Expert Tutoring and Natural Language Feedback in Intelligent Tutoring Systems

BY

XIN LU
B.S., Harbin Institute of Technology, China, 2000
M.S., Harbin Institute of Technology, China, 2002

THESIS

Submitted as partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Chicago, 2007

Chicago, Illinois

ACKNOWLEDGMENTS

It is a pleasure for me to acknowledge everyone who has helped me in my pursuits. First, and most, is the gratitude I have for my friend and advisor: Barbara Di Eugenio. She provided me supports in all kinds of way and led me to the achievements in my research. Professor Stellan Ohlsson, like my second advisor, provided light at the end of tunnel at every crucial moment in my research. I am grateful to other committee members Martha Evens, Bing Liu and Tom Moher. Also I extend a very special thank to everyone in the UIC Intelligent Tutoring System group. I am a very lucky person to have worked with such a diverse and intelligent group of people. My great colleagues Andrew Corrigan-Halpern, Trina C. Kershaw, Bettina Chow and Davide Fossati contributed a lot to my work. I also want to thank my colleagues, friends in natural language processing lab. In the past five years they gave me the most friendly and collaborative working environment. Finally, thank the most important persons in my life – my husband Yan Luo, and my parents Shuhua Yang and Guangbin Lu. They are always there with me to get through every moment in my life.

This work has been supported by Campus Research Board S02 and S03 from University of Illinois at Chicago, grant N00014-00-1-0640 from the Office of Naval Research, and partly, by awards IIS0133123 and ALT 0536968 from the National Science Foundation. The contents does not reflect the position or policy of the government and no official endorsement should be inferred.

XL

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

CHAPTER PAGE

LIST OF TABLES

LIST OF TABLES (Continued)

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANOVA	Analysis of Variance
CAR	Classification Association Rule
CBA	Classification Based on Associations
CTAT	Cognitive Tutor Authoring Tools
FSM	Finite State Machine
KCD	Knowledge Construction Dialogue
ICAI	Intelligent Computer-Aided Instruction
IS	Information State
ITS	Intelligent Tutoring System
MUT	Multiple Utterance Turn
NL	Natural Language
TDK	Tutor Development Kit
WME	Working Memory Element

SUMMARY

When people recognized that intelligent tutoring systems can provide great benefits of one-on-one instruction with lower cost and more flexibility in time and location, they also found that current intelligent tutoring systems are still not able to provide learning equally effective for the users, as expert human tutors do. To make the intelligent tutoring systems act more like the expert human tutors, previous studies proved that natural language interface could be one of the keys. But it is still not clear what type of feedback an ITS should provide and how to implement the feedback generation to engender significantly more learning than unsupervised practice.

In this research, I demonstrate the utility of a computational model of expert tutoring in generating effective natural language feedback in intelligent tutoring systems. To set up a basis for computationally modelling expert tutoring, a comprehensive study of the difference between one expert tutor and two non-expert tutors in effectiveness, behavior and language is presented. The findings from this study show that the expert tutor is more effective and uses more complex strategies than the non-expert tutors. And the difference also shows in individual moves, the organizations of a single turn and the patterns of interaction. Based on the empirical results, I develop a rule-based model of expert tutoring which takes advantage of a machine learning technique, Classification based on Associations. The tutorial rules are automatically learned from a set of annotated tutorial dialogues, to model how the expert tutor makes decisions on tutor's attitude, domain concepts and problem scopes to focus on, and tutor moves. The results

SUMMARY (Continued)

of evaluation show that these rules have very good accuracy. To employ the model of expert tutoring in the natural language feedback generation for intelligent tutoring systems, I design a framework of feedback generation with 3-tier probabilistic planning. The 3-tier planning automatically generates, selects and monitors plans for generating effective tutorial feedback based on the rule-based model and the information state that keeps track of the interaction in the intelligent tutoring system. To evaluate the framework, 5 different versions of a model-tracing intelligent tutoring system for a particular task are implemented. One version does not provide any feedback. Three of them provides simple graphic and verbal feedback. The last version provides natural language tutorial feedback using the framework I developed. The evaluation results show that the last version is significantly more effective than the other versions and has no significant difference with the expert tutor in learning improvements. Therefore, the tutorial rules successfully model expert tutoring and the intelligent tutoring system using them generates effective feedback. In a word, my research provides a road map to model expert tutoring and generate effective natural language feedback in intelligent tutoring systems.

CHAPTER 1

INTRODUCTION

The purpose of this dissertation is to demonstrate the utility of a computational model of expert tutoring in generating effective natural language feedback in intelligent tutoring systems (ITSs).

Our approach to this goal encompasses four subgoals:

1. to do a comprehensive study of the differences between expert tutors and non-expert tutors;
2. to develop a computational model of expert tutoring based on the tutorial dialogues by using machine learning techniques;
3. to design a flexible natural language feedback generator that employs the model;
4. to implement an ITS that illustrates the effectiveness of the feedback generator.

1.1 Background

In 1984, Benjamin Bloom defined the "two-sigma problem," which states that the average student who received tutoring scored two standard deviations higher on standardized achievement tests than the average student who received traditional group-based instruction (Bloom, 1984). Providing a personal training assistant for each learner is beyond the training budgets of most organizations. However, a virtual training assistant that captures the subject matter and teaching expertise of experienced trainers provides a captivating new option. The concept, known as intelligent tutoring systems (ITSs) or intelligent computer-aided instruction (ICAI), has been pursued for more than three decades by researchers in education, psychology, and artificial intelligence. Today, prototype and operational ITS systems provide practice-based instruction to support corporate training, K-12 and college education, and military training (Ong and Ramachandran, 2000).

The goal of ITSs is to provide the benefits of one-on-one instruction automatically and cost effectively. Like training simulations, ITSs enable participants to practice their skills by carrying out tasks within highly interactive learning environments. However, ITSs go beyond training simulations by answering user questions and providing individualized guidance. Unlike other computer-based training technologies, ITSs assess each learner's actions within these interactive environments and develop a model of their knowledge, skills, and expertise. Based on the learner model, ITSs tailor instructional strategies, in terms of both the content and style, and provide explanations, hints, examples, demonstrations, and practice problems as needed. So there is always a major issue regarding what, when and how to deliver the instructional feedback in ITSs.

1.1.1 Natural Language in Intelligent Tutoring Systems

Quite a few researchers reported that natural language is important to learning. (Fox, 1993) observed that one-on-one tutoring involves a collaborative construction of meaning, a process that arises from a natural language interaction or dialogue between individuals. (Graesser and Person, 1994) found a positive correlation between the student's achievements and the quality of the student's deep questions, which supports the hypothesis that the real basis for the effectiveness of tutoring versus classroom teaching is the environment of natural language dialogue. (Chi et al., 1994) showed the importance of self-explanation in learning. To enhance the interactive learning in ITSs, natural language interfaces are brought into ITSs as a form of delivering instructional feedback. ITS researchers are still investigating whether the natural language interaction between students and an ITS does improve learning, and which features of a natural language interface to an ITS cause the improvement. Recently the first results have appeared, that show that the students learn more when interacting in natural language with an ITS. CIRCSIM-Tutor, which generates hints and questions that students can respond to appropriately, and understands students' responses, has been shown to help students learn more than reading a comparable text and also keeps students more active (Evens and Michael, 2006). When the PACT Algebra Tutoring System asked students to give explanations, the student scored better on the posttest than students who used PACT without this facility (Aleven et al., 1999). In a troubleshooting ITS (Di Eugenio et al., 2005), Di Eugenio et al. found that students learned more when given more abstract but also more directive feedback. However, it is still not clear what brings us these results. The reason is that it is not yet well understood what makes human tutoring effective, especially from the natural language point of view. So researchers have started to study the human tutoring process— what tutors do when they interact with a student and how they do it.

Tutors of different level of expertise may behave differently and have different effects on learning. Also, it's likely that expert tutors tend to use more complex tutorial strategies and language than novices (Putnam, 1987; Graesser et al., 2005; Lepper et al., 1997; Glass et al., 1999). So from the point of view of computationally modelling a dialogue, computational modelling expert tutoring will be more difficult (Glass et al., 1999). It will be good to know how much more effective expert tutors are than non-expert tutors, what aspects make them more effective and what features of their tutoring dialogues can be applied to ITSs. One recent result showed that the expert tutor did have better learning outcomes but it's still not known to what behavior to attribute this result (Chae et al., 2005).

1.1.2 Tutorial Feedback Generation in Intelligent Tutoring Systems

If we add a natural language interface to an ITS, the ITS turns into a tutorial dialogue system. Like general dialogue systems, dialogue management becomes an intrinsically complex task for designing an ITS. The function of tutorial dialogue management is to keep track of the dialogue history and contexts, and pick the topics and concepts and generate tutorial feedback. There are three major models of dialogue management:

1. Finite State Machines (FSMs): define a finite state automaton that contains all plausible dialogues. There is only a limited and well-defined amount of information available in each state of the network.

2. Form filling: specify the information that the dialogue system must obtain from the user as a set of forms composed of slots. The structural complexity of possible dialogues is limited by the form design and the intelligence of the form interpretation and filling algorithm.

3. Planning-based dialogue management: tailor tutorial dialogues by dynamic planning. This is the state-of-the art in tutorial dialogue management. The planning usually includes

content planning and discourse planning. Content planning chooses the concepts that will be included in the text to be constructed. Discourse planning determines the outline of the text to be uttered.

Although previous and ongoing work in tutorial dialogue systems strives to support unconstrained natural language input and multi-turn tutorial strategies, there remain limitations that must be overcome: teaching strategies, encoded as curriculum scripts, KCDs, or plan operators, are domain specific; the purely plan-based systems embed control in plan operators or, necessarily, conflate planning with student modelling and maintenance of the dialogue context; and most of current tutorial dialogue systems mix high-level tutorial planning with low-level communication management. These limitations can make systems difficult to maintain, extend, or reuse.

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. It is therefore worth considering dialogue systems not designed for tutoring. These systems aim for dialogue strategies that are independent of dialogue context management and communication management concerns. These strategies contain no domain knowledge; they query domain reasoners to fill in necessary details. Furthermore, in systems explicitly performing dialogue planning, control is never embedded in plan operators. My goal is to combine these beneficial features (modularity and re-usability) with the flexibility and educational value of tutorial systems with reactive planners.

1.2 Contributions

This dissertation makes contributions to the fields of learning science, artificial intelligence in education and natural language generation.

I did a comprehensive empirical study of one expert tutor versus two non-expert tutors and natural language feedback in one-on-one tutoring. My work

- validated the significant effectiveness of expert tutoring;
- developed an annotation scheme for the tutorial dialogues;
- annotated several tutoring dialogues of three tutors with different level of experience in one-on-one tutoring;
- provided various evidence to support the finding that expert tutors tend to use more varied strategies and more complex language;
- analyzed the difference between expert and non-expert tutors in various aspects, such as verbosity, individual moves, interaction patterns between tutor and student, and the organization of turns with multiple utterances;
- provided potential answers to what tutors should do to make their tutoring more effective.

I presented a method to computationally model tutoring from annotated tutorial dialogues and established a rule-based model of expert tutoring by using a machine learning technique – Classification Based on Associations (CBA) (Liu et al., 1998). I also demonstrated the advantages of using CBA in learning tutorial rules and presented a set of features for learning tutorial rules. In addition, I introduced CBA into template selection for surface realization of natural language feedback messages.

I designed an innovative framework of natural language feedback generation that integrates a dialogue management theory – Information State Theory (Larsson and Traum, 2000), and a planning theory – probabilistic planning (Blum and Langford, 1999). This framework can automatically synthesize plans from internal states and external resources, which saves a lot of effort for manually defining plan operators. I also implemented a natural language feedback generator that can be adapted to any ITS in any domain.

I developed a model tracing ITS for the letter pattern task with five different versions, which differ from one another in the kind of feedback they provide the student.

My work also provided a comparison of the ITS and human tutors in the same tutoring domain and some factors that affect learning.

1.3 Outline

This dissertation is organized as follows:

Chapter 2 gives a review of related research in the study of human tutoring and dialogue management in ITSs.

Chapter 3 introduces the tutoring domain – the letter pattern task, and describes a study of human tutoring in this domain. In addition to a description of the letter pattern task, it contains a brief introduction to the organization of each tutoring session. We collected tutoring data with three human tutors with different level of expertise in one-on-one tutoring and annotated the tutorial dialogues following an annotation scheme that we defined. Then I compared the expert tutor with the non-expert tutors in the frequency of utterances and words, individual tutor moves and student moves, interaction dialogue patterns between tutor and student and multi-utterance interactions.

Chapter 4 describes the rule-based model of expert tutoring. I first introduce the method – Classification Based on Associations (CBA) and the features used to learn tutorial rules from the expert tutoring dialogues. Then I describe the experiments to learn several sets of rules for different use. At last I report and illustrate the experiment results.

Chapter 5 contains the design of the ITS and the natural language feedback generator. I present a model-tracing ITS by using the authoring tool – TDK (Tutor Development Kit) (Koedinger et al., 2003), which I configured into four different versions which provide simple feedback at different levels. To generate effective tutorial feedback in the fifth version, I developed a framework of feedback generation based on the Information State theory of dialogue management (Larsson and Traum, 2000) and probabilistic planning (Blum and Langford, 1999).

Chapter 6 describes the evaluation of the five versions of the ITS for the letter pattern task. I report the post-test performance of the subjects with different version of the ITS and the user rating of the fifth version collected through a questionnaire. I also did several analyses to validate the learning improvements, compare the ITS with the human tutors and explore the factors that effect the learning improvements.

Chapter 7 contains the conclusions and a look into the future work.

CHAPTER 2

RELATED RESEARCH

Since researchers first recognized the effect of one-on-one tutoring and the importance of dialogue to learning, a lot of work has been done to investigate how human tutors perform one-on-one tutoring and how to manage tutorial dialogues in ITSs. In this chapter, I review some recent work.

2.1 Study of Human Tutoring

Recent research on tutoring has been done with tutors of different level of expertise in one-on-one tutoring: some with novice tutors, some with expert tutors, and some with both novice and expert tutors. I review this literature in terms of types of tutors: expert, novice and expert versus novice.

2.1.1 Expert Tutors

A number of researchers at Stanford have conducted a detailed examination of the overall goals, the general strategies, and the specific motivational and instructional techniques of demonstrably expert and effective human tutors (Lepper et al., 1997). In these research projects, they collected tutoring sessions on mathematics with several dozens of expert tutors. They identified "expert" tutors as the tutors who are highly effective in working with a variety of different students. Their observations showed that expert tutors seek both to inform and to inspire students; they give roughly equal attention and weight to motivational and to informational factors during the tutoring session; and their decisions as tutors are based on concurrent ongoing assessment or models of the student's affective and cognitive states. Lepper et al. recognized several motivational characteristics and strategies of the expert tutors: intelligent, nurturant, socratic, progressive, indirect, reflective, encouraging. One limitation of their work is that most of the students in the study initially had low self-confidence and high anxiety about their competence at the task. But the study is based on the hypothesis that the tutors focused less on student's feelings of self-esteem and competence than providing direct instruction and offering explicit feedback.

(VanLehn et al., 2003) analyzed approximately 125 hours of tutorial dialogue between two expert human tutors and physics students to see what features of the dialogue correlated with learning. They chose to examine two particular features: impasses and explanations. An impasse occurs when a student gets stuck, detects an error, or does an action correctly but expresses uncertainty about it. The results of this study showed that a student's understanding of a principle usually increases if the student reaches an impasse but tutorial explanations are associated with learning gains in only a few cases. They also found that explanations that were just deep enough to allow students to solve the post-test problems were more effective than deeper explanations. Based on the results, VanLehn et al. suggested that tutors should let impasses occur and only give short explanations.

2.1.2 Novice Tutors

(Graesser and Person, 1994) videotaped, transcribed and analyzed nearly 100 hours of naturalistic tutoring sessions on mathematics with 3 novice tutors. Although these novice tutors are effective in producing learning gains, the anatomy of these tutoring sessions revealed that these novice tutors do not use most of the ideal tutoring strategies that have been identified in education and the ITS community, such as the socratic method (Collins, 1985) or sophisticated motivational techniques (Lepper et al., 1997). These tutors generated dialogue moves that are sensitive to the quality and quantity of the preceding student turn. The dialogue moves include positive immediate feedback, neutral immediate feedback, negative immediate feedback, pumping for more information, prompting for specific information, hinting, elaborating, splicing in the correct content after a student error and summarizing (Graesser et al.,).

(Chi et al., 2001) studied human one-on-one tutoring to explain the effectiveness of tutoring by testing three hypotheses: a tutor-centered one, a student-centered one and an interactive one. Their study is based on tutoring sessions in a conceptual domain (the human circulatory system) with 11 novice tutors. For the first hypotheses on the effectiveness of the tutors' moves, they found that tutors seemed to control and dominate the tutoring sessions and one tutoring move, giving explanations, correlated significantly with students' learning. For the student-centered hypotheses, their findings showed that the students' constructive responses (giving scaffolded responses and reflection) are effective. However, the tutors' moves to elicit these responses did not correlate with students' learning. And while testing the interactive hypotheses, the evidence only confirmed that the students' were always interactive. To further assess the value of interactions, Chi et al. did another study to maximize opportunities for interactions by encouraging tutors to prompt the students. The results showed that students learned just as effectively even when tutors were suppressed from giving explanations and feedback. This is consistent with the findings of (VanLehn et al., 2003) – explanations are not always associated with learning.

2.1.3 Expert Tutors versus Novice Tutors

Regardless of the level of expertise of the tutor, all the above studies have claimed that one-on-one tutoring works. However, researchers started to wonder why some tutors are more effective than the others. So some studies were conducted to investigate the difference between expert tutors and novice tutors on the assumption that expert tutors are more effective than novice tutors.

The CIRCSIM–Tutor project first started by studying the effects of tutoring on cardiovascular physiology with two expert tutors (Evens and Michael, 2006). The two expert tutors have significant effect on helping students learn more than students who read a carefully chosen and edited section from a standard textbook. Then four novice tutors were recruited to conduct tutoring sessions with first year medical students. The students tutored by the novice tutors did improve their scores but less substantially than the students tutored by the expert tutors. By counting the occurrences of five "primitive dialogues acts" (tutor elicits, tutor informs, tutor acknowledges, student answers, tutor asks for confirmation) in novice and expert tutor transcripts, significant differences in tutor's behaviors were found (Glass et al., 1999): novice tutors spend a lot of time during a session telling student things (informs) and much less time asking the student questions (elicits); on the contrary, expert tutors spend proportionately more time asking questions and less time telling students things; novice tutors frequently ask the students if they understand the phenomena just discussed (asks for confirmation), but expert tutors almost never do so. Although some clear differences in the behavior of novice and expert tutors are identified, it is still not determined what impact particular differences have on student learning.

The North Carolina A&T State University Algebra Tutorial Dialogue Project collected over 50 one-hour transcripts of tutoring of college-level remedial algebra problems with several tutors with different levels of experience (Kim and Glass, 2004). A study of one expert tutor and one novice tutor was conducted to discover what behaviors constitute expertise (Chae et al., 2005). The expert tutor was found to ask more questions in response to statements, and correcting more errors than the novice tutor. In addition, the expert tutor did hinting differently compared to the novice tutor. The expert tutor hinted frequently at the start of tutoring the problem in a collaborative style (requiring several turns); on the contrary, the novice tutor did hinting mostly in response to a student impasse. This finding is consistent with the conclusion of (Graesser et al.,): novice tutors mostly react to the immediate previous turn. Like the studies in the CIRCSIM–Tutor project, although the expert tutor did have better learning outcomes than the novice tutor in this study, no conclusion on what behavior to attribute the expert tutor's better learning outcomes has been drawn.

Another recent study of expert and novice tutors focused on reading. Cromley and Azevedo collected and analyzed verbal protocols from 3 expert tutors and 3 novice tutors. They found that the expert tutors used a much higher proportion of cognitive scaffolding, and less instruction and motivational scaffolding than did the novice tutors (Cromley and Azevedo, 2005). Instruction includes giving explanations, giving the answer, using analogies, summarizing, and tutor planning. Cognitive scaffolding includes hints, previews, prompts. Motivational scaffolding provides students with various types of positive and negative feedback. They also compared question asking, content errors and responses to student errors between the expert tutors and the novice tutors. No significant differences were found. However, in this study they were not able to evaluate the learning outcomes since the students were adults with low levels of literacy who voluntarily participated in tutoring in the literacy center.

Although the above findings have shown some differences between expert and novice tutors, it is still unclear what makes human tutoring effective.

2.2 Dialogue Management in Intelligent Tutoring Systems

In the introduction I introduced three major models of dialogue management for intelligent tutoring systems. Existing tutorial dialogue systems perform dialogue management in an ad hoc manner. They adopt none of the models of dialogue processing in their pure form, mainly, because none of the models explain how to generate effective tutorial feedback (Zinn et al., 2002):

- **AUTOTUTOR** is an ITS for the computer literacy domain. AUTOTUTOR's dialogue management relies on a curriculum script, a sequence of topic formats, each of which contains a main focal question, and an ideal complete answer (Graesser et al., 1999). This dialogue management can be regarded as an adaptation of the form-filling approach to tutorial dialogue; to solve the feedback generation problem, it adds feedback moves to slots. A set of fuzzy production rules determines the category of the dialogue move to be selected. However, AUTOTUTOR does not support multi-turn strategies in the tutorial dialogue planning. Although the AUTOTUTOR dialogue manager performs well in the descriptive domain of computer literacy, it is unclear how well this approach will work in problem-solving domains such as algebra or circuit trouble-shooting. In these domains student answers will often require the tutor to engage the student in a multi-turn scaffolding or remediation sub-dialogue.

- The **CIRCSIM–Tutor with APE** (Khuwaja et al., 1994; Freedman, 1996) incrementally constructs and executes plans, and uses simple template driven generation for realizing elementary plan steps. It provides single-turn and multi-turn teaching strategies. A teaching strategy is represented as a data structure called an operator which consists of several slots. The goal slot is achieved if the operator is successfully executed; the precondition slot contains a number of constraints that must be true for an operator to be applicable; and the recipe slot contains a number of sub-goals that are generated by applying the operator. However, a major drawback of APE is that it embeds control in operators, unlike traditional planners, where control is separated from action descriptions. This makes writing operators difficult and puts an additional burden on the planner.
- The tutoring system **ATLAS-ANDES** teaches Newtonian mechanics. The dialogue manager of ATLAS-ANDES uses a combination of knowledge construction dialogues (KCDs), which are recursive FSMs, and a generative planner (Schulze et al., 2000). The grammar of a KCD is very similar to an AUTOTUTOR curriculum script, but unlike AUTOTUTOR, this planner supports multi-turns through the use of recursive KCDs. While AUTOTUTOR requires a pre-defined and hand-crafted curriculum script, the ATLAS-ANDES

approach allows on-the-fly generation of nested KCDs, using the APE discourse planner. A compiler maps KCDs into plan operators, which are used by APE to combine KCDs into larger recursive FSMs. However, this dialogue manager is domain-specific because of the complex and domain-dependent KCDs.

- **EDGE** is an explanation system in which students are asked to explain various electric circuits. Dialogue management in EDGE is purely plan-based (Cawsey, 1989). EDGE has single turn teaching strategies and strategies that can unfold over multi-turns and remediation plans that can deal with specific types of wrong answers. EDGE also incrementally builds and executes plans. Before each tutor turn, the deliberative planner expands the current unfinished step with either a complex sub-plan or an elementary plan step. Elementary plan steps are then executed using simple template driven generation.

To overcome the limitations that make a system less maintainable, extensible and portable, researchers started to consider combining the modularity and reusability of general dialogue systems with the flexibility and educational value of tutorial system with reactive planners. (Zinn et al., 2002) built the BEETLE system, which is a basic electricity and electronics tutorial learning environment with a natural language interface. To manage tutorial dialogue effectively, they proposed a 3-tier dialogue planning architecture based on Information State Theory. The three tiers are deliberative planning, context-driven plan refinement and action execution. It is primarily based on the TRINDIKIT dialogue system shell. (The original dialogue management toolkit based on Information State Theory.) The deliberative planning and execution monitoring modules are implemented in the open planning architecture (Currie and Tate, 1991). They identified the situation factors which are important in tutorial interactions, and determined how they impact on the settings for the variables that determine face threat (Moore et al., 2004). Then they used two values (A&A) to describe the situational factors:

- Autonomy: Tutors should let students do as much of the work as possible;
- Approval: Tutors should provide the students with positive feedback as much as possible.

The discourse planner uses the A&A values when choosing feedback strategies, and the surface generator uses them to choose among applicable realization rules. In the system, the two values are generated by a rule engine according to the situational modeler which models the aspects of the domain knowledge and student behavior, and the temporal aspects. They ran experiments to evaluate the BEETLE system with their situational modeler and the linguistic strategies with A&A values. And they found there was no significant differences between the ratings of the human tutor responses and the system preferred responses. They are still evaluating the full BEETLE system. Their experimental results may support that natural language feedback generation can benefit from the general dialogue theory and dialogue management technology. However, the plan operators in the BEETLE system are written manually, which still leaves a lot of burden in using this dialogue planning structure.

Managing tutorial dialogue is a complex task. Each existing tutorial dialogue system has developed its own model of tutorial dialogue management, which performs well in each specific ITS. It is necessary to develop a generic model, that can be employed in any ITS for any domain.

CHAPTER 3

STUDY OF HUMAN TUTORS: EXPERT VS. NON-EXPERT

To accurately model expert tutoring, I need to know the real difference between expert tutors and non-expert tutors in effectiveness, behavior and language. In this chapter I go through the study I have completed in comparing the expert tutoring with the non-expert tutoring in the letter pattern task. (Note: the work described in this chapter is the result of collective efforts of the UIC ITS group [1].)

3.1 The Letter Pattern Tutoring Task

Our tutoring domain concerns extrapolating complex letter patterns (Kotovsky and Simon, 1973), which is a well known task for analyzing human information processing in cognitive science. Students are taught how to solve some problems called "Sequence Extrapolation Problems." This type of problem is composed of a sequence of letters that follow a particular pattern. The student's task is to find the pattern and recreate a sequence with a given starting letter, so the new sequence follows that same pattern. Here is an example pattern:

A B M C D M

In this case the pattern is made up of two chunks, each with three letters. Within each chunk, the first two letters are two adjacent letters in the alphabet "going forward" and the third letter

[1] Section 3.4.3 and Section 3.4.4 state my own work.

stays constant across two chunks, so it is called “chunk marker.” If it starts with the letter “E,” the new sequence will be:

E F M G H M

From the letter “E,” we “go forward” one, so the next letter is “F.” Then M is repeated because it’s a chunk marker. Then from “F,” we “go forward” one again, we get “G” which is the starting letter of the second chunk. So we can complete this new sequence with “H” and “M.”

The patterns in the letter sequence are built from the relationships between the letters in the alphabet. These relationships can be thought of as the underlying rules of the pattern. There are four types of rules that are used to create patterns: repetition, forward, backward and progression. Repetition just states that a letter is the same as another letter. Forward and backward means the letter goes forward or backward in the alphabet. Forward and backward are parameterized according to the number of steps. Progression means either that the length of a chunk or that the steps forward or backward between letters are changing progressively. Also the pattern can have multiple levels of chunks, which make the pattern very complex. For example, in a pattern like:

A B B D D D G G G G

there are four chunks whose length is progressively increasing from 1 to 4 and between each two adjacent chunks the letters move progressively forward from 1 to 3.

Our data collection was divided into two parts: training and testing. During the training session, each student needs to go through a curriculum of 13 problems where the complexity of patterns increases. The training will improve the student's ability to solve letter pattern problems. To test performance, each student also needs to solve two post-test problems, each 15 letters long letter pattern, via a computer interface. The training and post-test problems are listed in Appendix A. There is no pre-test since all of our subjects are American native speakers and we assumed that they all have knowledge of the alphabet. So here performance equates post-test score.

3.2 Data Collection: How Effective Is the Expert Tutor?

To investigate the effectiveness of different levels of tutoring expertise, we ran experiments on the letter pattern domain with three different tutors:

- Expert: had years of experience in one-on-one tutoring;
- Lecturer: had years of experience in lecturing but little experience in one-on-one tutoring;
- Novice: had no experience in teaching or tutoring.

There is no clear definition of levels of tutoring expertise. Some studies defined their expert tutors as those who have years of tutoring experience in those domains. For example, in the CIRCSIM project, the tutors are physiology professors who have considerable experience teaching the materials one-on-one and in small groups (Glass et al., 1999); in the North Carolina A&T State University Algebra Tutorial Dialogue Project, the expert tutor is a professor in the Mathematics Department at NC A&T State University who has taught and tutored basic algebra for many years (Chae et al., 2005). Some other studies defined as expert tutors who have years of experience in one-on-one tutoring in similar domains or in general (Cromley and Azevedo, 2005; Groves et al., 2005). For example, Person [1] started a project in which the expert tutors are secondary level teachers who have over 5 years of general tutoring experience.

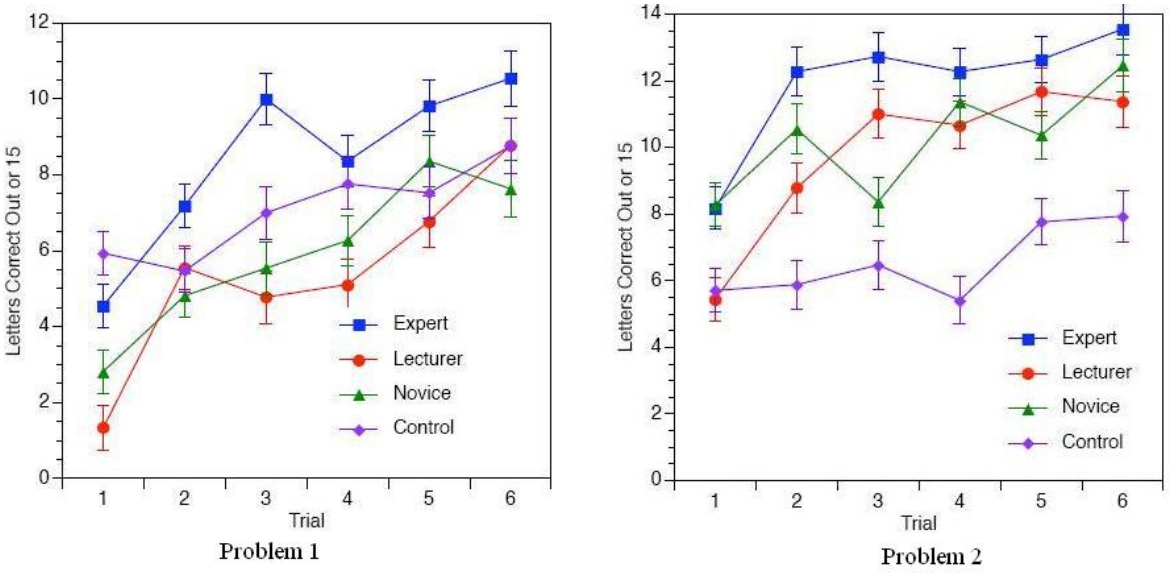
In our case, the letter pattern task doesn't require anything beyond the knowledge of the alphabet. Although none of the three tutors has tutoring experience in this particular domain, the expert tutor and the lecturer have years of research experience in the letter pattern task, which makes them qualified as more experienced tutors in this domain. For convenience, in the following sections the three tutors will be classified as:

non-expert tutors — the novice tutor and the lecturer;

more experienced tutors — the lecturer and the expert tutor.

There were 11 subjects in each condition. All of them went through all 13 training problems in one hour and then did two post-test problems. For each post-test problem, the same pattern, each subject had 6 trials, but each trial started with a different letter. The whole tutoring session of each subject was video-taped. Besides the three human tutoring conditions, there is also a control condition, in which the subjects did the post-test problems with no training at all but only read a short description of the domain. All the subjects in this study came from the psychology department subject pool, who are all over 18 years old, psychology major freshman and native speakers of American English.

[1] Natalie Person, a professor at Rhodes College, is studying expert tutors, P.C. (Person, 2005).



Problem 1

Problem 2

Figure 1. Average post-test scores

Are expert tutors really more effective than non-expert tutors? How much more effective indeed? To answer these questions, we compared the post-test scores of the 3 groups of subjects. The post-test score is the average number of letters correct out of a total of 15 letters in each trial for each problem. Figure 1 shows the average post-test score of each trail for each post-test problem.

On the whole, we found that the expert tutor is indeed much more effective [1]. Specifically (all the statistical results are based on ANOVAs [2]; when significant, ANOVAs are followed by Tukey's tests to determine which condition is significantly different from the others):

- The expert tutor is significantly more effective than the other two tutors on both post-test problems ($p < 0.05$ in both cases);
- Collectively, the tutors are significantly better than the control (no tutoring) on post-test problem 2 ($p < 0.001$);
- The expert tutor is significantly more effective than the control on both post-test problems ($p < 0.005$).

3.3 Data Annotation: Does the Expert Tutor Use More Complex Language?

In order to do further analysis on the dialogues of tutoring, we transcribed and annotated a subset of tutoring sessions that were video-taped. The dialogues on two specific problems in the curriculum were chosen:

- Problem 2: an easy pattern at the beginning of the curriculum, to show what the tutor did at the very beginning;

  T R P N L

[1] Thanks to Andrew Corrigan-Halpern for the analysis and the graphs of the post-test scores.

[2] Analysis of variance (ANOVA) is used to test hypotheses about differences between two or more means.

- Problem 9: a much more complex pattern, to show what the tutor did when the pattern was getting very complex;

  B D D F F F C C E E G G G C

For each tutor, six subjects' dialogues were chosen, where the same subject solved problems 2 and 9, for a total of 36 dialogue excerpts. These 36 dialogues were transcribed and annotated with tutor and student moves. The transcription guidelines are a small subset of the CHILDES transcription manual (MacWhinney, 2000). The annotation scheme for the tutor and student moves is based on the literature (Chi et al., 2001; Litman et al., 2004), and designed with simplicity in mind.

The tutor moves include four high level categories, reaction, initiative, support, conversation. Tutor reaction and initiative are also subcategorized.

- Reaction: the tutor reacts to something the student says or does, which is subcategorized as follows:

  Answering: answering a direct question from the student

  (Student:this same line?) Tutor: yeah the same one.

  Evaluating: giving feedback about what the student is doing

  (student: then it goes A C E) Tutor: right

  Summarizing: summarizing what has been done so far

  Ok, so you said the pattern was go forward two, then backward one. A C B

- Initiative is subcategorized as follows:

Prompting: prompting the student into some kind of activity, further subcategorized as:

- **General:** laying out what to do next

  Why don't you try this problem

- **Specific:** trying to get a specific response from the student

  What would the next letter be?

Diagnosing: trying to determine what the student is doing

Why did you put a D there?

Instructing: providing the student with information about the problem. Further subcategorized as:

- **Declarative:** providing facts about the problem

  Notice the two Cs here? They are separating different parts of the problem

- **Procedural:** giving hints or tricks about how to solve problem

  Start by counting the number of letters in each period

Demonstrating: showing the student how to solve the problem.

Watch this. First I count the number of letters between the G and J here.

- **Support:** the tutor encourages the student in his/her work without referring to particular elements of the problem

  Great job on the last problem. This next one is a little harder.

- **Conversation:** acknowledgments, continuers, and small talk

  all right

There are six categories of student moves which have been annotated:

- **Explanation:** explaining what the student said or did, reasoning, or thinking aloud

  and see I put them like together.

- **Questioning:** asking the tutor a question

  around the second C?

- **Reflecting:** evaluating own's understanding

  I don't really understand about the whole C thing.

- Reaction: reacting to something the tutor says, further subcategorized:

  - **Answering:** directly answering a tutor's question

    (Tutor: and where else do you have three letters?) Student: right here F.

  - **Action Response:** performing some action (e.g., writing down a letter) in response to the tutor's question or prompt

    (Tutor: okay what if I ask you to start with the letter P?) Student writes letters J H F D B

- **Completion:** completing a tutor's utterance

  (Tutor: that's right but if you think of these Cs as +/) Student: separators right right.

- **Conversation:** same as the one for tutor moves – acknowledgments, continuers, and small talk

Further details of the annotation scheme are in Appendix B.

Table I is an example fragment of transcript with the annotations of tutor and student moves.

TABLE I

A TRANSCRIPT FRAGMENT FROM THE EXPERT'S TUTORING

Line No.	Utterances	Annotation
38	**Tutor:** how'd you actually get the n in the first place?	Diagnosing
39	**Student:** from here I count from c to g and then just from n to r.	Answering
40	**Tutor:** okay so do the c to g.	Specific Prompting
41	**Tutor:** do it out loud so I can hear you do it.	Specific Prompting
42	**Student:** $c\ d\ e\ f$.	Explanation
43	**Student:** so it's three spaces.	Answering
44	**Tutor:** okay so it's three spaces in between.	Summarizing
45	**Student:** $n\ o\ p\ q$ and r.	Explanation
46	**Tutor:** okay.	Evaluating
47	**Tutor:** you obviously made a mistake the first time.	Evaluating
48	**Tutor:** one of the more obvious methods would be like just count backwards and double-check everything.	Procedural Instructing
...	...	
53	**Tutor:** you're pretty good at the ones that are just one space away.	Support
54	**Tutor:** you probably don't have to worry about those right?	Support
55	**Tutor:** but if there's a big gap you probably do want to double-check okay?	Summarizing
56	**Tutor:** questions?	Diagnosing
57	**Student:** uh not really.	Reflecting

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 (Carletta, 1996; Di Eugenio and Glass, 2004). Table II and Table III report the rates of inter-annotator agreement on tutor moves respectively across all categories per tutor, and for the individual categories and subcategories. Table IV reports the rates of inter-annotator agreement on student moves.

TABLE II

KAPPA VALUES FOR TUTOR MOVES BY TUTOR

Level	Novice	Lecturer	Expert	Overall
Full	**0.688**	0.553	0.452	0.528
High Level	**0.750**	**0.655**	0.597	**0.644**

Table II reports two results: for the full scheme (13 categories), and with no subcategorization for instructing and prompting (high level, 9 categories). In both cases, the dialogues with the novice are the easiest to annotate (with highest inter-annotator agreement), followed by those with the lecturer and then those with the expert. If we look at the Kappa values for each category and subcategory in Table III, in the novice tutor's dialogues, 8 out of 13 categories are reliable; but in the lecturer's only 6 and in the expert tutor's only 4 categories are reliable. Even in the categories which are reliable for all the three tutors, such as prompting and specific prompting, the Kappa values for the expert tutor's dialogues are much lower than the novice tutor's. These show that the expert dialogues are the hardest to code. This supports the intuition that expert tutors use more sophisticated strategies, but does not bode well for computational modelling of expert tutors: if it is harder to code expert dialogues, the data on which to train the natural language interface will be less reliable than for other types of tutors.

TABLE III

KAPPA VALUES FOR EACH TYPE OF TUTOR MOVE

Category	Subcategory	Novice	Lecturer	Expert	Overall
Answering		**0.83**	**0.81**	0.53	**0.75**
Evaluating		**0.71**	0.45	**0.60**	0.56
Summarizing		0.47	**0.62**	**0.60**	**0.60**
Prompting		**0.92**	**0.84**	**0.74**	**0.82**
	General	0.52	0.16	0.39	0.34
	Specific	**0.82**	**0.77**	**0.61**	**0.73**
Diagnosing		**1**	**0.71**	0.47	**0.63**
Instructing		**0.65**	0.59	0.48	0.55
	Declarative	**0.65**	0.39	0	0.33
	Procedural	0	0.23	0.39	0.37
Demonstrating		0.52	0	0.37	0.39
Support		0	0.50	0.39	0.39
Conversation		**0.71**	0.47	0.59	0.55

In the last column of Table III, we report the overall Kappa values for different categories and subcategories. Some categories are very reliable, such as prompting, and its subcategory specific prompting; some categories are acceptable, such as diagnosing; some categories are not, such as support and instructing. The former is not problematic for my analysis, since there are very few instances of support in the coded data. The latter instead is, since instructing is one of the categories where tutors differ. Only when we collapse instructing and demonstrating (see Instr-Demon), which in fact the annotators reported as hard to distinguish, we obtain an acceptable overall Kappa value (0.63). Measuring inter-annotator reliability involves more than a single number. (Di Eugenio and Glass, 2004) argued that using multiple reliability metrics with different methods can be more revealing than a single metric. On the other hand, what counts as sufficiently reliable inter-annotator agreement depends on the use which the annotated data will be put to (Passonneau, 2006). We will implement these tutor moves in an ITS. If these categories with low kappa values are able to improve learning, these categories have validity for tutoring. In our study, to make the annotation results on tutor moves more reliable for further statistical analysis, the two annotators met with a graduate student overseer and further discussed their disagreements. Finally they came to an agreed upon coding for all the dialogues.

TABLE IV

OVERALL KAPPA VALUES FOR EACH TYPE OF STUDENT MOVE

Category	Kappa
Explanation	**0.64**
Questioning	**0.89**
Reflecting	**0.65**
Answering	**0.80**
Action Response	**0.97**
Completion	0.43
Conversation	**0.71**

Table IV shows that the student moves are much easier to annotate. Only completion got a low Kappa value, probably due to very few occurrences.

On the tutorial dialogues of the expert tutor, I also annotated the tutor's attitude and the student's confidence, which are used to model expert tutoring in Chapter 4, since our expert tutor is more effective. The tutor's attitude and student's confidence each has three possible annotations: positive, negative and neutral. If the tutor explicitly agrees the student's response, the attitude is positive. If the tutor explicitly disagrees with the student's response, the attitude is negative. "Neutral" is used for the tutor's attitude, if it belongs to neither of these two cases. I annotated the student's confidence as "positive," when the student responsed very confidently; "negative," when the student seems unsure about his/her own response; "neutral" for the cases in between. Only one annotator did the annotation of the tutor's attitude and the student's confidence so no inter-annotator agreement was computed.

3.4 Analysis of Tutorial Dialogues: What Brings the Effectiveness?

In Section 3.2 I showed that the expert tutor is indeed more effective than the non-expert tutors. So the next question is: from the language point of view, what does the expert tutor do that is more effective? In the following sections, I now answer this question based on the analysis of the frequencies of words and utterances, tutor moves, student moves and interaction patterns.

TABLE V

TUTOR UTTERANCES PER PROBLEM

	Novice	Lecturer	Expert
Problem 2	**10.33**	17.00	34.83
Problem 9	**16.17**	69.50	69.83

3.4.1 Frequency of Words and Utterances

Table V illustrates the average number of tutor utterances per problem. Comparing the number of tutor utterances, we found (all the statistical results are based on ANOVAs, followed by Tukey's tests):

- a main effect of problem ($p < 0.05$): there are more utterances for problem 9 than problem 2;
- a main effect of tutor ($p < 0.05$): the novice has significantly fewer utterances than the other two, i.e., both expert and lecturer have longer dialogues with subjects;
- an interaction between problem and tutor ($p < 0.05$): the novice's utterances don't significantly increase, the other two tutors' do.

Table VI illustrates the average number of tutor and student words, and of tutor and student utterances, per tutor. Numbers in boldface refer to significant differences, which show 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

TABLE VI

AVERAGE NUMBERS OF WORDS AND UTTERANCES PER TUTOR

	Novice	Lecturer	Expert
Tutor Words	107.33	369.17	419.17
Student Words	55.00	209.00	83.00
Student words / Tutor words	0.51	0.57	**0.20**
Tutor Utterances	13.25	43.25	52.33
Student Utterances	7.74	29.50	17.67
Student Utterances / Tutor Utterances	0.58	0.68	**0.32**

literature. For example, (Chi et al., 2001) 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. This expert tutor appears to talk to much, but still, he is effective. Clearly the explanation lies somewhere else. The comparison of the frequency of words and utterances does not answer my question about why expert tutors are more effective, so we need to look further into the individual moves.

3.4.2 Tutor Moves

Table VII reports the percentages of moves by tutor. Note that the columns don't add up to exactly 100, because a few utterances were left without any tag, and viceversa, few utterances with more than one tag – annotators were allowed to use more than one code, although they

TABLE VII

PERCENTAGES OF TUTOR MOVES BY TUTOR

Category	Novice	Lecturer	Expert
Answering	**10.1**	5.4	1.4
Evaluating	16.4	12.9	7.8
Summarizing	**6.9**	16.7	16.6
General Prompting	4.4	3.3	4.1
Specific Prompting	17.6	**27.7**	13.9
Diagnosing	2.5	3.3	3.3
Declarative Instructing	**22.6**	6.2	4.0
Procedural Instructing	0.6	4.4	**17.2**
Demonstrating	6.3	0.0	11.1
Support	0.6	0.6	5.4
Conversation	9.4	16.9	10.5
Instructing+Demonstrating	29.6	**11.2**	33.4

were not encouraged to do so. We ran Chi-square [1] on the data in this table. Numbers in boldface refer to significant differences ($p < 0.04$). The table shows us that:

- the novice tutor directly answers student's questions more than the other tutors (consistently, the students with the novice tutor ask more questions. I will further discuss this in the next section);
- the more experienced tutors (the lecturer and the expert tutor) summarize more than the novice;

[1] Chi square is a non-parametric test of statistical significance. Typically, the hypothesis tested with chi square is whether or not two different samples are different enough in some characteristic or aspect of their behavior.

- the more experienced tutors use declarative instructing less than the novice;
- the expert tutor does procedural instructing, demonstrating and support more than the other tutors;
- when collapsing instructing and its subcategories with demonstrating (the annotators have difficulties to distinguish these categories), the lecturer does it significantly less than the other tutors.

However, the expert tutor does not always behave as one would expect him to: he does NOT prompt his students more, the lecturer does. [1]

3.4.3 Student Moves

TABLE VIII

PERCENTAGES OF STUDENT MOVES BY TUTOR

Category	Novice	Lecturer	Expert
Explanation	**7.5**	26.3	19.8
Questioning	**18.3**	8.4	6.8
Reflecting	14.2	16.5	13.9
Answering	25	27.1	35.4
Action Response	12.5	10.4	9.7
Completion	0	0.8	0.8
Conversation	22.5	10.6	13.5

[1] Thanks to Trina C. Kershaw for the analysis in Section 3.4.1 and Section 3.4.2.

Table VIII shows the percentages of student moves by tutor. I ran a Chi-square test on this data. There are significant differences between the novice and the two more experienced tutors in student's explanations and questioning:

- the students with the novice tutor explain much less than with the more experienced tutors;
- the students ask more questions of the novice tutor.

The behavior of the students with the novice tutor shows that the novice tutor behaves precisely as we would expected a novice tutor to: the novice tutor does lots of explicit instructing but does not prompt the student to do self-explanation. Also the students ask questions more frequently, perhaps because they feel more confusion when they are with the novice tutor; or perhaps because of social factors (the novice tutor is young and female, the other two tutors are older and male). The lecturer certainly behaves at least in one aspect as good tutors should: he does lots of prompting.

3.4.4 Interaction Patterns

The individual analysis of the tutor and student moves does not provide enough information 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 precisely because of the interaction between tutor and student. (Chi et al., 2001) discovered some interaction patterns from their study of human tutoring. For example, tutor scaffoldings elicited shallower follow-up than deep follow-up, which explains why students's responses to scaffolding correlated only with shallow learning. So the analysis of interaction patterns is able to answer the question – what does the expert tutor do that is more effective?

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: How do the expert tutor and the non-expert tutors react differently to each type of student move?

Table I (in Section 3.3) presents a fragment from a transcript of the expert's tutoring. A pair of moves from two different speakers that appear in sequence is an interaction pattern, which is called an "adjacency pair" in computational linguistics. For example, after the tutor's diagnosing in line 38, the student gives an answer in line 39. This forms a tutor-student interaction pattern — "T–diagnosing + S–answering." Then the tutor does a specific prompting, so line 39 and line 40 form a student-tutor interaction pattern — "S–answering + T–specific prompting." The student's explanations in line 42 and line 45 show that he is explaining his answer in line 39. Totally there are 72 possible types of tutor-student patterns and 72 possible types of student-tutor patterns, which are the combinations of 12 categories of tutor moves and 6 categories of student moves. (For the moment, I left out "Conversation"s in tutor moves and student moves, since conversation in general does not pertain to the subject matter.)

First I compared the total number of tutor-student patterns and student-tutor patterns and the number of pattern types. Table IX reports the number of interaction patterns and pattern types. Numbers in boldface refer to significant differences. (We use Chi-square as the significance test.) I found that in the tutoring dialogues from the novice tutor there are many fewer types of interaction patterns than in the dialogues from the other two tutors; the expert tutor has a similar number of pattern types in many fewer interactions than the lecturer. This supports the finding that expert tutors tend to use more varied tutorial strategies and language than the non-expert tutors (Glass et al., 1999).

TABLE IX

NUMBER OF INTERACTION PATTERNS AND TYPES, PER TUTOR

Interaction Pattern	Novice	Lecturer	Expert
Tutor-Student			
Types	**22**	37	39
Frequency	49	206	128
Ratio	0.45	**0.18**	0.30
Student-Tutor			
Types	**16**	31	38
Frequency	50	205	127
Ratio	0.32	**0.15**	0.30

3.4.4.1 Tutor-Student Interaction Patterns: Student's Reactions to Tutor Moves

I ran Chi-square on the frequencies of all tutor-student interaction patterns. Across all patterns, there are significant differences in student's reactions to tutor moves between the novice tutor and the other two tutors ($p < 0.01$). In each type of pattern that starts with a specific tutor move, each group of students reacts significantly differently ($p < 0.05$) to each type of tutor move with the exception of specific prompting. More specifically, I found:

- **Answering:** the novice tutor's answer is followed by student's questioning, not for the other two tutors;
- **Evaluating:** the lecturer's evaluating leads to many more student explanations but many fewer reflecting moves than the expert and novice tutor;
- **Summarizing:** with the novice tutor students almost never react to summarizing; the lecturer's summarizing leads to more student's reflecting; on the contrary, the expert tutor's leads to more student's explanation (e.g., in Table I, the expert tutor summarizes in line 44 and then in line 45 the student does explanation);
- **General Prompting:** the students with the expert tutor never have questions after his general prompting, but they do with the non-expert tutors (the novice tutor and the lecturer);
- **Specific Prompting:** the specific prompts from the more experienced tutors lead the students to explain much more than for the novice tutor (e.g., in Table I, the expert tutor does specific prompting in line 41 and then in line 42 the student gives an explanation); to the tutor's specific prompting, the students with the novice tutor respond with many more questions than with the other tutors;
- **Procedural Instructing:** the lecturer's procedural instructing leads to more reflecting (i.e. assessing one's own understanding); the expert tutor's leads to more explanation;

- **Demonstrating:** with the non-expert tutors, students hardly react to demonstrating; on the contrary, the expert tutor's demonstrating leads to any kind of student move.
- **Support:** with the non-expert tutors, students hardly react to support; on the contrary, the expert tutor's support leads to any kind of student move.

Comparing the expert tutor with the lecturer, although he does specific prompting significantly less than the lecturer and his students do less explanation than the lecturer's students, he tends to use more varied strategies to get the students to self-explain, instead of just specific prompting. Comparing the expert with the other two tutors, the expert's answering, general and specific prompting are possibly clearer to the students, since the students have no questions. Also demonstrating and support are the most interesting strategies that make the expert tutor different from the other tutors. Unfortunately the Kappa values of these two categories are very low. The former is hard to distinguish from instructing. The latter only has a few instances. Since the analysis is based on the annotations that were finally agreed upon by the two annotators through discussing with another supervisor, I can draw some preliminary conclusions on those categories with low Kappa values but none of them are very solid. However, as I mentioned in Section 3.3, whether these results are sufficiently reliable depends on the uses of the data. So we will know whether the two categories are really meaningful for tutoring after I implement them in an ITS. Table X summarizes the tutor-student interaction patterns in which the expert tutor is different from the non-expert tutors.

TABLE X

TUTOR-STUDENT INTERACTION PATTERNS OF THE EXPERT TUTOR

Tutor Move	Student Move
Summarizing	Explanation
Procedural Instructing	Explanation
Demonstrating	Explanation
Demonstrating	Reflecting
Support	Answering

3.4.4.2 Student-Tutor Interaction Patterns: Tutor's Reactions to Student Moves

Since at the moment we are more interested in providing feedback to the student rather than in interpreting the student's verbal input, it is more important to analyze how the tutor reacts to a student move. There are significant differences ($p < 0.02$) in tutor's reactions to student moves between all the tutors. Further I analyzed the student-tutor interaction patterns in the following two directions:

1. how the tutors react differently to each type of student move;
2. which student moves the tutors react to, using each type of tutor move.

In the first direction I found:

- **Explanation:** the novice tutor uses summarizing much less than the more experienced tutors; in response to a student's explanation, the lecturer uses specific prompting much more than the other moves and the other tutors;

- **Questioning:** the expert tutor does not answer immediately or directly, but the non-expert tutors do;
- **Reflecting:** the expert tutor uses much more procedural instructing, demonstrating and general prompting;
- **Answering:** the novice uses many fewer specific prompts but much more evaluating and declarative instructing — she appears to immediately deliver the knowledge or the solution;
- **Action Response:** the expert tutor uses many more summarizing and procedural instructing moves — actions involve procedures, so summarizing and procedural instructing moves may be more appropriate.

In the second direction (using each type of tutor move, which student moves the tutors react to), I found:

- **Evaluating:** the more experienced tutors evaluate the student's explanation more than the student's answer and they reflect more (e.g. in Table I, after the student's explanation in line 45 the expert tutor does evaluating in line 46);
- **Summarizing:** the more experienced tutors summarize more after a student's explanation, reflecting and action response — these involve more information to be summarized;
- **Specific Prompting:** the lecturer does specific prompting after any kind of student move instead of just in response to answering as the novice and expert tutors do;

TABLE XI

STUDENT-TUTOR INTERACTION PATTERNS OF THE EXPERT TUTOR

Student Move	Tutor Move
Explanation	Diagnosing
Summarizing	Diagnosing
Reflecting	General Prompting
Reflecting	Declarative Instructing
Reflecting	Procedural Instructing
Reflecting	Demonstrating
Action Response	Summarizing
Action Response	Procedural Instructing

- **Diagnosing:** the expert tutor diagnoses after any kind of student move, not just the student's reaction moves (answering and action response);
- **Declarative Instructing:** the expert tutor mostly does declarative instructing after the student's reflecting;
- **Procedural Instructing:** the more experienced tutors do more procedural instructing after the student's reflecting;
- **Demonstrating:** the expert tutor does more demonstrating after the student's reflecting, the lecturer never does demonstrating — in this particular domain, demonstration may be more useful.

Table XI summarizes the student-tutor interaction patterns in which the expert tutor is different from the non-expert tutors.

3.4.5 Multiple-Utterance Turns

While I was studying the interaction patterns, I observed that not all of tutor's specific prompting are immediately followed by any student move: 35.6% of the expert tutor's specific prompting is not immediately followed by any student move, which is much higher than that of the lecturer's (21.5%) and the novice's (25%). For example, in Table I, the expert tutor does specific prompting in line 40 but this specific prompting is followed by another specific prompting, instead of a student turn. This may be because most of the time the expert tutor does specific prompting in multi-utterances. This phenomenon also appears for other tutor moves, like from line 46 to line 48: in this single turn, the expert tutor uses three utterances and two categories of move.

Multi-utterances usually mean that in a single turn the tutor or the student make a sequence of moves (more than one) in succession without being interrupted. The number of utterances in a single turn is called the "length" of the multi-utterance turn. The utterances are segmented based on the CHILDES transcription manual (MacWhinney, 2000), which the transcribers used. So the first question is: what is the difference between the expert tutor and the non-expert tutors in lengths and frequencies of tutor multi-utterance turns and student multi-utterance turns? To answer this question, I counted the lengths and frequencies of tutor multi-utterance turns and student multi-utterance turns in each tutoring transcripts (for both problem 2 and problem 9 in the curriculum, three tutors, there are a total of 36 transcripts). Then I ran ANOVA on the counts to see whether there are significant differences between each pair of tutors and between the two problems.

Figure 2(A) shows the average lengths of multi-utterance tutor and student turns per problem. There is a significant difference in the average length of multi-utterance student turns between problem 2 and problem 9 ($p < 0.03$). Problem 9 is much more complex than problem 2 so the students use more utterances in a single turn.

Figure 2(B) shows the average lengths of multi-utterance tutor and student turns per tutor. The average length of the expert tutor's multi-utterance turn is significantly greater than the non-expert tutors' ($p < 0.005$). This means that the expert tutor talks more in each turn. The length of the expert tutor's multi-utterance turn varies from 1 to 22, but the maximum length of the Lecturer's is 9 and only two turns of the novice tutor have a length greater than 7. I ran Chi-square on the length distributions of the three tutors' turns and there are significant differences between tutors in length 1, length 3 and length 4 ($p < 0.05$). The expert tutor's turns with only one utterance are significantly fewer than the non-expert tutors, but his 3-utterance and 4-utterance turns are significantly more than the novice tutor. It supports that the expert tutor tends to talk more in each single turn.

The next question is how differently the expert tutor organizes his turn from the non-expert tutors. I analyzed the multi-utterance patterns of tutor turns with regards to how the tutors follow up differently each particular tutor move. First I looked at the differences between tutors as concerns which categories of tutor move are more likely followed by another tutor move. I ran a Chi-square test on the data in Table XII. (Numbers in boldface refer to significant differences.) I found:

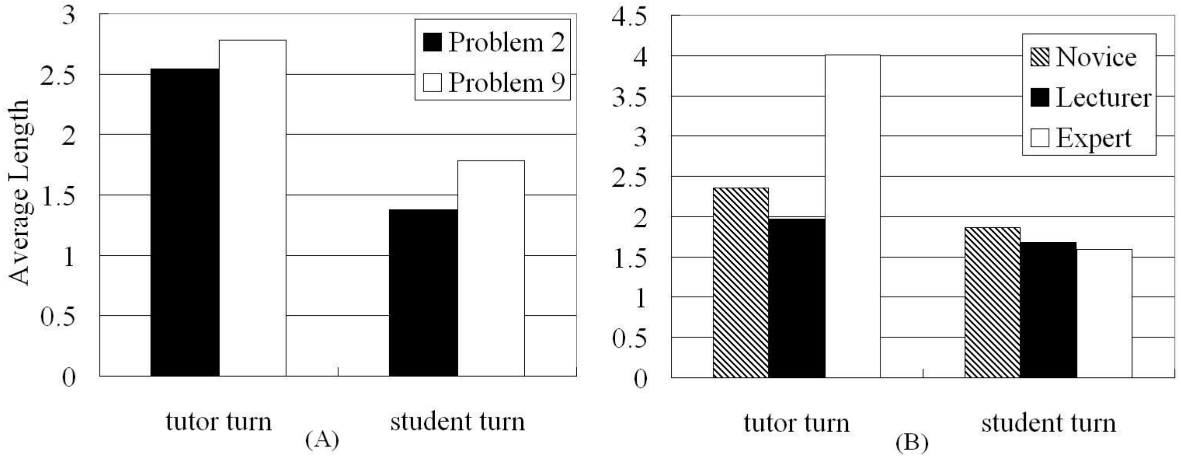


Figure 2. Average length of multi-utterance tutor and student turns, per problem(A) and per tutor(B)

- the novice tutor has significantly fewer summarizing moves, but many more declarative instructing moves followed by another move than the expert tutor and the lecturer ($p < 0.003$ in both cases);
- the expert tutor has significantly more procedural instructing and support followed by another move than the non-expert tutors ($p < 0.004$ in both cases);
- the lecturer has much more evaluating followed by another move than the novice and expert tutors ($p < 0.03$);
- the lecturer does not have demonstrating followed by another move but the novice and expert tutors do ($p < 0.03$);

TABLE XII

PERCENTAGES OF EACH CATEGORY OF TUTOR MOVE FOLLOWED BY ANOTHER TUTOR MOVE, PER TUTOR

Tutor Moves	Novice(%)	Lecturer(%)	Expert(%)
Answering	5	7.212	0.743
Evaluating	13.75	**22.6**	9.653
Summarizing	**11.25**	30.77	22.77
General Prompting	5	3.365	4.455
Specific Prompting	8.75	14.9	7.673
Diagnosing	3.75	0.962	2.723
Declarative Instructing	**40**	10.58	5.198
Procedural Instructing	1.25	7.212	**23.02**
Demonstrating	11.25	**0**	15.59
Support	0	0.481	**6.188**

Procedural instructing teaches the student how to solve a problem procedurally so it can seldom be completed in one single utterance. So I speculate that the expert tutor likes to use complete procedural instructing to help students. Before continuing the tutoring, the expert tutor also likes to encourage his student with support that would push students to move forward.

As for interaction patterns, it is more meaningful to find out that after each category of tutor move, how the expert tutor differs in the following move from the non-expert tutors. I ran a Chi-square test on the frequencies of all the multi-utterance patterns of the tutors. Across all patterns, there are significant differences in the moves following each category of tutor move between all the tutors ($p \approx 0$). More specifically, I found:

- **Answering:** the expert tutor does specific prompting much more than the non-expert tutors after answering — this shows our expert tutor often prompts and scaffolds students but usually, after answering student questions;
- **Evaluating:** the expert tutor and the lecturer do specific prompting much more than the novice tutor after evaluating; the expert tutor does procedural instructing much more than the non-expert tutors;
- **Summarizing:** the expert tutor does summarizing in multiple utterances much more than the non-expert tutors;
- **General Prompting:** the expert tutor does much less specific prompting than the non-expert tutors after general prompting;
- **Specific Prompting:**
  - the expert tutor and the lecturer do procedural instructing much more than the novice tutor after specific prompting;
  - all the three tutors do specific prompting in multiple utterances;
- **Diagnosing:** the expert tutor does much more procedural instructing and support than the non-expert tutors after diagnosing;
- **Declarative instructing:** the expert tutor does much more procedural instructing and demonstrating, but much less specific prompting than the non-expert tutors, after declarative instructing;

- **Procedural Instructing:** the expert tutor does procedural instructing in multiple utterances much more than the non-expert tutors; he also does much more demonstrating, but much less specific prompting than the non-expert tutors, after procedural instructing;
- **Demonstrating:** the lecturer never does demonstrating but the novice and expert tutors do demonstrating in multiple utterances;
- **Support:** the expert tutor does almost any kind of tutor move after support.

TABLE XIII

PATTERNS OF MULTI-UTTERANCE TURNS OF THE EXPERT TUTOR

Tutor Move	Tutor Move
Answering	Specific Prompting
Evaluating	Procedural Instructing
Summarizing	Summarizing
Diagnosing	Procedural Instructing
Diagnosing	Support
Declarative Instructing	Procedural Instructing
Declarative Instructing	Demonstrating
Procedural Instructing	Procedural Instructing
Procedural Instructing	Demonstrating
Support	Summarizing
Support	Procedural Instructing
Support	Support

Comparing the novice tutor with the expert tutor and the lecturer, she does declarative instructing after almost any kind of tutor move much more than the other two tutors. This supports our finding that the novice tutor tends to give out the information or tell the solution directly. These findings above hint at why the expert tutor is much more effective than the non-expert tutors even though he prompts less, talks more and leaves less talking to students, as compared to the lecturer: the expert tutor summarizes more completely, does procedural instructing and demonstrating more effectively and encourages students by support before moving on. Table XIII summarizes the patterns of multi-utterance turns in which the expert tutor is different from the non-expert tutors.

3.5 Discussion

Our analysis of tutorial dialogue moves, interaction patterns and multi-utterance turns explains why the expert tutor differs from non-expert tutors. The expert tutor is much more effective than the non-expert tutors because of the following behaviors and natural language features:

1. Instead of delivering information directly, he demonstrates or models the process for solving the problem (demonstrating, procedural instructing);
2. Before moving on, he finds success, and reinforces effort, in even minor accomplishment (support)— although there are not many supports in the tutoring dialogues, the expert tutor does it in various situations and much more frequently than the non-expert tutors;
3. Summarizes and reviews (summarizing);
4. Assesses the situation not only after a student's answer or action (diagnosing);
5. Uses questions to enhance problem solving (prompting).

CHAPTER 4

COMPUTATIONAL MODELING OF EXPERT TUTORING FEEDBACK

After highlighting the characteristics of our expert tutor, I am able to model expert tutoring to describe how at least one expert tutor gives natural language feedback to his students. In this chapter, I present a rule based model of how tutors generate their feedback. With all the dialogues, I use a machine learning technique — classification based on associations (CBA) — to learn tutorial rules for generating effective natural language feedback in ITSs.

4.1 Related Work

Based on Anderson's ACT-R theory (Koedinger et al., 2003), production rules can be used to realize any cognitive skill (more details on ACT-R will be given in Chapter 5). Therefore I can use production rules as a formalism to model expert tutoring computationally. The production rules can be designed manually or learned from the human tutoring transcripts. The dialogue management of AUTOTUTOR (Graesser et al.,) embeds a set of 15 fuzzy production rules to select the next dialogue move for the tutoring system. Fuzzy production rules are tuned to the quality of the student's assertions in the preceding turn, global parameters that refer to the ability, verbosity, and initiative of the student, and the extent to which the good answer aspects of the topic have been covered (Graesser et al.,). Although these production rules have achieved some good results for AUTOTUTOR, they are defined manually and only cover limited situations. Manual design of production rules needs extensive discussion with a set of expert tutors in a particular domain and a meta-analysis of all the previous studies in the learning science and ITS community.

In recent years, several researchers have applied machine learning techniques to transcript analysis, such as dialogue act prediction, cue word usage, planning rules and discourse segmentation. For example, Vander Linden and Di Eugenio used decision tree learning to learn micro-planning rules for preventative expressions (Linden and Di Eugenio, 1996a; Linden and Di Eugenio, 1996b). The CIRCSIM group has applied machine learning to discover how human tutors make decisions based on the student model (Zhou, 2000; Freedman et al., 1998).They used Quinlan's C4.5 learning algorithm (Quinlan, 1993) to find tutoring rules. C4.5 is a decision tree learning algorithm. To generate a decision tree, the algorithm will iteratively choose a feature to as a basis for branching the tree until a final decision is reached at a leaf node. They obtained about 80% to 88% accuracy across several choices:

1. Choosing a response strategy: 57 cases with 88% accuracy;
2. Choosing an explicit acknowledgement: 62 cases with 80% accuracy;
3. Choosing a realization within a topic: 18 cases with 83% accuracy;
4. Choosing a tutorial strategy: 23 cases with 87% accuracy.

However, since they only reported accuracy on training but not on testing, it's very hard to understand how good their approach is.

4.2 Method

Although there have been many attempts to apply machine learning to help discourse analysis, researchers are still investigating what machine learning techniques are suitable to the acquisition of knowledge for discourse processing. Currently quite a few machine learning techniques can achieve remarkable performance for classification. Classification based on associations (CBA) which integrates classification and association rule mining can generate class association rules and can do classification more accurately than C4.5 (Liu et al., 1998). (On the same datasets, CBA decreases the error rate from 16.7% for C4.5 to 15.6% on average.) CBA can generate understandable rules, find all possible rules that exist in data and discover interesting or useful rules specifically for an application. CBA also provides a feature selection module which partially reduces the burden of selecting features to improve performance. So in the next section, I propose to learn tutorial feedback rules by using CBA.

Association rule mining is an important data mining model studied extensively by the database and data mining community (Agrawal and Srikant, 1994; Klemettinen et al., 1994; Liu et al., 1998). Classification association rules (CAR) are association rules with target on the right hand side of the rules. A class association rule (CAR) is an implication of the form

$$X \rightarrow y, \text{where } X \subseteq I, \text{and } y \in Y. \tag{4.1}$$

X is a set of features. I is the set of all features. y is the target class, which will be a tutor move category in our domain. Y is the set of all classes. CBA also provides strength measurements for the CARs:

Support The rule holds with support *sup* if *sup*% of cases contain X or y.

Confidence The rule holds with confidence $conf$ if $conf$% of cases that contain X also contain y.

This means that an association rule is a pattern that states that when X occurs, y occurs with a certain probability. So when CBA does classification, more than one rule can fit a certain case and the final class will be derived from the rule with highest confidence. If the confidences of the rules are the same, the rule with highest support will be picked. Again if the supports are also equal, CBA will classify the case according to the rule which is generated earlier than the others. Of course, there will be some cases that no CARs can classify. CBA saves a default class to deal with this kind of situation.

When a tutor makes decisions for different cases, he/she may decide based on different sets of features and some of the features may not have precedence with respect to each other. Also some features may not exist for a certain case. For example, suppose I have three cases(T_1, T_2, T_3), five features(f_1: student move = “answering,” f_2: correctness of student move = “wrong,” f_3: correctness of student move = “partially correct,” f_4: hesitation time = 30 s, f_5: student move = “reflecting”), two categories of tutor move (evaluating, specific prompting) to predict and the following 4 rules:

R1: f_1: student move = "answering," f_2: correctness of student move = "wrong" → evaluating

R2: f_3: correctness of student move = "partially correct," f_4: hesitation time = 30 s, f_5: student move = "reflecting" → specific prompting

R3: f_1: student move = "answering," f_4: hesitation time = 30 s → evaluating

R4: f_2: correctness of student move = "wrong," f_5: student move = "reflecting" → specific prompting

The features in the three cases are:

T_1**:** f_1: student move = "answering," f_2: correctness of student move = "wrong"

T_2**:** f_3: correctness of student move = "partially correct," f_4: hesitation time = 30 s, f_5: student move = "reflecting"

T_3**:** f_1: student move = "answering," f_2: correctness of student move = "wrong," f_4: hesitation time = 30 s, f_5: student move = "reflecting"

None of the three cases has all five features. For T_1 and T_2, R1 and R2 can predict the tutor moves without any conflicts: when the student gives a wrong answer, the tutor evaluates it; when the student's move is partially correct with a reflection of his/her understanding and there is a 30 second pause, the tutor does specific prompting. In T_3, the student makes a move which is both an answer and a reflecting, the answer is wrong and there is a 30 second pause. R1, R3 and R4 can all be applied but R4 predicts a tutor move "specific prompting," which conflicts with the prediction of R1 and R3 "evaluating." Since CBA orders the rules based on confidence and support, this conflict will be solved by applying R1, which has the highest precedence in the rule set, without removing R2 and R3, which may be useful for other cases. This is very useful, especially when there are not a large number of annotated transcripts. Some rules are not the best choice for certain cases but may be good choices for other cases.

4.3 Feature Selection

The features used in the rules correspond to the annotations of the tutoring dialogues. Since the annotation scheme is also based on discussions involving the expert tutor, the features reflect the information that the expert used to make choices in giving feedback to his students. Table XIV lists all the features and their possible values. CBA will automatically generate rules with a subset of these features. Among the moves, 8 moves starting with "T-" are tutor moves and 6 moves starting with "S-" are student moves. Since our data set is relatively small, I only use 8 higher level tutor moves to reduce the model complexity. I need to account for the fact that within one utterance speakers may talk about more than one letter relationship and scope. Using binary "Yes" and "No" for the different types of letter relationships and relationship scopes allows us to do so, e.g., I can classify an utterance as "forward = yes" and "repeat = yes" to signify that both relationships are discussed (this utterance would be classified as "no" as regards all other relationship types if no other is discussed). The last feature is the student's knowledge state on each type of letter relationship, which is not annotated but computed from the values of other features within each dialogue excerpts. The computation is based on this formula:

$$k = \lfloor \frac{p \times 0.5 + w}{t} \times 5 \rfloor \tag{4.2}$$

where p is number of student actions/inputs which are partially correct, w is number of student actions/inputs which are wrong and t is total number of student actions/inputs. So the knowledge state k ranges from 0 to 5. The higher the value, the worse the performance on this type of letter relationship is. The reason to compute this feature is that the expert tutor gives feedback not only based on student's most recent performance but also the overall performance.

4.4 Experiments and Results

Tutorial dialogues are time series data, which means that the prediction of what the expert tutor should do now should be based on information of the last one or more utterances. I did several experiments that include different lengths of history. I found that using the features from only the last utterance gave us the highest prediction accuracy. This may be because we only have a total of 12 dialogues. The longer the history we consider, the sparser the data. So in the experiments we present here, only the features from the last utterance are used. In the 12 dialogues, there are a total of 388 transitions. However, this collection of transitions is very unbalanced. For some prediction categories, there are only 5% or fewer transitions. For example, for 2% of the transitions we have tutor move "answering." As discussed in (Chawla et al., 2002; Japkowicz, 2000), in such cases the dominant class of the target feature will be favored. We did synthetic over-sampling on the minority categories using the SMOTE algorithm (Chawla et al., 2002) to balance the number of transitions across different prediction categories.

Using 12 dialogues with 6-way cross validation, we did 4 experiments to learn tutorial rules for choosing the tutor's attitude, the letter relationship that the tutor will talk about, the relationship scope within the problem that the tutor will focus on, and the tutor move. Tutor's feedback to students can be divided into 3 categories according to the tutor's attitude towards students: positive, neutral and negative. Positive feedback is often given when a student does something correct. Negative feedback is often given when a student does something wrong. Another study of tutoring in the letter pattern task (Corrigan-Halpern, 2006) shows that giving positive and negative feedback at appropriate time and scope will improve learning. The letter relationship is the basic concept in the letter pattern task. The relationship scope concerns the coverage of each type of letter relationship. During tutoring, tutors need to choose the concepts to teach students and discuss with them, and also need to decide how to break down the problem and choose an appropriate coverage. These steps are often known as "choosing the topic" in ITSs (Evens and Michael, 2006). Choosing the topic and choosing the strategy often go together dependent on the context in which the decision has to be made. A tutor move is akin to a response strategy.

Table XV reports the number of rules and the precision, recall, F-score and accuracy of training and testing in learning these four sets of tutorial rules. In the table, N is for the number of rules; P is for the precision; R is for the recall; F is for the F-score; A is for the accuracy. These measures are defined as follows:

$$Precision = \frac{TP}{TP + FP} \tag{4.3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.4}$$

$$F - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4.5}$$

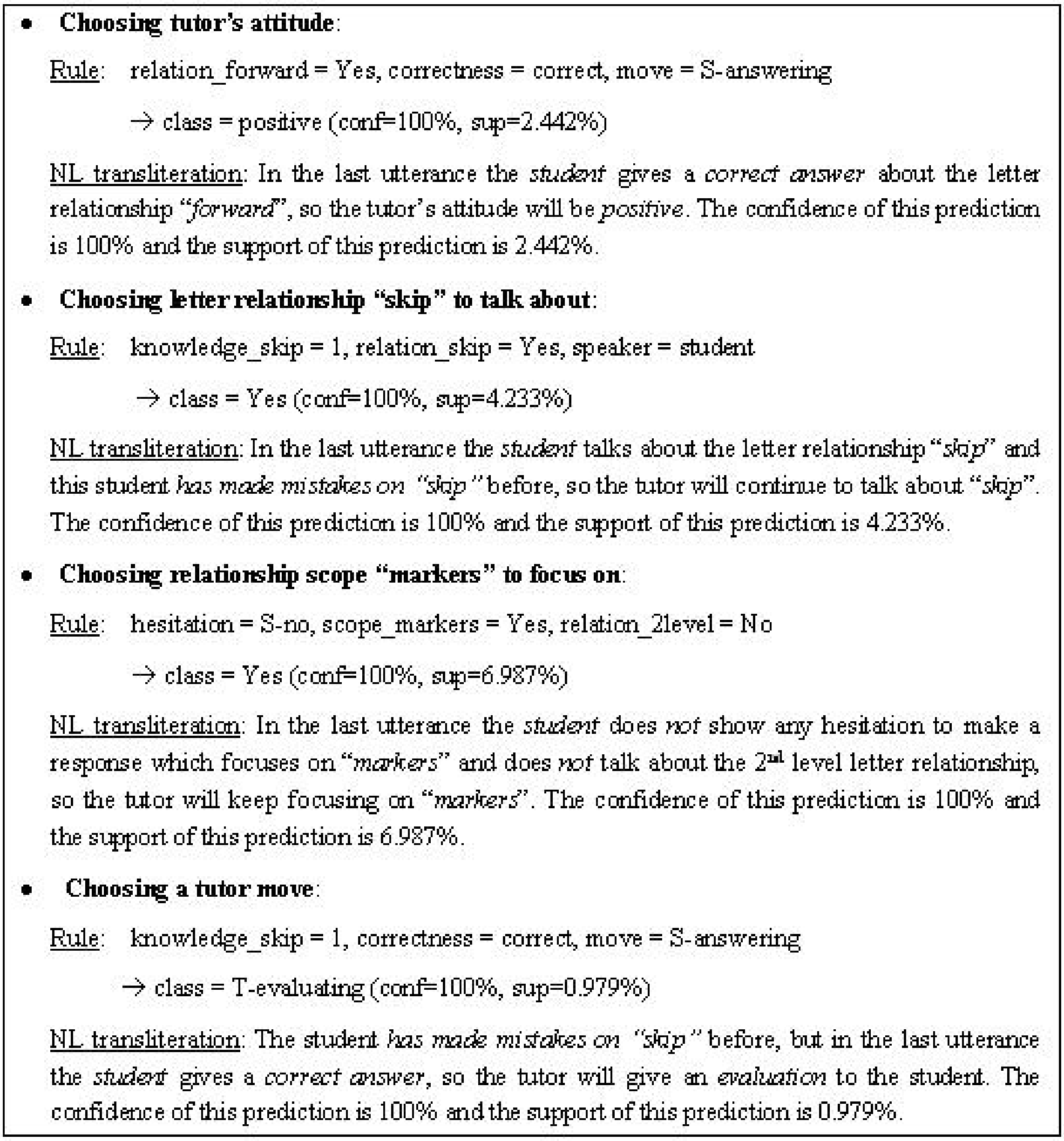
$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4.6}$$

Here “TP” is the number of true positive predictions; “FP” is the number of false positive predictions; “TN” is the number of true negative predictions; “FN” is the number of false negative predictions. Precision is the proportion of correct predictions for a certain category to all the predictions for this category. Recall is the proportion of correct predictions for a certain category to all the transitions for this category in the data set. In Table XV, there are no results for the letter relationship “progress(skip)” because in our data this type of letter relationship never appears.

4.5 Discussion

The different sets of rules learn different classifications; the number of classes is different for each set. For the tutor’s attitude, the rules learn to choose among positive, neutral and negative. For the tutor move, the rules learn to choose among 8 higher level tutor moves, which means that in this experiment I do not distinguish between “general prompting” and “specific prompting” or between “declarative instructing” and “procedural instructing.” As concerns the letter relationship to talk about and the relationship scope to focus on, the rules learn, for each type, whether to choose it or not (recall that the classification is binary here). If the decision is “Yes,” this type of relationship or this relationship scope will be covered. In Figure 3 some example rules and their natural language transliterations are listed.

The results in Table XV show that the tutorial rule learning obtains very good precision, recall and F-score in both the training and testing data. As concerns tutor moves, the results



- **Choosing tutor's attitude**:

  Rule: relation_forward = Yes, correctness = correct, move = S-answering

  → class = positive (conf=100%, sup=2.442%)

  NL transliteration: In the last utterance the *student* gives a *correct answer* about the letter relationship "*forward*", so the tutor's attitude will be *positive*. The confidence of this prediction is 100% and the support of this prediction is 2.442%.

- **Choosing letter relationship "skip" to talk about**:

  Rule: knowledge_skip = 1, relation_skip = Yes, speaker = student

  → class = Yes (conf=100%, sup=4.233%)

  NL transliteration: In the last utterance the *student* talks about the letter relationship "*skip*" and this student *has made mistakes on "skip"* before, so the tutor will continue to talk about "*skip*". The confidence of this prediction is 100% and the support of this prediction is 4.233%.

- **Choosing relationship scope "markers" to focus on**:

  Rule: hesitation = S-no, scope_markers = Yes, relation_2level = No

  → class = Yes (conf=100%, sup=6.987%)

  NL transliteration: In the last utterance the *student* does *not* show any hesitation to make a response which focuses on "*markers*" and does *not* talk about the 2nd level letter relationship, so the tutor will keep focusing on "*markers*". The confidence of this prediction is 100% and the support of this prediction is 6.987%.

- **Choosing a tutor move**:

  Rule: knowledge_skip = 1, correctness = correct, move = S-answering

  → class = T-evaluating (conf=100%, sup=0.979%)

  NL transliteration: The student *has made mistakes on "skip"* before, but in the last utterance the *student* gives a *correct answer*, so the tutor will give an *evaluation* to the student. The confidence of this prediction is 100% and the support of this prediction is 0.979%.

Figure 3. Example tutorial rules

for “summarizing,” “prompting,” and “instructing” are relatively low, as they have F-scores only around 0.3 or 0.4. There are three possible reasons:

- The data set is too small;
- There are other undiscovered features that are important for choosing tutor moves;

Comparing the accuracies of performance on training to the ones from the CIRCSIM group, I obtain much higher accuracies as concerns tutor’s attitude, the letter relationship which the tutor will talk about and the relationship scope within the problem which the tutor will focus on. On the training set (Recall, they didn’t evaluate on a test set), the accuracy for tutor moves is not as high. However, it is sufficient as a basis for the experiments, since our ultimate evaluation measure is whether the natural language feedback generated based on these rules can improve learning.

TABLE XIV

FEATURES AND POSSIBLE VALUES FOR LEARNING TUTORIAL RULES

Feature		Possible Values
Speaker		Student, Tutor
Move		T-answering, T-evaluating, T-summarizing, T-prompting, T-diagnosing, T-instructing, T-demonstrating, T-support,
		S-explanation, S-questioning, S-reflecting, S-answering, S-action response, S-completion
Student Correctness		Correct, partial, wrong
Tutor's attitude		Positive, neutral, negative
Student's confidence		Positive, neutral, negative
Hesitation		T-long, T-medium, T-short, T-no,
		S-long, S-medium, S-short, S-no
Letter relationship	2nd level	Yes, No
	Forward	Yes, No
	backward	Yes, No
	Marker	Yes, No
	Repeat	Yes, No
	Progress(length)	Yes, No
	Progress(skip)	Yes, No
	skip	Yes, No
Relationship Scope	Whole	Yes, No
	In 1st level	Yes, No
	2nd level	Yes, No
	Markers	Yes, No
Knowledge state on each letter relationship	2nd level	1,2,3,4,5
	Forward	1,2,3,4,5
	backward	1,2,3,4,5
	Marker	1,2,3,4,5
	Repeat	1,2,3,4,5
	Progress(length)	1,2,3,4,5
	Progress(skip)	1,2,3,4,5
	skip	1,2,3,4,5

TABLE XV

PRECISION, RECALL, F-SCORE AND ACCURACY OF TRAINING AND TESTING IN LEARNING TUTORIAL RULES

Prediction Category		N	Training				Testing			
			P	R	F	A (%)	P	R	F	A (%)
Tutor's attitude	positive	111	0.916	0.923	0.919	93.628	0.849	0.910	0.877	89.49
	neutral		0.932	0.907	0.917		0.905	0.805	0.852	
	negative		0.962	0.984	0.972		0.935	0.970	0.951	
Letter relationship	2nd level	79	0.942	0.956	0.949	94.768	0.847	0.902	0.872	86.916
	forward	78	0.964	0.946	0.955	95.568	0.887	0.895	0.889	89.004
	backward	50	0.977	0.977	0.977	97.695	0.940	0.976	0.957	95.819
	marker	79	0.942	0.938	0.940	93.972	0.833	0.896	0.862	85.816
	repeat	63	0.968	0.990	0.979	97.894	0.913	0.949	0.929	92.874
	progress(length)	58	0.971	0.981	0.976	97.696	0.936	0.967	0.951	95.020
	progress(skip)	-	-	-	-	-	-	-	-	-
	skip	80	0.943	0.944	0.943	94.318	0.849	0.846	0.847	84.799
relationship scope	whole	88	0.961	0.939	0.950	95.036	0.867	0.892	0.879	87.950
	in 1st level	82	0.954	0.960	0.957	95.669	0.895	0.905	0.899	90.155
	2nd level	77	0.953	0.959	0.956	95.573	0.876	0.936	0.904	90.132
	markers	79	0.924	0.962	0.942	94.109	0.824	0.913	0.861	85.448
Tutor move	Answering	189	0.971	1	0.985	78.261	0.915	1	0.954	56.789
	Evaluating		0.782	0.791	0.785		0.629	0.655	0.636	
	Summarizing		0.808	0.614	0.696		0.377	0.305	0.331	
	Prompting		0.812	0.669	0.731		0.360	0.276	0.301	
	Diagnosing		0.733	0.878	0.798		0.697	0.829	0.744	
	Instructing		0.811	0.652	0.722		0.500	0.381	0.410	
	Demonstrating		0.607	0.838	0.704		0.532	0.688	0.580	
	Support		0.856	0.816	0.832		0.712	0.748	0.725	

CHAPTER 5

DELIVERY OF NATURAL LANGUAGE FEEDBACK IN INTELLIGENT TUTORING SYSTEMS

The second goal of my research is to deliver effective natural language feedback in ITSs. I implemented five versions of the ITS for the letter pattern task to find out how to deliver effective natural language feedback in ITSs.

5.1 Building an Intelligent Tutoring System for the Letter Pattern Task

While collecting and analyzing the human tutoring data, I was also developing an ITS for training students to solve the letter pattern problems.

Since there are quite a few ITS authoring tools available and I am more interested in the functional evaluation of an ITS, I decide to use an authoring tool to build the tutoring system. In 1999, Tom Murray summarized and analyzed the research and development state of the art for ITS authoring systems (Murray, 1999). He published a seven-part categorization of two dozen authoring systems is given. In 2003, the categorization was updated, as shown in Table XVI (Murray, 2003).

Among all these tools, Category 4 is more appropriate for my task since the letter pattern extrapolation is a cognitive science task. Domain knowledge can be encoded as rules and tutoring strategies can be designed and applied but are not hard-encoded in the tools like those

TABLE XVI

ITS AUTHORING TOOLS BY CATEGORY

	Category	Example Systems
1	Curriculum Sequencing and Planning	Swift/DOCENT, IDE, ISD Expert, Expert CML
2	Tutoring Strategies	Econ, GTE, REDEEM
3	Device Simulation and Equipment Training	DIAG, RIDES, SIMQUEST, XAIDA, Instructional Simulator
4	Expert Systems and Cognitive Tutors	Demonstr8, DIAG, D3 Trainer, Training Express, **TDK (CTAT)**
5	Multiple Knowledge Types	CREAM-Tools, DNA, ID-Expert, IRIS, XAIDA, Instructional Simulator, IDVisualizer
6	Special Purpose Systems	IDLE-Tool/IMap/Indie, LAT, BioWorld Case Builder, WEAR
7	Intelligent/Adaptive Hypermedia	CALAT, GETMAS, InterBook, MetaLinks, TANGOW, ECSAIWeb

in Category 2. Finally I chose the Tutoring Development Kit (TDK) [1], which was the most advanced tool in Category 4 when I started building the systems.

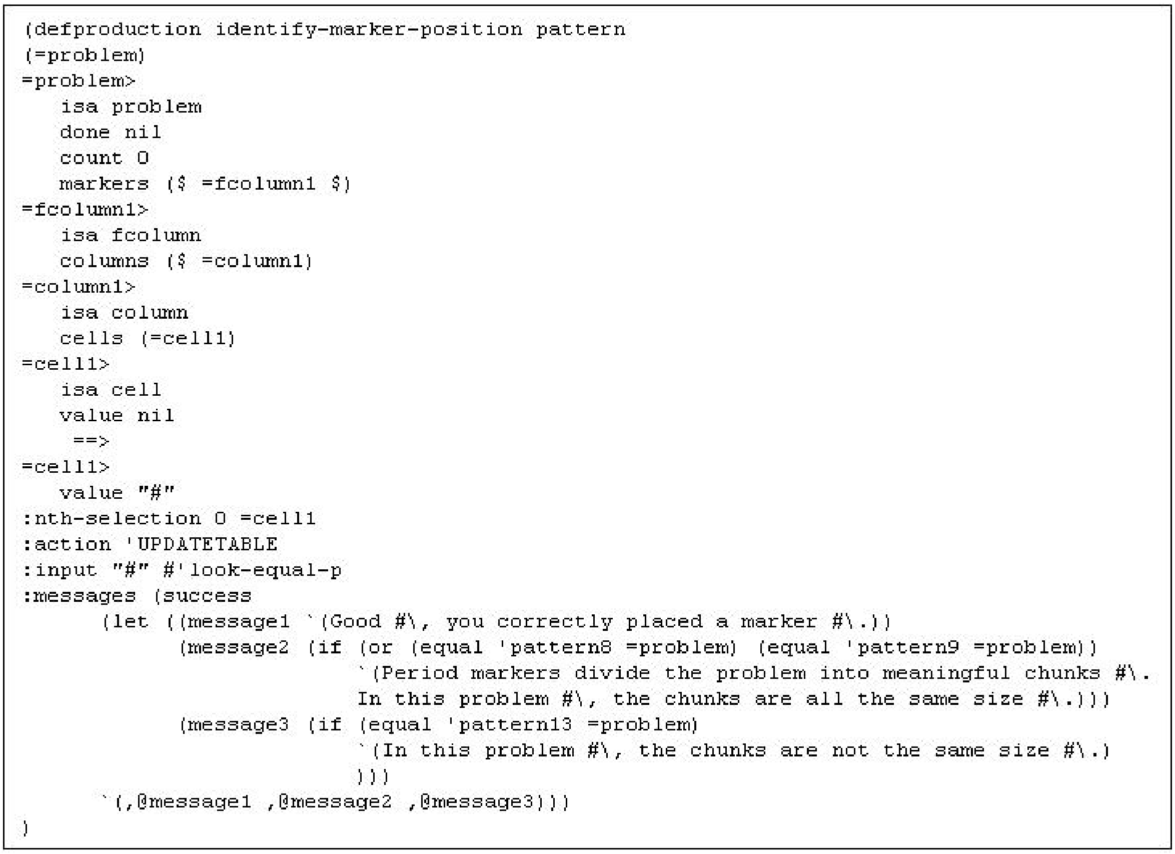
TDK (Koedinger et al., 2003), is based on the ACT-R theory (Anderson et al., 1990), which claims "Cognitive skills are realized by production rules." There are two long-term memory stores, declarative memory and procedural memory. The basic units in declarative memory are chunks and the basic units in procedural memory are production rules. Production rules are the units by which a complex skill is acquired. Each production rule covers a range of situations,

[1] TDK was still under development while I was using it to build the systems. Its production rule system is built in Common LISP and its student interface development package is in JAVA. The current version of TDK is called CTAT (Cognitive Tutor Authoring Tools), which is completely in a Java environment.

not a single situation. A rule usually consists of two parts — condition and action. Declarative knowledge is the working memory of a production system. In TDK, an element of declarative knowledge (chunk) is called "working memory element" (WME). Working memory elements are made up of pairs of "slots" and "values," which match the variables in the condition part (if-part) of the production rules. Performance knowledge is tied to particular goals and contexts by the "if-part." Figure 4 shows an example of the production rules written for the letter pattern task by TDK. The first line defines the name of the rule and the rule set which it belongs to. The lines before "==>" compose the condition part of the rule, which checks variables and their associated WMEs. Variables names are strings whose initial letters are "=." Each variable is associated with a WME, which contains pairs of slots and values. For example, "=cell1" is a variable. The condition part checks whether its first slot "isa" equals "cell" and its second slot "value" equals "nil." The lines after "==>" compose the action part of the rule, which updates the working memory elements, checks the input and generates messages.

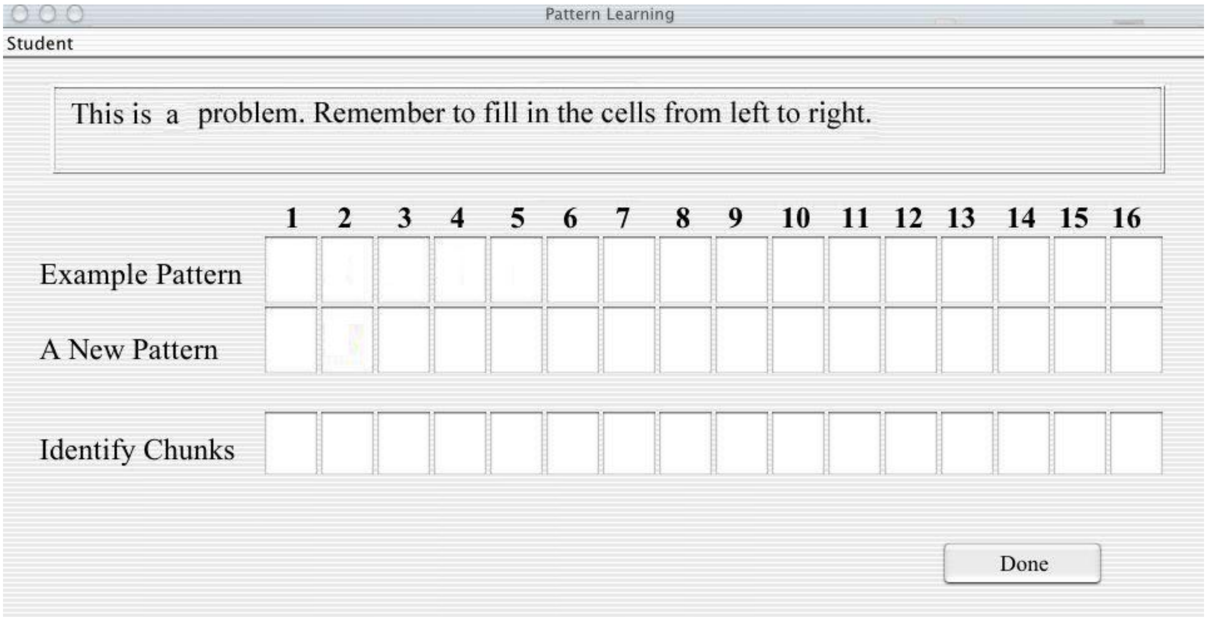
Rules in TDK are written in LISP. There are two kinds of rules in the TDK production system: correct rules model the solution(s) for each problem and buggy rules capture possible errors. The form of rules is the same but buggy rules use "defproduction-bug" to define the rules instead of "defproduction," and a "bug" flag to generate messages instead of "success." Also "hint" can be used to generate messages in both correct and buggy rules. When using "hint," the messages will only show up when the student asks for a hint by clicking a button.

In the ITS for the letter pattern task, 32 production rules, which include 16 correct rules and 16 buggy rules, were written. All the rules capture the extrapolation of all the letter relation-



```
(defproduction identify-marker-position pattern
(=problem)
=problem>
   isa problem
   done nil
   count 0
   markers ($ =fcolumn1 $)
=fcolumn1>
   isa fcolumn
   columns ($ =column1)
=column1>
   isa column
   cells (=cell1)
=cell1>
   isa cell
   value nil
    ==>
=cell1>
   value "#"
:nth-selection 0 =cell1
:action 'UPDATETABLE
:input "#" #'look-equal-p
:messages (success
      (let ((message1 `(Good #\, you correctly placed a marker #\.))
            (message2 (if (or (equal 'pattern8 =problem) (equal 'pattern9 =problem))
                          `(Period markers divide the problem into meaningful chunks #\.
                          In this problem #\, the chunks are all the same size #\.)))
            (message3 (if (equal 'pattern13 =problem)
                          `(In this problem #\, the chunks are not the same size #\.)
                          )))
      `(,@message1 ,@message2 ,@message3)))
)
```

Figure 4. A production rule in the ITS



This is a problem. Remember to fill in the cells from left to right.

Figure 5. Student interface of the letter pattern ITS

ships (relationships within a chunk and between chunks or high level chunks) and the inference of the letters to create a new pattern. All the problems in the curriculum can be covered by these production rules. The messages that the ITS returns are based on templates, however, they vary with different problems to avoid some repetitions and reflect different knowledge.

Figure 5 shows the student interface of the ITS. It was designed by taking into account ease of use and the limitations of the TDK interface development package. The text window at the top gives some basic instruction and information about which problem the student is working on. There are three rows of cells: The first row (*Example Pattern*) presents the pattern that needs to be extrapolated; the second row (*A New Pattern*) is the working row for the student to recreate the pattern by input letters with the starting letter – the first cell of this row is filled automatically with the letter the extrapolation must start from; the third row (*Identify Chunks*) can be used by students to identify chunks in an abstract way, as a way of parsing the pattern. So if the example pattern is the one I discussed in Chapter 2 –"ABMCDM," the student could write down 11#22#, identifying the three chunk markers (#) and two chunks of two letters each. This row is optional, which means the student could use it as extra help to complete the pattern.

I also built an instructional program that introduces the domain and also shows how to use the ITS. The students need to go through this program before they start to interact with the ITS.

5.2 Delivery of Simple Capsulated Feedback Messages

TABLE XVII

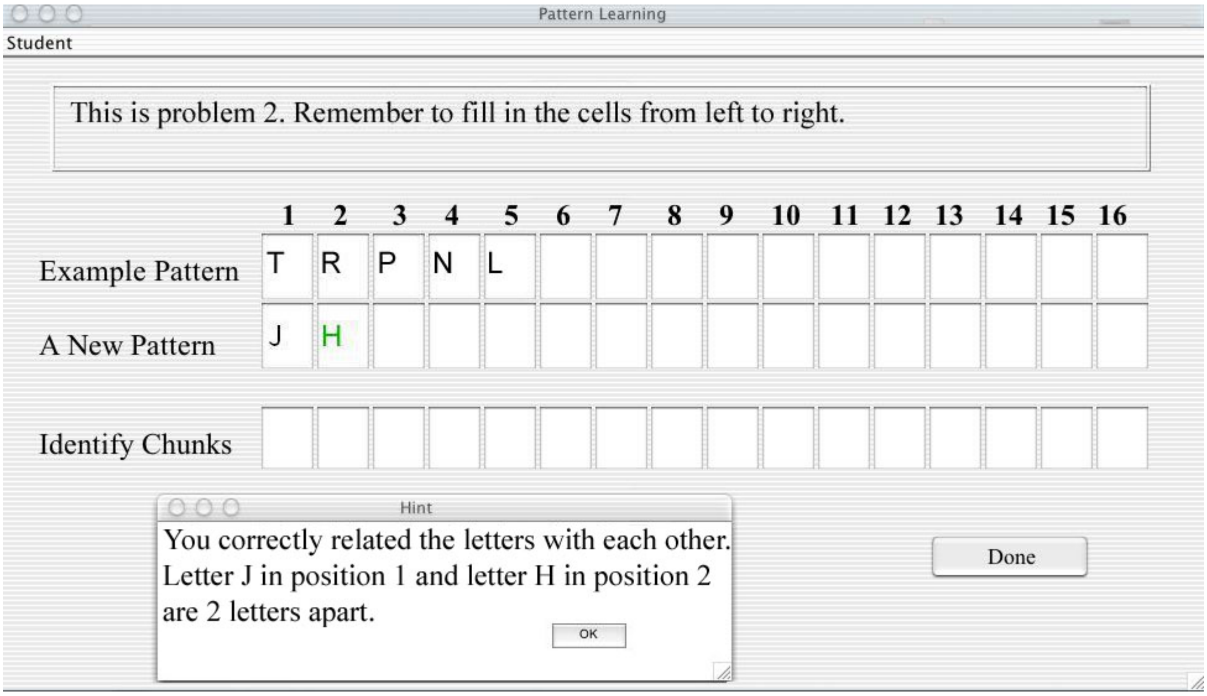
FEEDBACK TYPES OF FOUR VERSIONS OF THE BASELINE ITS

	Version / Feedback	Color	Positive Verbal	Negative Verbal
1	No feedback	No	No	No
2	Color only	Yes	No	No
3	Negative	Yes	No	Yes
4	Positive	Yes	Yes	No

By means of TDK, I developed four versions of the baseline ITS that provide students with different kinds of feedback. Table XVII represents the feedback types of the four versions.

Feedback is given for each input letter. Positive and negative verbal feedback are natural language feedback messages which are given out when the student makes a correct action or a wrong action. The positive feedback messages confirm the correct input and explain the relationships which this input is involved in. The negative feedback messages flag the incorrect input and deliver hints. The feedback messages were inspired by the expert tutor's language, besides I was attempting to avoid repetitions. But they were not computationally modelled on the expert tutor's language yet. Color feedback is a graphic feedback: the input turns green if it is correct, otherwise it turns red. In the no-feedback version, the input will turn blue no matter whether it is correct or not, just to show that the system accepts the input. Figure 6 and Figure 7 shows the user interface of the ITS for version 1 (Positive Feedback) and version 2 (Negative Feedback). The interface is common to all four versions, but the only difference is the form of the feedback.

Figure 6 shows a screen shot of problem 2 in the curriculum in the positive feedback version. (In the figure, the color of letter "H" in the second column is green.)There are five letters in this pattern. From the first letter to the right, each letter goes backward 2 in the alphabet. Therefore the second letter "H" in the new pattern row is a correct input so the letter turns to green and a message "You correctly related the letters with each other. Letter J in position 1 and letter H in position 2 are 2 letters apart." appears. The first sentence in the message stays common for all problems. The second sentence keeps the same form but the letter and its position vary with each problem.



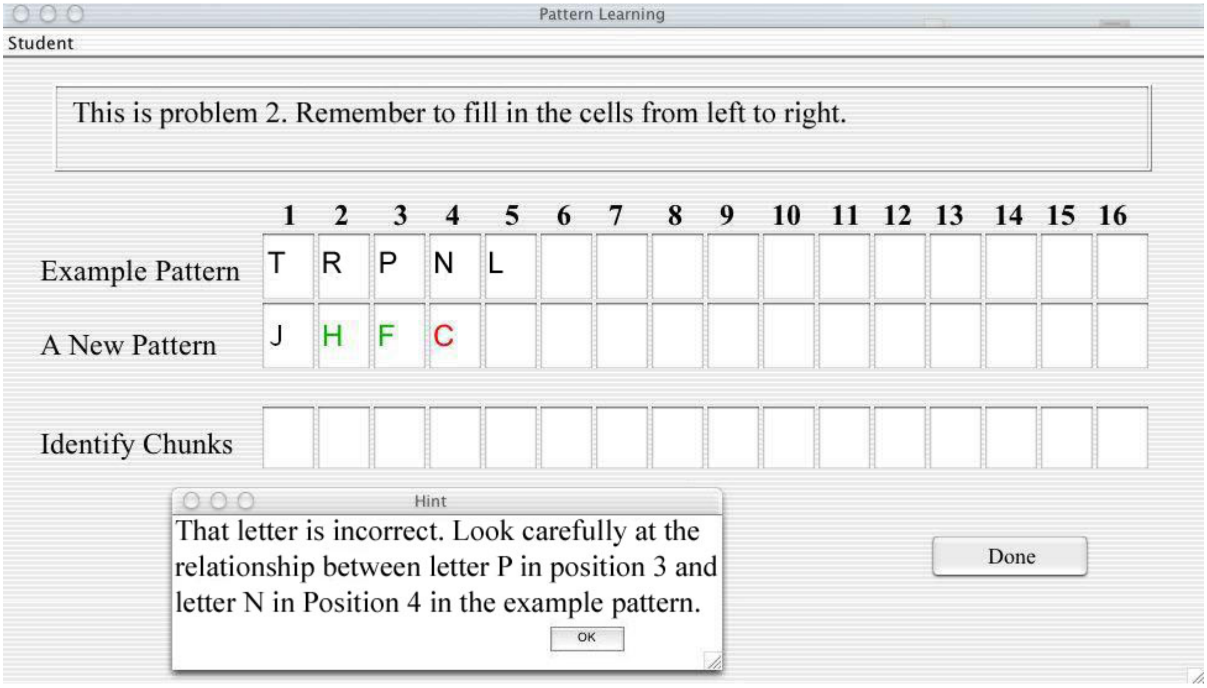
This is problem 2. Remember to fill in the cells from left to right.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Example Pattern	T	R	P	N	L											
A New Pattern	J	H														
Identify Chunks																

You correctly related the letters with each other. Letter J in position 1 and letter H in position 2 are 2 letters apart.

Figure 6. System interface with positive verbal feedback

Figure 7 shows the screen shot of problem 2 in the curriculum in the negative feedback version. (In the figure, the color of letter “H” in the second column and letter “F” in the third column is green; the color of letter “C” in the fourth column is red.) The two green letters in the new pattern row are correct inputs. The fourth letter in the new pattern row should be “D,” so the input letter “C” is incorrect. Then the letter turns red and a negative message “The letter is incorrect. Look carefully at the relationship between letter P in position 3 and letter N in position 4 in the example pattern” appears. As for the positive feedback version,



This is problem 2. Remember to fill in the cells from left to right.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Example Pattern	T	R	P	N	L											
A New Pattern	J	H	F	C												
Identify Chunks																

That letter is incorrect. Look carefully at the relationship between letter P in position 3 and letter N in Position 4 in the example pattern.

Figure 7. System interface with negative verbal feedback

the first sentence in the message stays common for all problems. The second sentence keeps the same form but the letter and its position vary with each problem.

5.3 Generating Effective Natural Language Tutorial Feedback

The second goal of my research is to develop a real natural language feedback generator for the ITS. Natural language generation usually includes content planning (decide what to say), discourse planning (decide how to say) and surface realization. Since I am more interested in the content and discourse level of natural language feedback rather than the surface level, for surface realization I used a template-based method. In Section 5.1, I have introduced the production rules in the baseline ITSs. The production rules based on TDK provide the capability to generate template-based messages that take into account what kind of processing is needed to use a particular kind of feedback. However, the messages have to follow the special syntax in TDK and can not include other variables than the working memory elements defined by means of TDK, or any other functions written outside of TDK. To generate more sophisticated natural language feedback modelled from expert tutoring dialogues, I built a feedback generator to provide more flexibility for natural language feedback generation in the ITS.

To enable natural language interaction in ITSs, tutorial dialogue management in current ITSs has played a very important role. In the computational linguistics and spoken dialogue systems communities, researchers have developed some dialogue theories and dialogue system technology, which are not designed specifically for tutoring. These systems aim for dialogue strategies that are independent of dialogue context management and communication management concerns. Information State (IS) theory is one of the dialogue theories widely used in dialogue systems and has been introduced into tutorial dialogue systems recently (Zinn et al., 2002). It can be combined with the traditional dialogue models, such as finite-state dialogue models and classical plan based models. The key idea of this theory is identifying the relevant aspects of information in dialogue, how they are updated, and how updating processes are controlled (Larsson and Traum, 2000). An information state theory of dialogue modelling consists of:

- a description of the *informational components* of the theory of dialogue modelling, including aspects of common context as well as internal motivating factors (e.g., participants, common ground, linguistic and intentional structure, obligations and commitments, beliefs, intentions, user models, etc.)

- *formal representations* of the above components (e.g., lists, sets, typed feature structures, records, Discourse Representation Structures (DRSs), propositions or modal operators within a modal logic, etc.)
- a set of *dialogue moves* that will trigger the update of the information state. These will generally also be correlated with externally performed actions, such as particular natural language utterances. A complete theory of dialogue behavior will also require rules for recognizing and realizing the performance of these moves, e.g., with traditional speech and natural language understanding and generation systems.
- a set of update rules, that govern the updating of the information state, given various conditions of the current information state and performed dialogue moves, including (in the case of participating in a dialogue rather than just monitoring one) a set of selection rules, that license choosing a particular dialogue move to perform given conditions of the current information state
- an update strategy for deciding which rule(s) to select at a given point, from the set of applicable ones. This strategy can range from something as simple as "pick the first rule that applies" to more sophisticated arbitration mechanisms, based on game theory, utility theory, or statistical methods.

Based on the Information State theory, I developed a natural language feedback generator and integrated it with tutorial rules learned from the expert tutoring. Figure 8 is the overall

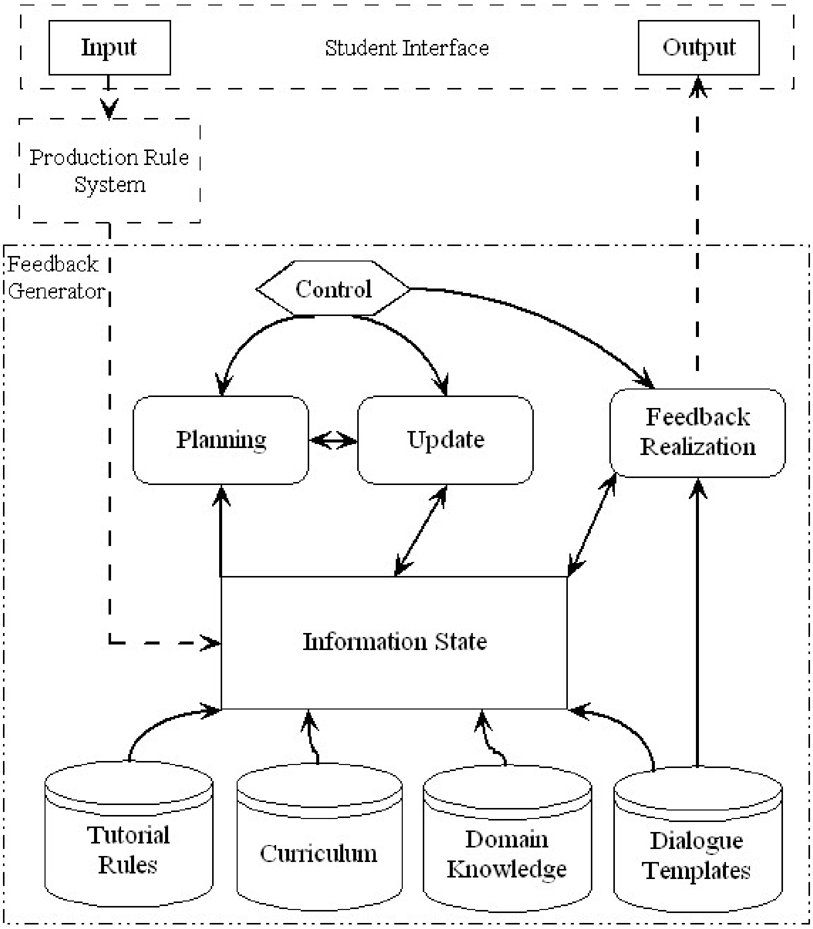


Figure 8. The natural language feedback generator

framework of the natural language feedback generator. In the figure, the solid arrows represent the interactions within the generator; the dashed arrows represent the interactions with the external components. There are three major modules, update, plan and feedback realization. Each of the modules can access the information state, which captures the overall dialogue context and interfaces with external knowledge sources (e.g., curriculum, tutorial rules) and the production rule system. The IS represents the information necessary to distinguish it from other dialogues, representing the cumulative additions from previous actions in the dialogue, and motivating future action. In particular, the IS in the feedback generator for the letter pattern ITS contains:

1. *Speaker*: either the student or the system, who is the producer of the current move;
2. *Domain concepts*: the letter relationships in the letter pattern that the speaker is talking about;
3. *Problem scopes*: the relationship scope within the letter pattern that is covered by the speaker;
4. *Speaker's move*: the system move or the student's action response. In Section 5.1, we have introduced the student interface for the letter pattern task. The student does not input any natural language but just letters or numbers which are used to complete the problem. So the student's move here is always the action response;
5. *Speaker's attitude*: the attitude of the tutorial move from the system or the confidence of the student – positive, negative or neutral;

6. *Correctness*: correctness of the student's response;

7. *Student's input*: the letters or numbers that the student input;

8. *Student's selection*: the cell which the student input a letter or a number into;

9. *Hesitation time*: the time from when the system is ready to accept an input to when the student inputs a letter or a number. The time that the system is ready is often the time when a new problem starts or the student closes the pop-up message. The time recorded by the system is continuous but the hesitation time used in the tutorial rules are categorical. So in order to match the hesitation time, I categorized the continuous hesitation time into four categories – "no," "short," "medium" and "long." I did clustering on the continuous hesitation time that I collected from the experiments of the four baseline ITS. Based on the clustering results, the time ranges of "no," "short," "medium" and "long" are 0∼11 seconds, 12∼31 seconds, 32∼65 seconds and 66 seconds or longer.

10. *Student's knowledge state*: the student's knowledge state on each type of letter relationship, which is computed as in Equation 4.2 in Section 4.3.

Most of these components match the features used in learning tutorial rules in Chapter 4, which allows an easy integration of the tutorial rules and the feedback generator.

In the following sections, I describe the three modules in this natural language feedback generator, how they cooperate with each other and with the external resources.

5.3.1 The Plan Module

The plan module generates plans for planning content and discourse structure of the natural language feedback. In the feedback generator, 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, plan selection and plan monitoring. Figure 9 shows the overview of the plan module. In the figure, the solid arrows represent the interactions within the framework; the dashed arrows represent the interactions with the external components.

The **plan generation** tier automatically synthesizes plans from the tutorial rules based on the information state and other external resources. A plan usually contains the following elements:

1. *Preconditions*: conditions that must always be true just prior to the execution of this plan. The tutorial rule whose left-hand side matches the preconditions is chosen to generate this plan;

2. *Goals*: the interface elements which the student is focused on. In the letter pattern ITS, they are the student's selections – the cells where the student inputs letters or numbers;

3. *Effects*: expected results after the execution of this plan. In the letter pattern ITS, the expected results are often that the student's inputs in the student's selections are correct;

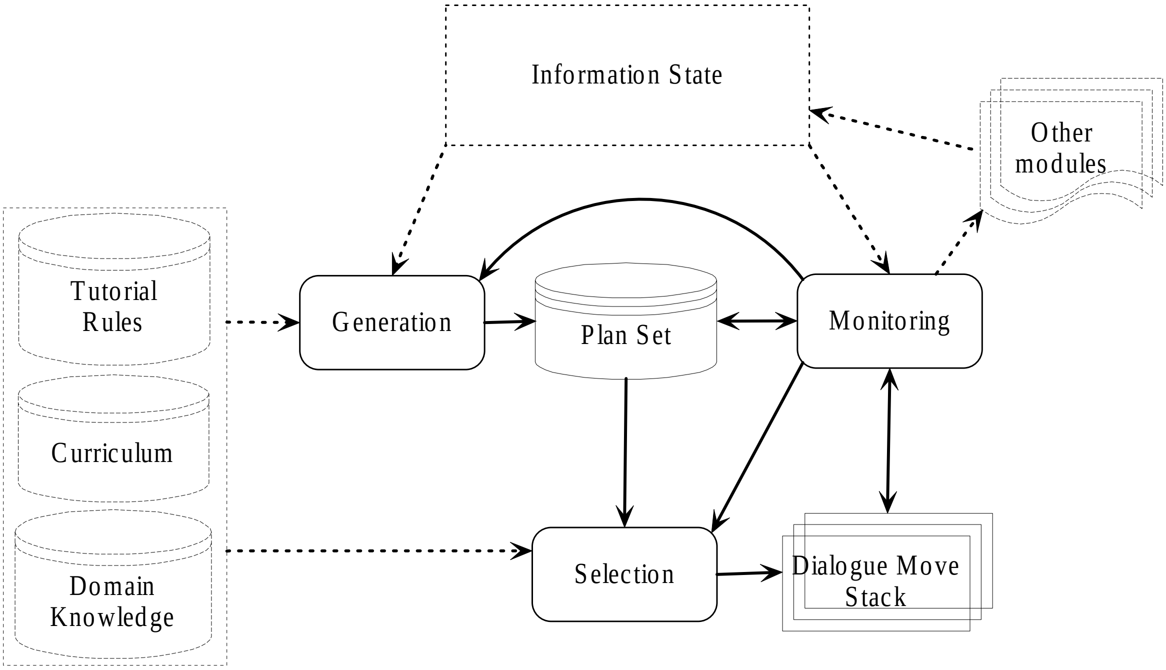


Figure 9. The framework of 3-tier planning

4. *Contents*: the information that is related to the content of the message to be generated, such as domain concepts and the problem scopes. In the letter pattern ITS, they are the letter relationships and the relationship scopes;
5. *Actions*: dialogue moves, templates when planning how to realize a feedback message for a particular tutoring move, or plans when supporting hierarchical planning. In the letter pattern ITS, the actions are a set of tutoring moves, which are the same as the tutor moves in the human tutoring dialogues, or a template, which is used to realize a feedback message (a template is selected in the plan selection tier, which will be described next);
6. *Modifiers*: additional information to help generating appropriate messages. In the letter pattern ITS, the tutorial attitude of each message is the modifier;
7. *Confidence*: primary strength measurement of the plan. It is the same as the confidence of the tutorial rule that is used to generate this plan;
8. *Support*: secondary strength measurement of the plan. It is the same as the support of the tutorial rule that is used to generate this plan.

The generated plans are in the plan set of the plan module. In the plan set, each plan corresponds to a tutorial rule in the tutorial set for choosing tutoring moves, which matches the current information state. The corresponding tutor move is put into "actions." The corresponding confidence and the corresponding support are put into the "confidence" and "support" of the plan. The condition part of the corresponding tutorial rule is put into "preconditions." The attitude of the tutoring move is chosen by going through the tutorial rules for choosing the tutor's attitude and is put into "modifiers." Then "Contents" of the plan are the letter relationships and the relationship scopes that are decided by the tutorial rules for choosing letter relationships and choosing relationship scopes. In the letter pattern ITS, the ITS can give feedback messages successively if the student does not input anything in a certain period of time (the hesitation time is "long"). So when generating a single plan, several tutoring moves are put into "actions." The first move in the plan is decided based on the current information state. And then for the rest of the moves, each tutoring move is decided based on the updated information state, in which the *hesitation time* changes to "long" and the *speaker's move* changes to the previous move. These moves except the first move would be executed if the student didn't input anything for a "long" time after seeing the feedback message generated from the previous tutoring move. Each tutoring move is associated with an attitude, which is decided based on the same updated information state using the tutorial rules for choosing tutor's attitude and is put into "modifiers." In the letter pattern ITS, there are four tutoring moves for each plan, because the average length of the multi-utterance turn of the expert tutor is 4 as I have discussed in Section 3.4.5.

The **plan selection** tier has the responsibility of selecting a plan for the ITS, selecting a template for each tutoring move that is used to accomplish the current plan and put tutoring moves into the dialogue move stack. Sometimes the generation tier generates several plans which all fit the current information state. There are two strength measurements (confidence and support) associated with each plan. If the last tier generated more than one plan, the plan selection first chooses the plan with highest confidence. If their confidences are the same, the plan with the highest support is chosen. If their supports are also the same, this tier chooses the first plan generated by the last tier.

TABLE XVIII

NUMBER OF POSSIBLE TEMPLATES AND ACCURACY OF CHOOSING A TEMPLATE FOR EACH TYPE OF TUTORING MOVE

Tutoring Move	Number of Templates	Accuracy
Confirming	2	77.778%
Evaluating	13	76.364%
Summarizing	19	75.676%
Prompting	12	71.552%
Diagnosing	5	90.476%
Instructing	21	74.101%
Demonstrating	11	58.571%
Support	4	88.235%

For each tutoring move, maybe several templates can be used to accomplish the current plan. For the letter pattern task, I wrote a total of 50 templates according to the expert tutor dialogues. (More details can be found in Appendix C.) For each type of tutoring move, there are several possible templates which can be used to realize the feedback message. In the expert tutor dialogues, I aligned each tutoring utterance with the corresponding template for realizing this utterance. Then as for tutor moves, I used the same features that are used for modeling expert tutoring feedback in Chapter 4 and learned a set of tutorial rules for choosing a template by using CBA. Table XVIII lists the number of templates for each type of tutoring move and the accuracies of the tutorial rules for choosing a template. In this table, there is a type of tutoring move –"confirming" instead of the human tutor move "answering" which is used in Chapter 3 and Chapter 4. This is because in the letter pattern ITS, the student does not input any natural language to ask a real question. The ITS only confirms the student's input when the student shows hesitation like what happened with the human tutor. For example, when the student asks the human tutor "Is this a T?" the human tutor answers "Yeah, this is a T." When the student hesitates for a while, the ITS uses a "confirming" move to give a feedback message like "Yeah, this is a T." In this plan selection tier, template selection is performed by using the tutorial rules for choosing a template. One template is chosen for one tutoring move. Each tutoring move in the selected plan is put into the dialogue stack after copying the elements "preconditions," "goals," "effects," "contents" and putting the selected template into the element "templates."

The **plan monitoring** tier checks whether the effects have been obtained after each tutoring move. If not, but the student's selection is unchanged, the next move from the dialogue move stack will be executed until the dialogue move stack becomes empty. Then another plan is selected from the plan set and the tutoring moves with the plan are pushed into the dialogue move stack. If every plan in the plan set has been selected or the student's selection has changed, the plan monitoring tier must re-generate the plans. For example, suppose that the current goal was to help the student figure out which letter was a chunk marker and that, just after the ITS gave a hint to the student on how the marker usually appears, the student jumped to fill in another letter within a chunk instead of writing down the chunk marker. At this time,

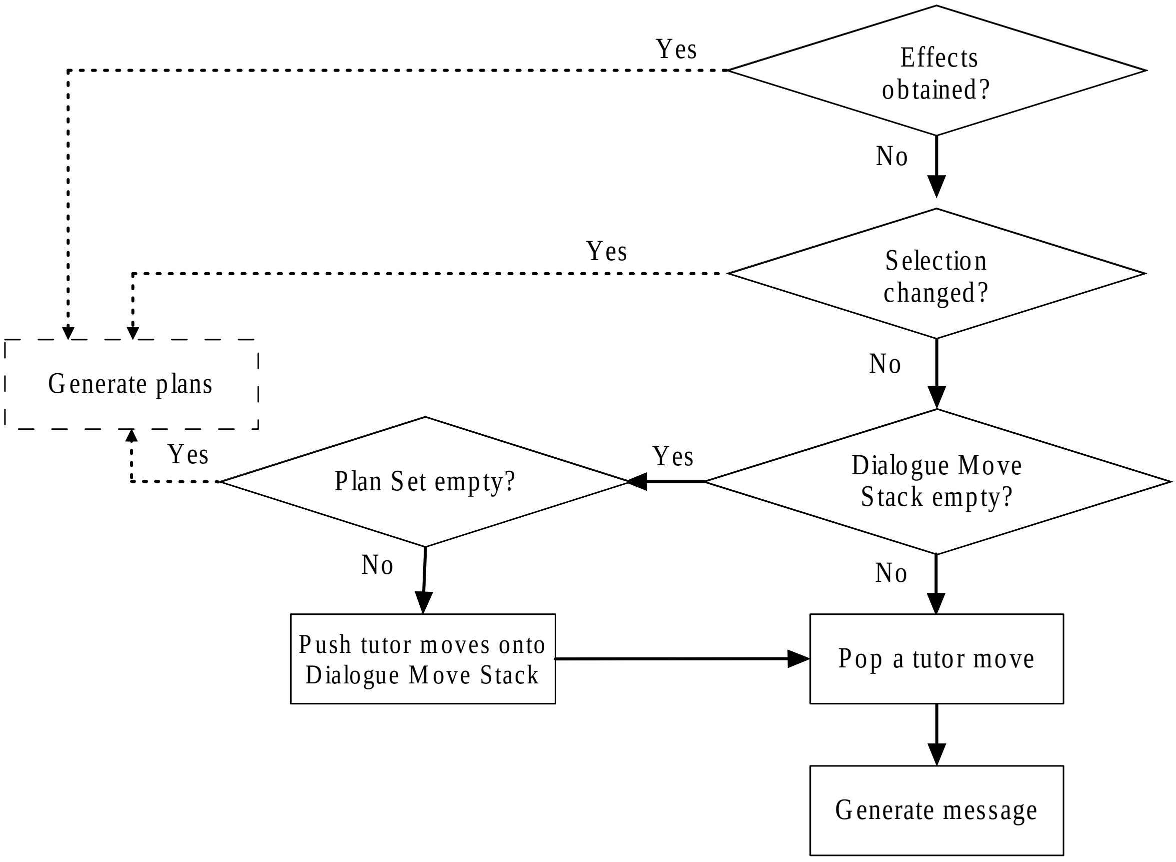


Figure 10. Plan monitoring

the system had to give out a message to pull the student back to the current goal. So the plan monitoring will call the plan generation to come up with a new plan or repair the old plan. In a plan, there are usually a sequence of tutoring moves and they will be put in a dialogue move stack. The system will execute the move from the top of the stack and push a new tutoring move onto the stack. So one possible way to deal with the example situation is to push the move which has just been executed back onto the dialogue move stack. Figure 10 shows the procedure of plan monitoring. In the figure, the solid arrows represent the flows within the monitoring tier; the dashed arrows represent the flows with the external components.

In this 3-tier framework, the strength measurements play a very important role: plans are generated with them; plans are chosen and realized according to them. The strength measurements come from the tutorial rules that are learned from the expert tutoring dialogues. They represent the possibility of a tutorial rule that has been frequently used during expert tutoring. So the tutorial rules are actually probabilistic rules and the plans are probabilistic plans. The 3-tier planning framework implements probabilistic planning (Blum and Langford, 1999) for feedback generation. Probabilistic planning is an extension of nondeterministic planning with information on the probabilities of nondeterministic events. Probabilities are important in quantifying the costs and success probabilities of plans when the actions are nondeterministic. It is important to maximize the probability of reaching the goals, and hence it is vitally important to use information on the probabilities of different effects of operators. Probabilistic planning approaches are directly applicable (Blum and Langford, 1999) and work in this area has shown that compact representations, like rules, are essential for scaling probabilistic planning to large worlds (Boutilier et al., 2000; Pasula et al.,).

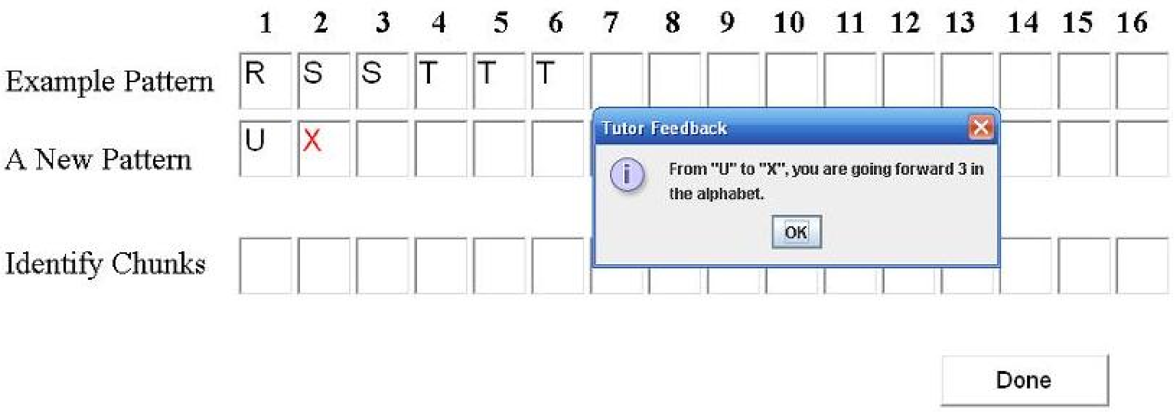
5.3.2 The Update Module

The update module maintains the context. As a new student action is made, it updates the IS according to the information collected by the production rule system. Then the plan module generates or revises the system plan and selects the next tutoring move based on the newly updated IS. The update module updates the tutoring move history by taking the newly selected tutoring move. At last the feedback realization module transforms this unaccomplished move into natural language feedback.

5.3.3 The Feedback Realization Module

The feedback realization model is in charge of the surface realization of the feedback message, which is the final step to deliver a message to the student interface. What is passed to this module is a tutoring move with all the elements (as introduced in Section 5.3.1) including a template to realize the feedback message.

All the templates for the letter pattern ITS are listed in Appendix C. In a template, there are variables (between a pair of "<" and ">") and functions (between a pair of "(" and ")") which are replaced with corresponding texts based on the elements with the tutoring move. The variables include column numbers, user inputs, letters in the "Example Pattern," correct letters in the "New Pattern" row, correct letters, numbers or markers in the "Identify Chunks" row, letter relationships and length of the pattern. There are five functions:



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Example Pattern	R	S	S	T	T	T										
A New Pattern	U	X														
Identify Chunks																

From "U" to "X", you are going forward 3 in the alphabet.

Figure 11. A feedback message in the "model" version ITS

n<var> : this function decides whether to precede the variable with "a" or "an" based on the first word that is used to replace the variable;

chunk_lengths : this function lists the number of letters in each chunk;

currentChunk_examples : this function lists all the example letters within the chunk in which the current input belongs;

list_letters<var1><var2> : this function lists all the letters in the alphabet from the first variable to the second variable;

explain<var1><var2> : this function gives an explanation of the letter relationship, which is described by the two variables.

The feedback message shown in Figure 11 was generated using the 11th template:

> From “<reference_pattern>” to “<input>,” you are going <input_relation> <input_number> in the alphabet.

The tutoring move of this message is “summarizing.” Based on the plan elements with this tutoring move, the variable “<reference_pattern>” is replaced with “U,” which is the letter “X” makes reference to; the variable “<input>” is replaced with the input “X.” The other two variables are computed by using “U” and “X.” There are two letters “V” and “W” between “U” and “X.” So the rest two variables are replaced with “forward 3.”

When subject is working on this pattern, if the feedback realization module was called to generate a feedback message by using the 21th template:

> What I think is helpful is you noticed the chunks with (chunk_lengths) letters.

The feedback realization module would call the function that computes the number of the chunk lengths in a pattern and then use the result numbers to replace the function “(chunk_lengths)” in the template so that the feedback message would be:

> What I think is helpful is you noticed the chunks with 1 2 3 letters.

CHAPTER 6

EVALUATION

The goal of developing five different versions of the ITS for the letter pattern task is to find out answers to the following questions:

1. Does the interaction with an ITS improve learning?
2. Does the student learn more when receiving feedback than with unsupervised practice?
3. What type of feedback does an ITS need to provide to engender significantly more learning than unsupervised practice?

In this chapter, I describe the experiments, report the results, analyze and discuss the findings from the experimental evaluation of the five versions of the ITS.

6.1 Experiments

Based on how feedback is generated, I named the five versions of the ITS for the letter pattern task as follows:

1. **No feedback**: The ITS only provides an interface so that the student can practice solving the 13 problems in the curriculum, but does not provide any kind of feedback.
2. **Color only**: The ITS provides graphic feedback by turning the input green if it is correct or red if it is wrong.
3. **Negative**: In addition to the color feedback, the ITS provides feedback messages when the input is wrong.

4. **Positive**: In addition to the color feedback, the ITS provides feedback messages when the input is correct.

5. **Model**: In addition to the color feedback, the ITS provides feedback messages generated by the feedback generator that is based on the model of expert tutoring.

To evaluate the five versions of the ITS, I ran a between-subjects study in which each group of subjects interacted with one version of the system. (The number of subjects for each group was: no feedback [N=33], color only [N=37], negative [N=36], positive [N=33], and model [N=38]). The subjects were trained to solve the same 13 problems in the curriculum that were used in the human tutoring condition. They also did the same post-test (2 problems, each pattern 15 letters long). For each post-test problem, each subject had 10 trials but each trial started with a new letter. As in the study of human tutors, I also had a control condition [N=32] in which the subjects did the post-test problems with no training at all but only read a short description of the domain. With the "model" version ITS, each subject also completed a questionnaire about their opinions about the ITS. All the subjects in this study came from the same subject pool as the one used for the study of human tutors in Chapter 3. The number of subjects are different across different groups because some subjects did not complete the post-test problems or because of some technical problems that happened during training.

6.2 Estimating Subject's Pre-Tutoring Ability

In our study, subjects did not take any pre-test to assess their pre-tutoring ability in the letter pattern task. Although for this task no special domain knowledge is required and subjects came from the same subject pool, which ensured that subjects have the similar pre-tutoring ability to a certain extent, it still leaves a concern the subjects in different groups might be at different levels of pre-tutoring ability. If it were true that some groups of subjects are at different levels of pre-tutoring ability compared to the others, any conclusions about different learning gain by interacting with different versions of the ITS would become invalid. To validate the pre-tutoring ability of the subjects, I used the information from each tutoring session. At the beginning of tutoring, how well a subject performs can show us his/her prior ability. Another study of tutoring in the letter pattern task (Corrigan-Halpern, 2006) also discussed the same approach. Among the 13 problems in the curriculum, the first 3 problems are categorized into the first level which is very basic and directly reflects the knowledge of the English alphabet. If a subject spent less time in the first 3 problems, this is likely to mean that s/he has higher pre-tutoring ability or some previous experience with solving this kind of puzzle. So I used the time spent in the first 3 problems to estimate pre-tutoring ability. On all the five groups of subjects with the ITSs, I ran a multiple regression [1] using "time spent in the first 3 problems" as the predictor variable to see whether it is correlated to the post-test score.

The result of the multiple regression shows that time spent in the first 3 problems is highly correlated to the post-test score ($p < 0.03$): the less the time spent in the first 3 problems, the higher the post-test score. The time spent on the first 3 problems accounts for 2.8% of the variance of the post-test scores of the five groups with the ITS. As many studies in tutoring

[1] The general purpose of multiple regression is to learn more about the relationship between several independent or predictor variables and a dependent or criterion variable. Multiple regression can establish that a set of independent variables explains a proportion of the variance in a dependent variable at a significant level (through a significance test of R^2), and can establish the relative predictive importance of the independent variables (by comparing beta weights).

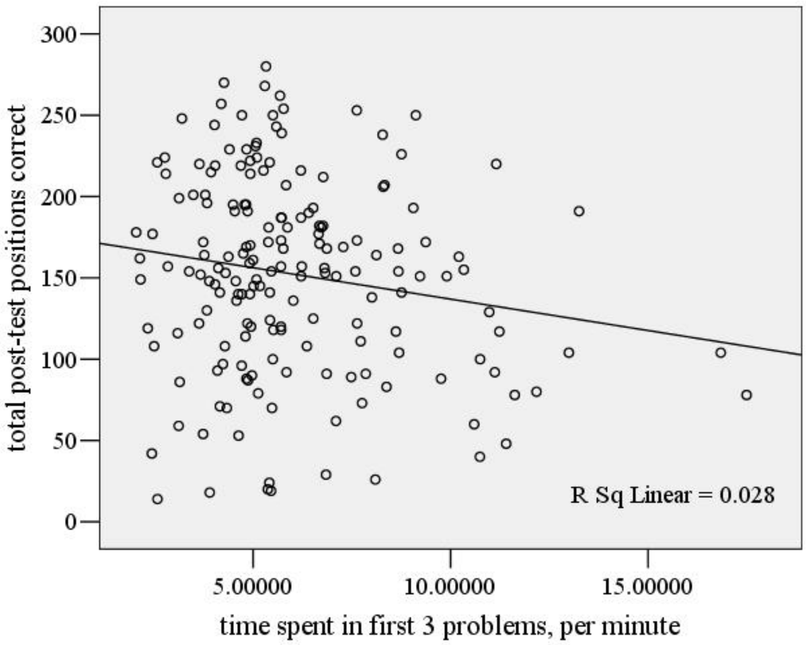


Figure 12. Linear regression plot of the correlation between the time spent in the first 3 problems and the total post-test scores

have reported, pre-tutoring ability is highly correlated to post-tutoring ability. These results confirmed my belief that “time spent in the first 3 problems” is the right metric to estimate subject’s pre-tutoring ability in the letter pattern task. Table XIX reports the time in minutes that the five groups of subjects with the ITS spent in the first 3 problems. The mean and the range of the time of the five groups are very similar. And no significant difference was found when ANOVA and a Turkey test was performed. Therefore, I conclude that the five groups of subjects with different versions of the ITS have the same level of pre-tutoring ability in the letter pattern task.

TABLE XIX

MEAN, MINIMUM AND MAXIMUM OF THE TIME SPENT IN THE FIRST 3 PROBLEMS OF THE SUBJECTS WITH THE FIVE VERSIONS OF THE ITS

Version		Mean	Minimum	Maximum
1	No feedback	5.29	2.45	16.84
2	Color only	6.17	2.13	17.49
3	Negative	6.16	2.04	13.25
4	Positive	6.20	2.49	11.16
5	Model	6.34	3.20	11.24

6.3 Results

Table XX reports the average post-test scores of six groups of subjects with the five versions of the ITS and the control condition. The post-test performance is measured by the average

TABLE XX

AVERAGE POST-TEST SCORES OF THE ITS

Versions		Number of subjects	Post-test Score		
			Problem 1	Problem 2	Total
0	Control	32	36.50	32.84	69.34
1	No feedback	33	58.21	75.27	133.48
2	Color only	37	68.32	66.30	134.62
3	Negative	36	70.33	66.06	141.83
4	Positive	33	75.06	79.00	154.06
5	Model	38	91.95	101.76	193.71

number of letters correct out of a total of 150 letters (in 10 trials) for each problem per subject. Comparing the post-test performance, the main findings are (all the statistical results are based on ANOVAs, followed by Tukey's tests):

- A main effect of the ITS ($p \leq 0.05$): The subjects who interacted with the ITS did significantly better in both the post-test problems than the subjects in the control condition.
- No main effect of simple feedback ($p > 0.05$): The subjects who are trained by the three versions of the ITS with simple feedback (color only, negative, positive) did not have significantly higher post-test scores than the subjects with the "no feedback" version in either of the post-test problems.
- No main effect of simple capsulated feedback message ($p > 0.05$): the subjects who are trained by the two versions of the ITS with simple capsulated feedback messages (negative, positive) did not have significantly higher post-test scores than the subjects with the two

versions without simple capsulated feedback messages (no feedback, color only) in either of the post-test problems.

- A main effect of modeled natural language feedback message ($p < 0.05$): the subjects who interacted with the "model" version of the ITS did significantly better in the total post-test scores of the two problems than the subjects with any other version of the ITS.

We can then conclude that the feedback modeled on the expert tutoring is significantly more effective than other types of feedback in the ITS. Although I found some significant results in the post-test performance, there are still three issues that need to be addressed:

1. How well did the subjects with the ITS perform in the post-test comparing to the subjects with the human tutors?
2. What factors brought the significant differences in the post-test performance?
3. What did the subjects think of the modeled feedback messages?

6.3.1 Comparison between the Intelligent Tutoring System and the Human Tutors

Since in the study of human tutors there are only 6 trials for each post-test problem, the first 6 trials per problem from the ITSs are used to run a comparison with the human tutors. Figure 13 shows the post-test performance of all five groups of subjects with the ITS, all three groups of subjects with the human tutors and the group of the control condition. The post-test performance is the average number of letters correct out of total 90 letters (in 6 trials) for each problem per subject. The error bars in the figure represent the standard deviations.

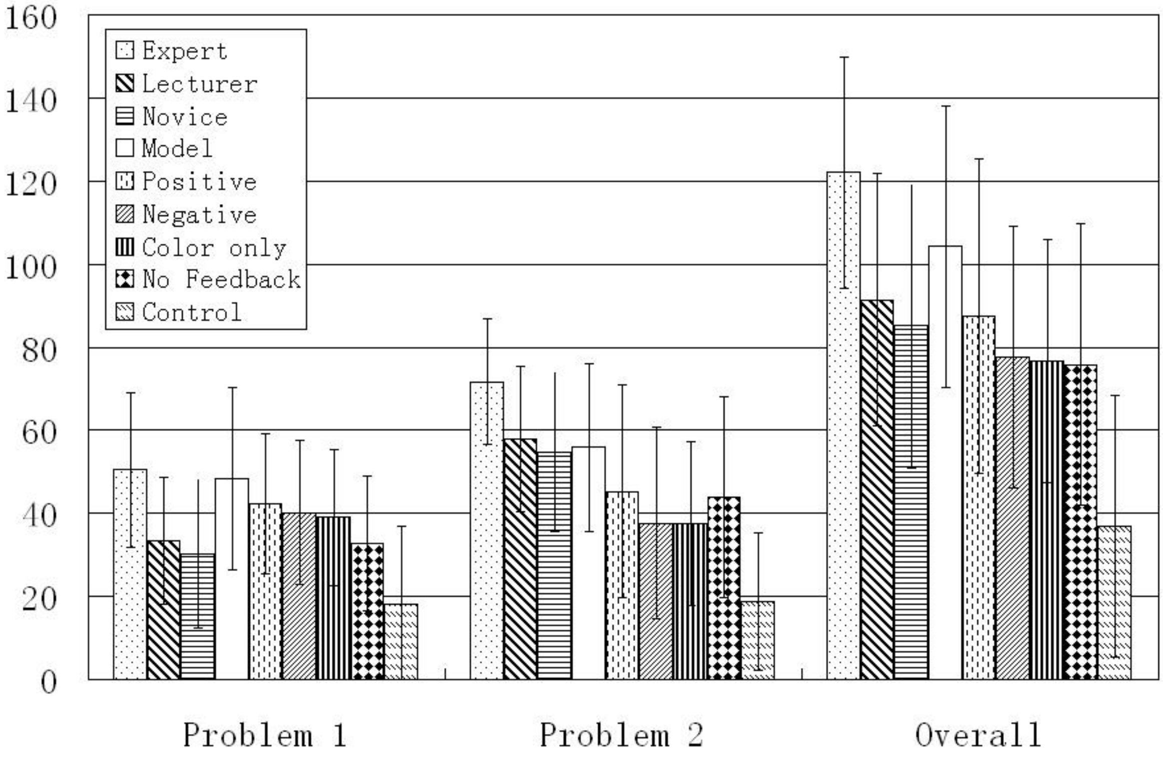


Figure 13. Post-test performance of tutors and five versions of the ITS

In this comparison, I found one significant result: the group of subjects with the expert tutor did significantly better in the post-test than the groups with the four versions of the baseline ITS (no feedback, color only, negative, positive), but did not do significantly better than the group with the "model" version ITS. There are some additional observations, although not significant:

- The groups with the four versions of the ITS which provides feedback (color only, negative, positive, model) had higher post-test scores in problem 1 than the groups with the novice tutor and the lecturer;
- The group with the "model" version ITS had higher total post-test scores than those with the novice tutor and the lecturer;
- The group with the "positive" version ITS had slightly higher total post-test scores than that with the novice tutor.

6.3.2 Searching for the Factors that Affect the Post-Test Performance

Table XXI reports the minimum and maximum of the total post-test scores of the subjects with the five versions of the ITS. There is a wide variance in the total post-test scores of all the groups. Then the question becomes what factors cause the wide variance. The answer to this question will help us to understand how the "model" version ITS help the subjects gain higher post-test scores. To find out the answer, I ran multiple regressions on all the five groups of subjects. Other than the time spent in the first 3 problems that I discussed in the last section, there are 6 potential predictor variables:

TABLE XXI

MEAN, MINIMUM AND MAXIMUM OF THE TOTAL POST-TEST SCORES OF THE SUBJECTS WITH THE FIVE VERSIONS OF THE ITS

Version		Mean	Minimum	Maximum
1	No feedback	133.48	14	243
2	Color only	134.62	26	239
3	Negative	141.83	29	250
4	Positive	154.06	19	268
5	Model	193.71	79	280

1. **Training time**: the time that each subject spent in interacting with the ITS;
2. **Total bugs**: the total number of incorrect inputs that each subject made while interacting with the ITS;
3. **Bugs in the "New Pattern" row**: the number of incorrect inputs that each subject made in the "A New Pattern" row while interacting with the ITS;
4. **"Identify Chunks" row**: whether the subject used the "Identify Chunks" row or not while interacting with the ITS;
5. **Problems using the "Identify Chunks" row**: the number of problems in which the subject made inputs in the "Identify Chunks" row while interacting with the ITS;
6. **Problems having bugs**: the number of problems in which the subject made incorrect inputs while interacting with the ITS.

Table XXII reports the results of the multiple regression. The results show that:

TABLE XXII

REGRESSION RESULTS FOR THE FIVE VERSIONS OF THE ITS

Predictor Variable	R^2	β	p
Training time	0.031	-0.177	0.018
Total bugs	0.114	-0.337	0
Bugs in the “New Pattern” row	0.158	-0.398	0
“Identify Chunks” row	0.012	-0.107	>0.05
Problems using the “Identify Chunks” row	0.002	0.040	>0.05
Problems having bugs	0.106	-0.325	0

1. **Training time** accounts for 3.1% of the variance: the shorter the training time, the higher the post-test score;

2. **Total bugs** accounts for 11.4% of the variance: the fewer the total bugs, the higher the post-test score;

3. **Bugs in the “New Pattern” row** accounts for 15.8% of the variance: the fewer the bugs in the “New Pattern” row, the higher the post-test score;

4. **“Identify Chunks” row**: whether the subject used the “Identify Chunks” row does not affect the post-test score;

5. **Problems using the “Identify Chunks” row**: the number of problems in which the subject used “Identify Chunks” row does not affect the post-test score;

6. **Problems having bugs** accounts for 10.6% of the variance: the fewer problems having bugs, the higher the post-test score.

TABLE XXIII

MEAN, MINIMUM AND MAXIMUM OF THE NUMBER OF EACH TYPE OF TUTORING MOVE AND TUTORING ATTITUDE IN THE MODEL VERSION ITS, PER SUBJECT

Tutoring Move/Attitude	Mean	Minimum	Maximum
Confirming	6.32	0	30
Evaluating	43.29	24	129
Summarizing	49.32	25	90
Prompting	3.50	0	33
Diagnosing	19.08	0	54
Instructing	9.32	2	36
Demonstrating	2.89	0	23
Support	36.95	11	66
Positive	101.00	72	204
Negative	10.95	1	31
Neutral	58.71	11	137

Since the subjects with the “model” version of the ITS did significantly better in the post-test than the subjects with the other versions, the next step is to find out among the subjects with the “model” version what else happened during training that causes the variance of the post-test scores. I counted the frequences of each type of tutoring move and each type of tutoring attitude, which are listed in Table XXIII. There is a wide variance in the type of tutoring move and the type of tutoring attitude of the feedback messages that each subject received. So the potential predictor variables for the post-test score of the subjects with the “model” version ITS are the number of each type of tutoring move and the number of each type of tutoring attitude. When running multiple regressions, each predictor variable is entered into each regression after entering the predictor variables that account for the variance of the post-test in all versions of the ITS. I first entered the *time spent in the first 3 problems*, then the *training time* and the *bugs in the "New Pattern" row*, and at last one of the potential predictors for the "model" version ITS. I used the *bugs in the "New Pattern" row* because it accounts for a higher percentage of the variance than the *total bugs* and the *problems having bugs.* The three variables are not independent from each other although they are all predictor variables which affect the post-test score. Among all the tutoring moves and attitudes, I found that only the number of "evaluating" moves affects the post-test score ($p<0.03$): it accounts for 12.7% of the variance. (The more "evaluating" tutoring moves, the higher the post-test score.)

TABLE XXIV

EXAMPLE MESSAGES WITH CORRESPONDING TUTORING MOVES AND ATTITUDES

Move	Attitude	Example Message
Confirming	Positive	This is a "P."
Evaluating	Neutral	OK.
Summarizing	Negative	From "R" to "T," you are going forward 2 in the alphabet, but...
Prompting	Positive	How did you get that?
Diagnosing	Negative	Are you doing backward 3?
Instructing	Positive	Actually you would get the patterns that way just by knowing the local relationships.
Instructing	Negative	Look for what this "C" is related to, maybe not the one that you thought.
Demonstrating	Neutral	So from "F," count forward 3 in the alphabet, you'll get the letter in the new pattern.
Demonstrating	Positive	"L M N O P Q," so "Q" is forward 5 from "L" in the alphabet.
Support	Negative	This is a tough one.

TABLE XXV

AVERAGE NUMBER OF EACH TYPE OF TUTORING MOVE WITH EACH TYPE OF TUTORING ATTITUDE, PER SUBJECT

	Tutoring Attitude		
Tutoring Move	Positive	Negative	Neutral
Confirming	4.2	0	2
Evaluating	37	1	5.9
Summarizing	23	3.9	22
Prompting	1	0	2.3
Diagnosing	12	1	5.8
Instructing	2.5	1	5.8
Demonstrating	1.9	0	1
Support	20	3.9	13

Each feedback message that the subject received was generated based on a type of tutoring move and a tutoring attitude. For example, a message can be an “instruction” with a “positive” attitude. So other than the number of moves and attitudes, I can also have the number of the combinations of each type of move and each type of attitude as the potential predictor variables. Some example messages with some combinations of tutoring move and attitude are listed in Table XXIV. There are no “negative confirming,” “negative prompting” or “negative demonstrating” moves in the experiments. Table XXV reports the average number of each type of tutoring move with each type of tutoring attitude. “Evaluating” as used in the following regressions includes all the “evaluating” moves with any tutoring attitude. It was not distinguished by the tutoring attitude because an “evaluating” move can predict more variance of post-test than when combined with a tutoring attitude. On the basis of the regression result with *time spent in the first 3 problems*, *training time*, *bugs in the "New Pattern" row* and *number of evaluating*, I ran multiple regressions with each type of combination as another predictor variable. I found the number of "positive instructing" moves marginally affects the post-test score ($p=0.06$): it accounts for 7.5% of the variance (fewer "instructing"s with "positive" attitude, the higher the post-test score). Again on the basis of the regression result with *time spent in the first 3 problems*, *training time*, *bugs in the "New Pattern" row*, *number of evaluating* and *number of positive instructing moves*, I ran multiple regressions with each type of combination as another predictor variable. I found the number of "neutral demonstrating" moves affects the post-test score ($p<0.05$): it accounts for 8.0% of the variance. (The more "demonstrating" moves with "neutral" attitude, the higher the post-test score.) No more significant results were found on the basis of all the above predictor variables.

On the whole for the "model" version ITS, I can explain 44.8% of the variance of the post-test score using *time spent in the first 3 problems*, *training time*, *bugs in the "New Pattern" row*, *number of evaluating*, *number of positive instructing* and *number of neutral demonstrating moves*: for higher post-test score, during the interaction with the ITS, the subject spent shorter time in the first 3 problems. They also spent shorter time in total, input fewer incorrect letters in the "New Pattern" row. In addition they received more evaluations, received fewer positive instructing, received more neutral demonstrating. From Table XXV we can see that there are many more positive evaluations than evaluations with other attitudes. So the subjects often receive evaluations when they input correct letters. This is consistent with that the fewer bugs, the subjects obtain higher post-test score, since "fewer bugs" means "more correct inputs."

The tutoring move “instructing” provides the student with information about the problem. On the contrary, “demonstrating” shows the student the process of solving the problem. Although instructing and demonstrating were reported as hard to distinguish by the annotators (the inter-annotator agreements are low, as reported in Chapter 3), the significant results with the ITS show that instructing and demonstrating are the tutoring moves that affect learning. The observations also support the findings in the study of human tutors in Chapter 3: the novice tutor uses more declarative instructing to provide the facts about the problem; instead, the expert tutor, who is significantly more effective than the novice tutor, demonstrates how to solve the problem more.

6.3.3 Rating of the Modeled Feedback Messages

With the “model” version ITS, each subject also completed a questionnaire which is used to collect subjective evaluations of the ITS. The questionnaire asks each subject to give a rating (from 1 to 5) of 7 aspects of the ITS and write down general comments at the end (More details can be found in Appendix D). The 7 aspects were:

1. How often did the subject read the feedback message (1 for “never,” 5 for “every time”);
2. The understandability of the feedback message (1 for “difficult,” 5 for “easy”);
3. Whether the feedback message was useful during training (1 for “not useful,” 5 for “very useful”);
4. How often does the subject feel that the feedback message was misleading (1 for “always,” 5 for “never”);

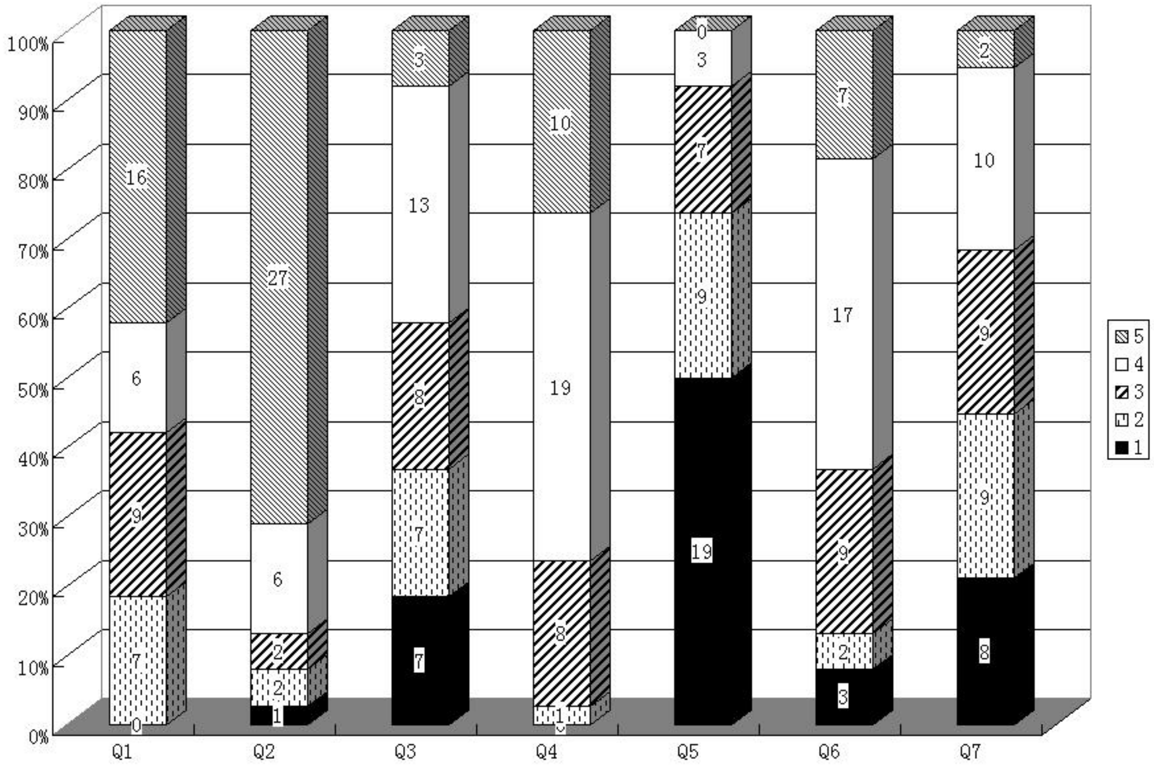


Figure 14. Ratings for the 7 issues in the questionnaire

5. Whether the subject felt that the feedback message was repetitive (1 for “very repetitive,” 5 for “not repetitive”);

6. Whether the subject thought the ITS helpful for the post-test (1 for “not helpful,” 5 for “very helpful”);

7. Whether the subject thought the feedback message in the ITS helpful for the post-test (1 for “not helpful,” 5 for “very helpful”).

Figure 14 reports the distribution of the ratings for the 7 issues. The Y-axis represents the percentages of the ratings. The numbers on the bars in the figure are the counts of each rating. From the figure, we can see that:

1. Most of the subjects (>70%) read the feedback messages often (Rate≥3);
2. Most of the subjects (>90%) had no difficulty understanding the feedback messages (Rate≥3);
3. Most of the subjects (>60%) thought the feedback messages useful during training (Rate≥3);
4. Almost none of the subjects (≈0%) felt the feedback messages misleading (Rate<3);
5. Most of the subjects (>70%) felt the feedback messages repetitive (Rate<3);
6. Most of the subjects (>80%) thought that the interaction with the ITS was helpful for the post-test (Rate≥3);
7. A little bit more than half of the subjects (>50%) thought that the feedback messages from the ITS during training were helpful for the post-test (Rate≥3).

So according to the subjects' ratings, the natural language feedback generator generated useful, understandable, not misleading, but repetitive feedback messages. The messages are repetitive because they were generated using a limited number of templates. Some of the subjects did not agree that it was the feedback message from the ITS during training that helped them on the post-test although they thought the interaction with the ITS helpful. This is possible because subjects tend to give an overall evaluation of the feedback but in fact during training only a few useful feedback messages may make a big difference later.

6.4 Discussion

The above results and analyses, not only answered the questions that are raised at the beginning of this chapter, but also led us to some interesting findings about the ITS.

The interaction with the ITS does improve learning. Judging from the post-test performance, the subjects who interacted with the ITS performed significantly better than the ones who didn't. Although there is no pretest to compute the learning gain in the letter pattern task, the analysis to estimate the pre-tutoring ability using the time spent in the first 3 problems, validates the observations that the subjects with different versions of the ITS have the same level of pre-tutoring ability. Therefore, we can assume that the post-test scores (which we have) are a good approximation of learning gains (which we don't have). This further supports our claims that the interaction with the ITS improves learning and that the ITS with different kinds of feedback improves learning to different degrees.

The interaction with the ITS is helpful, but the ITS that provides simple capsulated feedback messages is not more helpful than the ITS that does not provide verbal feedback messages. Furthermore, the ITS that provides simple feedback is not more helpful than the ITS that just supports unsupervised practice. On the contrary, the ITS that provides feedback messages generated by the rule-based model of expert tutoring is significantly more helpful than the other versions of the ITS. Compared with the human tutors, only the expert tutor is significantly more helpful than the ITS with simple feedback. However, there is no significant difference between the expert human tutor and the ITS with feedback that models expert tutoring. This means that the tutorial rules learned through the expert tutoring dialogues by using CBA successfully modelled expert tutoring. Although there is no significant difference in the post-test scores between the other two human tutors and the ITS, the subjects with the ITS that provides feedback performed better in the first post-test problem. It is possible that the subjects with the ITS adapted themselves to the computer interface of the post-test more easily. Among all the four versions of the baseline ITS, the "positive" version improved learning slightly more than the other three versions of the baseline ITS and even beat the novice tutor. Even if the result is not significant, this leads me to hypothesize that more positive feedback may be more effective. In another study of tutoring in the letter pattern task (Corrigan-Halpern, 2006), subjects given positive feedback performed better on the assessment task than subjects receiving negative feedback. This supports the principle that feedback is maximally effective if it can be processed easily and quickly. Positive feedback may be processed with minimal effort, because it provides information about actions the student has already performed correctly. The concept of the positive and negative feedback is similar to the tutor's attitude that is used in learning tutorial rules in Chapter 4 and the tutoring attitude that is used in the "model" version ITS. Therefore I counted the tutor's attitude in the expert tutor's dialogues, and found that the ratio of the positive attitude to the negative attitude is about 3.9 to 1. From Table XXV, we can also see that the number of positive attitudes that the "model" version provided in the feedback messages is about 10 times the number of negative attitudes. All the evidence supports that providing more positive feedback is more helpful to learning.

The "model" version ITS is evaluated from both a subjective and objective point of view. With respect to the subjective rating from the subjects who interacted with the "model" version

ITS, the feedback messages are easy to understand, not misleading, and useful during training but repetitive. Although most of the subjects agreed that the interaction with the ITS was very helpful for the post-test, only about half of them thought that the feedback messages contributed the most. Based on objective statistical analysis, the significant result of the "model" version ITS shows that the feedback generator can generate effective feedback messages using the rule-based model of expert tutoring. Among all the types of feedback messages, I found that the following three types contributed to learning improvements: "evaluating," "instructing" with "positive" attitude and "demonstrating" with "neutral" attitude. The findings suggest that ITSs might be able to improve learning effectiveness by

- acknowledging more what the student is doing;
- showing the student how to solve the problem more often;
- reducing the positive instructions that just provide the student with information about the problem.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

In this chapter, I summarize the accomplishments of this research and propose short-term goals and long-term directions for future research.

7.1 Conclusions

I began this work by stating that my goal was to demonstrate the utility of a computational model of expert tutoring in generating effective natural language feedback in ITSs.

I accomplished this goal by dividing it into four subgoals:

1. Comprehensively studying the difference between expert tutors and non-expert tutors;
2. Developing a computational model of expert tutoring based on the tutorial dialogues by using machine learning techniques;
3. Designing a flexible natural language feedback generator that employs the model;
4. Implementing an ITS which illustrates the effectiveness of the feedback generator.

Starting from collecting, annotating and analyzing human tutoring dialogues, I obtained on an empirical basis a characterization of how an expert tutor gives feedback to engender better learning outcomes. From the annotated dialogues, I developed a rule-based model of expert tutoring by using Classification Based on Associations. Then these rules were employed in a natural language feedback generator with 3-tier probabilistic planning. The model-tracing

ITS that provides natural language feedback generated by this generator helped students learn more. More specifically the major conclusions of my work are the following:

- Features of expert tutoring:
  - The expert tutor is more effective than the non-expert tutors (the lecturer and the novice tutor).
  - The expert tutor uses more complex language and more varied strategies.
  - The expert tutor demonstrates or models the process for solving the problem instead of delivering information directly.
  - The expert tutor finds every grain of success in the student's answers, and reinforces the efforts of the student in even minor accomplishments.
  - The expert summarizes and reviews what he has done and what the student has done.
  - The expert tutor assesses the situation, not only after a student's answer or action, but also after a student's explanation or summarization.
  - The expert tutor uses questions to enhance problem solving.
- Computational modeling and feedback generation:
  - Production rules are a good formalism to computationally model tutoring.
  - Classification based on associations (CBA) performs well in learning tutorial rules from annotated tutorial dialogues, even when we only have a small set of annotated tutorial dialogues.

  - The tutorial rules learned for choosing the tutor's attitude and choosing the topic are very accurate. Although the rules for choosing the tutor move are not as accurate, the feedback messages generated based on these rules improve learning.
  - CBA can also be used for selecting templates to realize natural language feedback messages.
  - The Tutoring Development Kit (TDK) provides a convenient way to develop a model-tracing ITS for a cognitive science task – extrapolating complex letter patterns. It also supports the delivery of simple graphical and verbal feedback.
  - Natural language feedback generation can benefit from Information State theory. The 3-tier probabilistic planning framework based on this theory can automatically synthesize plans from the tutorial rules.

- Empirical findings about the ITS:
  - The natural language feedback generator generates useful, understandable feedback messages that are not misleading.
  - The interaction with an ITS improves learning.
  - The student does not learn more when receiving simple graphical feedback or simple capsulated verbal feedback than from practice.
  - The student learns more when receiving natural language feedback messages generated based on the rule-based model of expert tutoring.

- There is no significant difference between the expert human tutor and the ITS with feedback that models expert tutoring. Therefore, the tutorial rules learned from the annotated tutorial dialogues by using CBA successfully modeled expert tutoring.
- The time spent in the interaction with the ITS has a significant effect on the performance in the post-test: The shorter the time spent in the first few problems and the shorter the total time spent with the ITS, higher post-test score obtained by the student.
- Performance in practing problems during the interaction with the ITS affects performance in the post-test: The fewer mistakes with the ITS, higher the post-test score obtained by the student.
- To improve learning, the ITS should provide more "evaluation" of what the student is doing, more often "demonstrate" to the student how to solve the problem, and reduce the number of the positive instructions that just provide the student with information about the problem.

All the above conclusions are based on a small data set, one single expert tutor and one single domain. They clearly need to be validated on a larger data set, or with different tutors and/or in different domains. However, our expert tutor is indeed an effective tutor so his behavior reflects what tutors do to help students learn better. On the other hand, our tutoring task is a cognitive science task that reflects our ideas about the general processing of how a human solves problems, so the findings in this domain should be easily extended to any other domain that involves problem solving.

7.2 Future Work

7.2.1 Improvements on Current Work

Most machine learning techniques, including CBA, benefit from larger data sets. In order to model expert tutoring more accurately, a larger collection of annotated tutorial dialogues is needed. We have only transcribed and annotated a small portion of tutorial dialogues from the 11 one-hour tutoring sessions with the expert tutor. So the next step would be to transcribe and annotate more dialogues of the expert tutor. Although this process is tedious, there are several advantages in doing it. The annotated tutorial dialogues are only for the two problems in the curriculum which do not cover all the letter relationships and more complex higher level relationships, such as progression in skipping letters. Having more tutorial dialogues transcribed will improve the coverage of the tutorial rules. In addition to obtaining more accurate tutorial rules, we will be able to improve the naturalness of the feedback messages, such as reducing the repetition that the subjects complained about. In recent years, researchers in the natural language generation field started to develop more flexible template-based realization, called "Stochastic Surface Realization" (Oh and Rudnicky, 2000; Ratnaparkhi, 2002; Langkilde and Knight, 1998). This type of surface realization not only maintains the advantages of template-based methods but also introduces corpus-based methods into natural language generation. If there is a larger data set, we can employ this type of surface realization in our feedback generator to generate more natural feedback messages.

Recently our research in another domain has shifted the study of human tutoring from expert tutors to effective tutors (Ohlsson et al., 2007), because studies of expert tutors have several weaknesses. Only a few expert tutors have been studied (del Soldato and du Boulay, 1995; Evens and Michael, 2006; Lepper et al., 1997). In addition, there is no clear definition of expert tutor. In most studies, tutors are often identified as "expert" on the basis of indirect indicators such as how long they have been tutoring. However, with respect to the goal of identifying the dimensions of tutoring that are causally related to high learning gains, what studies of human tutoring should really focus on is effective tutors, or effective tutoring sessions. The measure of effectiveness is some measure of learning outcomes. For the letter pattern task, we can regroup our tutoring sessions based on post-test performance and then repeat our study by exploring the difference between effective and non-effective tutoring sessions. Finally we can also obtain a rule-based model of effective tutoring and employ it into the feedback generation for the ITS.

7.2.2 Potential Long Term Directions

The framework of the natural language feedback generation which is proposed in this dissertation, is not specific to the letter pattern task. With suitable tutorial rule sets for the synthesis of plans and adapted template sets, the feedback generator can be used to generate tutorial feedback for ITSs in other domains. A couple of years ago, we started another project in another tutoring domain – basic data structures and algorithms. We have completed collecting data with two tutors. When the baseline ITS is ready, the feedback generator can be integrated into it to provide natural language feedback to students.

Although the ITS for the letter pattern task can provide students with effective natural language feedback, it does not fully support natural language interaction between the ITS and students. This is because currently it does not have a natural language understanding module to accept student's natural language inputs. The ability to understand student's language input would enhance the effectiveness of tutoring since the ITS would be able to assess the student's understanding more accurately and give more effective feedback. One of the features used for learning tutorial rules is the student move. If the ITS could understand the student's language input and categorize it into a type of student move, the generator could synthesize better plans from the tutorial rules since the decision would be made based on one more feature. Clearly understanding and categorizing student's input is a complex task and many researchers have been working on related issues (Graesser et al., 1999; Evens and Michael, 2006). Therefore, I would put developing a natural language understanding module into the long term plans for future research.

APPENDICES

Appendix A

DATABASE OF PATTERNS USED IN LETTER PATTERN TUTORING

A.1 PATTERNS/PROBLEMS TO BE USED IN TUTORING

1. E G I K *Analysis: Forward 2 relation*
2. T R P N L *Analysis: Backward 2*
3. D F H H J L N N *Analysis: Forward-3, repetition of every 3rd letter*
4. R S S T T T *Analysis: Progression on the number of repetitions; forward-1.*
5. A B B D D D G G G G *Analysis: Double progression on number of letters and forward-N.*
6. T T S R R Q P P *Analysis: Alternation of single versus repetition; backward-1.*
7. B C C E H H L Q Q *Analysis: Alternation of single letter and repetition; progression on forward-N.*
8. A B D G X X B C E H X X *Analysis: Four letter chunks with XX as chunk marker; repetition and recurrence on XX; progression on forward-N within chunks; forward-1G of entire chunk.*
9. B D D F F F C C E E G G G C *Analysis: Six letters chunks with chunk marker C; forward-2; progression on number of letters; forward-1; extra difficulty: spurious repetition.*
10. C E H L M L H E C *Analysis: Reflection; chunk marker; progression on forward-N/backward-N.*

11. O K H F Y D F I M Y *Analysis: Four-letter chunk; Y as chunk marker; progression on forward-N or backward-N within chunk; reflection combined with backward-2 between groups.*

12. M C D M F E M M C F M L I M *Analysis: Embedded chunks with 2 letters in the innermost chunk, 4 in the outer; forward-1 within first inner chunk, backwards-1 within second; then the pattern is stretched to forward-2 and backward-2.*

13. D T D E S S D E F T D E F G S S *Analysis: Alternation of chunk markers between single, double, S versus T; progression/stretching on number of letters in chunk; forward-1 within chunk.*

A.2 POST-TEST PROBLEMS

1. A B C X C B A Y Y D E F X F E D *Analysis: Embedded periods with outer chunk in 7 letter groups, and the inner chunk in 3-letter groups; X and Y Y as alternating chunk markers; reflection within a larger chunk; forward-1 and backward-1 within a chunk; forward-2 of entire chunk.*

2. A C Z D B Y Y D F X G E W W G *Analysis: Interleaving two patterns; pattern-1 has a chunk of 4 letters; the first two letters are related via forward-2; the second two letters via backward-2; the two groups via forward-1 from the last of first group to the first of second group; successive chunks are moved forward-3 down the alphabet; pattern 2 alternatives between 1 letter and 2 letters and goes backward-1 in the alphabet.*

Appendix B

THE ANNOTATION SCHEMES

B.1 The Annotation Scheme for Tutor Moves

- Tutor reaction: The tutor reacts to something the student does. The important thing to keep in mind when using this code is that whatever the tutor says is preceded by something the student says or does.

  - **Answering**: The student asks a question, and the tutor answers it. This category is used only when the tutor responds to direct questions. Before using this category, make sure that a question from the student precedes it.
  - **Evaluating**: The tutor gives the student feedback about what he/she is doing, specifically, a right/wrong evaluation. Evaluating should only be used within the context of a particular problem; it should not be used to evaluate the work done by the student across the set of problems.

    Example:

    Student: then it goes A C E

    Tutor: right

    Use Evaluate in this case

    Tutor: you've been doing really good so far

    Do NOT use Evaluate

Appendix B (Continued)

- Tutor reaction and/or tutor initiative

  Summarizing: The tutor summarizes what has been done so far. This can be either summarizing what the student has done so far or summarizing what the tutor has done so far. The key here is that the tutor is repeating something that the student knows, either because the student has said it, or because the tutor has said/demonstrated it. Only use Summarizing in context of the current problem.

  Examples:

  Tutor: Ok, so you said the pattern was go forward two, then backward one. A C B

  Tutor: Remember that I said to find the period markers first.

- Tutor initiative: The tutor speaks to the student without specifically responding to something the student has done or said.

  - **Prompting**: The tutor is prodding the student into some kind of activity. This may be general or specific.

    * **General**: The tutor is coaxing the student into general activity, or towards an action. Basically, the tutor is laying out what to do next.

      Example:

      Tutor: Why don't you try this problem?

Appendix B (Continued)

    - **Specific**: The tutor is trying to get a specific response from the student. The tutor may draw the student's attention towards a particular area of the problem and then ask him/her to do something in that area. The tutor may also question the student in hopes of drawing out a specific action.

      Examples:

      Tutor: What would the next letter be?

      Tutor: Look at these letters here. Can you find any pattern?

- **Diagnosing**: The tutor asks the student a question or makes a statement with the purpose of determining what the student is doing or thinking. The key to this category is that the tutor is seeking information about a student's behavior.

  Example:

  Tutor: Why did you put a D there?

- **Instructing**: The tutor is providing the student with information about the problem, either declarative (facts about the problem) or procedural (hints about how to solve the problem). The key here is that the tutor does not show the student how to solve the problem; rather the tutor just provides information.

    - **Declarative**: The tutor provides facts about the problem. This can be through telling the solution, pointing out relations between the periods or other parts, etc. The key here is that the tutor is telling the student something that he or she does not already know.

Appendix B (Continued)

Example:

Tutor: Notice the two Cs here? They are separating different parts of the problem.

∗ **Procedural**: The tutor gives the student hints or tricks about how to solve the problem. The tutor can also give the student specific directions. The tutor might tell the student to check the alphabetical relationships between the periods, for example.

Examples:

Tutor: Start by counting the number of letters in each period

Tutor: Remember to check for relationships within and between periods

- **Demonstrating**: This category goes beyond Instructing in that the tutor shows the student how to solve the problem through demonstrating correct actions and analysis of the problem. The attention of both the tutor and the student will be on solving the problem. This will often require a series of steps. The way to separate this category from Instructing is to ask yourself if the tutor is merely providing information, or going beyond that and being very directive in solving the problem.

Example:

Tutor: Watch this. First I count the number of letters between the G and J here. G H I J: three.

- **Support**: The tutor encourages the student in his/her work without making any specific problem-relevant comments. The tutor may agree that the problems are hard, or tell the student he or she is doing a good job so far, but will not mention anything about the problem at hand.

  Example:

  Tutor: Great job on the last problem. This next one is a little harder.

- **Conversation**: The tutor may talks to the student to set up a dialogue; for example, talking about the weather or other small talk. Additionally, the tutor may just say "mm hmm" or "uh-huh" to keep the dialogue going between student and tutor. The key to using this category is that the tutor does not make any reference to the problem at all. If you are not sure, check what the tutor is saying versus the videotape.

B.2 The Annotation Scheme for Student Moves

- **Explanation**: This category is operationalized as any utterance that went beyond the information given (e.g. tutor's questions or instructions), namely, an inference of new knowledge. So what he/she says may not be fit for the particular prompting but go beyond that. We usually code a student turn as an explanation when the utterance shows the student was explaining what he/she just did or said, doing reasoning or "think aloud," or making an indirect answer to a tutor's question. This category may appear in a series of answers to tutor's prompting or follow a sequence of student's utterances in the dialogue

Appendix B (Continued)

or some actions can't read from the dialogue but can be seen from the video. If it belongs to the last, coders need to write down the actions at the same time.

- **Questioning**: The student asks the tutor questions.
- **Reflecting**: reflecting consists of comprehension monitoring statements that might be made in response to tutors's comprehension gauging questions, or some other tutoring moves, or to express the degree of his/her understanding to the problem.
- Student Reaction: The student reacts to something the tutor says. The important thing to keep in mind when using this code is that whatever the student says is preceded by something the tutor says immediately or is spread in several turns after tutor's prompting. For this category and its following subcategories, the coder also need to code the correctness of the student's reaction as correct (c), partially correct (p), wrong (w). To code the correctness, the coder needs to know the correct answer in advance and also needs to look at the video carefully to know the problem proposed by the tutor. If the coder can not figure out the correctness of the student's reaction, just code the category leaving the correctness aside. However, usually the coder can get the value from the following tutor's turns.
  - **Answering**: The student answers in words directly to a tutor's question or prompting. To a particular prompting, the answer may appear as a series of turns or appear after a few other responses. Also the answer can be in a question form which shows

  the student was not sure about the correctness instead of expecting an answer from the tutor.

  - **Action response**: To a tutor's question or prompting, the student takes some actions as responses. It may happen at the same time the student is answering to a question or prompting. The student may only make action responses to a question or prompting. Usually what the student does is pointing to somewhere or write letters on the paper. So this category is applied to a student turn which does not appear in the dialogue transcripts but in the video. To code for this category, coders need to add a student turn into the transcripts with the short description about the student's actions.

- **Completion**: The student completes a tutor's utterance. It may happen at the same time the tutor is talking or while the tutor pauses.

- **Conversation**: The student's responses to what the tutor talks to set up a dialogue. Additionally, the student may also say "um," "ok," "oh" like the tutor to keep the dialogue going.

Instruction to write a short description of the student's actions:

Usually the student's actions belong to one of the following four categories:

1. Write letters

   If the student writes some letters down, the action will be described as "write letters XXXXXX."

Appendix B (Continued)

2. Mark letters

   If the student draw lines or other marker under or above some letters, the action would be “mark letters XXXXXX.”

3. Point to a letter or a set of letters

   If the student points to a letter or a set of letters using finger or pen, the action would be “point to letters X” or “point to letters XXXXX.”

   Note: Even if the student only writes, marks or points to one letter, we use the plural form of “letter.”

4. Remove letters

   If the student removes some letters by scratching or some other ways, the action would be “remove letters XXXXXX.”

The other problem is where to put the description. There are three different situations:

1. The action was happening at the same time of a particular student verbal turn. We put the description into the “Action” column in the same line of this student turn.

2. The action was happening at the same time of a series of student verbal turns. We put the description into the “Action” column in the line of the starting turn of that series.

3. The action was happening independently (we code it as “action response” category). First, we insert one line between the preceding turn and the following turn of this action response. Then, we put the description into the “Action” column in the new line that we just inserted.

Appendix C

THE TEMPLATES FOR THE SURFACE REALIZATION OF THE FEEDBACK MESSAGES

1. This is a(n<input>).
2. Yeah, this is a(n<input>).
3. We are going to compare "<current_example>" in column <current_column> to "<reference_example>" in column <reference_column>.
4. So from "<reference_pattern>," count <skip_relation> <skip_number> in the alphabet, you'll get the letter in the new pattern.
5. Look at the whole sequence, you'll find there are chunks with (chunk_lengths) letters.
6. (currentChunk_examples), they are <skip_relation> <skip_number> in the alphabet.
7. (list_letters<reference_example><current_example>), so "<current_example>" is <skip_relation> <skip_number> from "<reference_example>" in the alphabet.
8. It serves as a marker but it's also really close to the other letters.
9. Here is the marker. Leave it alone.
10. Let's say this could relate to column <reference_column>.
11. From "<reference_pattern>" to "<input>," you are going <input_relation> <input_number> in the alphabet.

Appendix C (Continued)

12. This is a tough one.

13. That's OK. Don't worry about it.

14. That's what most people find.

15. You are pretty good at these.

16. Some place there is a mistake.

17. That's great.

18. Good job.

19. Yeah exactly.

20. That's right.

21. What I think is helpful is you noticed the chunks with (chunk_lengths) letters.

22. Let's count the total number of letters in the pattern. There are <pattern_length> letters in this pattern.

23. We are just going to keep this the same.

24. You know the relationship.

25. Then repeat the marker.

26. You got these repeated "<current_example>"s.

27. Look for what this "<current_example>" is related to, maybe not the one that you thought.

28. You have to have a <skip_relation> <skip_number> from "<reference_pattern>."

Appendix C (Continued)

29. Even though you know how to do this stuff, you could make mistakes, so you should be especially careful to double check.

30. (explain<skip_relation><skip_number>)

31. Finding a way to look at the letters will help you do it more quickly. For example, use your fingers to count.

32. Not all the patterns are divided down the middle but it's a good place to think about.

33. If it's a long pattern it's helpful to think about dividing it into some smaller patterns.

34. Sometimes we'll move the markers and sometimes we won't. If the marker's close then we could move it along with the other letters. If the marker's far, then we'll just keep it as it is.

35. If you have a relationship between letters, you may want to find this exact same relationship someplace else.

36. It makes it a little more challenging just keeping track of the markers, because it has to match.

37. Actually you would get the patterns that way just by knowing the local relationships.

38. How many steps from "<reference_example>" to "<current_example>"?

39. Look carefully at this pattern and see what's happening.

40. What is the letter going <skip_relation> <skip_number> from "<reference_pattern>"?

41. Where do you see the same relationship in the pattern?

Appendix C (Continued)

42. Count the total number of letters in the pattern. There are <pattern_length>, right? Can you divide the pattern down in the middle?

43. From "<reference_pattern>" to "<input>," you have a <input_relation> <input_number> in the alphabet. Right?

44. So in this case the marker is close to the other letters in the pattern, right?

45. What is the relationship between "<reference_example>" and "<current_example>"?

46. Do you think this column is related to column <reference_column> or something else in the pattern?

47. Are you doing <skip_relation> <skip_number>?

48. How did you get that?

49. Where does this "<current_example>" come from?

50. You know that you need to do <skip_relation> <skip_number>, right?

Appendix D

QUESTIONNAIRE FOR THE “MODEL” VERSION ITS

1. Did you read the verbal feedback that the tutoring system provided?

   1 2 3 4 5

   Never Every Time

2. Did you have any difficulty understanding the verbal feedback?

   1 2 3 4 5

   A Lot Not At All

3. Did you find the verbal feedback useful while interacting with the tutoring system?

   1 2 3 4 5

   Not At All Very Useful

4. Did you ever find the verbal feedback misleading?

   1 2 3 4 5

   Always Never

5. Did you find the verbal feedback repetitive?

   1 2 3 4 5

   Very Repetitive Not At All

6. Did you find the interaction with the tutoring system helpful for your performance in the post-test?

Appendix D (Continued)

1 2 3 4 5

Not At All Very Helpful

7. Did you find the verbal feedback from the tutoring system helpful for your performance in the post-test?

   1 2 3 4 5

   Not At All Very Helpful

8. Please comment on any of the preceding questions or any other issue.

CITED LITERATURE

Agrawal, R. and Srikant, R.: Fast algorithms for mining association rules. In Proceedings of the 20th International Conference of Very Large Data Bases, VLDB, eds. J. B. Bocca, M. Jarke, and C. Zaniolo, pages 487–499. Morgan Kaufmann, 1994.

Aleven, V., Koedinger, K., and Cross, K.: Tutoring answer explanation fosters learning with understanding. In Proceedings of AI-ED'99, pages 199–206, Amsterdam, 1999. IOS Press.

Anderson, J. R., Boyle, C. F., Corbett, A. T., and Lewis, M. W.: Cognitive modeling and intelligent tutoring. Artificial Intelligence, 42(1):7–49, 1990.

Bloom, B. S.: The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. Educational Researcher, 13(6):4–16, 1984.

Blum, A. and Langford, J.: Probabilistic planning in the graphplan framework. In Proceedings of the Fifth European Conference on Planning, pages 319–332, 1999.

Boutilier, C., Dearden, R., and Goldszmidt, M.: Stochastic dynamic programming with factored representations. Artificial Intelligence, 121(1–2):49–107, 2000.

Carletta, J.: Assessing agreement on classification tasks: The kappa statistic. Computational Linguistics, 22(2):249–254, 1996.

Cawsey, A.: Explanatory dialogues. Interaction with Computers, 1(1):69–92, April 1989.

Chae, H. M., Kim, J. H., and Glass, M.: Effective behaviors in a comparison between novice and expert algebra tutors. In Proceedings of Sixteenth Midwest AI and Cognitive Science Conference, pages 25–30, Dayton, 2005.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P.: Smote: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16:321–357, 2002.

Chi, M. T., de Leeuw, N., Chiu, M. H., and LaVancher, C.: Eliciting self-explanations improves understanding. Cognitive Science, 18:439–477, 1994.

CITED LITERATURE (Continued)

Chi, M. T., Siler, S. A., Jeong, H., Yamauchi, T., and Hausmann, R. G.: Learning from human tutoring. Cognitive Science, 25(4):471–533, 2001.

Collins, A.: Teaching reasoning skills. Thinking and Learning Skills, 2:579–586, 1985.

Corrigan-Halpern, A.: Feedback in Complex Learning: Considering the Relationship Between Utility and Processing Demands. Doctoral dissertation, University of Illinois at Chicago, 2006.

Cromley, J. G. and Azevedo, R.: What do reading tutors do? a naturalistic study of more and less experienced tutors in reading. Discourse Processes, 40(2):83–113, 2005.

Currie, K. and Tate, A.: O-plan: The open planning architecture. Artificial Intelligence, 52:49–86, 1991.

del Soldato, T. and du Boulay, B.: Implementation of motivational tactics in tutoring systems. Journal of Artificial Intelligence in Education, 6(4):337–378, 1995.

Di Eugenio, B., Fossati, D., Yu, D., Haller, S., and Glass, M.: Aggregation improves learning: experiments in natural language generation for intelligent tutoring systems. In Proceedings of the 42nd Meeting of the Association for Computational Linguistics, Ann Arbor, MI, 2005.

Di Eugenio, B. and Glass, M.: The kappa statistic: A second look. Computational Linguistics, 30(1):95–101, 2004.

Evens, M. and Michael, J.: One-on-one tutoring by humans and computers. Mahwah, NJ, Lawrence Erlbaum Associates, 2006.

Fox, B.: The human tutorial dialogue project. Hillsdale, NJ, Lawrence Erlbaum Associates, 1993.

Freedman, R.: Interaction of Discourse Planning, Instructional Planning and Dialogue Management in An Interactive Tutoring System. Doctoral dissertation, Northwestern University, 1996.

Freedman, R., Zhou, Y., Glass, M., Kim, J. H., and Evens, M. W.: Using rule induction to assist in rule construction for a natural-language based intelligent tutoring system. In Proceedings of the 20th Annual Conference of the Cognitive Science Society, Madison, WI, 1998.

CITED LITERATURE (Continued)

Glass, M., Kim, J. H., Evens, M. W., Michael, J. A., and Rovick, A. A.: Novice vs. expert tutors: a comparison of style. In Proceedings of Tenth Midwest Artificial Intelligence and Cognitive Science Conference, Bloomington, IN, 1999.

Graesser, A., Person, N. K., and Harter, D.: Teaching tactics and dialog in autotutor. International Journal of Artificial Intelligence in Education, 12(3):257–279.

Graesser, A. C., Person, N., Lu, Z., Jeon, M. G., and McDaniel, B.: Learning while holding a conversation with a computer, chapter Technology-Based Education: Bringing Researchers and Practitioners Together, pages 143–167. Greenwich, CT, Information Age Publishing, 2005.

Graesser, A. C. and Person, N. K.: Question asking during tutoring. American Educational Research Journal, 31(1):104–137, 1994.

Graesser, A. C., Wiemer-Hastings, K., Wiemer-Hastings, P., and Kreuz, R.: Autotutor: A simulation of a human tutor. Journal of Cognitive Systems Research 1, pages 35–51, 1999.

Groves, M., Rego, P., and O'Rourke, P.: Tutoring in problem-based learning medical curricula: the influence of tutor background and style on effectiveness. BMC Medical Education, 5(20), June 2005.

Japkowicz, N.: The class imbalance problem: Significance and strategies. In Proceedings of the 2000 International Conference on Artificial Intelligence, volume 1, pages 111–117, 2000.

Khuwaja, R. A., Evens, M. W., Rovick, A. K., and Michael, J. A.: Architecture of circsim-tutor (v.3): A smart cardiovascular physiology tutor. In Proceedings of the 7th Annual IEEE Computer-Based Medical Systems Symposium, pages 158–163. IEEE Computer Society Press, 1994.

Kim, J. H. and Glass, M.: Evaluating dialogue schemata with the wizard of oz computer-assisted algebra tutor. In Proceedings of Intelligent Tutoring Systems: 7th International Conference (ITS 2004), Maceio, Brazil, 2004. Springer.

Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H., and Verkamo, A. I.: Finding interesting rules from large sets of discovered association rules. In Proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM'94), eds. N. R. Adam, B. K. Bhargava, and Y. Yesha, pages 401–407. ACM Press, 1994.

CITED LITERATURE (Continued)

Koedinger, K. R., Aleven, V., and Heffernan, N. T.: Toward a rapid development environment for cognitive tutors. In Proceedings of the 12th Annual Conference on Behavior Representation in Modeling and Simulation, 2003.

Kotovsky, K. and Simon, H.: Empirical tests of a theory of human acquisition of information-processing analysis. British Journal of Psychology, 61:243–257, 1973.

Langkilde, I. and Knight, K.: Generation that exploits corpus-based statistical knowledge. In Proceedings of COLING-ACL, pages 704–710, 1998.

Larsson, S. and Traum, D. R.: Information state and dialogue management in the trindi dialogue move engine toolkit. Natural Language Engineering, 6(3–4):323–340, 2000.

Lepper, M. R., Drake, M. F., and O'Donnell-Johnson, T.: Scaffolding techniques of expert human tutors, chapter Scaffolding student learning: Instructional approaches and issues, pages 108–144. Brookline, 1997.

Linden, K. V. and Di Eugenio, B.: A corpus study of negative imperatives in natural language instructions. In Proceedings of the 17th International Conference on Computational Linguistics, pages 346–351, 1996.

Linden, K. V. and Di Eugenio, B.: Learning micro-planning rules for preventative expressions. In Proceedings of the 8th International Workshop on Natural Language Generation, pages 11–20, Sussex, UK, 1996.

Litman, D. J., Rose, C. P., Forbes-Riley, K., Vanlehn, K., Bhembe, D., and Silliman., S.: Spoken versus typed human and computer dialogue tutoring. In Proceedings of the 7th International Conference on Intelligent Tutoring Systems, Alagoas, Brazil, 2004.

Liu, B., Hsu, W., and Ma, Y.: Integrating classification and association rule mining. In Proceedings of Knowledge Discovery and Data Mining, pages 80–86, New York, August 1998.

MacWhinney, B.: The CHILDES project. Tools for analyzing talk: Transcription Format and Programs, volume 1. Mahwah, NJ, Lawrence Erlbaum, 3rd edition, 2000.

Moore, J. D., Porayska-Pomsta, K., Varges, S., and Zinn, C.: Generating tutorial feedback with affect. In Proceedings of the Seventeenth International Florida Artificial Intelligence Research Symposium Conference (FLAIRS). AAAI Press, 2004.

CITED LITERATURE (Continued)

Murray, T.: Authoring intelligent tutoring systems: An analysis of the state of the art. International Journal of Artificial Intelligence in Education, 10:98–129, 1999.

Murray, T.: An Overview of Intelligent Tutoring System Authoring Tools: Updated analysis of the state of the art, chapter 17, pages 491–544. The Netherlands, Kluwer Academic Publishers, 2003.

Oh, A. and Rudnicky, A.: Stochastic language generation for spoken dialogue systems. In Proceedings of the ANLP/NAACL 2000 Workshop on Conversational Systems, pages 27–32, Seattle, 2000. Association for Computational Linguistics.

Ohlsson, S., Di Eugenio, B., Chow, B., Fossati, D., Lu, X., and Kershaw, T. C.: Beyond the code-and-count analysis of tutoring dialogues. In Proceedings of the 13th International Conference on Artificial Intelligence in Education, 2007.

Ong, J. and Ramachandran, S.: Intelligent tutoring systems: The what and the how. Learning Circuits, 2000.

Passonneau, R.: Measuring agreement on set-valued items (masi) for semantic and pragmatic annotation. In Proceedings of 5th International Conference on Language Resources and Evaluation, pages 831–836, Genoa, Italy, May 2006. European Language Resources Association.

Pasula, H. M., Zettlemoyer, L. S., and Kaelbling, L. P.: Learning probabilistic relational planning rules. In Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling, pages 73–82.

Person, N.: Why study expert tutors? Presentation at the Annual Project Meeting of the Office of Naval Research, 2005.

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

Quinlan, R. J.: C4.5: Programs for Machine Learning. Los Altos, CA, Morgan Kaufmann, 1993.

Ratnaparkhi, A.: Trainable methods for surface natural language generation. Computer Speech and Language, 16:435–455, 2002.

CITED LITERATURE (Continued)

Schulze, K., Shelby, R., Treacy, D., Wintersgill, M., VanLehn, K., and Gertner, A.: Andes: A coached learning environment for classical newtonian physics. The Journal of Electronic Publishing 6, pages 133–142, 2000.

VanLehn, K., Siler, S., and Murray, C.: Why do only some events cause learning during human tutoring. Cognition and Instruction, 21(3):209–249, 2003.

Zhou, Y.: Building A New Student Model to Support Adaptive Tutoring in a Natural Language Dialogue System. Doctoral dissertation, Illinois Institute of Technology, 2000.

Zinn, C., Moore, J. D., and Core, M. G.: A 3-tier planning architecture for managing tutorial dialogue. In Proceedings of Intelligent Tutoring Systems 6th. Intl. Conference, ITS 2002, eds. G. G. S.A. Cerri and F. Paraguacu, volume LNCS 2363, pages 574–584, Biarritz, France, 2002. Springer-Verlag Berlin Heidelberg.

VITA

NAME: Xin Lu

EDUCATION: Ph.D., Computer Science
University of Illinois at Chicago, Chicago, Illinois, 2007

M.S., Computer Science
Harbin Institute of Technology, Harbin, China, 2002

B.S., Computer Science
Harbin Institute of Technology, Harbin, China, 2000

EXPERIENCE: Research Assistant, Natural Language Processing Lab
University of Illinois at Chicago, Chicago, Illinois, 08/2002-07/2007

Teaching Assistant,Department of Computer Science
University of Illinois at Chicago, Chicago, Illinois, 2005-2006

Group Leader, Speech Technology Group, Machine Translation Lab
Harbin Institute of Technology, China, 03/2000- 07/2002

Research Intern, Speech Group
Microsoft Research Asia, Beijing, China, 11/2000- 03/2001

HONORS AND AWARDS: Best student paper award, International Conference on Intelligent Text Processing and Computational Linguistics, 2007.

Grace Hopper travel fellowship, Department of Computer Science, University of Illinois at Chicago, 2006.

WISE (Women in Science and Engineering) travel award, University of Illinois at Chicago, 2006.

The second place award, International Contest in Mathematical Modeling, 1999.

Dean's fellowship in three consecutive years, Harbin Institute of Technology, China, 1997,1998,1999.

People fellowship, Harbin Institute of Technology, China, 1999.

Baoshan Steel educational fellowship, Harbin Institute of Technology, China, 1997.

Freshman fellowship, Harbin Institute of Technology, China, 1996

PROFESSIONAL ACTIVITIES:

Program committee of the International Workshop on Communication between Human and Artificial Agents at the IEEE/WIC/ACM International Conference on Intelligent Agent Technology, 2006, 2007.

Refereed reviewer of the Workshop on Language-enabled Educational Technology, ECAI 2006.

Refereed reviewer of International Conference on Intelligent Text Processing and Computational Linguistics, 2006, 2007.

Coordinator of AI seminar in Department of Computer Science, University of Illinois at Chicago, 2006.

PROFESSIONAL MEMBERSHIP:

Student member of Association of Computational Linguistics

Student member of Asia-Pacific Society for Computers in Education

Student Member of the International Artificial Intelligence in Education Society

PUBLICATIONS:

Ohlsson,S., Di Eugenio, B., Chow, B., Fossati, D, Lu, X., and Kershaw, T.C.: Beyond the Code-and-Count Analysis of Tutoring Dialogues. In Proceedings of the 13th International Conference on Artificial Intelligence in Education, 2007.

Lu, X., Di Eugenio, B., and Ohlsson, S.: Learning Tutorial Rules Using Classification Based on Associations. In Proceedings of the 13th International Conference on Artificial Intelligence in Education, 2007.

Lu, X., Di Eugenio, B., Kershaw, T.C., Ohlsson, S., and Corrigan -Halpern, A.: Expert vs. Non-expert Tutoring: Dialogue Moves, Interaction Patterns and Multi-Utterance Turns. In Proceedings of the 8th International Conference on Intelligent Text Processing and Computational Linguistics, Lecture Notes in Computer Science, 4394:

456–468, 2007, Springer-Verlag.

Lu, X.: Expert Tutoring and Natural Language Feedback in Intelligent Tutoring Systems. In Proceedings of Doctoral Student Consortium at the 14th International Conference on Computers in Education, 2006.

Di Eugenio, B., Kershaw, T.C., Lu, X., Corrigan-Halpern, A., and Ohlsson, S.: Toward a Computational Model of Expert Tutoring: A First Report. In Proceedings of the 19th International FLAIRS Conference, pages 503–508, 2006.

Di Eugenio, B., Lu, X., Kershaw, T.C., Corrigan-Halpern, A., and Ohlsson, S.: Positive and negative verbal feedback for Intelligent Tutoring Systems. In Proceedings of the 12th International Conference on Artificial Intelligence in Education, 2005.

Ohlsson, S., Corrigan-Halpern, A., Di Eugenio, B., Lu, X., and Glass, M.: Explanatory Content and Multi-Turn Dialogues in Tutoring. In Proceedings of the 25th Annual Meeting of the Cognitive Science Society, 2003.

Lu, X., Liu, Z., Zhao, T., and Wang, L: Dealing with Polyphone in Text-to-Speech Systems Using How-Net. In Proceedings of NCMMCS6, Shenzhen, China, 2001.

Lu, X., Zhao, T., Liu, Z., and Yang, M.: Automatic Detection of Prosody Phrase Boundaries for Text-to-Speech Systems. In Proceedings of International workshop for Parsing Technologies, Beijing, China, 2001.

Lu, X., Xu, S., and Zhang, G.: Mathematical Model of Combinatorial Investment. Journal of Harbin Institute of Technology, Harbin, China, 1999, Harbin Institute of Technology Press.

PRESENTATIONS: Tutorial Dialogues: Expert vs. Non-Expert Tutors. The 3rd Midwest Computational Linguistics Colloquium, 2006.