Expert Tutoring and Natural Language Feedback in Intelligent Tutoring Systems

Xin LU
Department of Computer Science, University of Illinois at Chicago, U.S.
xlu4@uic.edu

Abstract: Intelligent tutoring systems can provide benefits of one-on-one instruction automatically and cost effectively. To make the intelligent tutoring systems as effective as expert human tutors, my research aims at investigating what type of natural language feedback an intelligent tutoring system should provide and how to implement the feedback generation to engender significantly more learning than simple practice. This paper describes a comprehensive study of expert versus non-expert tutoring and a baseline intelligent tutoring system which provides different kinds of feedback. It then proposes a method to computationally model expert tutoring and a framework of effective natural language feedback generation with 3-tier probabilistic planning.

Keywords: intelligent tutoring system, natural language, expert tutoring

Introduction

As computers have become widespread in recent years, people recognize that intelligent tutoring systems (ITSs) can provide the great benefits of one-on-one instruction with lower cost and more flexibility in time and location. However, current ITSs are still not able to provide learning as effective for the users, as expert human tutors do. To bridge the gap between current ITSs and human tutors, previous studies proved that natural language (NL) interfaces could be one of the keys[1][2]. But it is still not clear what type of NL feedback, when and how to deliver it in ITSs to engender significantly more learning than simple practice. For example, we found that students learned more when given more abstract but also more directive in an ITS that teaches troubleshooting[2]; Litman et al.[3] found that there was no difference in learning gains of students who interacted with a mechanics ITS using typed text or speech. For the implementation of NL interfaces, existing tutorial dialogue systems performs dialogue management in an ad hoc manner. But none of their models explain how to generate effective tutorial feedback. The reason is, that it is not yet well understood what makes human tutoring effective and what is the most appropriate and convenient way to implement the effective tutoring language. This paper aims at answering these two questions.

1. Proposed Approaches and Related Work

This paper study has two goals: computationally modeling expert tutoring and effective tutorial feedback generation. For the first goal, a comprehensive study of the differences between expert and non-expert tutors in effectiveness, tutor and student moves and interaction patterns has been done. However, it's still halfway through to a computational model of expert tutoring. To accomplish this goal, I propose a further study

of human tutoring dialogues and a machine learning method to learn tutorial rules. For the second goal, I have developed a baseline intelligent tutoring system and evaluated four versions of the system which differ in the types of feedback they provide the student. To generate more sophisticated and effective NL feedback, I propose a framework of feedback generation with 3-tier probabilistic planning.

Our tutoring domain concerns extrapolating complex letter patterns[4], which is a well known task for analyzing human information processing in cognitive science. Given a sequence of letters that follows a particular pattern, the student is asked to find the pattern and create a new sequence from a new starting letter. For example, the pattern of the sequence "ABMCDM" is: "M" as a chunk marker separates the whole sequence into two chunks of letters progressing according to the alphabet. Then with a starting letter "E", to maintain this pattern, the student needs to finish the sequence as "EFMGHM". Only knowledge of the alphabet is required in this domain. We collected dialogues in this domain. During the training session, each student goes through a curriculum of 13 problems of increasing complexity. The training will improve the student's ability in solving letter pattern problems. To test the performance, each student also needs to solve two post-test problems, each 15 letters long, via a computer interface.

1.1 Study of Human Tutors: Expert versus Non-Expert

One recent result showed that the expert tutor did have better learning outcomes than the novice tutor but it's still not known to what behavior this result attributes[5]. To accurately model expert tutoring, I need to know the real difference between expert tutors and non-expert tutors in effectiveness, behavior and language. Three questions are addressed:

- Does the expert tutor use more varied strategies and more complex language?
- How much more effective is the expert tutor as compared to non-expert tutors?
- What brings effectiveness in tutoring?

1.1.1 Methods and Results

To investigate the effectiveness of expert tutors, we ran experiments in the letter pattern domain with three different tutors: one expert; one novice; and one lecturer who is experienced in teaching, but not in one-on-one tutoring. We also have a control group of subjects who did the post-test problems with no tutoring at all but only read a short description of the domain. There are 11 students in each group, who are all psychology majored freshmen and native speakers in English. We found that the expert tutor is significantly more effective than the other two tutors and than control on both post-test problems. (See Figure 1 and more details can be found in [6].)

The dialogues on two specific problems in the curriculum were transcribed and annotated from the videotapes which recorded the tutors' interaction with the subjects. For each tutor, six subjects' dialogues were transcribed and annotated with the tutor and student moves by utterance. The annotation scheme is based on the literature, [3][7], and designed with simplicity in mind.

The tutor moves include four high level categories, reaction, initiative, support, conversation. Reaction is sub-categorized into answering, evaluating and summarizing. Initiative is sub-categorized into prompting (general, specific), diagnosing, instructing (declarative, procedural) and demonstrating. Corresponding to the tutor moves, there are seven categories in our student moves: explanation, questioning, reflecting, answering, action response, completion and conversation. Two independent groups, each group with

two annotators, coded the tutor moves and the student moves on all the dialogues. The Kappa coefficient is used to evaluate agreement[8][9]. After several round of annotation, the intercoder agreement on most of the categories reached an acceptable level (perfect agreement $0.8<Kappa\leq1$, or substantial agreement $0.6<Kappa\leq0.8$). By looking at the Kappa values by tutor, we found that the dialogues with the novice tutor are easiest to annotate (with the highest intercoder agreement), followed by those with the lecturer and then those with the expert tutor. This supports the intuition that expert tutors use more sophisticated strategies and language.

We counted the number of utterance and words and ran ANOVAs on the ratio of student words to tutor words and student utterances to tutor utterances. We found that the expert tutor's subjects do not talk more: the ratio of student utterances to tutor utterances is significantly lower for the expert tutor ($p<0.05$), and so is the ratio of student words to tutor words ($p<0.001$). This contrasts with the expectations of expert tutors' behavior from the literature. For example, [7] argues that subjects learn best when they construct knowledge by themselves, and that as a consequence, the tutor should prompt and scaffold subjects, and leave most of the talking to them.

We also ran Chi-squares on the frequency of individual tutor and student moves. In the analysis, some findings support some predictions[7][10]:

- The expert tutor and the lecturer summarize more than the novice;
- Students with the expert tutor and the lecturer do more explanations than the students with the novice tutor.

A finding contradicts some predictions[7]: the expert tutor does less specific prompting and his students explain less than the lecturer. And there are also some interesting findings:

- The expert tutor does not answer more questions from his students; the novice tutor does and her students ask more questions;
- The expert tutor does more procedural instructing, demonstrating and supporting;
- The novice tutor does more declarative instructing.

Declarative instructing provides facts about the problem. Procedural instructing provides hints to the student how to solve the problem rather than just provides information.

The individual analyses on the tutor and student moves are not enough for us to derive a computational model of expert tutoring. On the other hand, it is likely that one-on-one tutoring is more effective than classroom lecturing because of the deep interaction. Our next step was to compare the expert tutor to the non-expert tutors in interaction patterns. A pair of moves from two different speakers which appear in sequence is an interaction pattern, which is called "adjacency pair" in computational linguistics. For example, the student does an answer and then follows a tutor’s summarizing, that is called a student-tutor interaction pattern. My analysis concerns the following two issues:

- Tutor-Student Interaction Pattern: What's the difference between each group of students' behaviors after each type of tutor move?
- Student-Tutor Interaction Pattern: Do the expert tutor and the non-expert tutors react differently to each type of student move?

Table 1. Interaction Patterns of the Expert Tutor

Tutor-Student Interaction Patterns		Student-Tutor Interaction Patterns	
Tutor	Student	Student	Tutor
Summarizing	Explanation	Explanation	Diagnosing
Procedural Instructing	Explanation	Summarizing	Diagnosing
Demonstrating	Explanation	Reflecting	General Prompting
Demonstrating	Reflecting	Reflecting	Declarative Instructing
Support	Answering	Reflecting	Procedural Instructing
		Reflecting	Demonstrating
		Action Response	Summarizing
		Action Response	Procedural Instructing

Table 1 summarizes the tutor-student and student-tutor interaction patterns in which the expert tutor is different from the non-expert tutors ($p<0.05$).

1.1.2 Future Plans

While I was studying the interaction patterns, I observed that not all of tutor's specific prompting are immediately followed by any student move. This may be because often the expert does specific prompting in multiple utterances. I am currently studying the difference between expert and non-expert tutors in patterns of multi-utterance turns. This study will enhance our investigation of expert tutoring versus non-expert tutoring.

1.2 Learning Tutorial Feedback Rules

After highlighting what makes the tutoring expertise, I will be able to model the expert tutoring. With all the dialogues, I will then use machine learning techniques to learn tutorial rules for generating effective NL feedback in ITSs. Through machine learning, not only can I learn useful rules from large numbers of transcripts but also the rules can be adapted automatically when I introduce more transcripts later. The CIRCSIM group has applied machine learning to discover how human tutors make decisions based on the student model[11]. But they only applied it to a very small set of data and a very limited use in their ITS.

Classification based on associations (CBA) which integrates classification and association rule mining can generate understandable rules, find all possible rules that exist in data and discover interesting or useful rules specifically for an application[12]. An association rule is a pattern that states the features and the targets that occur with certain probabilities. These probabilities are expressed as two strength measurements (confidence and support), which can help solve the prediction conflicts without removing useful rules, especially when we don't have a large number of annotated dialogues. The features that I am going to include to predict the next tutor moves will be: student and tutor move history, correctness of student move, hesitation time, student's input, domain concepts and student's knowledge state.

1.3 Delivery of Natural Language Feedback in ITSs

1.3.1 Four Versions of the Baseline ITS

While collecting and analyzing the human tutoring data, I was also developing an ITS for training students to solve the letter pattern problems. The ITS is built on the basis of ACT-R Theory, which claims cognitive skills are realized by production rules[13]. The production rules usually contain correct rules modeling correct solutions and buggy rules modeling possible mistakes. Tutors built through this kind of production rules are model-tracing tutors. I used the Tutor Development Kit (TDK)[14] to develop our letter pattern ITS.

I developed four versions of the ITS with different kinds of feedback provided to the student. In the no feedback version, each letter the subject inputs turns blue, with no indication and no message regarding whether it is correct or incorrect; in the neutral version, the only feedback subjects receive is via color coding, green for correct, red for incorrect; in the positive version, they receive feedback via the same color coding, and in addition, verbal feedback on correct responses only; in the negative version, they receive feedback via the

same color coding, and in addition, verbal feedback on incorrect responses only. The language in the positive and negative conditions was inspired by (but not closely modeled on, or using tutorial rules learned from) the expert tutor's language.

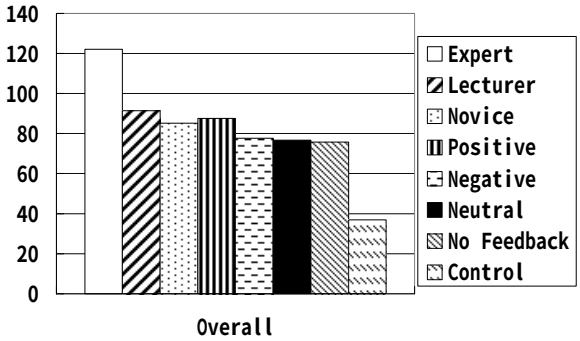

Figure 1. Post-Test Performance

To evaluate the four versions of the ITS and make comparison with the human tutoring, we ran a between-subjects study in which each group of subjects interact with one version of the system. With the ITS the subjects were trained to solve the same 13 problems in the curriculum that had been used in the human tutoring. Then they also did the same post-test (2 problems, each 15 letters long pattern). The post-test performance is the average number of letters correct out of total 180 letters (in 6 trials) for two problems per subject. Figure 1 reports the post-test performance for all groups of subjects with four versions of the ITS and the human tutors. We ran ANOVAs on the post-test performance. No significant differences were found between the groups of the four versions of the ITS. But they all did significantly better than the control group and all did significantly worse than the group with the expert tutor. The ITS had the students practice solving letter pattern problems but the feedback messages are too simple to lead to any significant improvement similarly to what happened with the expert tutor. The group with the positive verbal feedback version did slightly better than with the other three versions of the ITS and even beat the group with the novice tutor. Even if the result is not significant, this leads us to hypothesize that more positive feedback is better.

1.3.2 Generating Natural Language Feedback

The messages generated by means of TDK are too limited. There is great benefit to be gained from integrating dialogue theories and dialogue system technology that have been developed in the computational linguistics and spoken dialogue systems communities with the wealth of knowledge about student learning and tutoring strategies that has been built up in the ITS community. Therefore, I am planning to use Midiki (the MITRE Dialogue Kit) as the system shell of my NL feedback generator. Midiki is based on the Information State theory of dialogue management, which identifies the relevant aspects of information in dialogue, how they are updated, and how updating processes are controlled[15].

There will be three modules in the generator:

1. **The plan module** generates plans for planning content and discourse structure of the NL feedback. A plan is a structured collection of tutoring moves designed to accomplish a single task. The plan module generates plans based on the IS and the external resources (tutorial rules, curriculum and domain knowledge), using a 3-tier planning framework. The three tiers are:

- **Plan generation** automatically synthesizes plans from the tutorial rules based on the information state and other external resources.
- **Plan selection** selects a plan for the ITS and select a template for each tutoring move which is used to accomplish the current plan.

- **Plan monitoring** checks whether everything is going according to the plan after each tutoring move. If not, this tier must revise the plan or re-plan everything.

2. **The update module** maintains the context.
3. **The Feedback Realization Module** generates NL messages using a set of templates.

With this NL feedback generator, I will have the last version of the ITS for the letter pattern task. And I will run one last group of subjects to evaluate it. Of course, the feedback generator can be used in other domains. In our lab, we are planning to use it in an ITS for introductory computer science.

2. Contributions

I have been working on this project for four years. My work will contribute a comprehensive study of expert and non-expert tutoring dialogues, a method to computationally model expert tutoring, a framework for natural language feedback generation with 3-tier probabilistic planning and an ITS which provides different kinds of feedback.

Acknowledgments

This work is supported by grant N00014-00-1-0640 from the Office of Naval Research.

References

[1] Fox, B. (1993) The Human Tutorial Dialogue Project. Lawrence Erlbaum Associates, Hillsdale, NJ.

[2] Di Eugenio, B., Fossati, D., Yu, D., Haller, S. and Glass, M. (2005) Aggregation Improves Learning: Experiments In Natural Language Generation For Intelligent Tutoring Systems. The 42nd Meeting of the Association for Computational Linguistics, Ann Arbor, MI.

[3] Litman, D. J., Rose, C. P., Forbes-Riley, K., Vanlehn, K., Bhembe, D. and Silliman, S. (2004) Spoken Versus Typed Human And Computer Dialogue Tutoring. 7th International Conference on Intelligent Tutoring Systems, Alagoas, Brazil.

[4] Kotovsky, K. and Simon, H. (1973) Empirical Tests Of A Theory Of Human Acquisition Of Information-Processing Analysis. *British Journal of Psychology*, 61, 243-257.

[5] Chae, H. M., Kim, J. H., and Glass, M. (2005) Effective behaviors in a comparison between novice and expert algebra tutors. 16th Midwest AI and Cognitive Science Conference, 25-30.

[6] Di Eugenio, B., Kershaw, T. C., Lu, X., Halpern, A. C. and Ohlsson, S. (2006) Toward a Computational Model of Expert Tutoring: a First Report. 19th International conference of FLAIRS, Melborne Beach, FL.

[7] Chi, M. T., Siler, S. A., Jeong, H., Yamauchi, T. and Hausmann, R. G. (2001) Learning From Human Tutoring. *Cognitive Science*, 25, 4, 471-533.

[8] Carletta, J. (1996) Assessing Agreement On Classification Tasks: The Kappa statistic. *Computational linguistics*, 22, 2, 249-254.

[9] Di Eugenio, B. and Glass, M. (2004) The Kappa Statistic: A Second Look. *Computational linguistics*, 30, 1, 95-101.

[10] Landsberger, J. (2005) Feedback To Improve Study Guides. http://www.studygs.net

[11] Evens, M. and Michael, J. 2006. One-on-One Tutoring by Humans and Computers. Lawrence Erlbaum Associates, Mahwah, NJ.

[12] Liu, B., Hsu, W. and Ma, Y. (1998) Integrating Classification and Association Rule Mining. Knowledge Discovery and Data Mining, New York, NY, 80-86.

[13] Anderson, J. R., Boyle, C. F., Corbett, A. T. and Lewis, M. W. (1990) Cognitive Modeling And Intelligent Tutoring. *Artificial Intelligence*, 42, 1, Elsevier Science Publishers Ltd.

[14] Koedinger, K. R., Aleven, V. and Heffernan, N. T. (2003) Toward a rapid development environment for cognitive tutors. 12th Annual Conference on Behavior Representation in Modeling and Simulation.

[15] Larsson, S. and Traum, D. R. (2000) Information State And Dialogue Management In The TRINDI Dialogue Move Engine Toolkit. *Natural Language Engineering*, 6, 3-4, 323-340.