Developing pedagogically effective tutorial dialogue tactics: Experiments and a testbed

Kurt VanLehn, Pamela Jordan, Diane Litman

Learning Research & Development Center and Intelligent Systems Program
University of Pittsburgh, Pittsburgh PA, USA
vanlehn@cs.pitt.edu

Abstract

Although effective tutorial dialogue strategies are well understood, tutorial tactics that govern brief episodes of tutoring, such as a single step, are not. Because better tactics seem to be crucial for further improving pedagogical effectiveness, we have begun investigating the effects of varying tutorial tactics. In this paper we describe two planned experiments and the testbed we have created to support this experimentation.

1. Introduction

One-on-one, face-to-face human tutoring is widely considered to be an extremely effective method of instruction. Compared to classroom instruction, human tutors have raised students' learning gains by two standard deviations, which is a very large effect size [1]. This has inspired both analyses of human tutorial dialogues [2, 3] and development of tutoring systems [4] with both standard form/menu interfaces [5, 6] and natural language (NL) interfaces [7, 8].

Effective tutorial strategies are now well understood. When tutors are helping students solve complex, multi-step problems, such as the physics problem shown in the upper right pane of Figure 1, both human tutors and recent computer tutors follow the same two-phase strategy. During phase 1, the tutor and the student solve the problem together, step by step. In Figure 1, they have just completed one step (writing an equation) and are starting on a new step (solving it for v1). During phase 2, the tutor and student reflect on the problems' solution. They highlight the solution's main steps, review any confusions that the student may have had during the solution, and consider how the solution varies when the problem statement is varied in certain ways. In short, the overall strategy consists of collaborative solution (phase 1) followed by discussion (phase 2).

Although tutorial strategy is no longer contentious, tutorial tactics are still not well understood. Tutorial tactics govern brief episodes of tutoring, such as a single step or a single post-solution discussion question and seem to be crucial for achieving further improvements in pedagogical effectiveness. We are focusing on two tactical decisions: (1) During problem solving, should the tutor simply *tell* the student the next step or *elicit* the step from the student with a prompt or question? (2) When a step has been completed during problem solving or a question has been answered during the post-solution discussion, should the tutor ask the student to explain it?

In both cases, there is theory about the right tactics to use, but implementing it is nontrivial. For the first tactical decision, whether to tell or elicit a step, the common wisdom is *model, scaffold, fade*[9]. When a student is completely unfamiliar with a step (e.g., they have never applied the definition of kinetic energy before), then the tutor should tell her the step (this is called *modelling the step* by Collins et al.) because trying to elicit the step will probably frustrate most students (but not all). Once a student has some familiarity, the tutor should have a content-appropriate prompt for the step, such as "Try applying the definition of kinetic energy." This is called *scaffolding* (temporary help). A still more competent student should receive much less help–the scaffolding *fades*. Once a student has mastered a step, then it doesn't matter who does it; the tutor can do the step if that makes the problem solving more efficient. In short, this simple decision of whether to tell or elicit a step depends in subtle ways on both the students' competence and on affective factors, such as their self-confidence and desire for autonomy. If the tutor makes bad decisions, the tutorial dialogue has the potential to vary between a boring demonstration and a frustrating "try to read the tutor's mind" game.

For the second tutorial tactic, whether to ask for a justification of a step, the common wisdom is that students need to be aware of the justification. But eliciting one may not always be best. If the student is already aware of it, then typing in a justification can be slow and frustrating. And sometimes it may be better to delay discussion of the justifications to phase 2, after the problem has been solved [3], especially if the justification is abstract, plan-based or long.

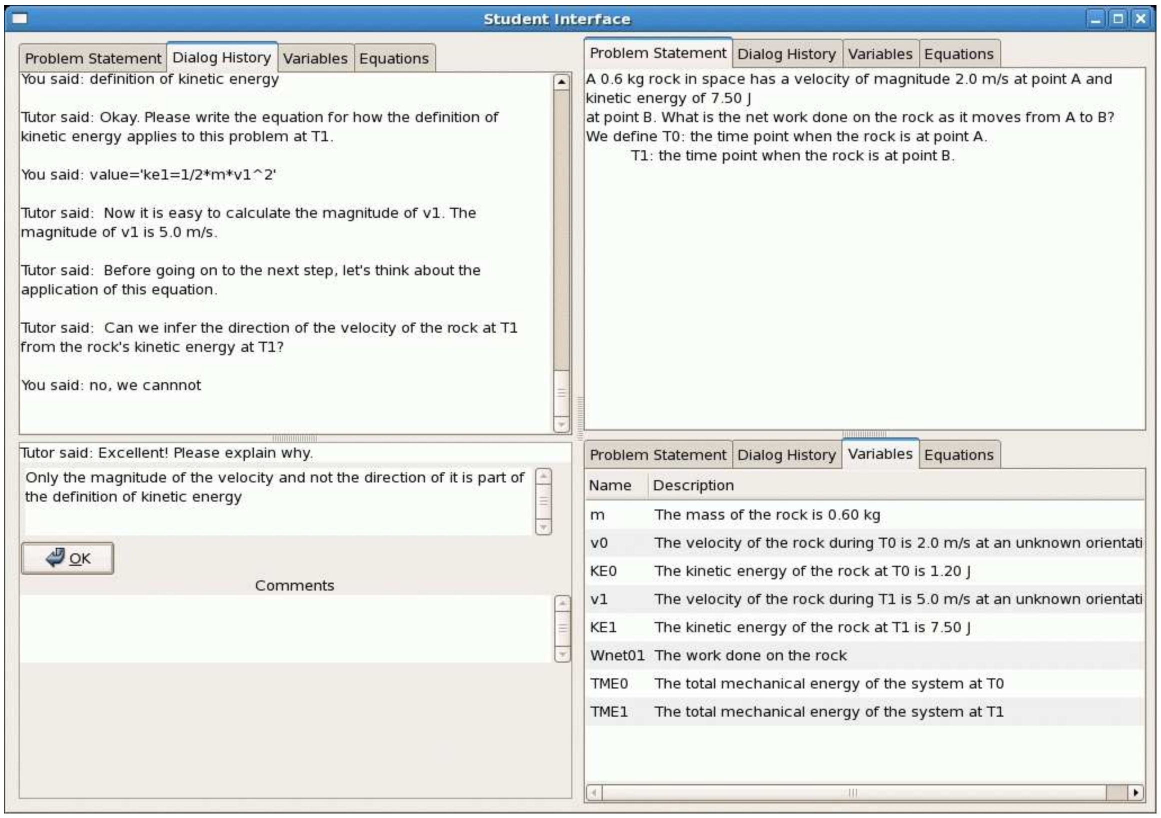
Given the subtlety of tutorial tactical decision making and the speed with which such decisions must be made, it is unsurprising that human tutors are only mediocre at it [10, 8]. Thus, we have switched from analyzing human tutorial dialogues to doing experiments that vary tutorial tactics and measure the resulting learning gains. This paper briefly introduces our first two experiments and Cordillera[1], a testbed for conducting such experiments. Our distant goal is to build tutoring systems whose dialogues are so well attuned to students that the systems will be even more effective than human tutors.

2. Learning Dialogue Policies

While prior studies have made progress in identifying desirable tutorial tactics, as discussed above, it is still unclear how to design a computer tutor that can optimally use such tactics. Work in designing spoken dialogue systems has proposed several data-driven

This work was funded by NSF grant 0325054.

[1]Cordillera is pronounced "Core-dee-yare-uh"



You said: definition of kinetic energy

Tutor said: Okay. Please write the equation for how the definition of kinetic energy applies to this problem at T1.

You said: value='ke1=1/2*m*v1^2'

Tutor said: Now it is easy to calculate the magnitude of v1. The magnitude of v1 is 5.0 m/s.

Tutor said: Before going on to the next step, let's think about the application of this equation.

Tutor said: Can we infer the direction of the velocity of the rock at T1 from the rock's kinetic energy at T1?

You said: no, we cannnot

Tutor said: Excellent! Please explain why.

Only the magnitude of the velocity and not the direction of it is part of the definition of kinetic energy

A 0.6 kg rock in space has a velocity of magnitude 2.0 m/s at point A and kinetic energy of 7.50 J
at point B. What is the net work done on the rock as it moves from A to B?
We define T0: the time point when the rock is at point A.
T1: the time point when the rock is at point B.

Name	Description
m	The mass of the rock is 0.60 kg
v0	The velocity of the rock during T0 is 2.0 m/s at an unknown orientati
KE0	The kinetic energy of the rock at T0 is 1.20 J
v1	The velocity of the rock during T1 is 5.0 m/s at an unknown orientati
KE1	The kinetic energy of the rock at T1 is 7.50 J
Wnet01	The work done on the rock
TME0	The total mechanical energy of the system at T0
TME1	The total mechanical energy of the system at T1

Figure 1: The student interface

methodologies for addressing this issue, including the use of Reinforcement Learning [11] to automatically learn the best action for a system to take at any state in a dialogue (e.g., [12, 13, 14]). Data for reinforcement learning algorithms are typically obtained by allowing an agent to explore an environment, in terms of perceiving a state and taking an action. The environment, in turn, provides a reward. Reinforcement learning finds an optimal policy from this data, by specifying the best action to take in every state such that the cumulative reward for the agent is maximized.

Past research into using Reinforcement Learning to improve spoken dialogue systems has commonly used Markov Decision Processes (MDP's) [11] to model a dialogue. A MDP is defined by a set of states $\{s_i\}_{i=1..n}$, a set of actions $\{a_k\}_{k=1..p}$, and a set of transition probabilities which reflect the dynamics of the environment $\{p(s_i|s_j,a_k)\}_{i,j=1..n}^{k=1..p}$: if the model is at time t in state s_j and takes action a_k, then it will transition to state s_i with probability $p(s_i|s_j,a_k)$. Additionally, an expected reward $r(s_i,s_j,a_k)$ is defined for each transition. When casting the dialogue control problem in this formalism, the model parameters $\{p(s_i|s_j,a_k)\}_{i,j=1..n}^{k=1..p}$ are typically estimated from a corpus of dialogues; then, a simple dynamic programming approach is used to learn the optimal control policy π^*, i.e. the set of actions the model should take at each state, to maximize its expected cumulative reward.

While policy development is very important, choosing the best features to model the state is equally important, since it impacts the actions a system will choose to take. In prior work [15], we used a previously collected corpus of spoken computer tutoring physics dialogues to train an MDP model of tutoring interactions, then used this model to learn whether considering more complex state features impacted the optimization (with respect to improving student learning) of two types of tutor tactics: whether or not to generate feedback, and what type of question to ask. However, because our data was not collected to be exploratory with respect to tutor tactics, the tutor often only used one type of action in many dialogue states, which severely limited the types of questions that our research could investigate. Thus, our first experiment involves generating an exploratory corpus more suited to reinforcement learning (and other probabilistic approaches), which will then be used to test whether machine learning can optimize tutorial dialogues.

3. Experimental Design

We are conducting our experiments within the Physics work-energy domain. From the Physics literature, we collected a database of 127 quantitative and qualitative problems and listed the knowledge components[2] that must be applied to solve each problem. We then identified 30 knowledge components to teach and selected seven quantitative problems that covered these knowledge components to use for student training. We also selected problems from the database to create a 37 item test[3] to derive expected re-

[2]*Knowledge component* is a generalization of everyday terms like concept, principle, fact, or skill, and cognitive science terms like schema, production rule, misconception, or facet.

[3]The pre and post-test are identical.

wards for use in Reinforcement Learning.

We have written dialogues for Cordillera that have many action choices (branches) that are under the tutor's control. In particular, for a state or step the tutor chooses (1) whether to tell or elicit it, and (2) whether to elicit a justification for it or not. In Figure 1, the dialogue history shows a number of *tell* actions with a few *elicits* and the pending question in the input pane at the bottom shows a *justify* action choice. As discussed above, these choices are important and yet difficult to make so as to maximize learning, motivation and speed. We want to find out if machine learning can infer effective policies for making such action choices.

In the first experiment these choices are made randomly. We expect that students will learn some knowledge components slowly and others rapidly, depending on what choices the tutor makes with that particular student and knowledge component. For instance, when solving the problem of Figure 1, if the tutor starts by asking some students, "Please define variables for the rock's kinetic energy," some students may be so confused that they need a multi-turn subdialogue to find out what the tutor means by "kinetic energy" and why two variables are needed for it. This subdialogue is likely to be so lengthy and complex that these students learn almost nothing from it. Thus, this tutorial dialogue choice will slow their learning of the *time point* knowledge component. On the other hand, learning may be much faster if the tutor's random choice led it to tell the student the time points (e.g., "Because the problem asks for the net work done on the rock, it is likely that the rock's kinetic energy changes. Thus, let us use KE0 for its kinetic energy at time T0 and KE1 for its kinetic energy at time T1."). Because the action choices are made randomly during this experiment, a student might get an unlucky set of choices for one knowledge component, which causes it to be learned slowly, and a lucky set of choices for another, which causes it to be learned quickly. Thus, during this first experiment, we will collect data on which action choices the tutor makes and how that affects a student's learning of individual knowledge components.

During the second experiment, we will test whether the policies inferred by machine learning are actually pedagogically effective. The experiment will have two groups of students: treatment and control. The treatment group will use the version of Cordillera with the learned policies. The control group will use the random-choice version of Cordillera that was used in experiment 1. We hypothesize that the treatment group will learn faster than the control group. That is, their learning gains will be larger than the control group's learning gains.

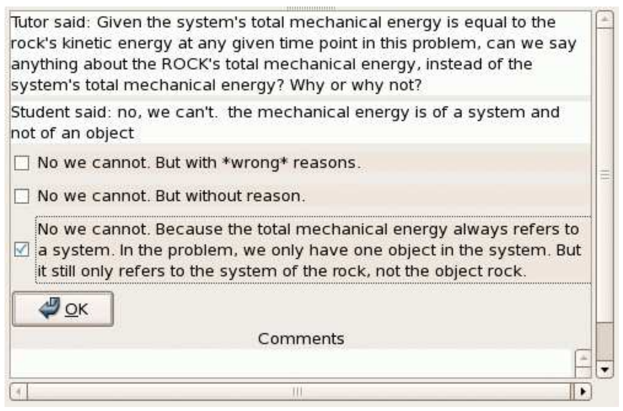
4. Cordillera

To construct Cordillera, we used the TuTalk [16, 17] NL tutorial dialogue toolkit as the foundation. TuTalk enables system designers to focus on the development of the content to be presented to students and was built to support dialogues in which a tutor tries to elicit the main line of reasoning from a student by a series of questions. This style of dialogue was inspired by CIRCSIM-Tutor's directed lines of reasoning [7]. In addition, TuTalk is modular so that core modules such as NL understanding can be replaced or supplemented as needed. We made two modifications; (1) we replaced language understanding with a human wizard and (2) embedded GUI menus and forms for student input.

The simplest dialogue one can write for TuTalk can be represented as a finite state machine. Each state contains a single tutor turn. The arcs leaving the state correspond to all classifications of a student's response turn. When creating a state, the author enters the text for a tutor's turn and defines several classes of student responses as transition arcs, and indicates which state each arc leads to. An arc can also "call" a finite state network, which allows authors to create hierarchical dialogues.

In TuTalk, the NL associated with states and arcs is represented in concept definitions. In the simplest case, a concept is a set of NL phrases. For instance, the set for concept NEG-ACK might be "Not quite", "Well, not exactly", "No". When a string is input, the dialogue manager asks the understanding module to determine what concepts it represents and determines transitions on the basis of the concept labels returned. Likewise when a concept is to be expressed, the dialogue manager asks the generation module to determine how to best express it in NL.

We embedded a GUI command language into the concept definitions that the Cordillera interface interprets. As a result, a Cordillera dialogue can interleave requests for student input to be via GUI or NL. The variables in the bottom right pane of Figure 1 were defined either by the student using a form interface or provided by the tutor (*elicit* vs. *tell*).


Tutor said: Given the system's total mechanical energy is equal to the rock's kinetic energy at any given time point in this problem, can we say anything about the ROCK's total mechanical energy, instead of the system's total mechanical energy? Why or why not?

Student said: no, we can't. the mechanical energy is of a system and not of an object

☐ No we cannot. But with *wrong* reasons.

☐ No we cannot. But without reason.

☑ No we cannot. Because the total mechanical energy always refers to a system. In the problem, we only have one object in the system. But it still only refers to the system of the rock, not the object rock.

Figure 2: The wizard's NL classification pane

To reduce the confounds of imperfect NL understanding on our experiments, we replaced the NL understanding module with a human interpreter which we call the language understanding wizard. The wizard's interface mirrors that of the student except that the student input pane is replaced by the student's response and a set of check-boxes for classifying the student's response. See Figure 2 for an example of the wizard's classification pane.

TuTalk also provides a number of advanced features. The most important of these features for Cordillera is the one in which authors can specify states to be skipped if certain simple conditions hold. For experiment 1, we marked justifications as states that can be randomly skipped.

5. Current Status

We completed two rounds of pilot testing on the dialogues to check that the content presented is understandable and helpful to students. We started experiment 1 at the beginning of the summer and expect be ready to begin experiment 2 by mid Fall.

6. References

[1] B. S. Bloom, "The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring.," *Educational Researcher*, , no. 13, pp. 4–16, 1984.

[2] A. C. Graesser, N. Person, and J. Magliano, "Collaborative dialog patterns in naturalistic one-on-one tutoring," *Applied Cognitive Psychology*, , no. 9, pp. 359–387, 1995.

[3] S. Katz, G. O'Donnell, and H. Kay, "An approach to analyzing the role and structure of reflective dialogue," *International Journal of Artificial Intelligence in Education*, , no. 11, pp. 320–343, 2000.

[4] K. VanLehn, "The behavior of tutoring systems," *International Journal of Artificial Intelligence and Education*, , no. 16, 2006.

[5] J. R. Anderson, A. T. Corbett, K. R. Koedinger, and R. Pelletier, "Cognitive tutors: Lessons learned," *The Journal of the Learning Sciences*, vol. 4, no. 2, pp. 167–207, 1995.

[6] K. VanLehn, C. Lynch, K. Schultz, J. A. Shapiro, R. H. Shelby, and L. Taylor, "The Andes physics tutoring system: Lessons learned," *International Journal of Artificial Intelligence and Education*, vol. 3, no. 15, pp. 147–204, 2005.

[7] M. Evens and J. Michael, *One-on-One Tutoring by Humans and Computers*, Lawrence Erlbaum Associates, Inc., 2006.

[8] K. VanLehn, A. C. Graesser, G. T. Jackson, P. Jordan, A. Olney, and C. P. Rose, "When are tutorial dialogues more effective than reading?," *Cognitive Science*, vol. 31, no. 1, pp. 3–62, 2007.

[9] A. Collins, J. S. Brown, and S. E. Newman, "Cognitive apprenticeship: Teaching the craft of reading, writing and mathematics," in *Knowing, learning and instruction: Essays in honor of Robert Glaser*, L. B. Resnick, Ed., pp. 453–494. Lawrence Erlbaum Associates, Hillsdale, NJ, 1989.

[10] M. T. H. Chi, S. Siler, H. Jeong, T. Yamauchi, and R. G. Hausmann, "Learning from human tutoring," *Cognitive Science*, vol. 25, pp. 471–533, 2001.

[11] R. Sutton and A. Barto, *Reinforcement Learning*, The MIT Press, 1998.

[12] E. Levin and R. Pieraccini, "A stochastic model of computer-human interaction for learning dialog ues," in *Proc. of EUROSPEECH '97*, 1997.

[13] M. Walker, "An application of reinforcement learning to dialogue strategy select ion in a spoken dialogue system for email," *JAIR*, vol. 12, 2000.

[14] S. Singh, D. Litman, M. Kearns, and M. Walker, "Optimizing dialogue managment with reinforcement learning: Experimen ts with the njfun system," *JAIR*, vol. 16, 2002.

[15] J. Tetreault, D. Bohus, and D. Litman, "Estimating the reliability of mdp policies: a confidence interval ap proach," in *NAACL*, 2007.

[16] P. W. Jordan, M. Ringenberg, and B. Hall, "Rapidly developing dialogue systems that support learning studies," in *Proceedings of ITS06 Workshop on Teaching with Robots, Agents, and NLP*, 2006.

[17] P. W. Jordan, B. Hall, M. Ringenberg, Y. Cui, and C.P. Rosé, "Tools for authoring a dialogue agent that participates in learning studies," in *Proceedings of AIED 2007*, 2007.