RMT: A Dialog-Based Research Methods Tutor with or without a Head

Peter Wiemer-Hastings[1], David Allbritton[2], and Elizabeth Arnott[2]

[1] School of Computer Science, Telecommunications, and Information Systems
peterwh@cs.depaul.edu
[2] Department of Psychology {dallbrit|earnott}@depaul.edu
DePaul University
243 South Wabash Avenue
Chicago, Illinois, 60604, USA

Abstract. RMT (Research Methods Tutor) is a dialog-based tutoring system that has a dual role. Its modular architecture enables the interchange and evaluation of different tools and techniques for improving tutoring. In addition to its research goals, the system is intended to be integrated as a regular component of a term-long Research Methods in Psychology course. Despite the significant technical challenges, this may help reduce our knowledge gap about how such systems can help students with long-term use. In this paper, we describe the RMT architecture and give the results of an initial experiment that compared RMT's animated agent "talking head" with a text-only version of the system.

1 Introduction

Research on human to human tutoring has identified one primary factor that influences learning: the cooperative solving of example problems [1]. Typically, a tutor poses a problem (selected from a relatively small set of problems that they frequently use), and gives it to the student. The student attempts to solve the problem, one piece at a time. The tutor gives feedback, but rarely gives direct negative feedback. The tutor uses pumps (e.g. "Go on."), hints, and prompts (e.g. "The groups would be chosen ...") to keep the interaction going. The student and tutor incrementally piece together a solution for the problem. Then the tutor often offers a summary of the final solution [1]. This model of tutoring has been adopted by a number of recent dialog-based intelligent tutoring systems.

Understanding natural language student responses has been a major challenge for ITS's. Approaches have ranged from encouraging one-word answers [2] to full syntactic and semantic analysis of the responses [3–5]. Unfortunately, it can take man-years of effort to develop the specialized lexical, syntactic, and conceptual knowledge to make such language-analysis successful which limits how far these approaches can spread.

The AutoTutor system took a different approach to the natural language processing problem. AutoTutor uses a mechanism called Latent Semantic Analysis (LSA, described more completely below) which is automatically derived from

a large corpus of texts, and which gives an approximate but useful similarity metric between any two texts [6]. Student answers are evaluated by comparing them to a set of expected answers with LSA. This greatly reduces the knowledge acquisition bottleneck for tutoring systems. AutoTutor's tutoring style is modeled on human tutors. It maintains only a simple model of the student, and uses the same dialog moves mentioned above (prompts and pumps, for example) to do constructive, collaborative problem solving with the student. AutoTutor has been shown to produce learning gains of approximately one standard deviation unit compared to reading a textbook [7], been ported to a number of domains, and has been integrated with another tutoring system: Why/AutoTutor [7].

This paper describes RMT (Research Methods Tutor) which is a descendant of the AutoTutor system. RMT uses the same basic tutoring style that AutoTutor does, but was developed with a modular architecture to facilitate the study of different tools and techniques for dialog-based tutoring. One primary goal of the project is to create a system which can be integrated into the Research Methods in Psychology classes at DePaul University (and potentially elsewhere). We describe here the basic architecture of RMT, our first attempts to integrate it with the courses, and the results of an experiment that compares the use of an animated agent with text-only tutoring.

2 RMT Architecture

As mentioned above, RMT is a close descendant of the AutoTutor system. While AutoTutor incorporates a wide variety of artificial intelligence techniques, RMT was designed as a lightweight, modular system that would incorporate only those techniques required to provide educationally beneficial tutoring to the student. This section gives a brief description of RMT's critical components.

2.1 Dialog Manager

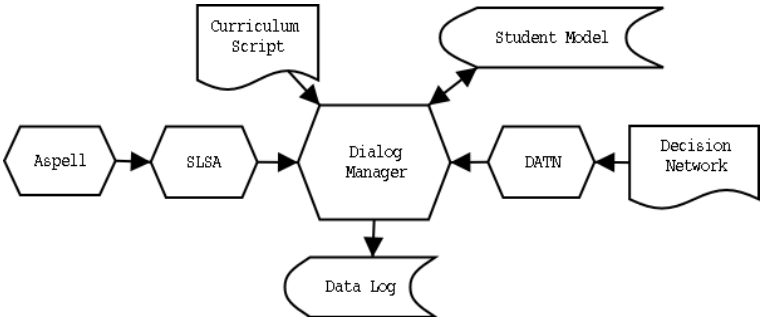

Fig. 1. RMT Architecture

As shown in figure 1, the dialog manager (DM) is the central controller of the system. Because RMT is a web-based system, each tutoring session has its own dialog manager, and the DM maintains information about the parameters of the tutoring session and the current state of the dialog. The DM reads student responses as posts from a web page, and then asks the Dialog Advancer Transition Network (DATN) to compute an appropriate tutor response.

Each tutor "turn" can perform three different functions: evaluate the student's previous utterance (e.g. "Good!"), confirm or add some additional information (e.g. "The dependent variable is test score."), and produce an utterance that keeps the dialog moving. Like AutoTutor, RMT uses pumps, prompts, and hints to try to get the student to add information about the current topic. RMT also asks questions, summarizes topics, and answers questions.

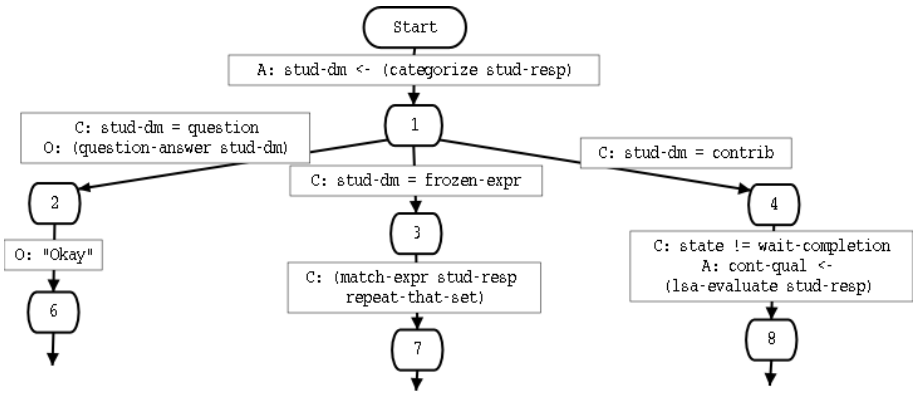

Fig. 2. A partial decision network

The DATN determines which type of response the tutor will give using a decision network which graphically depicts the conditions, actions and system outputs. Figure 2 shows a segment of RMT's decision network. For every tutor turn, the DATN begins processing at the Start state. The paths through the network eventually join back up at the Finish state, not shown here. On the arcs, the items marked C are the conditions for that arc to be chosen. The items labeled A are actions that will be performed. For example, on the arc from the start state, the DATN categorizes the student response. The items marked O are outputs — what the tutor will say next. Because this graph-based representation controls utterance selection, the tutor's behavior can be modified by simply modifying the graph.

2.2 Understanding Student Contributions

RMT uses Latent Semantic Analysis (LSA) to evaluate student contributions. LSA was first developed for information retrieval — selecting query-relevant

texts from a database. LSA has also been shown to perform well at finding synonyms, suggesting appropriate texts for students to read, and even grading student essays [8]. AutoTutor was the first system to use LSA to "understand" student responses in an interactive tutoring system [6], and it has subsequently been incorporated or evaluated for use by several other systems [3, 2, for example].

LSA evaluates a student response by comparing it to a set of expected answers. This works well in the tutoring setting because the tutor asks most of the questions and knows what types of answers (good and bad) the student is likely to produce. Due to space constraints, a complete description of LSA in a tutoring task is not included here. For more detail, please see [6]. One current research direction in the RMT project is to explore different applications of LSA, including segmenting input sentences into subject, verb, and object parts and comparing each separately.

2.3 Additional Functionality

Logging For data collection purposes, RMT borrows a piece of wisdom from a very successful reading tutor called Project LISTEN, "Log everything" [9]. As it interacts with a student, RMT stores information about each interaction in a database. The database collects and relates the individual utterances and a variety of other variables, for example, the type and quality of a student response. The database also contains information about the students and the tutoring conditions that they are assigned to. Thus, in addition to providing data for the experiments described below, we will be able to perform *post hoc* analyses by selecting relevant tutoring topics. (For example, "Is there a difference in student response quality on Mondays and Fridays?")

Talking Heads As AutoTutor does, RMT uses an animated agent with synthesized speech to present the tutor's utterances to the student. In principle, this allows the system to use multiple modes of communication to deliver a richer message. For example, the tutor can avoid face-threatening direct negative feedback, but still communicate doubt about an answer with a general word like "Well" with the proper intonation. Furthermore, in relation to text-only tutoring, the student is more likely to "get the whole message" because they can not simply skim over the text.

Curriculum Script A number of studies have shown that human tutors use a "curriculum script", or a rich set of topics which they plan to cover during a tutoring session [1]. RMT's curriculum script serves the same function. It is the repository of the system's knowledge about the tutoring domain. In particular, it contains the topics that can be covered, the questions that the tutor can ask, the answers that it expects it might get from the students, and a variety of dialog moves to keep the discourse going. RMT's curriculum script currently contains approximately 2500 items in 5 topics. We believe that this gives us a reasonable

starting point for using the tutoring system throughout a significant portion of a quarter-long class.

2.4 Pilot Testing and Results

We are currently preparing RMT for full-scale introduction into research methods classes at DePaul University. In Fall, 2003, we performed a pilot test of the system to ensure that there were no major glitches with it, that students would understand how to interact with it using a web browser, and to determine student attitudes toward the system.

Three versions of RMT were pilot tested with 26 volunteers enrolled in Introductory Psychology: a text-only interface (N = 8) and two versions using synthesized speech with animated agents, "Merlin" (N = 9) and "Miyako" (N = 9). Merlin is a cartoon-like character with many animations. Miyako is more human-like figure, but has limited movement. Each student completed one module on the topic of research validity, then answered both open-ended and Likert-scaled questions about the tutor interface, tutorial content, and tutor effectiveness.

Student responses to open-ended questions included positive comments about several specific aspects of the tutor's pedagogical design, including: the feedback the tutor provided about their answers; receiving hints and prompts that lead the student to the right answer; and having multiple chances to provide the correct answer to a question.

Although the pilot data does not speak to the actual effectiveness of the tutor in terms of objective measures of student learning, we did obtain student ratings of the effectiveness of both the curriculum script content and the tutor as a whole. The three conditions (text-only, Merlin, Miyako) did not differ in students' ratings of the tutorial content, but did differ in ratings of overall tutor effectiveness. On six-point scales, students indicated they expected to learn more from the text-only version of the tutor (mean = 2.5) than from the Merlin (mean = 3.7, $p = 0.09$ by LSD paired comparisons) or Miyako (mean = 3.9, $p < .05$) versions. As found in [10], these results suggest that more research is needed in the area of likeability and pedagogical effectiveness of agents.

In Winter Term 2004, we made the system available to the students in the research methods classes for the first time. The delivered system used the Miyako agent instead of Merlin because we were concerned that the students would not take the cartoonish Merlin character seriously. We used a different speech engine (Lernout & Hauspie British English) because it produced less irritating speech. Approximately 100 students signed up to voluntarily use the system. Unfortunately, they had to wait to use the system for about a week after they signed up while we registered them with the system. We believe that that delay along with the lack of any overt incentive for the students to use the system led to a disappointing outcome: only 6 students ever logged into the system even one time. In the Spring term, we offered extra credit to students who used the system, and 4 students completed all the requirements. In the future we plan to integrate the tutoring system more closely with the curriculum and have the teachers be more involved in promoting the system. In the next section, we

present the results of a study that we performed using Intro Psych subject pool participants.

3 Experiment

Our design was a 2 x 2 factorial, with agent (the Miyako head vs. text only) and task version (traditional tutoring task vs. simulated employment as a research assistant, described in more detail below) as between-subjects factors. Students were randomly assigned to the conditions except that participation in the agent conditions required the ability to install software on their Windows-based computer. As a result, more students interacted with the text-only presentation rather than the Miyako animated agent. 101 participants took the pretest. 23 were assigned to the "Miyako" agent, 78 to text-only presentation. 59 were assigned to the research assistant task version, and 42 to the tutor task version.

Each participant had one or two modules available (experimental design, reliability) to be completed.[3] We first reviewed the transcripts to code whether each participant had completed each module. We discarded data from participants who were non-responsive or who had technical difficulties.

Many students appeared to have difficulty installing the speech and agent software and getting it to work properly. A 2 x 2 between-subjects ANOVA comparing the number of modules completed (0, 1 or 2) for the four conditions in the study also suggested that there were significant technical issues with the agent software. Although there was no significant difference in the number of modules completed by participants in the two task versions (RA = .69; tutor = .81 modules completed), participants in the Miyako agent condition completed significantly fewer modules (.47) than those in the text-only condition (1.0), $F(1, 97) = 8.57, p < .01$.

Our primary dependent measure was gain score, defined as the difference between the number correct on a 40-question multiple-choice post-test and an identical pre-test. All analyses of gain scores included pre-test score as a covariate, an analysis which is functionally equivalent to analyzing post-test scores with pre-test scores as a covariate [11].

We first examined whether completion of the tutor modules was associated with greater gain scores compared to students who took the pre- and post-tests but did not successfully complete the modules. Of the 75 participants who completed both the pre-test and the post-test, 28 completed both modules, 26 completed one module, and 21 did not satisfactorily complete either module before taking the post-test. In a one-way ANCOVA, gain scores were analyzed with number of modules completed as the independent variable and pre-test score as the covariate. The main effect of number of modules was significant, $F(2, 71) - 9.50, p < .001$. Although the mean pre-test to post-test gain score for those completing two modules (4.4 on a 40-item multiple-choice test) was greater than that for those who completed no modules (2.4), participants who completed

[3] One week into the experiment, we found that students were completing the first topic too quickly, so we added another.

only one module showed no gain at all (gain = -.3). Only the difference between the mean gain for one module (-.3) versus 2 modules (4.4) was statistically significant, as indicated by non-overlapping 95% confidence intervals.

Breaking down the effects on gain scores for each of the two modules, it appeared that the "reliability" module significantly improved learning, but the "experimental design" module did not. Students who completed the reliability module had higher gain scores (4.4) than those who did not (0.9), and this difference was significant in an ANCOVA in which pre-test score was entered as the covariate, $F(1, 72) = 14.17, p < .001$. A similar analysis for the experimental design module revealed non-significantly lower gain scores for students who completed the experimental design module than those who did not, with mean gains of 2.1 vs. 2.4 respectively, $F(1, 72) < 1$.

The reliability module was considerably longer than the experimental design module, so time on task may be partly responsible for the differences in effectiveness between the two modules.

We next examined the effects of our two primary independent variables, agent and task version, on gain scores. For these analyses we included only participants who had successfully completed at least one module after taking the pre-test and before taking the post-test. Of the 54 participants who completed at least one module, 6 interacted with the Miyako agent and 48 used the text-only interface. Students were more evenly divided between the two task versions, with 25 in the tutor and 29 in the research assistant version.

Gain scores were entered into a 2 x 2 ANCOVA with agent and task version as between-subjects factors and pre-test score as the covariate. Gain scores were greater for students using the text-only interface (mean = 2.6, N = 48) than for those interacting with the Miyako agent (mean = -1.5, N = 6), $F(1, 49) = 3.70, p = .06$. Neither the main effect of task version nor the agent * task version interaction was significant, $Fs < 1$.

Because of the low number of participants interacting with the animated agent the effect of agent in the this analysis must be interpreted with caution, but it is consistent with our other findings indicating that students had difficulty with the Miyako agent. We suspect that technical difficulties may have been largely responsible.

4 Discussion

In this section, we describe some of the aspects of the system that may have contributed to the results of the experiment. In particular, we look at the the tutoring modules that were used, the animated agent, and the task version.

Modules We initially included only one module in the experiment because we thought it would take the participants somewhere between 30 and 60 minutes to complete it. We chose the "experimental design" module because we thought it would be accessible to intro psych students. Because we added the second module, "reliability", partway through the experiment and because the two modules

are significantly different, we can not say whether the gain difference for the number of modules completed was caused by the amount of interaction with the tutor, or due to some effects of the particular modules.

It could also be the case that the subject pool students had enough familiarity with the experimental design material that they performed better on the pre-test, and therefore had less opportunity for gain.

The Agent There were two significant weakness of the agent used here that may have affected our results. First, there may have been software installation difficulties. The participants were using the system on their own computers in their homes, and had to install the agent software if they were assigned to the agents version. The underlying agent technology that we used, Microsoft Agents, requires three programs to be installed from a Microsoft server. The participants could have had difficulty following the instructions for downloading the software or could have been nervous about installing software that they did not search out for themselves.

Second, the particular animated agent that we used was rather limited. A good talking head should be able not just to tap into the social dynamics present between a human tutor and student, but also provide an additional modality of communication: prosody. In particular, human tutors are known to avoid giving explicit negative feedback because that could cause the student to "lose face" and make her nervous about offering further answers. Instead, human tutors tend to respond to poor student answers with vague verbal feedback ("well" or "okay") accompanied by intonation that makes it clear that the answer could have been better [12].

Unfortunately, the agent that we used was essentially a shareware agent that had good basic graphics, but relatively no additional animations that might display the tutor's affect that goes along with the verbal feedback. Furthermore, the text-to-speech synthesizer that we used (Lernout & Hauspie British English) was relatively comprehensible, but we have not yet tackled the difficult task of trying to make the speech engine produce the type of prosodic contours that human tutors use. Thus, all of the tutor utterances are offered in a relatively detached, stoic conversational style.

Despite these limitations, we had hypothesized that the agent version would have an advantage over text-only for at least one reason: in the text-only version, the students might well just scan over the feedback text to find the next question. With audio feedback, the student is essentially forced to listen to the entire feedback and question before entering the next response. Of course, this may have also contributed to the lower completion rate of students in the agent version because they may have become frustrated by the relatively slow pace of presentation of the agent's synthesized speech.

Task Version As mentioned above, we tested two different task versions, the traditional tutor and a simulated research assistant condition. In the former, the

tutor poses questions,[4] the student types in an answer, and the dialog continues with both parties contributing further information until a relatively complete answer has been given. In the research assistant condition, the basic "rules of the game" are the same with one subtle, but potentially significant difference: instead of a tutor, the system is assuming the role of an employer who has hired the student to work on a research project. As previous research has shown, putting students into an authentic functional role — even when it is simulated — can greatly increase their motivation to perform the task, and thereby also increase their learning [13].

Unfortunately, in the current version of RMT, our simulation of the research advisor role is rather minimal. The only difference is in the initial "introduction" that the agent gives to the student. In the traditional tutor condition, the agent (or text) describes briefly how the tutoring session will progress with the student typing their responses into the browser window. In the research assistant version, the agent starts with an introduction that is intended to establish the social relationship between the research supervisor and student/research assistant. Unfortunately, there are no continuing cues to enforce this relationship. We intend to develop this aspect of the system further, but for the current evaluation we needed to focus on getting the basic mechanisms of the tutor in place along with the research methods tutoring content.

5 Conclusions

Because RMT is designed to be used in conjunction with classes on an everyday basis, there are obviously significant technical issues to overcome. In addition to the issues mentioned in the previous section, we plan on focusing on the natural language understanding mechanism to incorporate a variety of syntactic and discourse mechanisms in order to improve the system's understanding of the student replies.

We feel that in the long run, this type of system will be shown to be a valuable adjunct to classroom instruction. With a dialog-based tutoring system, the student can interact in a natural way using their own words. The process of constructing responses to the tutor's questions can in itself help the students "firm up the ideas" in their heads. However, it is also clear based on our experience that the tutoring system can not just be offered to the students. It must be an integrated component of the course.

While the results of our current experiment indicate that the use of an animated agent "talking head" does not increase learning (and in fact, appeared to lead to degradation of the students' knowledge), we feel that further research is warranted on this topic. The limitations of our current agent may have interfered with the student's attention to the material under discussion.

In any case, RMT has been shown to help students learn the rather difficult material covered in Psychology research methods classes. As we continue to

[4] As in human-human tutoring, students may ask questions, but rarely do [12].

develop and refine the system, we hope that it can eventually become another standard mechanism for augmenting the students' education.

References

1. Graesser, A.C., Person, N.K., Magliano, J.P.: Collaborative dialogue patterns in naturalistic one-to-one tutoring. Applied Cognitive Psychology **9** (1995) 359–387
2. Glass, M.: Processing language input in the CIRCSIM-tutor intelligent tutoring system. In Moore, J., Redfield, C., Johnson, W., eds.: Artificial Intelligence in Education, Amsterdam, IOS Press (2001) 210–221
3. Rosé, C., Jordan, P., Ringenberg, M., S. Siler and, K.V., Weinstein, A.: Interactive conceptual tutoring in Atlas-Andes. In: Proceedings of AI in Education 2001 Conference. (2001)
4. Aleven, V., Popescu, O., Koedinger, K.R.: Towards tutorial dialog to support self-explanation: Adding natural language understanding to a cognitive tutor. In: Proceedings of the 10th International Conference on Artificial Intelligence in Education. (2001)
5. Zinn, C., Moore, J.D., Core, M.G., Varges, S., Porayska-Pomsta, K.: The be&e tutorial learning environment (beetle). In: Proceedings of the Seventh Workshop on the Semantics and Pragmatics of Dialogue (DiaBruck 2003). (2003) Available at http://www.coli.uni-sb.de/diabruck/.
6. Wiemer-Hastings, P., Graesser, A., Harter, D., the Tutoring Research Group: The foundations and architecture of AutoTutor. In Goettl, B., Halff, H., Redfield, C., Shute, V., eds.: Intelligent Tutoring Systems, Proceedings of the 4th International Conference, Berlin, Springer (1998) 334–343
7. Graesser, A., Jackson, G., Mathews, E., Mitchell, H., Olney, A., Ventura, M., Chipman, P., Franceschetti, D., Hu, X., Louwerse, M., Person, N., TRG: Why/autotutor: A test of learning gains from a physics tutor with natural language dialog. In: Proceedings of the 25th Annual Conference of the Cognitive Science Society, Mahwah, NJ, Erlbaum (2003)
8. Landauer, T.K., Laham, D., Rehder, R., Schreiner, M.E.: How well can passage meaning be derived without using word order? a comparison of Latent Semantic Analysis and humans. In: Proceedings of the 19th Annual Conference of the Cognitive Science Society, Mahwah, NJ, Erlbaum (1997) 412–417
9. Mostow, J., Aist, G.: Evaluating tutors that listen. In Forbus, K., Feltovich, P., eds.: Smart Machines in Education. AAAI Press, Menlo Park, CA (2001) 169–234
10. Moreno, K., Klettke, B., Nibbaragandla, K., Graesser, A.: Perceived characteristics and pedagogical efficacy of animated conversational agents. In Cerri, S., Gouarderes, G., Paraguacu, F., eds.: Proceedings of the 6th Annual Conference on Intelligent Tutoring Systems, Springer (2002) 963–972
11. Werts, C.E., Linn, R.L.: A general linear model for studying growth. Psychological Bulletin **73** (1970) 17–22
12. Person, N.K., Graesser, A.C., Magliano, J.P., Kreuz, R.J.: Inferring what the student knows in one-to-one tutoring: The role of student questions and answers. Learning and Individual Differences **6** (1994) 205–229
13. Schank, R., Neaman, A.: Motivation and failure in educational simulation design. In Forbus, K., Feltovich, P., eds.: Smart machines in education. AAAI Press, Menlo Park, CA (2001) 37–69