International Journal of Artificial Intelligence in Education 21 (2011) 83–113
DOI 10.3233/JAI-2011-014
IOS Press

An Evaluation of Pedagogical Tutorial Tactics for a Natural Language Tutoring System: A Reinforcement Learning Approach

Min Chi, *Human-Sciences and Technologies Advanced Research Institute, Stanford University, CA, USA*
minchi@stanford.edu
Kurt VanLehn, *School of Computing, Informatics and Decision Science Engineering, Arizona State University, AZ, USA*
Kurt.Vanlehn@asu.edu
Diane Litman, *Department of Computer Science and Intelligent Systems Program and Learning Research and Development Center, University of Pittsburgh, Pittsburgh, PA, USA*
litman@cs.pitt.edu
Pamela Jordan, *Learning Research and Development Center, University of Pittsburgh, Pittsburgh, PA, USA*
pjordan@pitt.edu

Abstract. Pedagogical strategies are policies for a tutor to decide the next action when there are multiple actions available. When the content is controlled to be the same across experimental conditions, there has been little evidence that tutorial decisions have an impact on students' learning. In this paper, we applied Reinforcement Learning (RL) to induce two sets of pedagogical policies from pre-existing human interaction data. The NormGain set was derived with the goal of enhancing tutorial decisions that contribute to learning while the InvNormGain set was derived with the goal of enhancing those decisions that contribute less or even nothing to learning. The two sets were then tested with human students. Our results show that when the content was controlled to be the same, different pedagogical policies did make a difference in learning and more specifically, the NormGain students outperformed their peers. Overall our results suggest that content exposure and practice opportunities can help students to learn even when tutors have poor pedagogical tutorial tactics. However, with effective tutorial tactics, students can learn even more.

Keywords. Reinforcement learning, human learning, intelligent tutoring systems, pedagogical strategy

INTRODUCTION

Human one-on-one tutoring is one of the most effective educational interventions in that tutored students often perform significantly better than students in classroom settings (Bloom, 1984). Computer learning environments that mimic aspects of human tutors have also been highly successful. Intelligent Tutoring Systems (ITSs) have been shown to be highly effective in improving students' learning in the classroom. Classroom instruction with an ITS produces measurably larger learning gains than the same classroom instruction without an ITS (Anderson, Corbett, Koedinger, & Pelletier, 1995; Koedinger, Anderson, Hadley, & Mark, 1997; VanLehn, Lynch, & et al., 2005). One hypothesis as to the effectiveness of human or computer one-on-one tutoring is that it comes from the detailed management of "micro-steps" in natural language

[Problem Statement:]

A $0.6kg$ rock in space has a velocity of magnitude $2.0m/s$ at point A and kinetic energy of $7.50J$ at point B. What is the net work done on the rock as it moves from A to B? We define:
T_0: the time point when the rock is at point A.
T_1: the time point when the rock is at point B.

[Solution:]

Step 1: Applying the Definition of Kinetic Energy to solve for the rock's kinetic energy at T_0: $KE_0 = \frac{1}{2}mv_0^2$

Step 2: Applying the Definition of Total Mechanical Energy to solve for the rock-system's Total Mechanical Energy at T_0, $TME_0 = KE_0 + 0 + 0$

Step 3: Applying the Definition of Total Mechanical Energy to solve for the rock-system's Total Mechanical Energy at T_1, $TME_1 = KE_1 + 0 + 0$

Step 4: Applying the Change of Total Mechanical Energy for Non-isolated Systems to solve for the work done on the rock-system from T_0 to T_1: $Net_W = TME_1 - TME_0$

Fig. 1. A Four-step solution for training problem: P_4.

tutorial dialogue (Graesser, Person, & Magliano, 1995; Graesser, VanLehn, Rosé, Jordan, & Harter, 2001). A typical ITS, however, is step-based (VanLehn, 2006).

In domains like math and physics, solving a problem requires producing an argument, proof or derivation consisting of one or more inference steps; each step is the result of applying a domain principle, operator or rule. For example, to solve a physics problem, generally several physics principles need to be applied and some need to be applied more than once. Each principle application can be seen as *a step* in ITSs. Once a student enters a step, then the ITS gives feedback and/or hints.

For example, on an ITS such as Andes the solution for a quantitative physics problem P_4 involves four main steps (shown in Fig. 1) along with some minor steps that are omitted here for simplicity. Among the four steps, the principle "Definition of Total Mechanical Energy ($TME = KE + GPE + SPE$)" is applied twice, steps 2 and 3 respectively; the "Definition of Kinetic Energy ($KE = \frac{1}{2}mv^2$)" and the "Change of Total Mechanical Energy for Non-isolated Systems ($Net_W = TME_2 - TME_1$). " are applied once each.

Human tutors, by contrast, often scaffold students via a series of micro-steps that lead to a full step. In **[Step 1]** of the solution for P_4 above, for instance, a human tutor may take four **micro-level** steps (as shown in Fig. 2). In Fig. 2, each numbered line represents a dialogue turn. The labels **T** and **S** designate tutor and student turns respectively. In this example, the four micro-steps are selecting the principle to apply (lines 2 & 3), writing the corresponding equation (line 4), solving the equation (Line 5), and engaging in some qualitative discussion about the principle, definition of Kinetic Energy (line 6–8) respectively.

It is often hypothesized that students will learn more when solving problems by progressing through a series of micro-steps than through steps. When students are trained on step-based ITSs such as Andes, they may mainly focus on learning quantitative forms of the domain principles and ignore the qualitative perspective of physics concepts or domain principles. For instance, students may remember to use an equation like $KE = \frac{1}{2}mv^2$ without knowing that they are applying the definition of Kinetic Energy or that the kinetic energy of an object is not affected by the direction of the object's motion. As a result, they may fail to transfer their quantitative problem solving skills to qualitative problems such as: " While flying along, a jet releases an empty propellant tank which eventually crashes to the ground. Consider three cases: a) the jet is flying horizontally, b) the jet is climbing upwards at a 45 degree angle or, c) the jet is diving downwards at a 45 degree angle. In all 3 cases, the jet is flying at the same speed and it releases the propellant tank from the same height. Ignoring air friction, in which case is the tank's speed greatest as it hits the ground? (Correct Answer is: It doesn't matter. The speed is the same in all three cases.)"

1. **T:** So let's start with determining the value of KE_0.
2. **T:** Which principle will help you calculate the rock's kinetic energy at T0? Please provide the name of the principle, not an equation. **{ELICIT}**
3. **S:** Definition of kinetic energy
4. **T:** Yes, I agree. Now I will write the equation for applying the definition of kinetic energy to the rock at T0: KE0 = 1/2*m*v0^2 **{TELL}**
5. **T:** I calculate the value for KE0 = 0.5*0.6kg*2.0 m/s^2 = 1.2 J **{TELL},** KC_{23}
6. **T:** We do not know the direction of the rock's velocity at T0 from the problem statement. Would the direction of v0 affect the rock's kinetic energy at T0? **{ELICIT},** KC_{20}
7. **S:** No, it would not affect
8. **T:** Yes, I agree. This is because the kinetic energy only depends on the rock's mass and magnitude of velocity, not the direction of the rock's velocity. **{TELL},** KC_{20}
9. **T:** We've calculated KE0, recall that our plan is to calculate v1 next.

Fig. 2. An example dialogue.

Indeed, Halloun & Hestenes (1985) showed that even university physics students with high grades often score poorly on tests of qualitative physics. Additionally, in a review of research on problem solving, Maloney (1994) noted that there is surprisingly very little evidence on the role of problem solving in learning physics concepts. On the other hand, a large body of research on comparing experts and novices on physics problem solving indicates the importance of qualitative reasoning in physics instruction (Chi, Feltovich, & Glaser, 1981; Larkin, McDermott, Simon, & Simon, 1980). For example, experts tend to be fast at qualitatively understanding the problem situation and at representing the problem in terms of basic physics concepts before translating it into mathematical equations while novices often rush into applying equations with little or no qualitative description of the problem. Here we believe that by breaking a step into a series of micro-steps as in Fig. 2, students are exposed to more qualitative discussion and reasoning as shown in lines 6-8 and thus are more likely to learn domain principles in a deeper manner.

If the effectiveness of human one-on-one tutoring lies in tutors' ability to scaffold a series of micro-steps leading to a step entry, then we would expect human tutors to be more effective than step-based tutors as both require students to enter the same major steps. There have been several tests of this hypothesis.

Prior research on the impact of micro-steps on learning

Evens & Michael (2006) conducted a series of studies comparing four learning treatments in cardiovascular physiology. The no-tutoring group studied a text that included correct worked examples along with the reasoning for solving a pacemaker problem. The CIRCSIM group solved one training problem on a tutoring system, CIRCSIM, which presented a short text passage for each incorrect step. The CIRCSIM-tutor group solved the same training problem on a sophisticated natural language tutoring system, CIRCSIM-Tutor, which replaced the text passages in CIRCSIM with typed natural language dialogue. The human tutor group also solved the same training problem with expert human tutors. Results showed that the latter three groups out-performed the no-tutoring group, but the three treatments, CIRCSIM, CIRCSIM-Tutor and expert human tutors, tied with each other.

Reif & Scott (1999a) compared three groups in their study. The no-tutoring group did their homework and received no feedback until their homework was returned. The step-tutoring group did their homework on a step-based tutoring system; and the human tutor group did their physics homework problems with the

aid of a human tutor. In their experiment all groups were in the same physics class; the experiment varied only in the way that the students did their homework. The results again showed that the human tutor and the step-based tutoring system groups achieved learning gains but were not reliably different, and yet both were reliably larger than the gains of the no-tutoring group.

Finally, VanLehn, Graesser, & et al. (2007) conducted seven experiments in conceptual physics. In their experiments, all groups of students first studied a short textbook and then worked on several tasks that involved answering conceptual physics questions. For each question, the students wrote a short essay as their initial answer, were tutored on missing or incorrect steps, and then read a correct well-written essay. The treatments differed only in how students were tutored when the essay lacked a step or had an incorrect or incomplete step. There were five treatments. The human tutor group communicated via a text-based interface with an expert human tutor. The Why2-Atlas and Why2-AutoTutor groups used two Natural Language (NL) Dialogue-based Tutoring systems named Why2-Atlas and Why2-AutoTutor respectively. The canned text group read the same tutorial content as Why2-Atlas; in fact this group can be seen as training on a step-based tutoring system that uses text instead of dialogue for getting students to enter a step correctly. Finally, the no-tutoring group read from a textbook without answering conceptual questions.

The results across the seven experiments showed that the learning gains of the three tutoring groups and the canned text group were not reliably different, and were all higher than the read-only no-tutoring group except in one experiment, Experiment 4 (VanLehn, Graesser, & et al., 2007). Experiment 4, found that human tutoring was more effective than reading the canned text. However, upon reviewing transcripts of the tutoring, the authors concluded that the Experiment 4 materials were too far over the students' current level of competence, so reading the canned text's remedial text probably didn't suffice for comprehension, and yet a human tutor was able to help explain the content in novice terms (VanLehn, Graesser, & et al., 2007).

To summarize, once content was controlled to be the same across all groups, neither human tutors nor Natural Language (NL) tutoring systems designed to mimic human tutors, reliably outperformed step-based systems (Evens & Michael, 2006; VanLehn, Graesser, & et al., 2007). All three types of tutors, however, were more effective than no instruction (e.g., students reading material and/or solving problems without feedback or hints). Several techniques can be employed to control for content. For example, in some of the studies described here the domain content was controlled by ensuring students worked on the same training problems with the same human tutors or on a computer tutor that was scripted by the same human tutors (Evens & Michael, 2006; VanLehn, Graesser, & et al., 2007; Reif & Scott, 1999b). Additionally, content can be controlled to be equivalent by running a human tutoring group first, videotaping the tutoring sessions, and then having collaborating pairs of students watch those videotapes (Chi, Roy, & Hausmann, 2008).

One possible conclusion is that tutoring is effective, but that the micro-steps of human tutors and NL tutoring systems provide no additional value beyond conventional step-based tutors (VanLehn, 2008). On the other hand, such a conclusion may be premature. It could simply be that neither human tutors nor their computer mimics are good at making micro-step decisions. That is, the use of micro-steps could potentially be better, but human tutors (and their mimics) may lack the pedagogical skills to make appropriate decisions about which micro-steps to use when.

Do human tutors make effective tutorial decisions?

For any form of one-on-one tutoring, the tutor's behavior can be viewed as a sequential decision process where in, at each discrete step, the tutor is responsible for selecting the next action to take. For instance, some of the tutor turns in Fig. 2 are labeled {**ELICIT**} or {**TELL**}. This label designates a ***tutorial decision step*** wherein the tutor has to decide whether to elicit the requisite information with a question or to tell the student the information. In line 2 in Fig. 2, the tutor chooses to *elicit* the answer from the student by asking the question, "Which principle will help you calculate the rock's kinetic energy at T0? Please provide the name

of the principle, not an equation." If the tutor elected to tell the students, however, then he or she would have stated, "To calculate the rock's kinetic energy at T0, let's apply the definition of Kinetic Energy." Both actions cover the same target knowledge content.

Human tutors must make many rapid decisions in order to keep the tutorial dialogue flowing smoothly. Each of these tutorial decisions affects a student's successive actions and performance. It is often unclear how to make each decision most effectively, because its impact on learning may not be immediate or observed immediately, and more importantly decisions may not be independent of one another; in other words, the effectiveness of one decision also depends on the effectiveness of subsequence decisions.

Pedagogical strategies are defined as policies for deciding the next tutorial action when multiple options are available. It is commonly assumed that the effectiveness of human expert tutors is because they have effective pedagogical strategies (Graesser, Person, & Magliano, 1995). In order to exhibit effective pedagogical strategies, a tutor should adapt his or her behavior to students' needs including students' current knowledge levels, general aptitudes, emotional states and other salient features. However, previous research indicates that human tutors may not make such adaptations (Cade, Copeland, Person, & D'Mello, 2008; Chi, Siler, & Jeong, 2004; Katz, Connelly, & Wilson, 2007; Evens & Michael, 2006; Merrill, Reiser, Ranney, & Trafton, 1992; VanLehn, Siler, Murray, Yamauchi, & Baggett, 2003). For example, Chi, Siler, and Jeong (2004) found that human tutors do not seem to possess an accurate model of students' knowledge levels during the tutoring. Similarly, Putnam (1987) found that experienced tutors do not attempt to form detailed models of students' knowledge before attempting remedial instruction. Rather, each teacher appeared to move through a general curricular script irrespective of a student's state.

To summarize, although it is commonly assumed that human expert tutors have effective pedagogical strategies that lead them to make appropriate tutorial decisions, little evidence has been presented to date demonstrating this. Therefore, one explanation for the lack of difference among human tutors, Natural Language (NL) tutoring systems, and step-based tutoring systems in previous studies is that neither human tutors nor their computer mimics are always good at making micro-step decisions. In order to test this hypothesis, we investigated whether micro-step decision making matters to learning.

Primary research question

We focused on pedagogical strategies that govern tutorial interactions at the level of micro-steps. We use the term *"pedagogical tutorial tactics"* to refer to the pedagogical policies for selecting the tutorial action at each micro-step when there are multiple action options available. Our primary goal is to investigate whether pedagogical tutorial tactics impact students' learning.

In order to investigate the effect of pedagogical tutorial tactics on learning, it was necessary to separate tutorial decisions from instructional content, strictly controlling content so that it is equivalent for all students. It is generally difficult to control tutoring content with human tutors. Computer tutors, on the other hand, permit much greater control over, and tracking of, the tutorial content than do human tutors (Evens & Michael, 2006; VanLehn, Graesser, & et al., 2007; Reif & Scott, 1999b). In this project a Natural Language (NL) tutoring system, named Cordillera, was implemented to teach college students introductory physics.

Tutoring in domains like math and science is often structured as a two-loop procedure. An outer loop selects the problem or task the student should work on next, while the inner loop governs step level decisions during problem solving (VanLehn, 2006). In order to minimize content variation, all participants in this project solved the same training problems in the same order and followed the same major problem-solving steps for each problem. Moreover, the same micro-step *content* was presented to all students regardless of condition even though some students experienced the *tell* version and some experienced the *elicit* version.

In short, our primary research question is whether pedagogical tutorial tactics impact students' learning if the instructional content is controlled so that it is equivalent for all students. Next, we will briefly describe our general approach.

GENERAL APPROACH

Our approach was to apply a general data-driven methodology, Reinforcement Learning (RL), to induce pedagogical tutorial tactics or pedagogical policies directly from pre-existing training corpora. In the context of Reinforcement Learning, it is more appropriate to use the term "pedagogical policies"; while in the context of tutoring and learning it is more appropriate to talk about "pedagogical tutorial tactics". In this paper, the pedagogical policies induced by RL are the same as "pedagogical tutorial tactics" employed on tutoring systems. In the following we use both terms where it is proper.

One corpus used in this project was the Exploratory corpus. It was collected in 2007. 64 college students, the Exploratory group, were trained on a version of Cordillera, called random-Cordillera, where certain tutorial decisions were made randomly.

Previously, we investigated whether the RL-induced pedagogical tutorial tactics would improve students' learning (Chi, Jordan, VanLehn, & Litman, 2009). In that study a set of policies was induced from the Exploratory corpus. Those policies were named DichGain because when applying RL, we used dichotomized learning gains as the reward function so that there were only two levels of reward. The induced DichGain policies replaced the random policy in Cordillera and the new version was named DichGain-Cordillera. Apart from following different policies (random vs. DichGain), the remaining components of Cordillera, including the GUI interface, the training problems, and the tutorial scripts, were left untouched. DichGain-Cordillera's effectiveness was tested by training a new group of 37 college students in 2008. It was shown that no significant overall difference was found between the two groups on the pretest, posttest, or the Normalized Learning Gains (NLGs)[1] (Chi, Jordan, VanLehn, & Litman, 2009; Chi, 2009).

There were at least two potential reasons for this lack of difference. First, it might be caused by limitations in our RL approach; for example, in order to induce the DichGain policies, we defined only 18 features and used a greedy procedure to select a small subset of these for the state representation (Chi, Jordan, VanLehn, & Litman, 2009). Second, rather than randomly assigning students into the two groups, the Exploratory data were collected in 2007 while the DichGain data was collected in 2008.

Therefore, in this study we included multiple training datasets, a larger feature set and more feature selection approaches in our RL approach and ran a full comparison by randomly assigning students to two comparable groups. More specifically, we induced two sets of tutorial tactics: the Normalized Gain (NormGain) tactics were derived with the goal of enhancing tutorial decisions that contribute to learning while the Inverse Normalized Gain (InvNormGain) tactics were derived with the goal of enhancing those decisions that contribute less or even nothing to learning. The two sets of policies were then compared by having all students study the same materials and use versions of Cordillera with identical subject matter, training problems, tutorial scripts and user interface. Because all students studied the same content, we expected all students to learn, even those in the InvNormGain group. If our application of RL to induce pedagogical tutorial tactics is effective, then we expect that the NormGain students will learn *more* than their InvNormGain peers. This would occur if the micro-level decisions on ET and JS impact learning.

[1] $NLG = posttest - pretest/1 - pretest$. Here *posttest* and *pretest* refer to the students' test scores before and after the training respectively; and 1 is the maximum score.

In the sections that follow, the first describes the two types of tutorial decisions, the second explains how we applied RL to induce the pedagogical tutorial tactics in this study, the third describes our methods and includes an introduction to the Cordillera system, procedures, and so on, while the fourth reports our empirical results and some related log analysis. Finally, in the last section we present a post-hoc comparison of all four policy groups.

TWO TYPES OF TUTORIAL DECISIONS

We focused on two types of tutorial decisions: Elicit/Tell (ET) and Justify/Skip-justify (JS). We choose these two micro-step types because there is no widespread consensus on how or when these actions should be taken, as the literature review in the rest of this section shows.

Elicit/tell

During the course of one-on-one tutoring, the tutor often faces a simple decision, to *elicit* the next step information from a student, or to *tell* a student the next step directly. We refer to such tutorial decisions as *elicit/tell (ET) decisions*.

While a lecture can be viewed as a monologue consisting of an unbroken series of tells, human one-on-one tutoring is characterized by a mixture of tutor elicits and tells. Some existing theories of learning suggest that when making tutorial decisions, a tutor should adapt his or her actions to the students' needs based upon their current knowledge level, affective state, and other salient features (Vygotsky, 1971; Collins, Brown, & Newman, 1989; Koedinger & Aleven, 2007). Typically, these theories are considerably more general than the specific interaction decisions that system designers must make. This makes it difficult to instantiate these theories into specific pedagogical policies in ITSs. Therefore, when facing a decision whether to elicit or to tell a new step, most existing ITSs always decide to elicit first (Anderson, Corbett, Koedinger, & Pelletier, 1995; Koedinger, Anderson, Hadley, & Mark, 1997; VanLehn, Lynch, & et al., 2005).

Figure 3 presents a pair of sample dialogues comparing elicit and tell versions of a single tutorial dialogue extracted from Cordillera log files collected during this project. Both dialogues begin and end with the same tutor turns (lines 1 and 6 in (a) and 1 and 4 in (b)). However, in dialogue (a) the tutor chooses to elicit twice (lines 2-3 and 4-5 respectively) while in dialogue (b) the tutor decides to tell twice (lines 2 and 3). Note that the two dialogues cover the *same domain content*.

Justify/skip-justify

The second tutorial decision was whether to execute a justification step. During the tutoring process, human tutors sometimes ask students to *justify* a step they have taken or an entry they have made. It appears that the tutors' goal is to help students understand domain knowledge in a deeper way. They are prompting the student to provide a certain kind of self-explanation (Chi & VanLehn, 1991). The open question is whether or not the tutor should conduct an elaborate discussion of a problem solving step given that this discussion is not necessary for the solution. We refer to such tutorial decisions as *justify/skip-justify (JS) decisions*.

Much previous research including (Chi, de Leeuw, Chiu, & LaVancher, 1994; Conati & Van- Lehn, 2000; Aleven, Ogan, Popescu, Torrey, & Koedinger, 2004) found that asking students to justify their solution steps improves student learning. However, eliciting such a discussion may not always be desirable if, for example, the student is well aware of the rationale. If so, typing in a justification can be slow, frustrating, and distracting. Indeed, in domains like second language acquisition, Wylie et al found that tutors asking

(a) Elicit Version

1. **T:** So let's start with determining the value of v1.
2. **T:** Which principle will help you calculate the rock's instantaneous magnitude of velocity at T1? **{ELICIT}**
3. **S:** definition of kinetic energy
4. **T:** Please write the equation for how the definition of kinetic energy applies to this problem at T1 **{ELICIT}**
5. **S:** ke1 = 0.5*m*v1^2
6. **T:** From KE1 = 0.5*m*v1^2, ⋯

(b) Tell Version

1. **T:** So let's start with determining the value of v1.
2. **T:** To calculate the rocks instantaneous magnitude of velocity at T1, we will apply the definition of kinetic energy again. **{TELL}**
3. **T:** Let me just write the equation for you: KE1 = 0.5*m*v1^2. **{TELL}**
4. **T:** From KE1 = 0.5*m*v1^2, ⋯

Fig. 3. Elicit vs. Tell.

students to justify did not lead to better learning outcomes and significantly increased student training time when compared to a control group that was not asked to enter justifications (Wylie, Koedinger, & Mitamura, 2010). Additionally, Katz, O'Donnell, and Kay (Katz, O'Donnell, & Kay, 2000) found that in some cases it may be better to delay the justifications until the problem has been solved, especially if the justification is abstract, plan-based, or lengthy.

Figure 4 presents a pair of sample dialogues comparing justify and skip-justify versions of a single tutorial dialogue extracted from Cordillera's log files. In part (a), a justification is requested (lines 3-4); while in part (b), the justification request is skipped.

In summary, neither the ET (elicit/tell) nor JS (justify/skip-justify) steps are well-understood, and there is no widespread consensus on how or when they should be used. This is why one of our research objectives is to derive tutorial tactics for them from empirical data. If expert human tutors do not always make optimal, or effective, tutorial decisions, then hand-crafting rules based upon human data may be a suboptimal strategy. Inducing data directly from interaction data may be a more suitable method. In the next sections, we will briefly describe how we applied RL to induce the pedagogical tutorial tactics and then describe our study and finally present our results.

APPLYING RL TO INDUCE NORMGAIN AND INVNORMGAIN PEDAGOGICAL TUTORIAL TACTICS

RL and a Markov Decision Process (MDP)

Previous research on using RL to improve dialogue systems (e.g. (Levin & Pieraccini, 1997; Singh, Kearns, Litman, & Walker, 1999)) has typically used MDPs (Sutton & Barto, 1998) to model dialogue data. An MDP describes a stochastic control process and formally corresponds to a 4-tuple (S, A, T, R), in which:

(a) Justify Version

1. **T:** Can we infer the direction of the velocity of the rock at T1 from the rock's kinetic energy at T1? **{ELICIT}**
2. **S:** nope.
3. **T:** Excellent! Please explain why.**{JUSTIFY, ELICIT}**
4. **S:** Only the magnitude of the velocity and not the direction of it is part of the definition of kinetic energy.
5. **T:** Excellent! Now that we know v1,$\cdots$.

(b) Skip-justify Version

1. **T:** Can we infer the direction of the velocity of the rock at T1 from the rock's kinetic energy at T1? **{ELICIT}**
2. **S:** nope.
3. **T:** Excellent!**{Skip-JUSTIFY}**
4. **T:** Now that we know v1,$\cdots$.

Fig. 4. Justify vs. Skip-justify.

$S = \{S_1, \ldots, S_n\}$ is a state space.
$A = \{A_1, \ldots, A_m\}$ is an action space represented by a set of action variables.
$T : S \times A \times S \rightarrow$ ***[0, 1]*** is a set of transition probabilities between states that describe the dynamics of the modeled system; for example: $P(S_j|S_i, A_k)$ is the probability that the model would transition from state S_i to state S_j by taking action A_k.
$R : S \times A \times S \rightarrow R$ denotes a reward model that assigns rewards to state transitions and models payoffs associated with such transitions.
Additionally, $\pi : S \rightarrow A$ is defined as a policy.

The central idea behind this approach is to transform the problem of inducing effective pedagogical tactics into one of computing an optimal policy for choosing actions in an MDP. Note that in order for RL to be feasible, the number of states and actions should not be too large. On the other hand, when an MDP is constructed for tutorial dialogues, it can have millions of distinct dialogue states and tutor utterances. Here both states and actions in an MDP are abstract.

For instance, if we use only two features to represent the learning context: whether or not a student is engaged and whether or not the student already knows how to do the next step. All possible combinations of these two features would result in four states in the MDP. They are $\{NK, DK, NU, DU\}$ where N = "engaged", D = "disengaged", K = "next step probably known", U = "next step probably not known". Similarly, there might be just two actions in the MDP: "Elicit" in the MDP might denote any information-seeking questions asked by the tutor, and "Tell" represents all other tutorial utterances. Thus, "state" and "action" have different meanings in an MDP versus in a tutorial dialogue. In order to induce a policy from the MDP perspective, there must be deterministic functions for mapping from dialogue states to MDP states and from dialogue actions to MDP actions.

In this project we applied Policy Iteration (Sutton & Barto, 1998) to induce policies. In order to apply Policy Iteration, we need to learn transition probabilities T from a training corpus Γ, which is a collection of system-student tutorial dialogues. A system-student tutorial dialogue is generated as each student solves

a series of training problems on an ITS. For each tutorial dialogue, a scalar performance measure called reward, R, is calculated.

For example, a common choice for R in ITSs is student learning gain. In this project the reward function R is based on Normalized Learning Gain (NLG), which measures a student's gain *factoring out his/her incoming competence*. Following Singh et al. (1999), we can view each system-student interaction log d_i as a trajectory in the MDP state space determined by the system actions and student responses as follows:

$$s^1_{d_i} \xrightarrow{a^1_{d_i}, r^1_{d_i}} s^2_{d_i} \xrightarrow{a^2_{d_i}, r^2_{d_i}} \cdots s^{n_{d_i}}_{d_i} \xrightarrow{a^{n_{d_i}}_{d_i}, r^{n_{d_i}}_{d_i}}$$

Here $s^j_{d_i} \xrightarrow{a^j_{d_i}, r^j_{d_i}} s^{j+1}_{d_i}$ indicates that at the j_{th} turn in the tutorial dialogue d_i, the system is in MDP state $s^j_{d_i}$, executes MDP action $a^j_{d_i}$, receives MDP reward $r^j_{d_i}$, and then transitions to MDP state $s^{j+1}_{d_i}$. The number of turns in d_i is n_{d_i}. In this project, only terminal dialogue states have non-zero rewards because a student's learning gain is measured after the entire tutorial dialogue is completed.

For instance, suppose the MDP state space $= \{NK, DK, NU, DU\}$ as described above and the MDP actions are E, T where E = Elicit and T = Tell. Then the dialogue fragment of Fig. 2 *might* be represented as:

$$DU \xrightarrow{E,0} NU \xrightarrow{T,0} NU \xrightarrow{T,0} NK \xrightarrow{E,0} NK \xrightarrow{T,0}$$

Dialogue sequences obtained from the training corpus Γ then can be used to empirically estimate the transition probabilities T as: $T = \{p(S_j|S_i, A_k)\}^{k=1,\cdots,m}_{i,j=1,\cdots,n}$. More specifically, $p(S_j|S_i, A_k)$ is calculated by taking the number of times that the dialogue is in MDP state S_i, the tutor took MDP action A_k, and the dialogue was next in state S_j and dividing by the number of times the dialogue was in S_i and the tutor took A_k. The reliability of these estimates depends upon the size and structure of the training data.

Once an MDP model has been built, calculation of an optimal policy is straightforward. For this project we employed an RL toolkit developed by Tetreault and Litman (Tetreault & Litman, 2008), which uses a dynamic programming algorithm for policy iteration (Sutton & Barto, 1998). The code was originally built on the MDP toolkit written in Matlab (Chades, Garcia, & Sabbadin, 2005). The purpose of this algorithm is to handle the problem of reward propagation.

The RL toolkit developed by Tetreault and Litman requires all state features in the model to be discrete variables. However, most of the features involved in this project are numeric and had to be discretized before a suitable MDP could be constructed. Our discretization procedure used two clustering procedures: the TwoStep procedure which bounded the number of clusters in SPSS and the K-means procedure which used K-means clustering to locate optimal cluster centers. Other discretization procedures such as a simple median split can also be applied.

So far we have described the abstract methodology by which pedagogical policies are induced when $< S, A, R >$ are defined and T is estimated from a given training corpus Γ. While this approach is theoretically appealing, the cost of obtaining human tutorial dialogues makes it crucial to limit the size of the MDP state space so that all possible transitions are observed a sufficient number of times to obtain a reliable estimate T, while still retaining enough information in the states to represent accurately the human population and learning context.

So the main problem addressed here is to choose S, the set of MDP states, because we are already committed to using a reward function based on learning gains and our action space A is also clearly defined. More specifically, we will use $A = \{Elicit, Tell\}$ for inducing ET policies and using $A = \{Justify, Skip - Justify\}$ for inducing JS policies respectively. Once S is selected, it is a mechanical process to induce the MDP from the data and then calculate optimal policies.

In this project several approaches have been employed to select an appropriate MDP state, S. These approaches include applying a series of feature-selection methods to select a state representation from a large set of features, using multiple training corpora, and finding different MDPs and policies for different knowledge components. The rest of this section presents a few of the critical details of the process, but many others must be omitted to save space. Next, we will briefly describe our three approaches.

Applying feature selection to induce pedagogical tutorial tactics

For RL, as with all machine learning tasks, success depends upon an effective state representation S. Ideally it should include an abstraction of the relevant dialogue history that will be necessary to determine the effects of each action in each state. In particular, for each state in the MDP, every action taken at that state should occur enough times in the training corpus so that the distribution of resulting states can be reliably estimated. However, getting enough instances of each state-action pair isn't the only challenge. The state representation must be such that different actions from the same state tend to lead to different states; otherwise, a policy cannot have any impact on the rewards. The policy decides which action the tutor should take in a given state; if a different action choice doesn't affect the subsequent state, then the induced policy may not be effective. The challenge thus lies in identifying a set of features that allows an effective policy to be induced.

While much previous research on the use of RL to improve ITSs and Dialogue Systems has focused on developing the best policy for a *given* set of features (Beck, Woolf, & Beal, 2000; Iglesias, Martínez, Aler, & Fernández, 2009a,b; Iglesias, Martínez, & Fernández, 2003; Walker, 2000; Henderson, Lemon, & Georgila, 2005), our approach in this project was to begin with a large set of features to which a series of feature-selection methods was applied to reduce them to a tractable subset. More specifically 50 state features were defined based upon six categories of features considered by previous research (Moore, Porayska-Pomsta, Varges, & Zinn, 2004; Beck, Woolf, & Beal, 2000; Forbes-Riley, Litman, Purandare, Rotaru, & Tetreault, 2007) to be relevant for making tutorial decisions; and we applied twelve feature selection methods. More details on the 50 features and twelve feature selection methods can be found in (Chi, 2009).

To distinguish from the specific state representation S used in the RL and MDP formalisms, we use Ω to represent a large set of potential state representations. In other words, for any induced policy in this project, its corresponding state space S is a subset of Ω. More specifically our RL task becomes, for a given $< A, R >$ and Γ, how to select a subset S from a large set of potential states Ω that would generate the best policy. Moreover we assume that S and Ω are defined by Cartesian products of features so selecting S means selecting a subset of the features that define Ω.

In order to do feature selection, first we need to decide the maximum number of features to be used in defining S. The number should be small so that we have enough training data to cover the state-action pairs, yet large enough that the selected features encode enough information to make good decisions about actions. In order to determine the maximum number of features, it is necessary to consider the amount of available data and computational power. In this project, based on the minimum data available from the training corpora, we decided to use at most 6 binary features ($\hat{m} = 6$), which means that there can be as many as $2^6 = 64$ states in S.

Given a particular selection of features and a fixed training corpus and action representation, the methods mentioned earlier calculate the optimal policy. So for every S, there is exactly one optimal policy π. In order to compare policies from different Ss, we use a measure called the expected cumulative reward (ECR). Even though a policy is induced, different traversals of the state space may result in different paths and hence different rewards. The ECR of a policy is the reward averaged over a very large number of traversals. The higher the ECR value of a policy, the better the policy is supposed to perform. So our goal is to choose a set of features that maximizes the ECR of the resulting policy. ECR has been widely used as a criteria

for evaluating policies in the study of policy induction such as when applying RL to induce policies from simulated corpora (Janarthanam & Lemon, 2009; Williams & Young, 2007a,b).

To summarize, once <A, R> was defined and the system-student interactivity training corpus Γ provided, we applied twelve feature selection methods to select a subset of features from the large set defined in Ω. Recall that the number of features used to define S was limited to no more than 6. The goal is to find features that maximize the ECR of the resulting policy.

Three training corpora

In order to improve the effectiveness of the RL-induced policies, we used three training corpora. The Exploratory corpus $\Gamma_{exploratory}$ consisted of 64 complete tutorial dialogues from students who used a version of Cordillera that chose actions randomly; the DichGain corpus $\Gamma_{DichGain}$ contained 37 tutorial dialogues from students who used a version of Cordillera that chose actions according to a policy derived from the first round of RL and featured a dichotomous learning gain reward function; and the combined corpus $\Gamma_{combined}$ comprised a total of 101 dialogues from the Exploratory and DichGain groups.

$\Gamma_{exploratory}$ was collected for RL and designed to explore the feature space evenly and without bias. $\Gamma_{DichGain}$, by contrast, is similar to many other pre-existing corpora by following a set of specific pedagogical strategies. Inducing a successful policy from $\Gamma_{DichGain}$ would show the potential for applying RL to induce effective tutorial policies from most pre-existing data. $\Gamma_{combined}$, in theory, offers the benefits of both as well as an increased dataset.

When inducing NormGain and InvNormGain policies, rather than selecting one training corpus Γ a priori, all three were used. More specifically, a set of tutorial policies were derived from each training corpus separately and then the best policy from all sets was selected by ECR.

Inducing Knowledge Component (KC)-specific Policies

Finally, in order to improve the effectiveness of the RL-induced policies, we induced KC-specific policies. In the learning literature it is commonly assumed that relevant knowledge in domains such as math and science is structured as a set of independent but co-occurring Knowledge Components (KCs) and that KCs are learned independently. A KC is "a generalization of everyday terms like concept, principle, fact, or skill, and cognitive science terms like schema, production rule, misconception, or facet" (VanLehn, Jordan, & Litman, 2007). For ITSs these are atomic units of knowledge. It is assumed that a tutorial dialogue about one KC (e.g., kinetic energy) will have no impact on the student's understanding of any other KC (e.g., gravity). This is an idealization, but it has served ITS developers well for many decades, and is a fundamental assumption of many cognitive models (Anderson, 1983; Newell, 1994).

When dealing with a specific KC, the expectation is that the tutor's best policy for teaching that KC (e.g., when to Elicit vs., when to Tell) would be based upon the student's mastery of the KC in question, its intrinsic difficulty, and other relevant but not necessarily known factors specific to that KC. In other words an optimal policy for one KC might not be optimal for another. Therefore, one assumption made in this project is that *inducing pedagogical policies specific to each KC would be more effective than inducing an overall KC-general policy*.

The domain chosen for this project is the physics work-energy domain, which is a common topic in introductory college physics courses. Two domain experts, who are also knowledge representation experts (not the authors), identified 32 KCs in the domain. They had experience identifying KCs for a series of previous studies involving college physics. Note that a complicated domain like physics can often be broken into many KCs. Here the 32 identified KCs are believed to cover the most important knowledge in the domain.

Generally speaking, in a domain like math and physics, solving a problem mainly involves applying major domain principles. The major domain principles are more challenging and important than the other KCs since the student's overall learning performance depends more on learning domain principles. Among the 32 KCs, eight are major domain principles.

When inducing NormGain or InvNormGain policies, the decision was made to focus only on the eight primary KCs corresponding to the eight major domain principles. Therefore, the overall problem of inducing a policy for ET decisions and a policy for JS decisions is decomposed into 8 sub-problems for each decision type, one per KC. More specifically, to learn a policy for each KC, we annotated the tutorial dialogues with the KCs covered during each dialogue and the tutorial action decisions with the KCs covered by each action. For each KC the final kappa was ≥ 0.77, which is fairly high given the complexity of the task. A domain expert also mapped the pre- and post-test problems to relevant KCs. By doing so a KC-specific NLG score could be generated for each student.

Among the eight KCs, KC_1 does not arise in any JS decisions and thus only an ET policy was induced for it. For each of the remaining seven KCs a pairs of policies, one ET policy and one JS policy, were induced. So we induced a total of 15 KC-specific policies. During tutoring there were some decision steps that did not involve any of the eight primary KCs. For those steps, two KC-general policies, an ET policy and a JS policy, were induced. Cordillera applies a KC-general policy only when no KC-specific policy is relevant. Thus a total of 17 NormGain and 17 InvNormGain policies were induced.

Summary

In this project our primary research question was whether pedagogical tutorial policies for micro-steps would impact students' learning. In order to investigate our research question, we focused on two types of tutorial decisions: ET and JS. We applied RL to induce two sets of tutorial policies: the Normalized Gain (NormGain) policies and the Inverse Normalized Gain (InvNormGain) policies.

The induction of the NormGain and the InvNormGain policies shared the same general RL policy induction procedure. More specifically, in our policy induction procedure, twelve feature selection methods were used to select up to six features from a total of 50 features, three training corpora were investigated, and 15 KC-specific policies and two KC-general policies were induced. The only difference between the NormGain policies and the InvNormGain policies is the definition of the reward functions. Although the reward functions for inducing both sets were based on Normalized Learning Gain (NLG), the NormGain tutorial tactics were induced by using $\textit{the student's NLG} \times 100$ as the final reward; while the InvNormGain ones were induced by using $(1 - \textit{the student's NLG}) \times 100$ as the final reward. To sum, a total of 17 NormGain and 17 InvNormGain policies were induced.

An example induced policy

In this project we defined 50 features for Ω. After running the twelve feature selection methods, 17 NormGain and 17 InvNormGain policies were induced. One of the NormGain policies is shown in Fig. 5. The policy in Fig. 5 has three state features:

[StepSimplicityPS $[0, 0.38) \rightarrow 0; [0.38, 1] \rightarrow 1$**]:** encodes a step's simplicity level. Its value is estimated from the training corpus based on the percentage of correct answers given for the dialogue state. The discretization procedure binarized the feature into two values: 0 and 1. If less than 38% of the answers given were correct, then this is considered to be 'difficult' content and we set StepSimplicityPS = 0; Otherwise, StepSimplicityPS = 1.

[TuConceptsToWordsPS $[0, 0.074) \rightarrow 0; [0.074, 1] \rightarrow 1$**]:** represents the ratio of physics concepts to words in the tutor's utterances so far. The higher this value, the greater the percentage of physics content

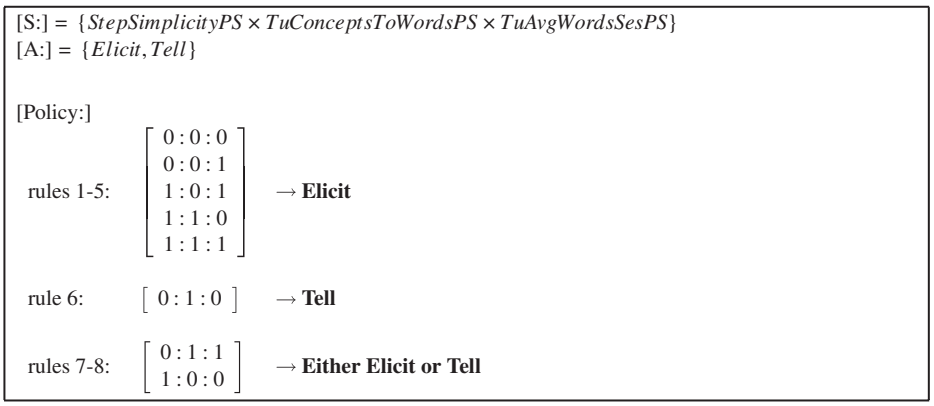

Fig. 5. An Example Of a NormGain policy for an ET Decisions.

that may have been included in tutor turns. Dialogue states with less than 7.4% on this measure have TuConceptsToWordsPS = 0 and 1 otherwise.

[TuAvgWordsSesPS $[0, 22.58) \rightarrow 0; [22.58, \infty) \rightarrow 1$**]:** encodes the average number of words in tutor turns in this session. This feature reflects how verbose the tutor is in the current session. The discretization procedure set the threshold at 22.58, so dialogue states where the tutor had an average of more than 22.58 words per tutor turn for the current session were represented with 1 for TuAvgWordsSesPS and 0 otherwise.

Since each of the three features was discretized into 2 values, a three-feature state representation resulted in a state space of $2^3 = 8$ states. Thus, 8 pedagogical rules are learned. Each rule reads as, "In <state> choose <action(s)>." If a rule has two actions as its choice for the "best action to do," that means that the ECRs tied during induction. Figure 5 shows that in 5 states the tutor should elicit (rules 1-5), in one state it should tell (rule 6); in the remaining 2 states either will do (rules 7-8).

For example, let's unpack rule 6 since it is the only situation in which the tutor should tell. For this rule to apply the state must be [0 : 1 : 0], which represents the values of the three corresponding features: StepSimplicityPS, TuConceptsToWordsPS and TuAvgWordsSesPS respectively. Rule 6 suggests that when the next dialogue content step is difficult (StepSimplicityPS is 0), the ratio of physics concepts to words in the tutor's turns so far is high (TuConceptsToWordsPS is 1), and the tutor has not been very wordy during the current session (TuAvgWordsSesPS is 0), then the tutor should tell. As you can see a three-feature policy is quite subtle and adaptive to the learning context.

The resulting 17 NormGain and 17 InvNormGain policies were added to Cordillera to yield two new versions of the system, named NormGain-Cordillera and InvNormGain-Cordillera respectively. The induced tutorial tactics were evaluated on human subjects to see whether the NormGain students would out-perform their InvNormGain peers.

METHODS

Cordillera

Cordillera is an NL Tutoring System teaching students introductory college physics (VanLehn, Jordan, & Litman, 2007), which is based upon the TuTalk NL tutorial dialogue toolkit (Jordan, Ringenberg, & Hall, 2006; Jordan, Hall, Ringenberg, Cue, & Rosé, 2007). TuTalk is an authoring tool which enables domain

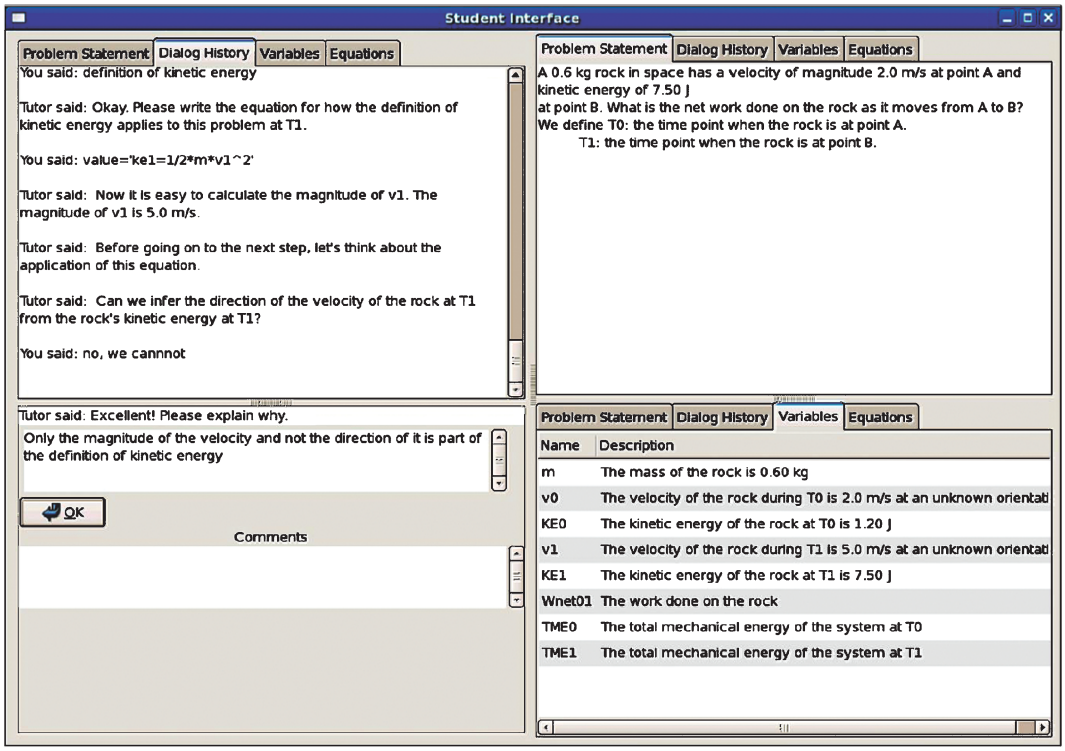

Fig. 6. Cordillera student interface.

experts to construct natural language tutoring systems without programming. Instead, domain experts focus on defining the tutoring content through writing tutorial scripts, which are then used for automating interactions. In other words, script authors determine the flow of the dialogue and the content of each tutor turn.

The student interface is used by students to read the tutor's tutorial instructions and to answer questions by means of natural language entries. Figure 6 shows a screen shot of the student interface. The Message Window, located in the bottom-left corner is where the dialogue interaction takes place. The remaining panes are the Dialogue History Pane (upper-left), Problem Statement pane (upper-right), and Variable Pane (lower-right). The Equations Pane, which lists equations entered by the student or the tutor so far, is not shown in the figure, but can be exposed by clicking on the appropriate tab.

To reduce confounds due to imperfect NL understanding, the NL understanding module in Cordillera was replaced with a human interpreter called the language understanding wizard (Bernsen & Dybkjaer, 1997). In this format, Cordillera works as a communications framework that connects a student interface to a wizard interface. The *only* task performed by the human wizards is to match students' answers to the closest response from a list of potential responses; they cannot make the tutorial decisions. From the wizard interface, information such as the student's identification or condition is not presented. In this study only one human wizard (the first author) was involved and in most wizard sessions 3-6 participants' answers were matched at the same time.

In this project, different versions of Cordillera were constructed, each of which differed only in terms of the pedagogical policies employed. The remaining components of the system, including the GUI interfaces and domain experts' tutorial scripts, were identical for all participants. In Cordillera the pedagogical policies are used to make two types of tutorial decisions.

Participants

Data was collected over a period of two months during the summer of 2009. Participants were 64 college students who received payment for their participation. They were required to have a basic understanding of high-school algebra. However, they could not have taken any college-level physics courses. Students were randomly assigned to the two conditions. Each took from one to two weeks to complete the study over multiple sessions. In total, 57 students completed the study (29 in the NormGain group and 28 in the InvNormGain group).

Domain and eight knowledge components

The domain chosen for this project, Physics work-energy domain, is a common topic of introductory college physics courses. As mentioned before, two domain experts identified 32 KCs for this domain, and we focused only on the KCs corresponding to the eight major domain principles shown in Table 1. In Table 1 the first column lists its corresponding KC number. The second column describes the name of the principle. The last column is the formula or mathematical expression of the principle.

The remaining 24 KCs primarily cover various definitions of major physics concepts and important physics facts in the domain. For example, major physics concepts include the definition of gravitational force (KC_2), spring force(KC_3) , normal force (KC_4), isolated system (KC_{25}), and so on. Important physics facts include examples such as that the unit for work is the Joule (J)(KC_{15}), that when an object slows to a stop and reverses direction its velocity is momentarily zero (KC_{30}), that the unit for velocity is m/s (KC_{31}), and so on.

Procedure

The participants in this study experienced the same procedure and materials as the participants in the Exploratory and DichGain studies conducted earlier. More specifically, all participants in this project experienced the same five standard phases: 1) background survey, 2) pre-training, 3) pre-test, 4) training, and 5) post-test. Unless specified explicitly in the following, the procedure, reading contents, training materials, GUI, test items, and so on were identical across all groups and in each phase there were no time limits.

Table 1
Major principles of work and energy

KC	Principle description	Expressions
KC_1	Weight Law (w)	$W = mg$
KC_{14}	Definition of Work (W)	$W = Fdcos(\alpha)$
KC_{20}	Definition of Kinetic Energy (KE)	$KE = \frac{1}{2}mv^2$
KC_{21}	Gravitational Potential Energy (GPE)	$GPE = mgh$
KC_{22}	Spring Potential Energy (SPE)	$SPE = \frac{1}{2}kd^2$
KC_{24}	Total Mechanical Energy (TME)	$TME = KE + GPE + SPE$
KC_{27}	Conservation of Total Mechanical Energy (CTME)	$TME_1 = TME_2$
KC_{28}	Change of Total Mechanical Energy for Non-isolated Systems (TMENC)	$Net_W = TME_2 - TME_1$

The background survey asked students for demographic information such as gender, age, SAT scores, high school GPA, experience with algebra, calculus, physics, and other information.

Following the background survey, students read the physics textbook during the pre-training and took the pre-test. The physics textbook was only available during phase 2, pre-training.

In the training phase, students were first trained to solve a demonstration problem, which did not include physics content, on Cordillera. The sole purpose of this step was to familiarize them with the GUI interface. Both the NormGain group and the InvNormGain group then solved the same seven training problems in the same order on their versions of Cordillera. Except for the policies (NormGain vs. InvNormGain), all components of Cordillera, including the GUI interface and the tutorial scripts, were identical for all students.

Finally, students took the post-test. The pre- and post-tests were identical. Both contained a total of 33 problems selected from the physics literature by two domain experts (not the authors). The 33 problems covered 168 KC applications. The tests were given online and consisted of both multiple-choice and open-ended questions. Open-ended questions required the students to derive an answer by writing or solving one or more equations. Once an answer was submitted, students automatically proceeded to the next question without receiving any feedback on the correctness of a response. Students were not allowed to return to prior questions.

As mentioned above, normalized learning gains (NLGs) were used as the reward functions when inducing NormGain and InvNormGain tutorial policies. Therefore, we used identical pre- and post-tests in order to avoid the need to rescale the test to make their scores compatible. Students were not informed that the tests would be identical at any point; they received no feedback on their test answers or test scores; and the minimum time between the pre- and posttest was one week.

Grading

All tests were mixed together and graded in a double-blind manner by a single experienced grader (the first author). In a double-blind procedure, neither the students nor the grader know who belongs to which group. To keep the grading consistent, students' test answers were graded problem-based. More specifically, all students' answers on one test problem (both pre- and posttest) were graded together. The grader was not informed who an answer belonged to nor which test (pretest or posttest) the answer belonged to.

For all identified relevant KCs in a test question, a KC-based score for each KC application was given. In the following sections, when we need to compare the NormGain group to the InvNormGain group, we use the sum of these KC-based scores. A later analysis (not presented here) showed that using other scoring rubrics resulted in essentially the same pattern of results as this scoring rubric Chi (2009). The tests contained 33 test items which covered 168 KC occurrences. For comparison purposes all test scores were normalized to fall in the range of [0, 1].

RESULTS

Learning performance

Overall learning performance

Random assignment appears to have balanced the incoming student competence across conditions. There were no statistically significant differences between the two conditions in the pretest scores $t(55) = 0.71$, $p = 0.484$. Additionally, no significant differences were found between the two conditions

on the math SAT scores and the total training time spent on Cordillera: $t(39) = 0.536$, $p = 0.595$ and $t(55) = -.272$, $p = 0.787$ respectively.

A one-way ANOVA was used to test for learning performance differences between the pre- and posttests. Both conditions made significant gains from pretest to posttest: $F(1, 56) = 31.34$, $p = 0.000$ for the NormGain condition and $F(1, 54) = 6.62$, $p = 0.013$ for the InvNormGain condition. Table 2 compares the pretest, posttest, adjusted-posttest, and NLG scores between the two conditions. In Table 2, the adjusted posttest scores for each condition were calculated by running an ANCOVA using the pretest score as the covariate. The second and third columns in Table 2 list the means and standard deviations of the NormGain and InvNormGain groups' corresponding scores. The fourth column lists the corresponding statistical comparison. The fifth column lists the effect size (Cohen's d), which is the mean of the experimental group minus the mean of the control group, divided by the groups' pooled standard deviation. Table 2 shows that there was no significant difference between the two groups on pretest scores. However, there were significant differences between the two groups on the posttest, adjusted-posttest, and NLG scores. Across all measurements, the NormGain group performed significantly better than the InvNormGain peers. The effect size was large.

Since KC-based tutorial tactics were induced, it would be interesting to compare the two groups' performance on a KC by KC basis. So next we will investigate whether students learned on all eight primary KCs.

KC-based learning performance

In the pretest on a KC by KC basis, no significant difference was found between the two conditions across all eight primary KCs except that on KC_{27}, the NormGain group scored only marginally higher than the InvNormGain group: $t(55) = 1.74$, $p = 0.088$ (see Table 3). In order to account for varying pretest scores, adjusted posttest scores were calculated by running an ANCOVA using the corresponding pretest score as the covariate.

On a KC by KC basis, Table 3 summarizes the comparisons on the pretest and adjusted posttest scores between the two conditions. The third and fourth columns in Table 3 list the means and standard deviations of the NormGain and InvNormGain groups' pretest or adjusted posttest scores on the corresponding KC. The fifth column lists the corresponding statistical comparison. The sixth column lists effect size (Cohen's d). In short, Table 3 shows that the NormGain condition out-performed the InvNormGain across all primary KCs (in bold) except for KC_{28}, on which no significant difference was found between the two groups.

Summary of learning performance

As expected, both NormGain and InvNormGain groups had significant learning gains after training. More importantly, although no significant difference was found in time on task, MSAT scores, and pretest scores, the NormGain group out-performed the InvNormGain group on the posttest and NLG scores. On a KC by

Table 2
Normgain vs. InvNormGain on various test scores

	NormGain	InvNormGain	Stat	Cohen's d
Pretest	0.42 (0.16)	0.39 (0.23)	$t(55) = 0.71$, $p = 0.484$	0.15
Posttest	0.65 (0.15)	0.54 (0.20)	$t(55) = 2.32$, $p = 0.024$	0.65
Adjusted Posttest	0.63 (0.095)	0.55 (0.095)	$F(1, 54) = 10.689$, $p = 0.002$	0.86
NLG	0.41 (0.19)	0.25 (0.21)	$t(55) = 3.058$, $p = 0.003$	0.81

Table 3
Between-group comparison on pretest and adjusted posttest scores across primary KCs

KC	TestScore	NormGain	InvNormGain	Stat	d
KC_1	Pretest	0.42 (0.15)	0.39 (0.22)	$t(55) = 0.66$, $p = 0.51$	0.16
	Adjusted Posttest	0.64 (0.12)	0.54 (0.12)	$F(1, 54) = \mathbf{9.80}$, $p = \mathbf{0.0028}$	**0.85**
KC_{14}	Pretest	0.43 (0.23)	0.44 (0.25)	$t(55) = -0.17$, $p = 0.86$	–0.04
	Adjusted Posttest	0.65 (0.17)	0.53 (0.17)	$F(1, 54) = \mathbf{6.47}$, $p = \mathbf{0.014}$	**0.72**
KC_{20}	Pretest	0.38 (0.17)	0.37 (0.22)	$t(55) = 0.31$, $p = 0.76$	0.05
	Adjusted Posttest	0.67 (0.11)	0.58 (0.11)	$F(1, 54) = \mathbf{10.30}$, $p = \mathbf{0.002}$	**0.83**
KC_{21}	Pretest	0.45 (0.20)	0.43 (0.24)	$t(55) = 0.35$, $p = 0.72$	0.09
	Adjusted Posttest	0.75 (0.13)	0.65 (0.13)	$F(1, 54) = \mathbf{7.62}$, $p = \mathbf{0.008}$	**0.78**
KC_{22}	Pretest	0.42 (0.25)	0.39 (0.26)	$t(55) = 0.41$, $p = 0.68$	0.12
	Adjusted Posttest	0.63 (0.17)	0.51 (0.17)	$F(1, 54) = \mathbf{7.77}$, $p = \mathbf{0.007}$	**0.72**
KC_{24}	Pretest	0.46 (0.15)	0.41 (0.23)	$t(55) = 0.89$, $p = 0.38$	0.26
	Adjusted Posttest	0.64 (0.11)	0.58 (0.11)	$F(1, 54) = \mathbf{4.22}$, $p = \mathbf{0.045}$	**0.56**
KC_{27}	Pretest	0.53 (0.21)	0.42 (0.24)	$t(55) = 1.74$, $p = 0.088$	0.5
	Adjusted Posttest	0.74 (0.18)	0.63 (0.18)	$F(1, 54) = \mathbf{5.88}$, $p = \mathbf{0.019}$	**0.62**
KC_{28}	Pretest	0.37 (0.20)	0.36 (0.26)	$t(55) = 0.13$, $p = 0.90$	0.04
	Adjusted Posttest	0.53 (0.17)	0.47 (0.17)	$F(1, 54) = 1.61$, $p = 0.21$	0.36

KC basis, the NormGain condition also out-performed the InvNormGain across all primary KCs (in bold) except one. Overall, the results show that the tutorial decisions on micro-steps made a significant difference in the students' learning.

To summarize, the results are consistent with the primary research hypothesis. The NormGain condition indeed out-performed the InvNormGain condition. In order to investigate why the NormGain tutorial tactics were more effective than the InvNormGain one, it will be necessary to dig into the logs and make a detailed comparison of the differences between the two sets of tutorial tactics. For example, the induced NormGain tutorial tactics might simply elicit more answers from the students or execute more justification steps during the tutoring. Therefore, the following section will investigate whether the NormGain and InvNormGain tutorial tactics resulted in different patterns of tutorial actions.

Log analysis

Overall log analysis

As mentioned earlier, we focused on two types of tutorial actions: Elicit/Tell (ET) and Justify/Skip-Justify (JS). To quantify the relative frequency of these decisions, an Interactivity ratio (I-ratio) and Justification ratio (J-ratio) are defined as:

$$\mathbf{I-ratio} = \frac{N_{Elicit}}{N_{Elicit} + N_{Tell}} \tag{1}$$

$$\mathbf{J-ratio} = \frac{N_{Justify}}{N_{Justify} + N_{SkipJustify}} \tag{2}$$

The higher the I-ratio is, the more interactive the dialogue is (one view of interactivity). The higher the J-ratio is, the more likely the students would be presented a justification step. In order to characterize

Table 4
Overall characteristics of tutorial decisions in exploratory corpus

		NormGain (29)	InvNormGain (28)	Stats
1	Tell	63.759 (19.528)	63.250 (4.656)	$t(55) = 0.134$, $p = 0.894$
2	Elicit	198.586 (17.463)	204.000 (7.679)	$t(55) = -1.506$, $p = 0.138$
3	ET Decisions	262.345 (6.149)	267.250 (6.775)	$\mathbf{t(55) = -2.864, p = 0.006}$
4	Skip-Justify	9.345 (3.829)	11.000 (1.700)	$\mathbf{t(55) = -2.096, p = 0.041}$
5	Justify	42.517 (3.786)	40.321 (1.442)	$\mathbf{t(55) = 2.874, p = 0.006}$
6	JS Decisions	51.862 (0.833)	51.321 (1.156)	$\mathbf{t(55) = 2.030, p = 0.047}$
7	Overall Decisions	280.103 (4.126)	285.464 (6.995)	$\mathbf{t(55) = -3.539, p = 0.001}$
8	I-ratio	0.758 (0.073)	0.763 (0.018)	$t(55) = -0.395$, $p = 0.694$
9	J-ratio	0.820 (0.073)	0.786 (0.030)	$\mathbf{t(55) = 2.273, p = 0.027}$

the log data, the I-ratio, J-ratio and several other easily obtained measures were calculated. The goal is to see whether the NormGain tutorial tactics resulted in different tutorial behaviors from the InvNormGain policies when viewed from this shallow aspect.

Table 4 summarizes the shallow measures and compares them between the NormGain and InvNormGain tutorial dialogues. It also includes the I-ratio and J-ratio measures, which will be discussed in the following two sections. The other shallow measures include the average number of tell decisions (row 1), elicit decisions (row 2), ET decisions (row 3), skip-justify decisions (row 4), justify decisions (row 5), JS decisions (row 6), overall decisions (row 7), I-ratio (row 8) and J-ratio (row 9). Table 4, shows that except for the total number of tells (row 1) and elicits (row 2), the two groups differed significantly on the remaining measures. Although 5 of the measures were statistically different, the absolute differences between them were small. This suggests that these characteristics are not revealing what causes the learning gain differences between the two groups. Perhaps the I-ratio and J-ratio, which are discussed next, will shed light on how the NormGain policy caused more learning than the InvNormGain policy.

Comparing I-ratio across primary KCs

Although no significant difference was found between the two groups on the I-ratio overall, once the dialogue was broken into a KC by KC basis there were significant differences between the two groups on each of the eight primary KCs (see Table 5). In Table 5, row 2 shows that on KC_{14} the NormGain group got all elicits while the InvNormGain group got all tells. Among the rest of seven primary KCs, the NormGain condition was more likely to get elicits than the InvNormGain condition on KC_{20}, KC_{21}, and KC_{22}; and the InvNormGain condition was more likely to get elicits than the NormGain condition on KC_1, KC_{24}, KC_{27}, and KC_{28}.

Recall that Table 3 shows that the NormGain condition out-performed the InvNormGain across all primary KCs (in bold) except for KC_{28}, on which no significant difference was found between the two groups. Overall, our results seemingly suggest that elicits work better on KC_{14}, KC_{20}, KC_{21}, and KC_{22}, while tells work better on KC_1, KC_{24}, KC_{27}. There are many possible explanations for such phenomena. Based on the number of ET tutorial decisions made on each KC, we can classify the eight KCs into three categories. Next, we will explain the phenomena on a category by category basis.

KC_{20}, KC_{21}, and KC_{24} are high occurrence KCs in that each of them are involved in more than 50 ET tutorial decisions. Among the three KCs, KC_{20} and KC_{21} both involve multiplying several physics quantities while KC_{24} involves summing over physics quantities. Perhaps because summation is much

Table 5
Compare normgain vs. invnormgain on I-ratio across eight primary KCs

		NormGain(29)		*InvNormGain (28)*	*Stats*
1	KC_1	0.500 (0.000)	<	0.696 (0.157)	$\mathbf{t(55) = -6.72, p = 0.000}$
2	KC_{14}	1.000 (0.000)		0.000 (0.000)	
3	KC_{20}	0.897 (0.024)		0.696 (0.030)	$\mathbf{t(55) = 27.87, p = 0.000}$
4	KC_{21}	0.923 (0.030)		0.863 (0.045)	$\mathbf{t(55) = 5.95, p = 0.000}$
5	KC_{22}	0.888 (0.099)		0.543 (0.089)	$\mathbf{t(55) = 13.88, p = 0.000}$
6	KC_{24}	0.866 (0.028)	<	0.920 (0.029)	$\mathbf{t(55) = -7.21, p = 0.000}$
7	KC_{27}	0.484 (0.137)	<	0.651 (0.112)	$\mathbf{t(55) = -5.03, p = 0.000}$
8	KC_{28}	0.000 (0.000)	<	0.525 (0.108)	$\mathbf{t(55) = -26.08, p = 0.000}$

easier to understand than multiplication, this is why NormGain policies chose to elicit more frequently on the former two KCs and tell more frequently on KC_{24}. The next category is the medium occurrence KCs, which include KC_{22}, KC_{27}, and KC_{28}. Each of these three KCs occurred in the range of 17 to 28 ET decisions. Among the three KCs, KC_{22} involves multiplying several physics quantities while KC_{27} and KC_{28} involve equating or summing over physics quantities. As with the three high occurrence KCs, the multiply-vs-summation difference may explain why elicits were more effective for KC_{22} but not KC_{28}, and tells were more effective on KC_{27}. Finally, the remaining two KCs, KC_1 and KC_{14}, were involved in less than 10 ET decisions and thus they are the low occurrence KCs. Our results suggest that tells seem to work better on KC_1 while elicits seem to work better on KC_{14}. While both KC_1 and KC_{14} involve multiplication of physics quantities, the concept of gravitational force (KC_1) may not be completely new to many of the participants, even physics novices, while the definition of work would be more likely to be unfamiliar to these physics novices. Overall, these are only hypothetical explanations and further research is needed for the reasons behind such results.

Comparing J-ratio across primary KCs

Similarly, the J-ratio can be examined on a KC by KC basis. Only seven primary KCs (KC_1 was not involved in JS decisions) were involved (see Table 6). Surprisingly, on two KCs, KC_{22} (row 4) and KC_{28} (row 7), both NormGain and InvNormGain tutorial tactics achieved the same results, executing all justification steps. There are at least two potential explanations. One possible explanation is that the JS decisions on these KCs may not matter to the students' learning. The other possible explanation is that the source training corpora used to induce these two KC-specific policies might not be exploratory enough.

On KC_{14}, however, following the NormGain tutorial tactics resulted in skipping all justification steps, but following the InvNormGain tutorial tactics resulted in executing all justification steps. For the remaining four KCs, no significant difference was found between the two conditions on KC_{20} (row 2) and KC_{24} (row 5). Only a marginally significant difference was found between the two groups on KC_{27}. On KC_{21}, however, the NormGain group was significantly more likely to get justification steps than the InvNormGain group.

Summary of Log Analysis

Overall, following the NormGain tutorial policies did not generate more interactive tutorial tactics than following the InvNormGain ones. But once broken into a KC by KC basis, the NormGain tutorial tactics resulted in different I-ratio for every one of the primary KCs. On the other hand, following the NormGain

Table 6
Compare NormGain vs. InvNormGain on J-ratio across eight primary KCs

		NormGain (29)	InvNormGain (28)	Stats
1	KC_{14}	0.000 (0.000)	1.000 (0.000)	
2	KC_{20}	1.000 (0.000)	0.994 (0.022)	$t(55) = 1.467$, $p = 0.148$
3	KC_{21}	0.815 (0.216)	0.573 (0.096)	$\mathbf{t(55) = 5.445, p = 0.000}$
4	KC_{22}	1.000 (0.000)	1.000 (0.000)	
5	KC_{24}	0.876 (0.024)	0.871 (0.005)	$t(55) = 1.071$, $p = 0.289$
6	KC_{27}	0.046 (0.140)	0.000 (0.000)	$t(55) = 1.736$, $p = 0.088$
7	KC_{28}	1.000 (0.000)	1.000 (0.000)	

tutorial tactics seemed more likely to execute a justification step but once broken into KC by KC basis, the NormGain and InvNormGain tutorial tactics' J-ratio were only significantly different on KC_{21} and KC_{14} (The NormGain tutorial tactics skipped all of them while the InvNormGain executed all of them).

Summary on NormGain vs. InvNormGain

To summarize, the findings confirmed our primary hypotheses in this project: the pedagogical tutorial policies applied at the micro-step level affected students' learning. Moreover, the use of RL to derive tutorial tactics from existing data proved to be feasible and successful. On the other hand, the results also suggested that content exposure with the Cordillera system, irrespective of the micro-step policies employed, was, indisputably, an important factor in students' learning because even the InvNormGain students learned significantly. Nonetheless, micro-step pedagogical policies also made a significant impact.

However, it is not clear as to what it was about the NormGain tutorial tactics that caused the NormGain students to learn more effectively than the InvNormGain group. By simply analyzing the log file in a relatively shallow way, it seems that it was not that the NormGain tutorial tactics were simply more interactive or generated more justification steps. This is consistent with the conjecture that interactivity is not, necessarily, the most important determiner of students' learning. For example, although no significant difference was found between the two conditions in terms of the number of elicitation prompts and tells they received or the I-ratio, the NormGain students nonetheless learned significantly more than the InvNormGain students. Additionally, once broken into a KC by KC basis, the NormGain group had a significantly higher I-ratio than the InvNormGain group on KC_{14}, KC_{20}, KC_{21}, KC_{22} and the learned significantly more than the latter on all three KCs; however the InvNormGain students had a significantly higher I-ratio than the NormGain group on KC_1, KC_{24}, KC_{27}, and KC_{28}, but the former did not learn more than the latter group.

For JS decisions, the induced NormGain tutorial tactics indeed resulted in more justification steps in students' tutorial dialogues. However, once the tutorial decisions were broken into a KC by KC basis, the two groups differed significantly only on KC_{21} and KC_{14}. Therefore, future work is needed to investigate the induced tutorial tactics and find out what actually caused these learning differences.

The NormGain and InvNormGain tutorial tactics in this study were derived from the Exploratory and DichGain Corpora in previous studies. Therefore, it is possible to draw some hypotheses from observations by running a post-hoc comparison among the four groups. A cross-study analysis comparing the three studies will be presented in the next section.

COMPARISONS ACROSS ALL FOUR GROUPS

The preceding section focuses on two groups, NormGain and InvNormGain, because they were selected by random assignment from the same population and thus provide the most rigorous test of our hypotheses. In this section, we fold in an analysis of the results from two other groups, DichGain and Exploratory, in the hope that this wider view will shed some light on the main results. We begin by summarizing methodological differences among the groups, then compare their learning gains, then compare their I-ratio and J-ratio measures.

Study Variations

A total of 158 participants used four versions of Cordillera as part of the three studies: The Exploratory Group contained 64 students who used Random-Cordillera (Study 1); the Dichotic Gain (DichGain) Group was comprised of a total of 37 students who used DichGain-Cordillera (Study 2); The Normalized Gain (NormGain) group included 29 students who used NormGain-Cordillera and the Inverse Normalized Gain (InvNormGain) group included 28 students who used InvNormGain-Cordillera (Study 3). All of the participants followed the same procedure, used the same preparatory materials and problems, and interacted with Cordillera. They all completed a background survey, read a textbook covering the target domain knowledge, took a pre-test, solved the same seven training problems in the same order using Cordillera, and finally took a post-test. Only four salient differences existed across the three studies:

1. Although all of the participants were recruited in the same way, they were recruited in different years. In Study 3 the students were randomly assigned into the NormGain and InvNormGain groups (2009). On the other hand, in the first two studies participants were not randomly assigned to the Exploratory (2007) and DichGain groups (2008).
2. Interaction decisions were guided by different micro-step policies. Random-Cordillera made random decisions on micro-steps. The other three versions of Cordillera followed corresponding induced tutorial policies to decide which action to take.
3. Apart from a single question variation on Studies 2 and 3, all three studies used identical exams containing a total of 33 test questions. The one variation occurred as the result of the replacement of a single question, Q_{20}, which had been used in Study 1. It was judged to be too easy and was replaced with a more difficult question, Q^*_{20} that covered the same KCs for Studies 2 and 3. The remaining 32 test items were identical across all three studies.
4. A group of six human wizards were involved in Studies 1 and 2; but only one of the six wizards (the first author) was in Study 3.

Despite these differences, because the NormGain and InvNormGain groups trained in Study 3 were guided using tutoring tactics derived from the Exploratory and DichGain corpora, a post-hoc comparison among the four groups will allow us to observe the characteristics of the induced tutorial tactics from a wider point of view.

In order to establish test equivalence, Q_{20} and Q^*_{20} were excluded from the scores used below. As described in the previous chapter, the tests contained 33 test items which covered 168 KC occurrences. Removing $Q20$ reduced this total by 1 leaving 32 test items covering 166 KC occurrences. For comparison purposes both scores were normalized to 1.

Based on the procedure of induced tutorial tactics, it was expected that $NormGain > DichGain > Exploratory > InvNormGain$.

A one-way ANOVA showed that there were no significant differences among the four groups on overall training time: ($F(3, 147) = 1.531$, $p = 0.209$). More specifically, the average total training time in minutes

across the seven training problems, was $M = 278.73$, $SD = 67.38$ for Exploratory group, $M = 294.33$, $SD = 87.51$ for DichGain group, $M = 259.99$, $SD = 59.22$ for NormGain group, and $M = 264.57$, $SD = 67.60$ for InvNormGain group. Additionally, no significant difference was found among the Exploratory, the NormGain, and the InvNormGain groups on the Math SAT scores[2]: ($F(2, 83) = 0.520$, $p = 0.596$).

Learning performance

A one-way ANOVA was used to test for performance differences between the pre- and post-tests. Participants across four groups made significant gains from pre-test to post-test: $F(1, 314) = 41.82$, $p = 0.000$.

While no significant pre-test score difference was found among the four groups ($F(3, 154) = 0.38$, $p = 0.77$), there were significant differences among the four groups on the posttest and NLG scores: $F(3, 154) = 3.41$, $p = 0.02$ and $F(3, 154) = 5.30$, $p = 0.002$ respectively. Moreover, t-test comparisons showed that there was a significant difference between the NormGain group and all of the three remaining groups on the post-test scores and NLG scores (see Table 7). In Table 7, the first column lists the two groups in comparison and their corresponding mean and standard deviation scores. The second column lists the statistical result of the t-test comparison. The last two columns list the effect size and power of the comparison. For effect size, Cohen's d was still used. However, there were no significant differences among DichGain, Exploratory, and InvNormGain on all three test scores. Overall, our results suggest that $NormGain > DichGain = Exploratory = InvNormGain$ across two different performance metrics: post-test and NLG scores.

Log analysis

Having compared the individual groups' learning performance, this subsection will compare the log file variations across the four groups. Moreover, given the limited space, we will only present the comparison among the four groups on the I-ratio and J-ratio.

I-Ratio

Table 8 summarizes t-test comparisons on the I-ratio among the four tutorial corpora. In Table 8, the first two columns list the two groups in comparison and their corresponding mean and SD scores. The last column lists the statistical results of the *t*-test comparisons. From Table 8, the I-ratios for the four student groups were: 0.76 (NormGain), 0.76 (InvNormGain), 0.44 (DichGain), and 0.50 (Exploratory) respectively. Except for no significant difference between the NormGain and InvNormGain on the I-ratio, both groups were significantly more interactive than either the DichGain group or Exploratory group. Altogether, the result is $NormGain = InvNormGain > Exploratory > DichGain$ on the I-ratio.

High interactivity is a key characteristic of one-on-one human tutoring. It is commonly believed that more interactive would result in more learning. Our post-hoc comparisons also shows that the NormGain group was more successful than the Exploratory and DichGain groups and the former is also being more interactive than the latter two. However, our other post-hoc comparisons suggest that the more successful tutorial tactics were not necessarily more interactive than the less successful tactics. Comparisons between the NormGain and InvNormGain groups suggest that it is not the absolute level of interactivity that determines the students' success. The NormGain group was more successful than the others despite there being no significant difference in interactivity ratios between it and the InvNormGain group. Conversely, the InvNormGain group was no more successful than the Exploratory and DichGain groups despite being more interactive than either.

[2] This data was lost for the DichGain group due to a data management error.

Table 7
Compare four groups on overall learning performance

Group Name μ (σ)		*Stat*	*Cohen's d*	1− β
		Pretest		
NormGain	0.42(0.15)	$t(55) = 0.66$, $p = 0.507$	0.16	0.58
InvNormGain	0.39(0.23)			
NormGain	0.42(0.15)	$t(64) = 1.05$, $p = 0.299$	0.25	0.49
DichGain	0.38(0.17)			
NormGain	0.42(0.15)	$t(91) = 0.29$, $p = 0.792$	0.05	0.8
Exploratory	0.41(0.20)			
		Posttest		
NormGain	0.65(0.15)	$t(55) = 2.32$, $p = 0.024$	0.64	0.53
InvNormGain	0.54(0.20)			
NormGain	0.65(0.15)	$t(64) = 3.28$, $p = 0.0017$	0.82	0.46
DichGain	0.50(0.21)			
NormGain	0.65(0.15)	$t(91) = 3.17$, $p = 0.0069$	0.63	0.35
Exploratory	0.53(0.21)			
		NLG		
NormGain	0.42(0.19)	$t(55) = 3.15$, $p = 0.0026$	0.87	0.54
InvNormGain	0.25(0.21)			
NormGain	0.42(0.19)	$t(64) = 4.626$, $p = 0.000$	0.95	0.18
DichGain	0.22(0.23)			
NormGain	0.42(0.19)	$t(91) = 3.61$, $p = 0.0005$	0.84	0.33
Exploratory	0.22(0.26)			

Table 8
Pairwise comparison among four groups on I-ratio

Group 1		*Group 2*		*Group 1 vs. Group 2*
NormGain	0.76 (0.07)	InvNormGain	0.76 (0.02)	$t(55) = 0.395$, $p = 0.694$
NormGain	0.76 (0.07)	Exploratory	0.50 (0.03)	$t(91) = 24.72$, $p = 0.000$
NormGain	0.76 (0.07)	DichGain	0.44 (0.04)	$t(64) = 22.08$, $p = 0.000$
InvNormGain	0.76 (0.02)	Exploratory	0.50 (0.03)	$t(90) = 43.998$, $p = 0.000$
InvNormGain	0.76 (0.02)	DichGain	0.44 (0.04)	$t(63) = 36.34$, $p = 0.000$
Exploratory	0.50 (0.03)	DichGain	0.44 (0.04)	$t(99) = 7.967$, $p = 0.000$

Justify ratio

Table 9 summarizes t-test comparisons on J-ratio among the four tutorial corpora. In Table 9, the first two columns list the two groups in comparison and their corresponding mean and SD scores. The last column lists the statistical results of the t-test comparisons. Table 9 shows that the mean of J-ratios for the four student groups were: 0.82 (NormGain), 0.79 (InvNormGain), 0.43 (DichGain), and 0.53 (Exploratory). The difference was statistically significant: $F(3, 154) = 322.88$, $p = 0.000$. Table 9 presents the pair wise

Table 9
Pairwise comparison among four groups on J-ratio

Group 1		*Group 2*		*Group 1 vs. Group 2*
NormGain	0.82 (0.07)	InvNormGain	0.79 (0.03)	$t(55) = 2.27$, $p = 0.027$
NormGain	0.82 (0.07)	Exploratory	0.53 (0.06)	$t(91) = 18.95$, $p = 0.000$
NormGain	0.82 (0.07)	DichGain	0.43 (0.07)	$t(64) = 22.85$, $p = 0.000$
InvNormGain	0.79 (0.03)	Exploratory	0.53 (0.06)	$t(90) = 43.998$, $p = 0.000$
InvNormGain	0.79 (0.03)	DichGain	0.43 (0.07)	$t(63) = 26.65$, $p = 0.000$
Exploratory	0.53 (0.06)	DichGain	0.43 (0.07)	$t(99) = 7.894$, $p = 0.000$

t-test comparisons. It shows that on J-ratio, the result is: $NormGain > InvNormGain > Exploratory > DichGain$.

To summarize, NormGain tutorial policies resulted in substantially more justifications than InvNormGain tutorial policies. However, although the NormGain group had a higher ratio of justification prompts than the InvNormGain, Exploratory, or DichGain groups it is not the case that the absolute justification ratio guarantees learning. As with the interactivity ratio, the InvNormGain group received a higher justification ratio than the Exploratory or DichGain groups despite having been induced to enhance those decisions that contribute less or even none to the students' learning, and despite the absence of a significant difference in adjusted post-test scores or NLG between the groups.

Summary of post-hoc comparison

To summarize, a post-hoc comparison of learning performance across four groups shows that the NormGain group significantly outperformed all other three groups while no significant learning difference was found between the remaining three groups. These results were consistent both for the post-test scores and the normalized learning gains. These results support the prior analysis which showed that the NormGain tutorial tactics significantly improved students' learning compared with the InvNormGain ones.

However, the lack of a significant difference between the InvNormGain, DichGain, and Exploratory groups seemingly contradicts the initial predictions. The InvNormGain tactics were specifically induced to enhance those decisions that contribute less or even none to the students' learning. Therefore, a lower performance on the students' part there than in at least the DichGain group, which sought to enhance the tutorial decisions that contribute to the students' learning, was expected. One possible explanation for the lack of difference is that when tutorial tactics are done well, they add value to students' learning experience, but when they are done poorly, students are unaffected by them no matter how poor they are.

In other words, one possible explanation for the lack of difference is that the tutorial tactics employed by InvNormGain-, DichGain- and Random-Cordillera systems were ineffective and thus presented a minimum bar. By 'ineffective' it does not mean that they prevented the students from learning but rather that they were not able to make a positive impact on their learning above and beyond the baseline provided by Cordillera itself. Here the basic practices and problems, domain exposure, and interactivity of Cordillera set a minimum bar of students' learning that the tactics, however poor, cannot prevent. This is only a post-hoc explanation not a tested hypothesis, however it merits further study.

CONCLUSION

Human tutors often scaffold students via a series of micro-steps while a typical ITS is step-based (VanLehn, 2006). One hypothesis as to the effectiveness of human one-on-one tutoring comes from the detailed management of micro-steps (Graesser, Person, & Magliano, 1995; Graesser, VanLehn, Rosé, Jordan, & Harter, 2001). However, once content was controlled to be the same across all conditions, neither human tutors nor Natural Language (NL) tutoring systems designed to mimic human tutors reliably outperformed step-based systems (Evens & Michael, 2006; VanLehn, Graesser, & et al., 2007).

On the other hand, for any form of one-on-one tutoring, the tutor's behavior can be viewed as a sequential decision process wherein, at each discrete step, the tutor is responsible for selecting the next action to take. Although it is commonly assumed that human expert tutors have effective pedagogical skills, little evidence has been presented to date demonstrating that. Therefore, our explanation for the lack of difference among the human tutors, Natural Language (NL) tutoring systems, and step-based tutoring systems in previous research is that neither human tutors nor their computer mimics are good at making micro-step decisions.

In this project our primary research question is whether pedagogical tutorial tactics would impact students' learning if the instructional content was controlled to be equivalent for all students. In order to control the instructional content in this project, we used Cordillera.

To investigate our research question, we focused on two types of micro-step tutorial decisions Elicit vs.Tell (ET) and Justify vs. Skip-Justify (JS) and applied RL to induce two set of tutorial tactics: the Normalized Gain (NormGain) tactics were derived with the goal of making tutorial decisions that contribute to students' learning, while the Inverse Normalized Gain (InvNormGain) tactics were induced with the goal of making less beneficial, or possibly useless, decisions. The two sets were then empirically compared on real students. The students were randomly assigned to balanced conditions and received identical training materials and procedures apart from the tutoring tactics employed.

After spending the same amount of time on training, the NormGain group outperformed the InvNormGain group in terms of posttest scores and the normalized learning gain regardless of the grading criteria. This result suggests that the lack of a difference among the human tutors, Natural Language (NL) tutoring systems, and step-based tutoring systems in previous studies (Reif & Scott, 1999a; Evens & Michael, 2006; VanLehn, Graesser, & et al., 2007) is that neither human tutors nor their computer mimics are good at making micro-step decisions.

On the other hand, applying RL to induce pedagogical policies on ITSs is not new (Beck, Woolf, & Beal, 2000; Iglesias, Martínez, Aler, & Fernández, 2009a,b; Iglesias, Martínez, & Fernández, 2003; Martin & Arroyo, 2004; Tetreault, Bohus, & Litman, 2007; Tetreault & Litman, 2006). However, only a few studies evaluated the induced policy on real students to verify whether RL indeed fulfilled its promise (Beck, Woolf, & Beal, 2000; Iglesias, Martínez, Aler, & Fernández, 2009a,b; Iglesias, Martínez, & Fernández, 2003). As far as we know, none of these studies has empirically shown that the RL induced policy indeed made a difference in students' learning performance even though improving student learning is a primary goal for any ITS. Therefore, the better learning gains of the NormGain group over the InvNormGain group in this study indicated that applying RL to derive effective tutorial policies that would improve students' learning is feasible.

The analyses of the log data showed that the NormGain policies did not differ significantly from the InvNormGain policies in their overall interactivity (the ratio of Elicits to Tells). This is consistent with earlier studies showing that increasing the overall interactivity of natural language dialogue tutoring tends not to increase its effectiveness (VanLehn, 2008; Evens & Michael, 2006; VanLehn, Graesser, & et al., 2007; Reif & Scott, 1999b; Chi, Roy, & Hausmann, 2008; Rose, Moore, Allbritton, & Lehn, 2001; Johnson & Johnson, 1992).

However, there were significant and large differences in I-ratio on a KC by KC basis. For instance, KC_{14} was always elicited by the NormGain policy and always told by the InvNormGain policy, whereas KC_{28} was never elicited by the NormGain policy and elicited about half the time by the InvGainPolicy. This suggests that KCs really are learned independently in that some KCs benefit from high amounts of interaction and other KCs benefit from low amounts of interaction. For instance, perhaps some KCs are just easier to learn or more familiar initially, so it is better for the tutor to Tell them, whereas other KCs are difficult to learn or completely unfamiliar, so it is better for the tutor to Elicit them and students to learn by generating answers on their own.

This hypothesis offers an explanation for the surprising result that human tutors are often no better than step-based tutoring systems (Reif & Scott, 1999a; Evens & Michael, 2006; VanLehn, Graesser, & et al., 2007). Studies of human tutors suggest that they do not monitor students' competence at a fine-grained level (Chi, Siler, & Jeong, 2004; Putnam, 1987). They can indicate the overall competence of a student in a topic, but cannot reliably diagnose bugs and probably cannot indicate competence per KC. This suggests that they may not regulate their tutoring actions at a per-KC level either. That is, they may not say to themselves, "Now I need the student to apply the Change of TME for non-isolated systems (KC_{28}), which most students tend to mess up, and I've been asking a lot of hard questions recently, so I'm just going to show how to apply it myself." This particular policy rule is quite simple compared to the NormGain ones (see Fig. 5), but it illustrates what human tutors would have to consider if they were to be as effective as the NormGain policies suggest they can be. Moreover, they would have to make this decision very rapidly.

On the other hand, the J-ratio (relative number of Justify actions) was significantly higher for the NormGain group than the InvNormGain group. This shows that prompting for self-explanations tends to be effective for promoting learning, which is consistent with the self-explanation literature (Chi, de Leeuw, Chiu, & LaVancher, 1994; Conati & VanLehn, 2000; Aleven, Ogan, Popescu, Torrey, & Koedinger, 2004). However, once the totals are broken down by KC, the two groups *only* differed on two KCs, KC_{14} and KC_{21}. On KC_{21} the NormGain tutorial tactics skipped all of the justify actions while the InvNormGain tactics covered all of them; while on KC_{21} the J-ratio was higher for the NormGain group than the InvNormGain group. Moreover, a wider comparison among the NormGain, InvNormGain, DichGain, and Exploratory groups suggests that increased learning might not be due to receiving a higher number of justification steps. The InvNormGain students had significantly more justification steps than the DichGain, and Exploratory groups. However, the former did not learn more than the latter two groups.

Although more research is needed to understand what caused the NormGain tutorial policies to be more effective than the others, the take-home message is that effective tutorial policies seem to depend on details including the particular KCs involved in the micro-step. Computer tutors such as Natural Language tutoring systems can handle such details faster and better than human tutors, so it may not be long before they are more effective than human tutors. The post-hoc analysis in the prior section shows that all four groups learned significantly by training on Cordillera. This result indicates that the content exposure and practice opportunities can cause students to learn even from tutors with poor pedagogical tutorial tactics. However, it also indicates that with effective tutorial tactics students can learn even more.

ACKNOWLEDGEMENTS

NSF (#0325054) supported this work. We also thank the Learning Research Development Center at the University of Pittsburgh for providing all the facilities used in this work.

REFERENCES

Aleven, V., Ogan, A., Popescu, O., Torrey, C., & Koedinger, K. R. (2004). Evaluating the effectiveness of a tutorial dialogue system for self-explanation. In Lester, Vicari, & Paraguaçu (2004), (pp. 443-454).

Anderson, J. R. (1983). *The architecture of cognition*. Cambridge, Mass. : Harvard University Press.

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences*, 4(2), 167-207.

Beck, J., Woolf, B. P., & Beal, C. R. (2000). Advisor: A machine learning architecture for intelligent tutor construction. In *AAAI/IAAI*, (pp. 552-557). AAAI Press / The MIT Press.

Bernsen, N. O. & Dybkjaer, L. (1997). *Designing Interactive Speech Systems: From First Ideas to User Testing*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Bloom, B. S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 13, 4-16.

Cade, W. L., Copeland, J. L., Person, N. K., & D'Mello, S. K. (2008). Dialogue modes in expert tutoring. In Woolf, Aïmeur, Nkambou, & Lajoie (2008), (pp. 470-479).

Chades, M. C., Garcia, F., & Sabbadin, R. (2005). Mdp toolbox v2.0 for matlab.

Chi, M. (2009). Do *Micro-Level Tutorial Decisions Matter: Applying Reinforcement Learning To Induce Pedagogical Tutorial Tactics*. Ph.D. thesis, School of Art & Science University of Pittsburgh.

Chi, M., Jordan, P. W., VanLehn, K., & Litman, D. J. (2009). To elicit or to tell: Does it matter? In V. Dimitrova, R. Mizoguchi, B. du Boulay, & A. C. Graesser (Eds.) *AIED*, (pp. 197-204). IOS Press.

Chi, M. T. H., Feltovich, P., & Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. *Cognitive Science*, 5, 121-152.

Chi, M. T. H., de Leeuw, N., Chiu, M. H., & LaVancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive Science*, 18(3), 439-477.

Chi, M. T. H., Roy, M., & Hausmann, R. G. M. (2008). Observing tutorial dialogues collaboratively: Insights about human tutoring effectiveness from vicarious learning. *Cognitive Science*, 32(2), 301-342.

Chi, M. T. H., Siler, S., & Jeong, H. (2004). Can tutors monitor students' understanding accurately? *Cognition and Instruction*, 22(3), 363-387.

Chi, M. T. H. & VanLehn, K. (1991). The content of physics self-explanations. *The Journal of the Learning Sciences*, 1, 69-105.

Collins, A., Brown, J. S., & Newman, S. E. (1989). Cognitive apprenticeship: Teaching the craft of reading,writing and mathematics. In L. B. Resnick (Ed.) *Knowing, learning and instruction: Essays in honor of Robert Glaser, chapter* 14, (pp. 453-494). Lawrence Erlbaum Associates: Hillsdale New Jersey.

Conati, C., & VanLehn, K. (2000). Toward computer-based support of meta-cognitive skills: A computational framework to coach self-explanation. *International Journal of Artificial Intelligence in Education*, 11, 398-415.

Evens, M., & Michael, J. (2006). *One-on-one Tutoring By Humans and Machines*. Mahwah, NJ: Erlbaum.

Forbes-Riley, K., Litman, D. J., Purandare, A., Rotaru, M., & Tetreault, J. R. (2007). Comparing linguistic features for modeling learning in computer tutoring. In Luckin, Koedinger, & Greer (2007), (pp. 270-277).

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

Graesser, A. C., VanLehn, K., Rosé, C. P., Jordan, P. W., & Harter, D. (2001). Intelligent tutoring systems with conversational dialogue. *AI Magazine*, 22(4), 39-52.

Halloun, I., & Hestenes, D. (1985). The initial knowledge state of the college physics students. *Am. J. Phys*., 53(11), 1043-1055.

Henderson, J., Lemon, O., & Georgila, K. (2005). Hybrid reinforcement/supervised learning for dialogue policies from communicator data. *In IJCAI Workshop on K&R in Practical Dialogue Systems*, (pp. 68-75).

Iglesias, A., Martínez, P., & Fernández, F. (2003). An experience applying reinforcement learning in a web-based adaptive and intelligent educational system. *Informatics in Education*, 2(2), 223-240.

Iglesias, A., Martínez, P., Aler, R., & Fernández, F. (2009a). Learning teaching strategies in an adaptive and intelligent educational system through reinforcement learning. *Applied Intelligence*, 31, 89-106. ISSN 0924-669X. URL http://dx.doi.org/10.1007/s10489-008-0115-1, 10.1007/s10489-008-0115-1.

Iglesias, A., Martínez, P., Aler, R., & Fernández, F. (2009b). Reinforcement learning of pedagogical policies in adaptive and intelligent educational systems. *Knowledge-Based Systems*, 22(4), 266-270. ISSN 0950-7051.doi: 10.1016/j.knosys.2009.01.007. Artificial Intelligence (AI) in Blended Learning - (AI) in Blended Learning.

Janarthanam, S., & Lemon, O. (2009). User simulations for online adaptation and knowledge-alignment in troubleshooting dialogue systems. In *Proceedings of the 12th SEMdial Workshop on on the Semantics and Pragmatics of Dialogues*.

Johnson, H., & Johnson, P. (1992). Different explanatory dialogue styles and their effects on knowledge acquisition by novices. In *Proceedings of the Hawaii International Conference on System Sciences*, (pp. 47-57).

Jordan, P. W., Hall, B., Ringenberg, M. A., Cue, Y., & Rosé, C. P. (2007). Tools for authoring a dialogue agent that participates in learning studies. In Luckin, Koedinger, & Greer (2007), (pp. 43-50).

Jordan, P. W., Ringenberg, M. A., & Hall, B. (2006). Tools for authoring a dialogue agent that participates in learning studies. In *Proceedings of ITS06 Workshop on Teaching with Robots, Agents, and NLP*.

Katz, S., Connelly, J., & Wilson, C. (2007). Out of the lab and into the classroom: An evaluation of reflective dialogue in andes. In Luckin, Koedinger, & Greer (2007), (pp. 425-432).

Katz, S., O'Donnell, G., & Kay, H. (2000). An approach to analyzing the role and structure of reflective dialogue. *International Journal of Artificial Intelligence and Education*, 11, 320-343.

Koedinger, K. R., & Aleven, V. (2007). Exploring the assistance dilemma in experiments with cognitive tutors. *Educational Psychology Review*, 19(3), 239-264.

Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, 8(1), 30-43.

Larkin, J. H., McDermott, L. C., Simon, D. P., & Simon, H. (1980). Expert and novice performance in solving physics problems. *Science*, 208, 1335-1342.

Lester, J. C., Vicari, R. M., & Paraguaçu, F. (Eds.) (2004). *Intelligent Tutoring Systems, 7th International Conference,ITS 2004, Maceiò, Alagoas, Brazil, August 30 - September 3, 2004, Proceedings*, volume 3220 of *Lecture Notes in Computer Science*. Springer.

Levin, E., & Pieraccini, R. (1997). A stochastic model of computer-human interaction for learning dialogue strategies. In *EUROSPEECH* 97, (pp. 1883-1886).

Luckin, R., Koedinger, K. R., & Greer, J. E. (Eds.) (2007). *Artificial Intelligence in Education, Building Technology Rich Learning Contexts That Work, Proceedings of the 13th International Conference on Artificial Intelligence in Education, AIED 2007, July 9-13, 2007, Los Angeles, California, USA, volume 158 of Frontiers in Artificial Intelligence and Applications*. IOS Press.

Martin, K. N., & Arroyo, I. (2004). Agentx: Using reinforcement learning to improve the effectiveness of intelligent tutoring systems. In Lester, Vicari, & Paraguaçu (2004), (pp. 564-572).

Merrill, D. C., Reiser, B. J., Ranney, M., & Trafton, J. G. (1992). Effective tutoring techniques: A comparison of human tutors and intelligent tutoring systems. *The Journal of the Learning Sciences*, 2(3), 277-306.

Moore, J. D., Porayska-Pomsta, K., Varges, S., & Zinn, C. (2004). Generating tutorial feedback with affect. In V. Barr & Z. Markov (Eds.) *FLAIRS Conference*. AAAI Press.

Newell, A. (1994). *Unified Theories of Cognition*. Harvard University Press; Reprint edition.

Putnam, R. T. (1987). Structuring and adjusting content for students: A study of live and simulated tutoring of addition. *American Educational Research Journal*, 24(1), 13-48.

Reif, F., & Scott, L. (1999a). Teaching scientific thinking skills: Students and computers coaching each other. *Am.J. Phys.*, 67(9), 819-831.

Reif, F., & Scott, L. A. (1999b). Teaching scientific thinking skills: Students and computers coaching each other. *American Journal of Physics*, 67(9), 819-831.

Rose, C. P., Moore, J., Allbritton, D., & Lehn, K. V. (2001). A comparative evaluation of socratic versus didactic tutoring. In *Proceedings of the 23rd annual Meeting of the Cognitive Science Society, Edinburgh, UK*. URL http://www.cs.cmu.edu/cprose/pubweb/cogsci01.pdf.

Singh, S. P., Kearns, M. J., Litman, D. J., & Walker, M. A. (1999). Reinforcement learning for spoken dialogue systems. In S. A. Solla, T. K. Leen, & K.-R. Müller (Eds.) *NIPS*, (pp. 956-962). The MIT Press.

Sutton, R. S. & Barto, A. G. (1998). *Reinforcement Learning*. MIT Press Bradford Books.

Tetreault, J., & Litman, D. (2006). Using reinforcement learning to build a better model of dialogue state. In *Proceedings 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL), Trento*, Italy.

Tetreault, J. R., Bohus, D., & Litman, D. J. (2007). Estimating the reliability of mdp policies: a confidence interval approach. In C. L. Sidner, T. Schultz, M. Stone, & C. Zhai (Eds.) HLT-NAACL, (pp. 276-283). The Association for Computational Linguistics.

Tetreault, J. R., & Litman, D. J. (2008). A reinforcement learning approach to evaluating state representations in spoken dialogue systems. *Speech Communication*, 50(8-9), 683-696.

VanLehn, K. (2006). The behavior of tutoring systems. *International Journal Artificial Intelligence in Education*, 16(3), 227-265.

VanLehn, K. (2008). The interaction plateau: Answer-based tutoring < step-based tutoring = natural tutoring. In Woolf, Aïmeur, Nkambou, & Lajoie (2008), (p. 7).

VanLehn, K., Graesser, A. C., & et al. (2007). When are tutorial dialogues more effective than reading? *Cognitive Science*, 31(1), 3-62.

VanLehn, K., Jordan, P., & Litman, D. (2007). Developing pedagogically effective tutorial dialogue tactics: Experiments and a testbed. In *Proceedings of SLaTE Workshop on Speech and Language Technology in Education ISCA Tutorial and Research Workshop*.

VanLehn, K., Lynch, C., & et al. (2005). The andes physics tutoring system: Lessons learned. *Int. J. Artif. Intell. Ed.*, 15(3), 147-204. ISSN 1560-4292.

VanLehn, K., Siler, S., Murray, R. C., Yamauchi, T., & Baggett, W. B. (2003). Why do only some events cause learning during human tutoring? *Cognition and Instruction*, 21(3), 209-249.

Vygotsky, L. (1971). Interaction between learning and development. In T. M. Cole (Ed.) *In Mind in Society.*, (pp.79-91). Harvard University Press: Cambridge Massachusetts.

Walker, M. A. (2000). An application of reinforcement learning to dialogue strategy selection in a spoken dialogue system for email. *Journal of Aritificial Intelligence Research*, 12, 387-416. URL http://citeseerx.ist.psu.edu/viewdoc/summary? doi=10.1.1.43.1121.

Williams, J., & Young, S. (2007a). Partially observable markov decision processes for spoken dialog systems. *Computer Speech and Language*, 21(2), 231-422.

Williams, J., & Young, S. (2007b). Scaling pomdps for spoken dialog management. IEEE Trans. on Audio, *Speech, and Language Processing*.

Woolf, B. P., Aïmeur, E., Nkambou, R., & Lajoie, S. P. (Eds.) (2008). *Intelligent Tutoring Systems, 9th International Conference, ITS 2008, Montreal, Canada, June 23-27, 2008, Proceedings, volume 5091 of Lecture Notes in Computer Science*. Springer.

Wylie, R., Koedinger, K., & Mitamura, T. (2010). Is self-explanation always better? the effects of adding selfexplanation prompts to an english grammar tutor. In *Proceedings of the 31st Annual Conference of the Cognitive Science Society, COGSCI 2010, Amsterdam, The Netherlands*.