To Elicit Or To Tell: Does It Matter?

Min CHI[a], Pamela JORDAN[a], Kurt VANLEHN[b] and Diane LITMAN[a]
[a] *Learning Research Development Center, University of Pittsburgh*
[b] *Computer Science & Engineering, Arizona State University*

Abstract. While high interactivity has been one of the main characteristics of one-on-one human tutoring, a great deal of controversy surrounds the issue of whether interactivity is indeed the key feature of tutorial dialogue that impacts students' learning results. There are two commonly held hypotheses regarding the issue: a widely-believed *monotonic interactivity hypothesis* and a better supported *interaction plateau hypothesis*. Beyond a certain level of interactivity, the former hypothesis predicts (high > moderate) while the latter hypothesis predicts: (high = moderate). In this study, we proposed the *tactical interaction hypothesis* which predicts: (high + effective tactics) > (high + no/ineffective tactics) = moderate. Overall our results support this hypothesis. However, finding effective tutorial tactics is not easy. This paper sheds some light on how to apply Reinforcement Learning to derive effective tutorial tactics.

Keywords. Reinforcement Learning, Intelligent Tutoring Systems, Natural Language Tutoring Systems, Pedagogical Tutorial Tactics, Knowledge Component

Introduction

High interactivity is a key characteristic of one-on-one human tutoring. Whereas a classroom lecture can be viewed as monologue consisting of a long sequence of tutor instructions or *"tell"* acts, individual tutoring features much give and take and can be viewed as a mixture of tutor questions or *elicit* acts, students' responses, and tutor instructions. A common assumption, often referred as the *monotonic interaction hypothesis* [11][13], is that greater interactivity causes greater learning. Most Intelligent Tutoring Systems (ITSs), especially Natural Language (NL) tutoring systems, are designed to be highly interactive.

However, previous studies have shown that when the content of the instruction is strictly controlled to be equivalent in all conditions, highly interactive tutoring (such as human tutoring) is seldom more effective than moderately interactive instruction (such as step-based NL tutoring systems), even though both are often more effective than low interaction instruction (e.g. answer based instruction) [3][6][11]. A detailed review of the literature [13] distinguished between the widely-believed *monotonic interactivity hypotheses* and the better supported *interaction plateau hypothesis*. The former states that increase in interactivity causes an increase in learning while the latter states that increasing interactivity yields increasing learning until it hits a plateau, and further increases in interactivity do not cause noticeably increase in learning. Thus the key difference between the two hypotheses is whether high interaction methods would out-perform moderate ones: (high>moderate>low) or (high = moderate >low).

Most studies cited above made use of human tutors as their high interaction condition. However, they did not focus on the role of tutoring tactics in guiding interaction and its effects on tutor success. During tutoring, human tutors have to induce tutorial policies from their episodic memory of tutoring sessions and then execute them in real time with limited resources. Chi et al and others have argue that human tutors may not always select optimal tutorial actions [2][11]. In this study we propose an additional hypothesis: the *tactical interaction hypothesis*. It states that increasing interactivity yields increasing learning until it hits a plateau, and further increases in interactivity do not cause increases in learning unless they are guided by effective tutorial tactics. Tutorial tactics are policies used to select the optimal tutorial action at any given time from the available set. Thus, we hypothesize that: (high + effective tactics) > (high + no/ineffective tactics) = moderate.

To investigate the three hypotheses, we focused on two tutorial actions: elicit and tell. An elicit asks students a question about the problem at hand. Whereas a tell presents the information directly to the students. Figure 1 presents an example comparing elicit and tell versions of the same topics extracted from a log file we collected. Both tutorial dialogues start and end with the same tutor turns (lines 1 and 5 in (a) and (b)). However, the tutor chooses to elicit first then tell in (a) (lines 2-3 and line 4 respectively) and instead to tell first and then elicit in part (b) (line 2 and lines 3-4 respectively). Note that (a) and (b) cover the same corresponding content. Generally speaking, eliciting more information from the students during tutoring results in a more interactive tutorial dialogue. In this paper, we defined interactivity in terms of the elicit-tell ratio, which is defined as the number of elicits a student received divided by the number of tells he/she received in a given tutorial dialogue. The higher this value, the more interactive the tutorial dialogue.

1. **Tutor:** So let's start with determining the value of KE0.
2. **Tutor:** Which principle will help you calculate the rock's instantaneous magnitude of velocity at T1? ***{elicit}***
3. **Student: definition of kinetic energy**
4. **Tutor:** Let me just write the equation for you: ke1 = ½*m*v1^2. ***{tell}***
5. **Tutor:** From ke1 = ½*m*v1^2, we get v1^2=ke1/(0.5*m). We substitute…

(a) Elicit-Tell Version

1. **Tutor:** So let's start with determining the value of KE0.
2. **Tutor:** To calculate the rock's instantaneous magnitude of velocity at T1, we will apply the definition of kinetic energy again. ***{tell}***
3. **Tutor:** Please write the equation for how the definition of kinetic energy applies to this problem at T1 ***{elicit}***
4. **Student: ke1 = ½*m*v1^2**
5. **Tutor:** From ke1 = ½*m*v1^2, we get v1^2=ke1/(0.5*m). We substitute…

(b) Tell-Elicit Version

Figure 1. Elicit vs. Tell

Unlike the other two hypotheses, validation of the tactical interaction hypothesis relies on an important assumption that we have effective tutorial tactics. Most tutorial tactics for ITSs are encoded as hand-coded rules that seek to implement cognitive and/or pedagogical theories. The theories may or may not have been well-evaluated. Typically, system designers and domain expert design tutorial tactics by hand and make many nontrivial design decisions. It is often not easy to evaluate these decisions because the performance of these tutorial tactics depends on many other factors, such as the difficulty of domain content, the student's competence, the usability of the system, how easily the dialogues are understood, and so on. Previous research has primarily treated the specification of tutorial tactics as a system design problem: several versions of a system are created and the only difference among them is the tutorial tactics used. Data is collected with human subjects interacting with these different versions of the system and results from students' performance on different versions are statistically compared. Due to the costs of experiments, only a handful of policies are typically explored. Yet, many such other reasonable tutorial tactics are still possible.

In recent years, work on the design of dialogue systems has involved several data-driven methodologies. Among them, Markov decision processes (MDP) and Reinforcement Learning (RL) have been most widely applied. In this study, we applied

RL to semi-automatically induce effective tutorial tactics. We say semi-automatically because manual effort was necessary to identify the relevant feature states. We used a complex task domain where it is common to view the yet to-be-learned knowledge as comprised of several independently learned components, called Knowledge Components (KCs) [1]. 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"[12]. For the purposes of our tutoring system these are the atomic units of knowledge. Techniques exist to re-engineer the definition of KCs so that they are independently learnable [1], and this improves the overall effectiveness of the resulting tutoring system. It is commonly assumed that these KCs are learned independently. For example, various standardized tests are built based on the assumed independence among these KCs. Since KCs are assumed to be learned independently, we argue that tutorial tactics specific to each KC should also be induced independently. That is, when the tutor is about to mention a KC, whether to use an elicit or a tell should depend on the student's current mastery of that KC, its intrinsic difficulty, and the dialogue history. Thus, the best elicit/tell policy for one KC might not be optimal for another. In this study, we have eight primary KCs. We induced eight policies and conducted eight tests of three hypotheses, once per KC.

Later results indicated that every student in this study received at least some elicit prompts in each KCs, thus based on the standard of [13] the least interactive dialogues we collected are still moderately interactive. We expect that on all KCs:

1. If the *interactivity hypothesis* is correct, the group with higher elicit-tell ratios would learn more.
2. If the *interaction plateau hypothesis* is correct, students would learn equally well regardless of interactivity difference.
3. If the tactical interaction hypothesis is correct and our RL-based tutorial tactics are indeed effective, students trained with effective, more interactive tutorial instruction would learn more than those with less effective, lower interactive ones.

First we will briefly describe how we apply RL to the NL tutoring system we used in this study. Then we will describe our approach and finally present our results.

1. Applying RL to NL Tutoring Systems

RL is a machine learning method that centers on the maximization of expected rewards and has commonly used Markov Decision Processes (MDP's) [9] to model a dialogue. An MDP describes a stochastic control process whose state transitions possess the Markov property. An MDP formally corresponds to a 4-tuple (S,A,T,R), in which: $S = \{S_1,\ldots, S_n\}$, 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: $P^{a_k}_{Si,Sj}$ is the probability that the model would transition from state S_j to state S_i by taking action A_k. Finally, $R : S \times A \times S \rightarrow R$ denotes a reward model that assigns rewards to state transitions and models payoffs associated with such transitions. While, $\pi: S \rightarrow A$ is defined as a policy.

In order to effectively derive tutorial tactics using RL, we followed the following procedure (See [4] for details). In the first stage, we built a NL tutoring system named Cordillera-explore, in which decisions on when to elicit or tell were made randomly. Then a group of students, called the Exploratory group, were trained on it. More specifically, they 1) took a background survey, 2) read a textbook covering the target domain knowledge, 3) took a pretest, 4) trained on Cordillera-explore, and 5) took a posttest. Each individual student's interaction log together with his/her pre- and post-test scores is treated as one whole dialogue. The Exploratory corpus was the collection of these dialogues.

In the second stage, we used the Exploratory corpus to construct the MDP's 4-tuple <S, A, T, R>. Ideally, S should summarize the dialogue history compactly, employing the fewest features possible, while retaining the relevant information about the dialogue interaction. For RL, as with all machine learning tasks, success is dependent upon choosing an appropriate feature set for representing states. We used a procedure that began by defining a large set of features. In particular, we started by defining 18 features based upon the four categories of features considered by [7] to be relevant for human tutors when making their tutorial decisions: *autonomy, temporal situation, problem solving state,* and *performance*. We then select a small subset from them. For this study we only included features that could be computed automatically because hand coded dialogue features would be infeasible given that the feature values need to be available in real time when the learned policies are used to control the tutoring system.

For each dialogue in the Exploratory corpus, a scalar performance measure, the reward, was needed. We defined reward based on students' normalized learning gains (NLG). More specifically, students were split into two groups by the median value of their NLG. We gave the better-performing half of students' dialogues a positive reward of +100 and the remaining ones a negative reward of -100. The rewards were assigned in the final dialogue state. Following [8], we view each student's dialogue as a trajectory of chosen tutorial actions determined by dialogue context and system actions:

$$S1 \rightarrow (A1, R1) \rightarrow S2 \rightarrow (A2, R2) \ldots Sn \rightarrow (An, Rn)$$

Here Si → (Ai, Ri) → Si+1 means that at the ith turn, the current state (i.e, context of dialogue, students performance) was in state Si, the NL tutor executed action Ai and received reward Ri, and then the state changed to Si+1. The first state S1 reflects the student's performance on the pre-test. For each student, the reward is delayed, and thus we have R1… Rn-1 all equal to 0 and only the final reward Rn equals to either 100 or -100 depending on the student's NLG. The problem of deriving effective tutorial tactics thus becomes calculating the optimal policy for certain action decisions in an MDP. To calculate the best policy, we used Tetreault and Litman's tool since it has proven to be both reliable and successful [10]. In order to learn a policy for each KC, we annotated our tutoring dialogues (final kappa ≥ 0.77 for each of the eight KCs) and action decisions based on which KCs a tutor action or tutor-student pair of turns covered. Additionally, we have mapped students' pre- and post-test scores to the relevant KCs for each test item. While there are other KCs involved in the tutorial dialogue, they appeared significantly less frequently and often co-occurred with these eight main KCs. Thus, they are not our focus in this study.

The high cost of collecting human data precludes us from collecting a large Exploratory corpus. Given the complexity of task at hand, we therefore need to conduct effective feature selection. In this study, we followed a greedy-like search of the feature space. More specifically, for each of the 18 features, we employed the MDP to induce a single-feature policy. MDP generally requires discrete features, therefore we employed a median split to convert all numerical features into binary variables. Thus, for each KC, we have 18 single-feature-policies. For each of four categories of features, we selected the one feature which produced the single-feature policy with the highest Expected Cumulative Reward (ECR) in the category. ECR is calculated by normalizing the value of each state by the number of times it occurs in a dialogue and then summing over all states. The higher the ECR, the more effective the learned policy is expected to be. The four features selected were then used to induce a more complicated four-category-feature policy (see Figure 2 for an example). We then picked the one policy that has the highest ECR from the 19 learned policies: 18 single-feature policies and one four-category-feature policy. We call this resulting policy the Greedy-RL tutorial tactics. Figure 2 shows an example of one such policy for KC21 (definition of gravitational potential Energy).

Figure 2, line 1 (‘features’) indicates the four features involved in the policy: *duration, ProblemComplexity, tellsSinceElicit, and pctCorrectKCSession*. Line 2 (‘cutoff’) lists the median values used to convert the three corresponding features from real numbers to binary values. A total of 16 rules were learned: in 8 situations the tutor should elicit (line 4), in 5 it would tell (line 5); in the remaining 3 the tutor can do either (line 6). For example, “0:MED:1:0” (shaded in line 4) means when the *duration* since the most recent decision made on KC21 is less than 50s, the *ProblemComplexity* of the current problem is medium, the students has been told at least once since the most recent elicit (*tellsSinceElicit)*, and the student’s performance on KC21in today’s session is less than 71.79% correct, then the tutor should elicit the next step from the student. As can be seen, the derived four-category-feature tutorial tactics are quite subtle.

1. **'features'**=[duration,ProblemComplexity,tellsSinceElicit, pctOverallCorrectKC],
2. **'cutoff'** =[duration =’50.0' tellsSinceElicit ='0.0001' pctCorrectKCSession ='0. 7179'],
3. **'policy':**
4. **'elicit:** [0:MED:1:0, 1:COMP:1:0, 0:COMP:1:1, 0:MED:0:0, 0:COMP:1:0, 0:MED:1:1, 0:COMP:0:1, 1:COMP:0:1],
5. **'tell:** [1:MED:0:1, 1:MED:0:0, 1:MED:1:0, 1:MED:1:1, 0:MED:0:1]
6. **'else:** [0:COMP:0:0, 1:COMP:0:0, 1:COMP:1:1]

Figure 2. A Greedy-RL Policy On KC21: Gravitational Potential Energy

In the final stage, we replaced the random policy with these Greedy-RL policies and called the new system *Cordillera-GreedyRL*. We then trained a new group of students on the new system. All students went through the same training procedure as in Stage 1.

2. Approach

2.1. Participants

All participants in the training were required to have basic knowledge of high-school algebra, no experience with college-level physics, and were paid for their time. Each student trained during and completed the study in a period of two to three weeks. Data ~~was~~ collected in two stages. The first data collection with Cordillera-explore lasted over four months during the fall of 2007 and 64 students completed the experiment. The second data collection with Cordillera-GreedyRL lasted over three months during spring 2008 and 37 students completed the experiment. Therefore, we have a total of 101 students who finished the study: 64 were in the Exploratory group and 37 were in the GreedlyRL group.

2.2. Domains & Main Knowledge Components

The experiment used the Physics work-energy domain as covered in a first-year college physics course. The eight primary KCs were: the weight law (KC1), definition of work (KC14), Definition of Kinetic Energy (KC20), Gravitational Potential Energy (KC21), Spring Potential Energy (KC22), Total Mechanical Energy (KC24), Conservation of Total Mechanical Energy (KC27), and Change of Total Mechanical Energy (KC28).

2.3. Procedure

All students went through the same online training procedure: 1) background survey, 2) textbook, 3) pre-test, 4) training on the tutoring system, and 5) post-test. For each principle, the textbook provided a general description and reviewed some examples. The textbook was not available to students during any other phase of the experiment. Both tests were taken online, and once an answer was submitted, students automatically proceeded to the next question without any feedback on the correctness of their answer. Students were not allowed to return to earlier questions. The pre- and post-tests were identical. There were 33 problems selected from the Physics literature on the tests. In phase 4, students first walked through a demonstration problem with Cordillera. Then, all students solved the same seven problems in the *same* order. The seven training problems were ordered roughly by increasing complexity.

2.4. Grading Criteria

All tests were graded in a double-blind manner by a single experienced grader who was not familiar with the hypotheses being tested. The maximum score for each problem was 1. Additionally, the grader identified all relevant KCs and gave a score for each KC application. Each problem was assigned a difficulty weighting so that the total score possible on the test was 100 points for 33 problems. We evaluated the student's competence on each KC separately weighted by the problem difficulty. That is, given a problem containing KC21 with difficulty 6, the student would receive 6 points if they completed the KC21 correctly in that problem irrespective of their work on the other KCs in it. All KC-based scores were normalized by dividing with the corresponding total maximum possible scores.

3. Results

Our data ~~was~~ collected at different times and thus students were not randomly assigned to the groups. The Exploratory group had higher incoming competence than the Greedy-RL group as measured by pre-test score: $t(99) = 2.00$, $p < 0.05$. This fact is important because we have found that highly competent students often manage to learn regardless of instructional methods [4]. Our results show that students scored significantly higher in the posttest than pretest: $F(1, 64) = 12.71$, p=0.001 for the Exploratory group and $F(1, 37)$ =16.061, p=0.000 for the Greedy-RL group respectively. On a KC by KC basis, both conditions learned significantly on all the main KCs save for KC14 and KC28. On these KCs, no significant difference between pre- and post-test scores was found in the Exploratory group: $F(1, 64)$=0.251, p=0.617 for KC14 and F(1, 64)=2.80, p=0.097 for KC28; however, a significant difference was found in the Greedy-RL group: $F(1, 37)$=4.10, p=0.047 for KC14 and F(1, 37)=4.175, p=0.045 for KC28 respectively. Thus, following the Greedy-RL tutorial tactics on KC14 and KC28, students performed significantly better in the posttest than in the pretest but not when the decisions were randomly made. This suggests that the derived tutorial tactics may be effective.

Given the unbalanced incoming competence between the two groups, we used an ANCOVA to factor out pretest scores and compared the resulting adjusted posttest scores for the two groups We found that the Greedy-RL group had significantly higher adjusted post-test scores than the Exploratory group on just one KC: KC21, $F(1)$= 4.93, p<0.029. However, no significant differences were found between the two groups on the adjusted scores on the other seven KCs.

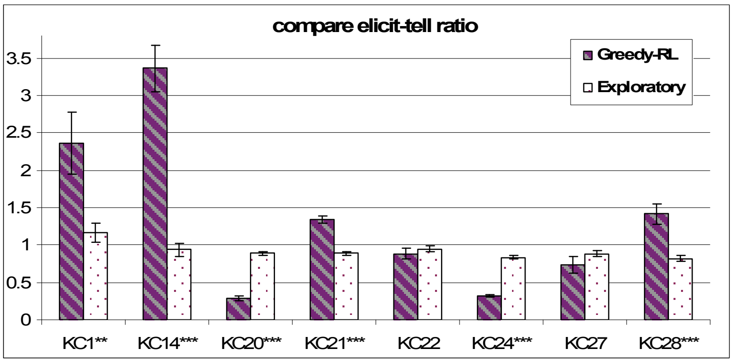

Figure 3. Compare the elicit-tell ratio across the two groups of students

We next investigated the interactive characteristics of the derived KC-based tutorial tactics by comparing the elicit-tell ratios between the two groups. For the Explore group, the ratios approached 1/1 for each KC, as Cordierra-Explore randomly chose between elicit and tell. As Figure 3 shows, the elicit-tell ratio of the Greedly-RL group varied depending on the policy, with some KCs getting more elicits than tells (KC1, KC14, KC21 and KC28) and some getting more tells than elicits (KC20 and KC24). For these KCs, the elicit-tell ratios were significantly different from the elicit-tell ratios of the Exploratory group (all are at the level $p \leq .001$). On KC22 and KC27, no significant difference was found between the two groups in terms of their elicit-tell ratios. Thus, on 6 of the 8 main KCs, the Greedy-RL policies clearly resulted in significantly different interactive patterns from the random selection.

4. Discussion

The *monotonic interactivity hypothesis* states that more interactivity lead to increased learning. If this is true, we would expect the Exploratory group to have learned more than the Greedy-RL group on KC20 and KC24. Similarly, on KC1, KC14, KC21, and KC28 the Greedy-RL group should learn more than the Exploratory. However, the only significant difference between the two groups in terms of adjusted post-test scores was on KC21. On the other 5 KCs, the groups did not differ. Thus, on 5 out of 6 KCs, the data does not provide much support for the monotonic interactivity hypothesis.

The *interaction plateau hypothesis* states that states that more interactivity beyond given point will not increase learning and thus students would learn equally well regardless of interactivity levels. If this is true, then we expect the students should learn equally across the board. However, on KC21 the more interactive group (Greedy-RL) learned more than the less interactive group (Exploratory). On the other 7 KCs, the groups did not differ, as reported earlier. Thus, neither of these two hypotheses is consistently supported across the board.

Finally, the *tactical interaction hypothesis* states that more interactivity beyond given point does not cause increases in learning unless it is governed by effective the tutorial tactics. If this is true and all our derived RL-based policy were indeed effective, we expect that on KC1, KC14, KC21, and KC28, Greedy-RL would learn more than the Exploratory while KC20 and KC24, no difference should been found. This hypothesis was supported on KC21, KC20 and KC24 but not on KC14, KC21, and KC28. One likely explanation is that the tutorial tactics we derived for KC14, KC21, and KC28 were not effective enough. This is likely given the fact that our state features were relatively restricted and our feature selection procedure is quite greedy. Additionally, in subsequent work we found a significant correlation between the predicted rewards for the derived policies used in this study and the actual improvements made. Moreover, applying improved feature selection methods has yielded policies with higher predicted rewards than the Greedy-RL policies employed in this study (at least double see [4] for a detailed discussion).

Therefore, we concluded that the *tactical interaction* hypothesis is still supported by our results but our application of RL in the study was not optimal. This is why we did not see clear difference between the two groups on KC1, KC14, and KC28. In particular, our features or our Greedy-RL method for selecting them may have been flawed (see [4] for a detailed discussion). Overall, our results suggest that the *tactical interaction hypothesis* may be right but deriving effective tutorial tactics is not easy. However, RL seems to be a effective, useful tool to derive tutorial tactics.

Additionally, we have identified a number of questions for future exploration. For example, we only included features that can be computed automatically; grader tagged features have not been tested. In future studies, we will employ a larger feature set, better policies, will randomly assign subjects to groups, and experiment with new policy induction mechanisms.

References

[1] Cen, H., Koedinger, K. R., & Junker, B. (2006). Learning Factors Analysis: A general method for cognitive model evaluation and improvement. In M. Ikeda, K. D. Ashley, T.-W. Chan (Eds.) Proceedings of the 8th International Conference on ITS, 164-175. Berlin: Springer-Verlag.

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

[3] Chi, M. T. H., Siler, S., Jeong, H., Yamauchi, T.,&Hausmann, R. G. (2001). Learning from human tutoring. Cognitive Science, 25, 471–533.

[4] Chi, M & VanLehn, K. (2008). Eliminating the Gap between the High and Low Students through Meta-Cognitive Strategy Instruction. 9th International Conference, ITS2008, pp 603-613.

[5] Chi, M, Jordon, P, VanLehn, K, & Litman, D (in progress) Applying Reinforcement Learning To Induce Effective Pedagogical Knowledge-Components Based Tutorial Tactics

[6] Katz, S., Connelly, J., & Allbritton, D. (2003). Going beyond the problem given: How human tutors use post-solution discussions to support transfer. JAIED, 13, 79-116.

[7] Moore, J.D, Porayska-Pomsta, K, Varges, S, Zinn, C (2004): Generating Tutorial Feedback with Affect. FLAIRS Conference

[8] Singh, S., Kearns, M. S., Litman, D. J., & Walker, M. A. (1999). Reinforcement learning for spoken dialogue systems. In Proc. NIPS 99

[9] Sutton, R. S. and Barto, A. G. (1998). Reinforcement Learning: An Introduction.A Bradford Book. The MIT Press.

[10] Tetreault, J., and Litman D (2008). A Reinforcement Learning Approach to Evaluating State Representations in Spoken Dialogue Systems. *Speech Communication Volume 50 , Issue 8-9 Pages 683-696*

[11] VanLehn, K., Graesser, A. C., Jackson, G. T., Jordan, P., Olney, A., & Rose, C. P. (2007). When are tutorial dialogues more effective than reading? Cognitive Science 31(1), 3-62.

[12] VanLehn, K., Jordan, P., Litma, D., (2007). Developing pedagogically effective tutorial dialogue tactics: Experiments and a testbed. In proceedings of SLaTE Workshop.

[13] VanLehn, K. (submitted). The interaction plateau: Less interactive tutoring is often just as effective as human tutoring.