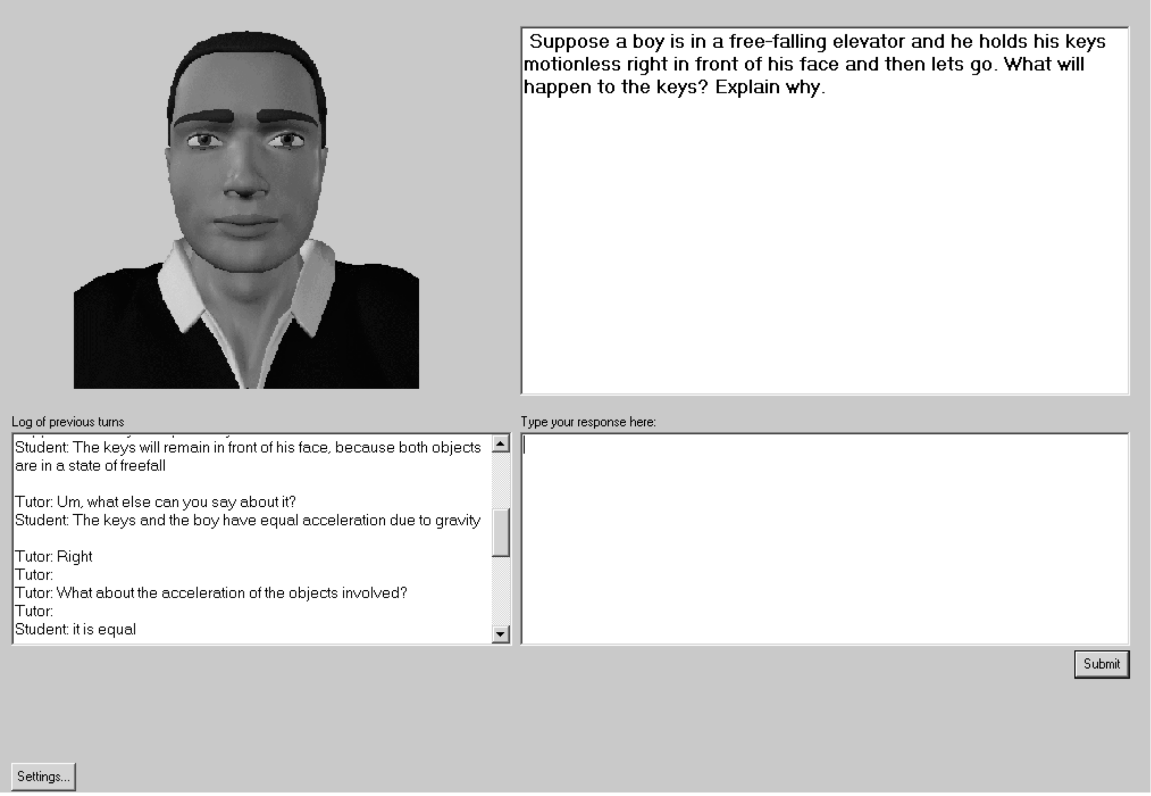
In 1996, the Tutoring Research Group (TRG) at the University of Memphis began to build the first intelligent tutoring system for the domain of Computer Literacy. The goals from the outset of the project were to develop a computer tutor that would simulate human tutors [3], [5], [6], [18]. To this end, AutoTutor has been developed and continues to evolve simulating the dialog moves of human tutors, all the while still maintaining a general system architecture that allows it to be domain independent.

SYSTEM OVERVIEW

AutoTutor works by having a conversation with the learner, and is visually manifested as a talking head that is a dialog partner for the student, as is seen in figure 1. AutoTutor and the student work together on properly answering the main question or problem which appears in the upper right hand box (see figure 1). Students respond by typing their answers into the text entry box on the lower right-hand side of the screen, and a dialog history box is presented underneath the talking head, on the lower left, for review as needed.

After the student provides an initial answer to a problem or main question, AutoTutor and the student engage in a multi-turn mixed-initiative dialog to collaboratively construct the best possible answer to the question or problem presented. AutoTutor questions, provides feedback, delivers hints, prompts the student for specific information, gives missing information, and answers student questions, which all model the dialog moves of human tutors. These conversations on a particular problem typically last in excess of 20-30 turns and a tutoring session consists of the coverage of 5-12 problems or questions.

AutoTutor is designed to potentially tutor on a variety of topics (presently the system handles computer literacy and conceptual physics at the college level). AutoTutor resides on a Pentium-based server platform, delivered across the web so it may reach the widest audience possible. The system is comprised of many modules and utilities as will be described below.



Suppose a boy is in a free-falling elevator and he holds his keys motionless right in front of his face and then lets go. What will happen to the keys? Explain why.

Log of previous turns

Student: The keys will remain in front of his face, because both objects are in a state of freefall

Tutor: Um, what else can you say about it?
Student: The keys and the boy have equal acceleration due to gravity

Tutor: Right
Tutor:
Tutor: What about the acceleration of the objects involved?
Tutor:
Student: it is equal

Figure 1. AutoTutor Interface

MODULES AND UTILITIES OF AUTOTUTOR

To facilitate all of the aforementioned goals, AutoTutor is comprised of 9 major components: (1) avatar/client module, (2) assessment module, (3) a speech act classification module (SAC), (4) Latent Semantic Analysis (LSA) utility, (5) a dialog management module, (6) a question answering utility (QUEST), (7) a curriculum script utility, (8) parser utility, and (9) a log utility. Many of these components have been discussed in previous publications; and therefore, will be mentioned briefly in this paper [1], [3], [7], [8], [11], [12], [15], [18], [19].

Avatar/Client Module

AutoTutor uses a three-dimensional animated agent that remains on the screen during the entire tutoring session, to deliver the tutoring dialog moves, gestures, and proper emotions. It was created using Curious Labs Poser 4 and is controlled by Microsoft Agent software. Dialog moves used by Why/AutoTutor during the tutoring session are synchronized with the agent's head movements, facial expressions, and hand gestures and serve both conversational and pedagogical functions [13]. The animated agent is embedded in a thin client that is delivered via web browser to the user of the system. This design allows all of the system architecture and computation demands to reside on remote servers allowing AutoTutor to reach a wider audience. Tutoring session data transfer can occur over a 56K modem connection in real time.

Speech Act Classifier Module and Parser Utility

When students interact with AutoTutor, their input is constrained to typing their conversational turns into a text entry box on the interface and submitted by hitting the "Enter" key. Upon submission AutoTutor analyzes for student initiative. That is to say, AutoTutor attempts to determine whether the student was providing an answer, asking a question, wanting the tutor to repeat, or expressing a lack of understanding. The Speech Act Classifier module (SAC) of AutoTutor does this using a sentence parser (Sekine's Apple Pie Parser [17]) utility and a series of rules and transformations that decide into which category the student's contribution falls. The SAC assigns the student's input into one of 23 speech act categories, including assertions, meta-cognitive statements, meta-communicative statements, and 17 question categories. These speech act categories enable AutoTutor to sustain mixed-initiative dialog as well as dictate how it will generate pedagogically effective and conversationally appropriate dialog moves.

Curriculum Script Utility

AutoTutor begins each tutoring session by introducing the functionality of the system, and then begins tutoring the learner using material found within the curriculum script utility. The curriculum script utility is a database of organized concepts, misconceptions, good answers, bad answers, teaching points, and question-answer units of various types. The curriculum script utility serves to organize the topics to be covered during the tutoring session. This content takes on the form of word phrases, sentences, questions, and paragraphs in a structured database, and is one of the few domain specific components.

Assessment Module and Latent Semantic Analysis Utility

AutoTutor's assessment module, using the Latent Semantic Analysis (LSA) utility, assesses student contributions and decides which portions of the ideal answer for the problem are missing from the student's contributions. In AutoTutor, we compare material found within the curriculum script utility to the student's contributions using LSA, a statistical technique that measures the conceptual similarity of two text sources [1], [8], [9]. We have found LSA to be accurate in assessing student turns, after it is pre-trained on a set of information for the domain on which the student is being tutored [5], [19].

Question Answering Utility (QUEST)

In order to answer a variety of student questions, the question answering utility of AutoTutor (QUEST) engages the speech act classifier module, the parser utility, and relevant query concepts. By doing so, it returns a relevant answer derived from an electronic textbook that is entered into the utility. QUEST can deal with a variety of questions, including deep comprehension questions, and depending on what the student asks, it can answer in terms of definitions, examples, causes, effects, and opinions about phenomena and concepts.

Dialog Management Module

To deliver pedagogically correct and conversationally appropriate dialog moves, AutoTutor uses a complex set of production rules to select which dialog pathways it should use [14]. The dialog management module, using these production rules and previously gathered information, determines the proper dialog move to continue the conversation in the appropriate tutorial manner.

When a student contribution is first received, AutoTutor determines student initiative using the SAC module and the context of the previous AutoTutor turn. AutoTutor then assesses the student's contribution quality in relation to the curriculum script as needed. The module adapts to student initiative and transitions to the next dialog pathway. This pathway may either continue with the tutoring lesson plan, as proscribed by the production rules and curriculum script, or respond to the student's initiative by answering the student's question.

Log Utility

All the information generated and gathered by the system and the student user is written to a database by the log utility so that analyses of the tutoring sessions can be analyzed later on in detail.

DOMAIN INDEPENDENCE

Many intelligent tutoring systems have been hand crafted and designed only to handle one domain. From the beginning AutoTutor has been kept modular in nature in order to potentially accommodate almost any topic for tutoring. This has largely been motivated by the fact that human tutors make use of similar or universal dialog moves that appear to be domain independent [2], [4], [10], [16]. The only difference that arises across these tutoring sessions is the content of those dialog moves. To make this concrete I provide an example below:

1) What can you say about the forces acting on the object?
2) Could you tell me more about the interaction between the neutron and the proton?

The two examples superficially appear to be different, and at the content level they are. Example 1 is in the physics domain and example 2 is from chemistry. If one examines their function in a tutoring session, we would find that they are equivalent in nature. Both hint at information that the user is missing in their answer to a problem or question. So while content maybe different the mechanics of tutoring adhere to universal principles and thus these principles are modeled in the architecture of AutoTutor.

As would be expected, the tutoring components of AutoTutor are domain independent and consist of the avatar/client module, assessment module, speech act classification module, dialog management module, parser utility, and the log utility. The other three components require domain content in order to function. The question-answering module pulls answers from its bank of information and thus needs domain specific content. The question-answering utility requires an electronic text of information on the topic, preferably in the form of an electronic textbook that has the content on which the student will be tutored. The curriculum script utility is a nontrivial component of the system, for it contains the organized, domain specific lesson plan that AutoTutor will draw from in order to tutor the student. All of the materials in the curriculum script utility thus need to be changed when AutoTutor tutors in another topic area. The Latent Semantic Analysis utility, while domain independent in nature, requires a body of domain information in electronic text form to pre train the system in advance of the tutoring session. This domain content is crucial so AutoTutor can properly assess answers on their quality.

CONCLUSIONS AND FUTURE WORK

It is in this manner that AutoTutor has maintained domain independence, allowing it to be changed in a relatively short amount of time across domains. Current work is being executed to build authoring tools to scaffold novice users in switching AutoTutor over to a new domain. This would ultimately allow a teacher to set up automated intelligent tutoring for any topic needed for the student, thus allowing AutoTutor to touch an even wider audience

ACKNOWLEDGEMENTS

The Tutoring Research Group (TRG) is an interdisciplinary research team comprised of approximately 35 researchers from psychology, computer science, physics, and education (See http://www.autotutor.org). This research conducted by the authors and the TRG was supported by grants from the National Science Foundation (SBR 9720314 and REC 0106965) and the Department of Defense Multidisciplinary University Research Initiative (MURI) administered by the Office of Naval Research under grant N00014-00-1-0600. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of ONR or NSF.

REFERENCES

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

[2] Fox, B. (1993). *The Human Tutorial Dialog Project*. Hillsdale, NJ: Erlbaum.

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

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

[5] Graesser, A.C., VanLehn, K., Rose, C., Jordan, P., and Harter, D. (2001). Intelligent tutoring systems with conversational dialogue. *AI Magazine* 22, 39-51.

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

[7] Graesser, A.C., Wiemer-Hastings, P., Wiemer-Hastings, K., Harter, D., Person, N., and Tutoring Research Group (2000). Using latent semantic analysis to evaluate the contributions of students in AutoTutor. *Interactive Learning Environments*, 8, 129-148.

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

[9] Landauer, T.K., Foltz, P.W., and Laham, D. (1998). An introduction to latent semantic analysis. *DiscourseProcesses*, 25, 259-284.

[10] Lepper, M. R., Woolverton, M., Mumme, D.L., and Gurtner, J.L. (1991). Motivational techniques of expert human tutors: Lessons for the design of computer-based tutors. In S.P. Lajoie and S.J. Derry (Eds.), *Computers as Cognitive Tools* (pp. 75-105). Hillsdale, NJ: Erblaum.

[11] Mathews, E.C., Jackson, G.T., Person, N.K., Graesser, A.C.. (2003). Discourse Patterns in Why/AutoTutor. *Proceedings of the 2003 AAAI Spring Symposia on Natural Language Generation in Spoken and Written Dialogue.* (pp.97-103). Palo Alto, CA: AAAI Press.

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

[13] Person, N.K., Craig, S., Price, P., Hu, X., Gholson, B., Graesser, A.C., & TRG (2000). Incorporating human-like conversational behaviors in AutoTutor. Proceedings of the Agents 2000 Conference.

[14] Person, N. K., Graesser, A. C., Harter, D., Mathews, E. C., & the Tutoring Research Group. (2000). Dialog move generation and conversation management in AutoTutor. Proceedings of the AAAI Fall Symposium: Building Dialogue Systems for Tutorial Applications (pp.45-51). Falmouth, MA: AAAI Press.

[15] Person, N.K., Graesser, A.C., Kreuz, R.J., Pomeroy, V., and TRG (2000). Simulating human tutor dialog moves in AutoTutor. *International Journal of Artificial Intelligence in Education*.

[16] Person, N. K., Graesser, A. C., Magliano, J. P., and Kreuz, R. J. (1994). Inferring what the student knows in one-to-one tutoring: The role of student questions and answers. *Learning and Individual Differences,* 6, 205-29.

[17] Sekine, S. (1997) Apple Pie Parser: a public domain system, version 5.9.

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

[19] Wiemer-Hastings, P., Wiemer-Hastings, K., and Graesser, A. (1999). Improving an intelligent tutor's comprehension of students with Latent Semantic Analysis. In S.P. Lajoie and M. Vivet, *Artificial Intelligence in Education* (pp. 535-542). Amsterdam: IOS Press.