USING JOINT ACTIONS TO EXPLAIN ACKNOWLEDGMENTS IN TUTORIAL DISCOURSE: APPLICATION TO INTELLIGENT TUTORING SYSTEMS

BY

STEFAN BRANDLE

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computer Science
in the Graduate College of the
Illinois Institute of Technology

Approved _______________________________
Advisor

Chicago, Illinois
May 1998

ACKNOWLEDGMENT

I wish to express my appreciation to the members of the CIRCSIM-Tutor research group who have taken this project from a twinkle in the eye to a useful running system. Michael Glass deserves a special mention for his work in collecting and organizing our documents, keeping the equipment running, and for bailing us out any number of times for a great number of things. I owe great thanks to those who spent many hours categorizing acknowledgments for me: Bhavana Vaidyanathan, Bruce Mills, Cecilia Widstrand, and Farhana Shah. Deborah Kennell deserves credit for her editorial attempts to get me to write in decent English. Last, Christina Fandrich gets credit for giving me the extra incentive to get done.

I am now joining the veritable throng of Ph.D. success stories who owe special thanks to Dr. Martha W. Evens, my advisor, for her patient guidance, support, and friendship. Thank you.

This work was supported by the Cognitive Science Program, Office of Naval Research under Grant No. N00014-94-1-0338, to the Illinois Institute of Technology and the associated AASERT Grants to the Illinois Institute of Technology. The content does not reflect the position or policy of the government and no official endorsement should be inferred.

S.S.B.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	**Definition**
	Cardiovascular
CC	Cardiac Contractility
CO	Cardiac Output
CVP	Central Venous Pressure
DR	Direct Response
HR	Heart Rate
IS	Inotropic State
MAP	Mean Arterial Pressure
RAP	Right Atrial Pressure
RR	Reflex Response
SS	Steady State
SV	Stroke Volume
TPR	Total Peripheral Resistance
	Other
CAI	Computer-Assisted Instruction
CST	CIRCSIM-Tutor
ITS	Intelligent Tutoring System
NL	Natural Language

CHAPTER I

INTRODUCTION

1.1 Problem Statement

Computers have erupted onto the human scene at an explosive pace. Most of the public is aware of the constant advances in the capabilities of the computer; what they are much less aware of is all the research required to make these advances possible. One of the very challenging areas of research is the study of how computers can be taught to interact with humans through the competent use of human language. This ability requires the capability both to generate language that can be understood by humans and also to process linguistic input from humans. A further complication is that linguistics has not finished describing human language; the field is certainly not able to propose a complete model of language use. The complete model then needs to be turned into a model suitable for implementation in computer systems. This effort requires a tremendous amount of research in essentially all areas of linguistics.

This thesis describes research in a small, but significant, area of human-computer-interaction (HCI) through natural language. It deals with the problem of how humans and computers can coordinate their communication through the use of acknowledgments and related phenomena. My primary goals are 1) to advance the understanding of coordination in communication by drawing together information about how coordination is understood in a number of different fields, 2) to describe coordination in tutorial dialogue, and 3) to propose a model of how intelligent tutoring systems (ITSs) should generate acknowledgments in language.

Human communication is powerful and successful, not entirely because of the immediate accuracy in understanding utterances, but in part due to the ability of the communicating parties to dynamically detect and correct problems as they arise. Acknowledgments and associated communication mechanisms are key to this process of monitoring communication to determine whether the quality of the communication is acceptably high. In studying natural language and human communication protocols, we face the problem that by the time we are old enough to become interested in the underlying mechanisms, those mechanisms have become so automatic that we have difficulty observing them. As I study natural language, I supplement the direct study of acknowledgments in our corpus of human communication transcripts by studying artificial communication protocols and then applying that knowledge back to the study of human protocols.

My natural language research group studies linguistic phenomena in the context of the continuing development of an intelligent tutoring system, CIRCSIM-Tutor. This intelligent tutoring system (ITS) uses natural language to tutor medical students on the human blood pressure regulation system. One of the problems facing ITSs such as ours is the loss of the subtleties of communication acknowledgment information such as facial expression, gesture, speech volume, timing and intonation. That is, these systems must deal with communication signal information loss. ITSs also face problems at higher acknowledgment levels. For instance, they must deal with determining what to expect in an utterance, recognizing whether an utterance matches what would normally be expected in the context, and determining the implications of what is actually said and what is not said. Accessing much of this information is beyond the current state of the art, but it is my belief that we can do more with the information available to us; namely the text typed in at the keyboard, some coarse timing information, and simple matching of inter-turn utterance expectations with what is actually typed by the student.

1.2 Intelligent Tutoring Systems with Natural Language Interfaces

1.2.1 Dreams of Intelligent Tutoring Systems. Science fiction writers have dreamed and written about computer-based education that would support high quality learning while accommodating individual student learning characteristics and scheduling needs [Cherryh, 1988; McCaffrey, 1990].

Even more down-to-earth visionaries have projected dreams of cost-effective intelligent tutoring systems – modeled on effective human tutoring behavior – which would go beyond automated page-turning to support dynamic teaching customized for each individual student. For instance, in a description of the "College of Utopia", Edward Lias speaks of round-the-clock availability, individually tailored textbooks, infinite patience, and global geographic and demographic availability as some of the possible benefits of introducing computers into education [Lias, 1982, p. 117, pp. 263-265].

1.2.2 Need for Effective Systems that Support Learning. At a meeting of the ICTE (International Conference on Technology in Education) [Brandle, Smith, and Robergé, 1996], I became aware that many organizations have massive ongoing training needs, especially large organizations such as branches of our military. Many people hope that ITSs can meet these needs by providing cost-sensitive, effective and flexible training. That is the reason why the Office of Naval Research is actively supporting a number of projects in ITS technology, including the CIRCSIM-Tutor project at the Illinois Institute of Technology.

Johnson [1988, p. 200], offers a list of typical reasons for seeking computer assistance in learning: 1) high flow of students, 2) expensive real equipment, 3) unavailable real equipment, 4) unsafe real equipment, 5) critical skill and knowledge must be developed, 6) training conducted at remote sites, 7) low availability of instructors, and 8) need for high volume of recurrent training. One legitimate way of summarizing this list of reasons for using computer-assisted instruction, is to present it as a search for means to overcome limitations in resource availability. Intelligent tutoring systems are one of the tools for overcoming limitations in resource availability.

Following is a list of important properties of intelligent assistants [Hoschka, pp. 2-3]: 1) domain competence, 2) assessment competence, 3) learning and adaptive behavior, 4) processing imprecise instructions, 5) explaining abilities, and 6) cooperation support. Each one of these properties is crucial to the success of ITSs. The natural language interface – and the study of acknowledgments – enhances an ITS's performance in five of the six properties listed above: assessment competence, learning and adaptive behavior, processing of imprecise instructions, explaining abilities, and cooperation support.

1.2.3 CIRCSIM-Tutor. Several CAI programs have been developed in the Physiology Department at Rush University. Rovick and Brenner, in 1983, developed HEARTSIM on PLATO to assist students in understanding blood pressure regulation. Joel Michael and Allen Rovick developed another program, CIRCSIM, in 1986, to teach the physiological algorithms that allow comprehension of blood pressure regulation. Shim [1991, p. 2] explains that although CIRCSIM was a "great advance in terms of Computer Aided Instruction systems," it had to work with rigid tutoring plans and was not capable of dynamically adjusting itself to the student. Michael and Rovick established a joint project, CIRCSIM-Tutor, with Martha Evens at the Illinois Institute of Technology to develop a system that would overcome these problems by adding a student model, mixed-initiative dialogues, and adaptive instruction.

The implementation focus of my research is the ongoing development of the CIRCSIM-Tutor interface.

1.3 Significance of Research

The Intelligent Tutoring Systems community agrees that we need more analysis of dialogue and the area of acknowledgment has barely been touched. This study has important implications for student modeling and other forms of user modeling. My knowledge of computer communications has helped to shed light on this issue.

1.3.1 Need for Empirical Study of Acknowledgments in Human Tutoring. Although various researchers have studied acknowledgments in human tutoring or teaching (e.g., Sinclair and Coulthard [1975], Fox [1993], and Novick and Sutton [1994]), and have made some published statements about empirical research on acknowledgments (e.g., Spitkovsky and Evens [1993]), more primary research describing existing tutoring acknowledgment behavior is needed [Graesser, 1996a]. My proposed study will help advance research in human tutoring discourse by adding empirically-based descriptive knowledge. This in turn may assist research in normative tactics for tutorial discourse.

1.3.2 The Loss of Bandwidth Issue in Student Modeling. Kurt van Lehn [1988] states that "a characteristic shared by many ITSs is that they infer a model of the student's current understanding of the subject matter and use this individualized model to adapt the instruction to the student's needs" [p. 58]. He goes on to describe a "diagnostic module" (elsewhere in the literature also called the student modeler) that attempts to infer the student model by analyzing the student's "observable behavior" [p. 58].

Van Lehn rates student modeler functionality in terms of a qualitative hierarchy based on the student modeler input. The student modelers are ordered on the basis of the observable behavior (bandwidth) used to make inferences about the student model in terms of bandwidth. The highest bandwidth corresponds to behavior that allows access to approximate mental states, medium bandwidth allows access to intermediate mental states, and the lowest bandwidth allows access only to final mental states. [pp. 58-59]. An example is the observation of a chess player doing think aloud playing, versus seeing the player testing moves by moving pieces around on the board, versus seeing only the final actual move (the case with playing chess against a computer). A serious difference between human tutors and an ITS lies in the amount of observable behavior, or available information bandwidth. For instance, just in the area of verbal behavior, humans have access to the speech intonation, rate, and stress behaviors jointly titled prosody ("a collection of phenomena relating to how sentences are spoken rather than what is spoken" [Allen, 1995, pp. 625-627]), whereas ITSs are hard pressed just to identify what is spoken.

I believe that we are losing information available to us in the input and wish to assist in developing bandwidth enhancement techniques for natural language interfaces. If we succeed in enhancing the available bandwidth, this added information should permit us to perform more sophisticated student modeling.

1.3.3 Enhance the Natural Language Capability of ITSs. Just as ITS capabilities in general are still quite limited [Woolf, 1990], so also is their Natural Language processing power. Current systems can generate separate utterances, but they are limited in discourse capabilities. One person who has contributed significantly in this area is Alison Cawsey at the University of Edinburgh. In a paper titled "The Structure of Tutorial Discourse" [Cawsey, 1989a, p. 1], she says:

> most of this work has concentrated on the pedagogic issues of topic control ... rather than on the linguistic structure of the text, or the detailed structure of interactions. In Woolf's thesis work, for example, a typical terminal node in her discourse structure is 'describe general knowledge'. Yet linguists have shown that a description has much internal structure, and may involve interactions with the user within the description.

She continues on to describe other work as better, but still inadequate, and presents a discussion on how all utterances are part of the overall "tutorial discourse." In other papers, such as "A Computational Model of Explanatory Discourse: Local Interactions in a Plan-Based Explanation" [Cawsey, 1989b], "Repair Work in Human-Computer Dialogue" [Cawsey and Raudaskoski, 1990] and "Understandable Explanations: The FUDGE Discourse Generator" [Cawsey, 1990], Cawsey continues the process of examining tutor-generated discourse.

My work assists in enhancing ITS natural language capabilities by providing additional knowledge regarding the generation and recognition of acknowledgments.

1.3.4 Contribute to Discourse Analysis by Applying Knowledge of Acknowledgments in Data Communication Protocols. As discussed earlier, studying language is quite difficult. This is partly due to the degree of fluency attained by researchers. This fluency makes many aspects of language so automatic as to become essentially invisible. In contrast, the domain of artificial communications protocols – specifically, data communication protocols – is very carefully designed and documented. Acknowledgments are critical to successful data communications, and are consequently well understood. Studying artificial communications protocols appears fruitful as a source for finding mechanisms that might also apply to human communication.

1.4 Goals

1.4.1 Empirical Study of Acknowledgments in Human Tutoring. I use discourse analysis techniques such as those detailed in [Sinclair and Coulthard, 1975] and [Sinclair, 1993] to carry out the coding (tagging of interesting information in transcripts) and study of transcripts of human keyboard-to-keyboard tutoring sessions. An advantage of studying keyboard-to-keyboard sessions is that transmitted information is largely restricted to the typed text, a limitation that closely parallels current ITS limitations and simplifies the overall study. Conclusions based on these transcripts should be more applicable to ITSs than transcripts of face-to-face tutoring sessions.

1.4.2 Generating Acknowledgments for the Tutee. From this study of human transcripts, I have built a model of that tells when to issue an acknowledgment, what to issue, and how to issue the acknowledgment.

1.4.3 Involving the Modeler. My work has many implications for student modeling. This study of acknowledgments has shown that we need more information from the student model. Expert tutors make use of a wide range of information about the student in delivering acknowledgments. This might even include some gross representation of the student's affective state.

1.5 Organization of the Thesis

Chapter II is an overview of previous research in intelligent tutoring systems, including CIRCSIM-Tutor. Chapter III presents a number of different approaches to the problem of coordinating communication, including the study of both natural language and artificial communications. It also proposes the idea that some forms of communication coordination are inherent in the nature of information. Chapter IV discusses several experiments in recognizing and categorizing acknowledgments and related phenomena. Chapter V presents a model of communication coordination that is suitable for implementation in intelligent tutoring systems. In Chapter VI, there is a short discussion of how this model for communication coordination depends on the system's ability to build a pedagogical model of the student. Chapter VII addresses the role of coordination in Circsim-Tutor's interface module. Chapter VIII contains my conclusions and addresses future research.

CHAPTER II

INTELLIGENT TUTORING SYSTEMS

2.1 Tutoring

The dictionary defines tutoring as "To act as a tutor to; to teach or instruct, esp. privately," and a tutor is defined as "One employed to instruct another privately in some branch or branches of learning; a private instructor" [New Webster's Dictionary, 1981]. Tutoring is teaching with the distinguishing feature that it is private – or personal – instruction. It is generally viewed as an opportunity for superior learning.

Forms of tutoring range from the very formal setting of a private tutor to the extremely informal sessions of a parent attempting to teach a child how speak. The essence of tutoring is one-on-one instruction characterized by significantly more interaction than is normal in a typical teaching environment. A great degree of learner engagement and the ability to probe the learning and provide immediate feedback are among the reasons that tutoring is so successful.

This chapter touches on the history of tutoring, and then reviews the endeavor to build electronic, computer-based tutors, which try provide the benefits derived from instruction by human tutors. It considers the possibility of developing tutoring systems that are capable of tutorial dialogue with humans, and in particular, the CIRCSIM-Tutor project.

2.1.1 The History of Tutoring. Tutoring has been a part of learning as far back as our records of education go. One of the best known ancient tutors was Socrates, who believed that the best way to attain knowledge was through a disciplined conversation that he called dialectic. Socrates believed that by this process of "intellectual midwifery" (his term), by asking questions, "by progressively correcting incomplete or inaccurate notions, one could coax the truth out of anyone" [Stumpf, 1982, p. 37]. We still speak today of this process as "Socratic dialogue."

At times in history, the popularity of tutoring reached the point that it became a paradigm of instruction in mass education. For instance, the Bell-Lancaster system in early nineteenth century England was one of the dominant forms of education [Allen, 1976, pp. 12-17]. This extremely successful learning system was based on using children to teach each other. Many Bell-Lancaster students showed spectacular progress in learning.

Tutoring is used extensively today as a "learning supplement." For example, our own institution, the Illinois Institute of Technology, has tutors and teaching assistants who assist other students with their difficulties in solving problems and mastering concepts.

Woolf [1990] cites claims of learning improvement of one standard deviation for tutoring by ITSs compared to traditional teaching, vs. improvements of two standard deviations for tutoring by human tutors.

My purpose here is not to try proving the benefits of tutoring; it is accepted by the education community and the effectiveness of many different types of tutoring has been demonstrated through numerous studies. (A good source of information and research on tutoring is the 1980 book by Ehly and Larsen, <u>Peer Tutoring for Individualized Instruction</u>.) My purpose is to start with the premises that tutoring is rightfully an important part of learning, that quality feedback and learner engagement are among tutoring's "active ingredients" responsible for effectiveness, and lead the reader to consider a few ways in which we might enhance the effectiveness of computers as tutors.

2.2 Intelligent Tutoring Systems

A lot of effort has been expended developing computer systems that can tutor effectively, and then trying to improve on those systems. My goal in this section is to consider some of the arguments for this work, and then to review some of the history of computer-based tutoring.

2.2.1 Rationale for Computer-Based Tutoring. Robert Taylor in the introduction to the book <u>The Computer in the School: Tutor, Tool, Tutee</u> [Taylor, 1980], argues that the computer can take on the three roles of tutor, tool, and tutee. In the tutor mode, the computer is presented as a flexible tutoring system, but one which is relatively crude and limited with a high ratio of hours spent in preparation to hours of instruction. The "computer as tool" refers to its passive role supporting various tasks such as calculation, statistical analysis, and word processing. By speaking of the computer as tutee, Taylor refers to various forms of programming whereby students teach the computer how to perform various tasks, e.g. tutoring other students, playing games, and drawing maps. He argues that successfully teaching the computer to perform a task induces higher quality learning. Intelligent tutoring systems would be categorized primarily as tutors, but it seems that a well designed ITS could function in all three modes, whether it operated in the different modes at different times, or even in all three simultaneously.

2.2.2 The History of Computer-Based Tutoring. Even though computers are a recent development in our history, there have been numerous efforts to add them to our set of educational tools. These have ranged from simple page turning systems to more ambitious projects such as Plato, and CIRCSIM-Tutor, the project in which I am currently involved.

Chambers and Sprecher [1984] present an overview of computer-assisted instruction (CAI) from the late 1950s through 1980. CAI came to be used as a learning supplement, especially for various forms of computerized drill. In many cases, whole courses were developed, based on CAI. PLATO, the best known CAI system, provided a custom authoring language called Tutor that supported learner directed learning as well as traditional lesson plan sequencing. Chambers and Sprecher believed that CAI showed promise; the first two points in their summary of CAI effectiveness evaluations are 1) "The use of CAI either improved learning or showed no differences when compared to the traditional classroom approach." and 2) "The use of CAI reduced learning time when compared to the regular classroom" [p. 12].

The time line for intelligent tutoring systems frequently starts in 1970 with the publication of Carbonell's Mixed-Initiative Man-Computer Instructional Dialogues [Carbonell, 1970].

Beverly Woolf, in "20 Years in the Trenches: What Have We Learned" [Woolf, 1990], compares the advances in the research and engineering of intelligent tutoring systems to that of advances in architecture over the centuries. Woolf presents a picture of slow but steady trial and error engineering enhancements of ITSs. A number of problems have been solved, or turned out to be nonproblems, and even though a number of problems remain to be solved (including "[effective inclusion] of student models, qualitative reasoning, machine learning, hypertext, or multimedia" [p. 243]), she argues that engineering will solve these in time, just as advances in architecture resolved their unsolvable problems. A number of concerns that were thought to stand in the way of success have been resolved. For instance, it was thought that the systems would be too expensive to build. Woolf states that experience from the 1980s showed that "once a framework is established, several tutors can be produced in a rather efficient manner" [p. 241]. She also addresses the question of preparation time needed and quotes Anderson's suggestion of 60 person-hours of preparation time per hour of training [p. 242]. One other allegation was that intelligent tutoring systems would be no more effective than traditional teaching methods. In response to this, Woolf says [p. 242]:

> In fact, Anderson has shown these systems can be one standard deviation more effective than lecture-style teaching methods (Anderson, 1988). For example, in a programming course at Carnegie-Mellon University, an intelligent tutor improved learning results, as measured by test performance, by 43% and reduced learning time by 30%. Using traditional lecture-style environments, students spent about 40 hours covering the first six lessons of a LISP course. Using the intelligent tutoring system and the lectures, students completed the lessons in only 15 hours.

It is a mark of the progress in ITSs that she complains "one standard deviation learning improvement for intelligent tutors as compared with lecture-style teaching is not good enough" [p. 243].Among the problems Woolf identifies, my work primarily addresses the issue of advances in discourse, especially ITS response selection and generation [pp. 247-248]. A secondary benefit is enhanced support for student modeling.

2.2.3 Issues in ITSs

- **Architecture**. The typical ITS consists of four primary groupings of functionality [Burns and Capps, 1988]: 1) the expert module contains information about the problem domain (the knowledge base) and can perform various operations within that domain, such as solving problems, 2) the tutor module — an expert system in the pedagogy domain — plans sessions at various levels of detail and handles other teaching/learning functions, 3) the student module is responsible for building a model of what the student knows and is capable of performing, and 4) the HCI interface and working environment do the actual communication with the user. My work relates primarily to interface issues, especially the natural language interface.

- **Research Grounded in Application**. Research into, and discussions of, theory are essential; without a good theoretical grounding, we tend towards producing ad hoc solutions of limited use. However, it is my belief that trying to implement theories into real live systems is profoundly healthy for the theory. Success strengthens theory by adding support to the plausibility of the theory. Failure strengthens theory by encouraging further analysis of unsuccessfully implemented theories and by promoting survival of the fittest theory.

Furthermore, many people pay little attention to primary research and theoretical activity. Presenting working ITSs should promote further interest in implementing systems and in supporting the theory and primary research behind them.

• **Tests of Intelligence**. Burns and Capps [1988] propose that ITSs must pass "three tests of intelligence" [p. 1]: 1) they must become good enough experts in the problem domain of instruction to reason and solve problems within that domain, 2) they must be capable of deducing the student's relevant knowledge within that domain, and 3) the system's teaching abilities must be effective enough to raise the student's performance towards the system's own performance level within the problem domain.

2.3 Natural Language Interfaces

So why are natural language interfaces important for intelligent tutoring systems? A good answer to that question could fill a book by itself, but let me start with a quote from the official CIRCSIM-Tutor history document [Michael and Rovick, 1996]:

> By the late '80s we realized that while CIRCSIM was an effective learning resource for our students, there were many kinds of errors and misconceptions that it was unable to remedy, in part, because it could not detect their presence.
>
> This conviction arose from the experience of tutoring students using CIRCSIM and observing that even when few if any errors were made, conversation with the students nevertheless revealed significant problems with their understanding of the baroreceptor reflex.
>
> It seemed to us that the solution to our problem would be the availability of a program that could hold a conversation, a dialogue, with student while they were engaged in solving a CIRCSIM problem.

This answer proposed that certain aspects of teaching, especially in tutoring, require better communication capabilities than the typical "ask a question, choose a response from among those listed, provide feedback on the response" or "fill in the blank" interaction. Achieving more sophisticated tutoring requires more sophisticated communication.

Another part of the answer is that this is basic research in natural language understanding and generation. This is one of the greatest areas of research that we face. The task is daunting, but the rewards are significant. Any work that advances our understanding of natural language, and improves our ability to build systems with natural language capabilities, is inherently worthwhile.

A third answer is that we would like to have a testbed for studying the effects and effectiveness of different tutoring protocols. A serious experiment requires that the tutors follow the protocols meticulously, but it can be hard to get humans to be sufficiently consistent. An intelligent tutoring system, with sufficient natural language ability, could be programmed to use any one of multiple protocols, and it would probably follow those protocols more consistently that most humans. This would permit a rigorous comparison of the pedagogical value of different tutoring protocols.

2.3.1 Text Generation: Canned Versus Dynamically Generated Output. Canned text is text that is built into the program as a set of strings that are issued based on a determination of what output is needed at a given time. An example would be the string "No, heart rate does not increase, it decreases." A more sophisticated implementation might store this as a template with slots for the variables. For instance, "No, $variable does not $change1, it $change2", when combined with the parameters 1) heart rate, 2) increase, and 3) decreases, would produce the same sentence as in the previous example. The programmer would write code to call the function with different parameters, depending on what output was desired. The problem is that this rapidly breaks down due to the complexity of real language. Supposed I wished to have the system say "No, neither heart rate nor inotropic state changes." It is clearly impossible to use the template above to generate this utterance, so a new template must be built. If the system developer wishes to be able to generate a wide range of utterances, this need for new templates continues until there is a large number of these templates, and even then there will be many utterances that cannot be generated.

We would like to be able to issue arbitrary "content specifications", rather like propositional statements, and have the text generator produce context-coherent dialogue.

In other words, we would like the underlying reasoning system, the planner, to issue something like "(negative-acknowledgment (change heart rate) (change inotropic state)" and have the text generation subsystem produce say "No, neither heart rate nor inotropic state changes."

This is a serious challenge. To the best of my knowledge, there is no existing tutoring system that produces real-time, totally dynamically generated output. CIRCSIM-Tutor does this successfully to a degree, but this reaching this goal will remain a challenge for the foreseeable future.

2.3.2 Input Understanding: Constrained Versus Unconstrained Input. Another natural language challenge that CIRCSIM-Tutor takes on is the processing of unconstrained input. By "unconstrained", I mean that the user is given a prompt and unlimited space for typing, and is permitted to enter absolutely anything. Most intelligent tutoring systems allow the user to select from among several choices by clicking on one of the choices. This keeps input processing very simple. A slightly more sophisticated system will permit the user to type in a word, a phrase, or an equation. This is more difficult to process, but it is still much simpler in that the questions are constructed so that there are usually only a few possible answers; the input understander only has to match it against the set of "official" answers to determine correctness, and to judge the answer correct, it usually demands an exact match against one of the official answers.

CIRCSIM-Tutor takes on a bigger challenge than these other systems when it permits unconstrained input. It tries to accept as valid any answers that a human tutor would accept as valid. That includes recognizing close synonyms, processing variant word order, performing spelling correction to compensate for the frequent typing errors, and judging whether to accept answers that are almost, but not exactly, right. To further complicate the task, the system has to deal with user initiatives. Imagine the following sequence of tutorial dialogue turns:

Tutor: "What are the determinants of cardiac output?"

Tutee: "How complicated an answer do you want?"

Tutor: "Just name those prediction table variables that determine cardiac output."

In this case, the tutee did not answer the question asked. Instead, the person initiated a new topic with a request for clarification of the original question. An intelligent tutoring system with unconstrained input has to deal with this sort of input as well, a very difficult task.

2.4 CIRCSIM-Tutor

2.4.1 History of CIRCSIM-Tutor. Researchers in the Physiology Department at Rush University have developed several CAI programs to enhance the learning of their medical students. Rovick and Brenner, in 1983, developed HEARTSIM for PLATO to assist students in understanding blood pressure regulation. Michael and Rovick developed another program, CIRCSIM, in 1986, to teach the physiological algorithms that allow comprehension of blood pressure regulation. Although CIRCSIM was a great advance in terms of Computer Aided Instruction (CAI) systems, it had to work with rigid tutoring plans and was not capable of dynamically adjusting itself to the student. Michael and Rovick established a joint project, CIRCSIM-Tutor, with Evens at the Illinois Institute of Technology to develop a system that would overcome these problems by adding a student model, mixed-initiative dialogues, and adaptive instruction. For more detail on the history of CIRCSIM-Tutor, see [Michael and Rovick, 1996].

2.4.2 Domain & Domain Issues. The goal of the tutoring is that the students develop a usable understanding of blood pressure regulation by the barroreceptor reflex. This reflex is activated as blood pressure increases or decreases. The central nervous system is notified and commands changes in parts of the blood circulatory system in order to maintain steady blood pressure. Unless the blood pressure perturbing factor overwhelms the system, this mechanism will maintain an approximately steady blood pressure.

The students are presented with a model of the blood pressure system that abstracts the response mechanism as the changes in seven variables measured during three phases: 1) Direct Response (DR), the change in the variables induced by the perturbation, 2) Reflex Response (RR), the change induced by the central nervous system intervention, and 3) Steady State (SS), the change in the variables relative to their values before the perturbation. The seven variables are as follows:

1. Inotropic State (IS) – the ion state (ion density) in the heart muscle. It determines the heart muscle contraction force.

2. Central Venous Pressure (CVP) – the pressure of the blood returning to the heart.

3. Stroke Volume (SV) – the volume of blood pumped out of the heart per stroke. SV is determined by CVP and IS.

4. Heart Rate (HR) – the number of heart beats per minute.

5. Cardiac Output (CO) – the volume of blood pumped out of the heart per minute. CO=SV*HR.

6. Total Peripheral Resistance (TPR) – a measure of resistance to blood flow.

7. Mean Arterial Pressure (MAP) – blood pressure in the arteries. MAP=CO*TPR.

2.4.2.1 Information about the Tutoring Protocols. A tutoring protocol is a set of rules about how tutoring is performed. The protocol our tutors currently use is as follows:

Solve the Direct Response (DR) stage
Collect & tutor the primary variable
First collect the student's prediction for primary variable and its value

Then tutor the primary variable and/or its value as needed
[Student must get this right before proceeding]

Collect & tutor the rest of the prediction table variables for DR
First collect all the student's predictions for the rest of the variables
After all the predictions are collected, tutor those predictions as needed

Solve the Reflex Response (RR) stage
Collect & tutor all the prediction table variables for RR
First collect the student's predictions for all prediction table variables
After all the predictions are collected, tutor those predictions as needed

Solve the Steady State (SS) stage
Collect & tutor all the prediction table variables for SS
First collect the student's predictions for all prediction table variables
After all the predictions are collected, tutor those predictions as needed

It is important to understand that under the current protocol the tutor carefully avoids making any judgment about the predictions until all predictions have been collected; the exception is the primary variable in DR. Consequently, the tutor's utterances are intended as judgments only when collecting the primary variable in DR and during the tutoring phase in each stage after collecting all the predictions. By contrast, closure and repair can and do occur at any point during the tutoring session. This restriction on judgment is one of the main ways in which the current protocol (Protocol 3) differs from the previous protocols. During the tutoring phases, the tutor may also go fishing for possible problems in the student's understanding. For more details about the three protocols see [Khuwaja, 1995].

2.4.3 Previous Work. Members of the CIRCSIM-Tutor project have studied different aspects of tutoring language. Hume [1995] studied hinting; Shah [1997], studied student initiatives and the tutor's responses; Sanders [1995] demonstrated the need for multiturn discourse planning; Freedman [1997] developed rules to describe discourse generation at several levels.

CHAPTER III

DEVELOPING A THEORY OF ACKNOWLEDGMENT

I started my study of acknowledgments in the laboratory rather than the library. By this I mean that rather than proceeding to the library to study the many branches of linguistics, behavioral sciences, and other disciplines, I sat down with tutoring transcripts on one part of a table and a computer with a running word processor on another part of the table, and started reading through the transcripts while performing type aloud thinking into the word processor. This approach to research poses the danger of inducing wild flights of speculative fancy in the mind thus unencumbered by the accumulated knowledge available in each discipline. On the other hand, it also offers the possibility of performing uncontaminated thinking – based on real data – that leads to a set of beliefs which are owned by the thinker and form a basis for reacting to the existing research and belief sets. Another advantage of this uninformed thinking is that the researcher has what is sometimes described as a "sufficient level of ignorance" to ask and follow some of the right naïve questions – questions that a more sophisticated thinker might discard, a priori, as invalid, silly or unproductive. While doing this preliminary thinking, I was impressed by the complexity of conversational behavior. For instance, the minimal utterance "OK" can mean "I acknowledge hearing what you said," "I understand what you said," or even "I agree with what you said." Then there are the questions about why the speaker uttered the string "OK," rather than any of the other strings that could have been chosen instead. It became progressively apparent to me that if I wanted a small field where I could do some quick research, wrap up a nice clean answer and call the case closed, this was not the place to be. However, I also concluded that providing even a modest effort in helping advance our knowledge of this very complex topic was worthwhile.

Once I had developed some ideas about what was happening with acknowledgments, it was then time to start doing library research to discover what other people had studied and concluded about this area of research. This chapter is about some of the discoveries from the library side of my research, and my reactions to what I read. The next chapter, Chapter IV, is about some of the applied research that I did, going back to the laboratory to attempt to get a better grasp on what was actually happening in tutorial dialogue, and what it meant.

I struggled a lot while writing the research chapter because I kept trying to introduce all the interesting authors and ideas that I had encountered. In the end, I went through and ruthlessly eliminated all but a few authors who seemed to bring the clearest contributions to my study of acknowledgments.

The following sections cover subdisciplines or areas of inquiry that have influenced my analysis of acknowledgments in human-to-human tutoring transcripts.

3.1 Linguistics and Discourse Theory

This section of the chapter presents a quick overview of some of the work in linguistics and discourse theory that I consider useful for the study of tutorial dialogue and acknowledgments.

3.1.1 Conversation Analysis. On the library side of my studies, I discovered Conversation Analysis (CA). A good introduction is presented in a book with the promising title, <u>Computers and Conversation</u> [Luff et al., 1990]. In the chapter "On the Analysis of Interaction: An Introduction to Conversation Analysis," Robin Wooffitt [1990] introduces a subfield of language studies that seems highly relevant to my study of tutoring discourse behavior.

According to Wooffitt [pp. 10-13], CA started out in the social sciences in the 1960s as Harvey Sacks worked with recordings of incoming telephone conversations at a suicide prevention center. He noticed a case in which, when the support person identified himself, the caller demonstrated a sudden apparent difficulty in hearing. Sacks came to wonder whether the caller was using this behavior to avoid giving his name in response. This led to further investigation of the ways in which speakers use utterances to achieve goals in communication with others. Sacks decided that human utterances are carefully crafted to some purpose and his goal was, as Wooffitt says, "to describe these procedures: to explain how a speaker came to use these words, in this way, on this occasion." [p. 12]

Wooffitt also points out that Sacks was very careful to clarify that he was describing verbal behavior and not delving into any analysis of putative psychological processes behind the verbal behavior [p. 12]. "CA deals primarily with regular patterns of conversational behaviour, and the object of analysis is to describe the systematic structural characteristics which underpin particular phenomena." [p. 20]

Harvey Sacks and his colleagues, Emanuel Schegloff and Gail Jefferson, established no codified set of prescribed methods. Wooffitt explains [p. 35]:

> Conversation analysts try to describe the methods which *people themselves* use to make sense when they talk to each other. Analysis of empirical materials proceeds without theoretical speculation as to the nature of conversation.

He adds that one cannot simply apply CA "and the results come flooding out" [p. 35], but that it is the result of "rigorous and painstaking examination" [p. 36]. This makes sense in that numerous other domains have problems that cannot be solved by numeric brute force (e.g., Ackermann's function in mathematics [Taylor, 1998, pp. 233, 298-299]), but which require careful analysis and thought before achieving results.

An important part of the CA tradition is the practice of always presenting the original data when discussing any results. This public availability of data helps ensure the validity of the conclusions. I believe that this practice is important and will endeavor to follow it by including significant amounts of original data in the appendices.

The relevance of Conversation Analysis to my studies is that it presents examples of how to study tutorial discourse, but more importantly, a number of researchers working in tutoring draw on these ideas (e.g., Barbara Fox, [1993]).

3.1.2 Discourse Analysis. Sinclair and Coulthard [1975] speak of how the mainstream of linguistics then tended to consider only problems within clauses, focusing on phonology, morphology, and intraclausal grammar. Sinclair and Coulthard state that though these studies brought significant linguistic progress, they were not a complete study of language. In particular, they generally ignored the study of language above the "rank of the clause" and paid little attention to context, both verbal and non-verbal. The transformationalists, in particular, performed little interclausal study.

In contrast, Sinclair and Coulthard were more interested in superclausal language, and paid a lot of attention to the verbal and non-verbal context. They mention how the path was opened for the discourse analysis by the insistence of the Generative Semantics Movement — Ross, the Lakoffs, and McCawley — that context is needed to understand utterances. Sinclair went further and argued that it was not possible to really understand an utterance without a thorough knowledge and understanding of the context. Various people in England, including Coulthard, were unhappy with the linguistic complexity approach used for studying language in schools, and were developing alternative methods of testing the functional power of a child's linguistic abilities. Sinclair and Coulthard brought this together with emerging work in the analysis of conversation, resulting in a research project to study linguistic aspects of teacher/pupil interaction.

The upshot of Sinclair and Coulthardt's work was that they developed a fairly sophisticated system of analysis to describe schoolroom discourse, a system which should also be applicable to tutorial discourse. (They attribute much of their methodology to Halliday's <u>Categories of a Theory of Grammar</u>.) The basic hierarchy they used consists of the following structures [pp. 24-27]: 1) lesson, 2) transaction, 3) exchange, 4) move and

5) act (see Table 1). In effect, they developed a grammar that describes the structure of pedagogical discourse. For instance, the lesson is an unordered series of transactions; a transaction is described as an ordered series of exchanges of the form: Preliminary exchange, one or more Medial exchanges, and a Terminal exchange. Exchanges are classified as either Boundary (marking the boundary of a lesson part), or Teaching, which covers a number of sequences such as Initiation by teacher, Response by student, Optional Feedback, and so on.

Table 1. Levels and Ranks

Non-Linguistic Organization	Discourse	Grammar
course		
period	LESSON	
topic	TRANSACTION	
	EXCHANGE	
	MOVE	sentence
	ACT	clause
		group
		word
		morpheme

Although Sinclair and Coulthard do not quite go that far, it seems possible that a formal grammar of tutoring might be developed, one which describes the grammar rules by using notation similar to that used to describe formal languages, e.g., BNF or equivalent notations used in computation theory [Taylor, 1998; Lewis and Papadimitriou, 1981]. The complexity of discourse makes it unclear how specific such a grammar could get, due to the constraints imposed by context. Even if we can't develop a "unified general grammar of everything", more detailed descriptive grammars should be possible within restricted domains.

The significance to my research of this work in discourse analysis is twofold: 1) I believe Sinclair and Coulthard provide a way of looking at discourse that contributes to building a useful foundation for the study of tutorial discourse, and 2) many other researchers interested in tutorial dialogue (e.g., Alison Cawsey) come from the discourse analysis tradition and use it in their study of human discourse, and the application to computer generation and understanding of dialogue. Studying discourse analysis renders their research more understandable.

3.1.3 Joint Action Theory. Herbert Clark, in his new book, Using Language [Clark, 1996], claims that "Language use is really a form of *joint action*." [p. 3] and spends the rest of the book presenting a case for this statement. He doesn't devote much space specifically to acknowledgments per se, but the material that he discusses under other labels such as signaling [pp. 155-190] actually represents the use of embedded acknowledgments and presents a useful framework for studying both discourse in general, and acknowledgments in particular. Before going into detail on his contributions to my study, I wish to make two important observations about this book: 1) it is the most comprehensive description of the use of language that I have seen; indeed, several people who attended the 1996 annual conference of the Society for Text and Discourse – where Clark was the opening speaker – had recommended his work to me (which led to an email exchange with him that changed the direction of my research), and 2) Clark places an emphasis on language as a situated joint action, that is very useful when considering the linguistic context of tutoring.

One of his six foundational propositions regarding the use of language is "The study of language use is both a cognitive and a social science" [p. 24]. He argues that using language is like playing a piano duet; there are individual performers who are producing individual performances and can be studied that way, but one must also study the performers as a pair. Studying either activity without the other yields only a partial understanding of what is happening. Clark suggests that cognitive scientists (this would also include most linguists) tend to study the individual performance, without consideration of the joint coordinated activity, and that social scientists tend to study language mainly as a joint activity, without consideration of the individual performances.

Clark presents discourse as a joint activity composed of joint actions. He focuses on the mutual coordination of their individual actions by the participants in linguistic activity.

Following is a set of terms – relevant to my studies – that Clark uses, along with explanations as needed.

Table 2. Clark's Levels of Language

Level	Speaker	Listener
4	Proposes joint action	Takes up proposal
3	Signals that p	Recognizes speaker means p
2	Signals something	Identifies the signal
1	Presents signal	Attends to the signal

- **Levels**. Clark's model of linguistic joint actions states that there are concurrent joint actions taking place on four separate levels (summarized in Table 2). From the lowest level to the highest, they are: 1) The speaker presents a signal and the addressees attend to the signal. 2) The speaker signals something to the addressees, who identify the signal. 3) The speaker signals that *p*, and the addressees recognize that the speaker means *p*. 4) Speaker proposes a joint project and the addressees take up the proposal [pp. 148-151]. The action at each of these levels is performed in order to make possible the action at the next higher level, i.e., the speaker presents a signal in order to signal something, in order to signal that something, in order to propose something. The addressee attends to the signal in order to identify the signal, in order to determine what the speaker means to communicate, in order to be able to take up the proposal. To provide an example, if I wish to have someone fetch me a sandwich, I can generate a set of vibrations in order to generate the signals for the sentence “Please get me a sandwich”, in order to signal that I would like the person to fetch me a sandwich, in order to get that person to fetch me a sandwich. The intended recipient of the message attends to the sound I’m generating, in order to determine what words I’m trying to signal, in order to determine what it is that I wish to communicate, in order to be able to decide whether to take up my request and fetch me the sandwich as requested, or perhaps to decline the request.

- **Joint Action**. This is an activity that results from the coordinated actions between individuals.

- **Discourse Coordination**. "There is coordination of both *content*, what the participants intend to do, and *processes*, the physical and mental systems they recruit in carrying out those intentions" [p. 59]. Clark says that joint actions can be divided into phases – entry, body, and exit – and the phases are what get coordinated. Entries and exits are coordinated by syntactic, morphological, and intonational markers.

- **Closure**. A joint action is complete when there is a mutual recognition of closure on that action. "It is a fundamental principle of intentional action that people look for evidence that they have done what they intended to do." [p. 222] He restates Norman's *Principle of closure*: "Agents performing an action require evidence, sufficient for current purposes, that they have succeeded in performing it." [p. 222] and then introduces the *Principle of joint closure*: "The participants in a joint action try to establish the mutual belief that they have succeeded well enough for current purposes" [p. 226]. Clark uses summoning an elevator as an example of what evidence is available for obtaining closure [p. 222]. When a person wishes to summon an elevator, he presses the “up” button. The direct evidence that he has depressed the button is that he sees and feels the button getting depressed. If for some reason the button doesn’t appear to be moving and there is no other evidence, the person will probably try pressing the button again. But the person didn’t press the button just for the sake of depressing it. Rather, he did it in order to call the elevator. The standard evidence that the elevator has been successfully called is that the “up” button lights up. If the evidence is missing, the person will probably try running through the whole elevator call cycle again. Clark refers to Norman’s insistence that devices that fail to provide proper indications of closure are just asking for problems interacting with the users.

Clark then presents some characteristics of evidence of closure [p. 223]. 1) Validity: he says that the evidence must be “reliable and interpretable”. To return to the elevator example, the “up” button should illuminate every time the elevator button is correctly depressed and only when it is depressed. Regarding interpretability, the lighting of the “up” button is much stronger evidence of closure on calling the elevator than hearing some vague clicking coming from the elevator shaft. A person realizes that a reliable “up” button won’t light unless it was successfully depressed because it is clearly tied to the summoning of the elevator, but remains uncertain about how to interpret the vague clicking noises. 2) Economy of effort: Clark asks how useful the button would be if it only lit up for a part of a second, or only lit up if the person pressed the button for a precise amount of time. The first situation would require user attention at precisely the right time and the second is too complicated. 3) Timeliness: The “up” button should illuminate immediately upon being depressed, not seconds or minutes later. The person can’t achieve closure on the action until the evidence of success is presented and will generally not be pleased with delays in closure. 4) Optimal evidence: The best evidence normally is not the one the offers the strongest possible validity, economy and timeliness, because it is too expensive and inefficient to do that. Clark says the all we really need is “evidence sufficient for current purposes”. He also points out that what evidence is sufficient depends on the context; people will feel less need to work to get proof that an elevator has been successfully summoned, but will most likely be willing to put a fair amount of effort into determining that cleanup of a toxic chemical has been completed.

• **Least Effort**. Clark states the principle of least effort: "All things being equal, agents try to minimize their effort in doing what they intend to do" [p. 224]. He says that we try to minimize effort in everyday activities, so it comes as no surprise that we try to reduce the amount of effort required to verify the completion of an action. Also, many actions are complete only when a criterion is reached. These "criterial actions" are complete only when a criterion is met. He says "Agents cannot perform them without adhering to the principle of closure. An inherent part of doing them is deciding when they are complete" [p. 224].

• **Opportunistic Closure.** "Agents consider an action complete just as soon as they have evidence sufficient for current purposes that it is complete" [p. 224]. He says that one implication of this is that "if agents can treat an action as a criterial action, they will" [p. 224].

Definition of Acknowledgment. As far as I can tell, Clark doesn't actually produce a definition of acknowledgment, but depends on the reader having a sufficiently similar visceral understanding of the term. I may be putting words in his mouth, but I believe that an acceptable definition of acknowledgment in the context of joint action and levels of joint action, is the provision of proof of communicative success at the four levels. Clark does speak of acknowledgments [p. 231] that comment specifically on the understanding of an utterance (level three) and of other acknowledgments, which he calls "assessments", which are uptakes of assertions (level four). I believe that he would also support the suggestion that acknowledgments can be provided at level one (attending to the speaker) and level two (decoding the signal). However, we do not typically issue acknowledgments for all levels. The general answer to that has to do with the principle of least effort; people won't generally expend more effort than they need to on acknowledgments. The more specific answer lies in the power provided by the following two principles.

- **Upward Completion**. "In a ladder of actions, it is only possible to complete actions from the bottom level up through any level in the ladder" [p. 147]. In other words, in order to reach the goal of getting the elevator to come, I must succeed at pushing the button (level one) in order to depress the button (level two). This could fail if I pressed the wall instead (level one failure), or the button were stuck (level two failure). I might succeed at depressing the button (level two success), but fail to activate the mechanism because it is broken (level three failure). I might succeed at activating the call mechanism (level three success), but fail to get an elevator because the elevator was turned off (level four failure). Lastly, I might succeed in calling the elevator (level four success), but not get it to come because the doors were propped open (level five failure).

- **Downward Evidence.** "In a ladder of actions, evidence that one level is complete is also evidence that all levels below it are complete" [p. 148]. To return to the elevator example, if I try to push the button and the elevator doors open in front of me to reveal the elevator, I don't wonder whether I managed to push on the button, depress the button, activate the call mechanism, or call the elevator.

- **Application to Acknowledgments**. I believe that all of the above principles and ideas apply to acknowledgments. The following applications are particularly worth noting.

- **Downward Evidence of Positive Acknowledgments**. The general principle of downward evidence applies to acknowledgments, and to positive acknowledgments in particular. In other words, a positive acknowledgment at level three (the listener understood what was being signaled), implies a positive acknowledgment at layer two (the listener recognized/decoded the signal) and a positive acknowledgment at layer one (the listener was paying attention). This makes sense in view of the principle of least effort. It is generally not possible to succeed at the level of uptake on the joint action (level four), without having understood the signal (level three), having decoded the signal (level two) and paid attention to the signal (level one). Consequently, a positive acknowledgment at one level also functions as an implicit acknowledgment at all lower levels. It is worth noting that this implies that positive acknowledgments should be issued at as high a level as possible to obtain maximum closure on lower levels and thus maximize efficiency.

• **Upward Evidence of Negative Acknowledgments**. Just as efficiency implies that acknowledgments should be issued at the highest possible level, so it also implies that negative acknowledgments are issued at the lowest possible level. In normal situations (e.g., not performing face saving), if the listener were unable to perform an uptake on the joint action (level four) because of a failure to pay attention (level one), it would not make sense to say "I disagree" or "I refuse" (level four), or “I don’t understand what you mean” (level three), or “I couldn’t figure out what you said” (level two), because saying that would lead the speaker to focus on resolving the problem at a level other than the level at which the problem really occurred.

While speaking of negative acknowledgments, it is also worth pointing out that the negative acknowledgment should be relevant. It would be inefficient to signal a level one problem ("I'm sorry, I wasn’t paying attention?") if the problem was one that could not be solved by repeating the signal (e.g., the listener cannot decode one of the words used).

3.1.4 Metacommunication. Clark (and others, e.g., Stubbs [1993]) divide communication into two tracks or channels. The first track is the one in which the business is transacted, such as the tutor requesting that the tutee make a prediction about the value of cardiac output and the tutee responding. The second track is the metacommunication track, the one in which the participants monitor the success of the communication, make adjustments in the communication parameters, detect and signal communication problems, and attempt to fix communication problems. Stubbs states that metacommunication is for "attracting or showing attention, controlling the amount of speech, checking or confirming understanding, summarizing, defining, editing, correcting, specifying topic" [pp. 50-53].

Acknowledgments are metacommunication – communication about communication, in the second track – not communication in the primary communication track.

3.1.5 Repair. When people recognize that there has been some failure in the communication, they typically start trying to fix the problem so that understanding is achieved. This process is called repair. Hirst et al. [1994] identify classes of situation where repair is needed: 1) misunderstandings and 2) non-understandings, and present models for how these problems are solved. They argue that though computers have a far from perfect command of language, we should not despair. They point out that humans also evidence a far from perfect command of language, but succeed in communicating anyway because of their successful techniques for dealing with failures in communication. Therefore, rather than trying to get computers to understand flawlessly, we should be developing models and techniques to allow computers to deal flexibly with these failures when they happen.

Hirst et al. define non-understanding (or not understanding) as "a participant's failure to find any complete and unique interpretation of an utterance" [p. 215]. The participant cannot resolve the meaning of the utterance, is left with either no interpretation or multiple possible interpretations, and is confused. They note that "the participant is aware that it has happened" [p. 215]. The other type of failure to understand, misunderstanding, occurs when the participant determines an interpretation and does not realize (right away, anyway) that this is not the correct interpretation. Self-misunderstanding occurs when the person who has the misunderstanding is the one who recognizes the problem. Other-misunderstanding occurs when one person realizes that another must have a misunderstanding. Both types of misunderstanding are diagnosed when a participant is unable to fit an utterance in satisfactorily with other utterances.

Barbara Fox [Fox, 1986] has studied tutoring extensively. That includes studying repair. To make a point about the importance, she includes an extended quote from John Seely Brown [pp. 4-5] whom she calls one of the field's leading theoreticians:

> A dialogue involves constant conversational repairs between two people. When someone doesn't understand what I've just said, I must try to diagnose not only what, but why, he didn't understand and then accordingly repair what I said. It is basically an adjusting process that goes on between the two of us communicating.... We have been misled into thinking that natural language, per se, is so powerful. Instead, I think it is the dialogue process that is so powerful, e.g. the notion of conversational repairs that occur between two people. If we can understand this process and how to capture it in man-machine communication, we will have made a major breakthrough on the perceived friendliness of machines.

I believe that this is an example of the centrality of metacommunication to the whole process of communication.

3.1.6 Back Channel. This important communication mechanism is used when the listener signals information back to the speaker while the speaker is still speaking. When communicating with someone, we could take strict turns in communicating. However, that leads to wasted effort and time in many cases. For instance, suppose that I explain something to a student and the student gets confused right after the first sentence. Following the "strict turns speaking" rule, the student doesn't signal anything until after I have finished speaking, several minutes later. At this point the student issues a negative acknowledgment to indicate that most of my effort was wasted due to an early communication failure. I then have to go back, repair the failure, and will probably have to repeat much of the information that had followed after the point of breakdown. It would be much more efficient to discover earlier on that there was a problem. In normal speech, if there is a problem, the addressee will start signaling to indicate a problem while I am still speaking. Alternatively, the listener could use back channel signaling to send the speaker positive acknowledgments that will permit the speaker to know that all is well and keep on sending information without pausing the receive an assessment of whether the communication is still succeeding. The back channel signal can be either verbal or based on some other nonverbal mechanism. Fox [1993] emphasizes the importance of back channel responses in tutoring.

3.1.7 Discourse Markers. Discourse markers are words like “OK”, “well”, “but”, “so”, and “I mean”. In *Discourse Markers*, Schiffrin [1987] defines markers as “sequentially dependent elements which bracket units of talk” [p. 31]. This is a fascinating book about how the special role that these words play in discourse. My particular interest is in the idea that certain words play a special role in bracketing, or delimiting, units of speech. One way of looking at discourse is as a set of joint actions that each contains a set of subsidiary joint actions, each of which in turns contains a set of subsidiary joint actions, etc., which eventually are individual actions, all of which are units of speech. So anything that serves to delimit units of speech, should also serve to delimit, or bracket, joint actions. Specifically, a very high percentage of all acknowledgments occur at these boundaries. Although I am not studying discourse markers per se, the fact that so much discourse information can be carried by one individual word seems particularly important to anyone working on natural language systems. By including particular words in the generated text, and by scanning the input for particular words, we should be able to enhance the system’s ability to communicate successfully with humans. In Chapter IV, I discuss the study of words that are relevant to issuing and recognizing acknowledgments. In particular, these words are important for marking closure, judgment, and repair.

3.2 Information Theory

I am not going to try covering information theory in any depth, but do want to mention some ideas from information theory that I believe to be useful in studying discourse. What follows is based upon Gardner’s The Mind’s New Science [1985, pp. 16-22].

3.2.1 Information Theory Concepts. Norbert Wiener was a mathematician at MIT, who became very involved with servomechanisms (system guidance devices, e.g., used to guide anti-aircraft artillery and keep airplanes on course). He started thinking "about the nature of feedback and of self-correcting and self-regulating systems, be they mechanical or human" [p. 20]. Wiener paid attention to the work of McCulloch and Pitts, the early researchers of natural and artificial nervous systems, and saw an analogy between his research in guidance systems and nervous systems. He concluded that these self-correction, self-regulating systems, were the basis for a whole new science, cybernetics. Cybernetics was all about control and communication theory. What interests me is the idea that the principles inherent in control and communication are universal principles; whether one studies them in the context of the human nervous system, or robotic control, or data communication, or human discourse.

Another M.I.T. scientist, Claude Shannon, was an electrical engineer who became very interested in information and developed the foundational idea of information theory: "that information can be thought of in a way entirely divorced from specific content or subjective matter as simply a single decision between two equally plausible alternatives" [p. 21]. Combining this with Wiener's contributions led to the study of information separately from its embodiment in a transmission device. There are rules that apply to all information anywhere, no matter what the context or the mechanism. Of course, context and transmission device are relevant, but this does introduce questions about what communications mechanisms – and acknowledgments, of course – are inherent in communication. The study of such mechanisms instantiated in one communications domain should produce results that are potentially applicable in any other communications domain. This provides support for the meaningfulness of my consideration of the artificial protocols used in computer data communications, although my chronology was reversed; I first became interested in studying data communications to learn about human communications on the intuitive belief that it was relevant, and then later discovered information theory.

Following are several terms that I believe are worth exploring here.

• **Signal Encoding**. You can't really do anything with pure, raw information; it can't be transmitted or used, unless it is encoded somehow. Our universe abounds with information storage and transmission mechanisms.

• **Channels and Modalities.** The information that we are most familiar with has been encoded in various energy forms, such as audio waves, light, and electrical currents. Most computers use information that has been encoded in digital binary mode, but live, direct speech is analog. I will refer to these information carriers as channels. The term, modalities, is also used.

• **Bandwidth Capacity.** Different channels have different information bearing capacities. Shannon and Nyquist [Tannenbaum, 1989] were the originators of various formulae that can be used to calculate the bandwidth available for a given channel.

• **Metacommunication.** An important part of communication is the ability to communicate about the communication itself, in other words, metacommunication. This includes protocols for transmission (such as information encoding choices and information chunking), flow control, and error control. Although I cannot prove it and don't presently know whether anybody has tried to prove it, it is my belief that communication is impossible without metacommunication. The two primary reasons are as follows: 1) The very act of communicating information is a metacommunicative act in that the recipient is able to infer metainformation such as the communication channel's bandwidth, error levels, controlling protocol, and also infer information about the transmitting entity's properties. 2) There are communication systems which have no direct feedback (radio broadcasting, lighthouses), but such a systems appear to be extremely limited in terms of communication abilities.

• **Flow Control.** Given that maximizing efficiency and throughput are two of the goals of communication, it is imperative that there be some mechanism for regulating the rate of transmission. This is usually implemented by choosing what is deemed to be a non-optimal but acceptable signaling rate at the beginning of a communications session, and then increasing or decreasing the signaling rate based on the ability of the channel and the communicating entities to support the signaling rate. If the message is sent through faster than the channel or the recipient entities can reliably handle, then the transmission rate must be reduced; not reducing it would waste the "signaling opportunity" by using the opportunity to send information that will probably have to be retransmitted due to lost information. Similarly, sending information through at lower than possible rates also wastes "signaling opportunity". Dynamically adaptive systems will constantly monitor the communication to determine the equally dynamic, current optimal signaling rate.

• **Error Detection and Correction.** Error-free communication is like frictionless work, a hypothetical abstraction that is useful at times, but which does not describe reality. Any successful communication system must include mechanisms to cover both error detection and correction. There is a whole subfield of communication theory dedicated to the efficient and effective detection and correction of errors.

• **Positive/Negative Feedback.** This topic relates to all of the above. Feedback – studied in control theory – describes the process of feeding information related to the output of a control system, back into the control system as one of the inputs into its control decision process. Positive feedback systems use the feedback to produce direct stimulation of the system's output. If the input signal increases, the output signal becomes stronger. A noxious example of this is the feedback heard as a loud squeal on public address systems. Positive feedback is of less interest to me in communication, compared to negative feedback systems. That is because negative feedback systems are "goal-seeking". They take the feedback signal as an input to determine whether the system is making progress towards the goal, or away from it. If it is heading towards the goal, then it continues to generate the same output. If it is moving away from or past the goal, then it changes its output so as to start heading back towards the goal. A central heating system is a good example of a negative feedback system. If the temperature drops below the target temperature, then the heating unit is activated so as to return the temperature towards the target. If the temperature reaches or goes over the target, then the heating unit is turned off so as to return the temperature towards the target. Communication (and information theory) depends heavily on negative feedback in order to succeed. Acknowledgments are an important part of the negative feedback mechanisms in communication.

3.3 Data Communication

One problem that arises in studying language is that human adults have used language for so long and with such resulting expertise that by the time we turn to studying language, we discover that much of our use of language is automatic and largely unconscious. In order to study this expert behavior, we have to use the linguistic equivalent of knowledge engineering with the painstaking development of techniques for codifying the expert's linguistic knowledge. This process of extracting information from the expert turns out to be extremely difficult. For instance, the expert often overlooks relevant information until the knowledge engineer recognizes that some relevant information must be missing and presses the expert for the missing facts, chains of reasoning, etc. [Waterman, 1986; Gaines, 1990].

A supplemental approach is to find an analogous and more easily studied domain, perform the analysis, and then try applying the results to the original domain. Information theory (discussed earlier in this chapter) suggests that the mechanisms for communicating information during human-human tutoring are specialized instances of general information communication mechanisms. This suggests that artificial communications protocols developed by humans for electronic data communications might be a fruitful analogous domain for study. These protocols have the virtue of being recently, precisely, and tediously developed by humans, with the consequence that the mechanisms remain fairly apparent in our thinking and are well documented. Another virtue is that they involve implemented theory; flaws in the theoretical development tend to become painfully evident when introduced into a communications system.

A major advantage of studying computer communications in order to gain insight into acknowledgment protocols is that it is a field that is so young that we are still learning the requirements for successful and efficient communication, and consequently are very much aware of the necessary mechanisms. Another advantage is that we have had to explicitly develop the acknowledgment protocols as we develop hardware and software, and thus we tend to be much more aware of the mechanisms.

Based on my experience with the management, programming, and formal study of computer communications, I have developed a list of kinds of acknowledgments, which I have formed into a taxonomy of acknowledgments in communications. A detailed treatment of computer communications is beyond the scope of this document, but here is a summary of the ideas. [Tanenbaum, 1989] contains a lucid treatment of computer communications.

Dealing with computers is notoriously prone to error and failure. This is especially true of computer communications where there are constant problems with data being damaged, data getting completely lost, computers dropping connections, etc. To guarantee reliable communication, the developers of communications systems must verify the progress and success of all communication and do so rigorously. So when data is sent from one computer to another, unless a clear positive acknowledgment is received at some point, the worst is believed and the data must be retransmitted. There are ways to cut down on the overhead of constantly verifying the successful communication of information, but the bottom line is that ultimately all communication not unequivocally verified is generally considered to have failed. This leads to painstaking development of communications protocols and makes the study of computer communications very rewarding as a technique of studying communication in general.

I have found the following acknowledgment categories in computer communications:

- **Positive Acknowledgment**. When a computer sends data packets to another computer as part of a file transfer, the receiving computer will send back acknowledgment packets signaling safe reception and indicating precisely which packet (or packets) is being acknowledged. Positive acknowledgments can be used at the single packet level, or can occur at higher levels to mark closure on a unit of actions or mark closure on the whole session.

- **Negative Acknowledgment.** During a file transfer, it is possible that a packet became corrupt. Depending on the details of the protocol in use, the receiving computer will send back a signal indicating that a bad packet was received and precisely which packet is being discussed. A negative acknowledgment can also be sent when the receiving computer determines that it should have received the next packet by now, so it declares the packet dead and notifies the other system that the expected packet never arrived.

- **Delayed Acknowledgment**. Acknowledgments can get delayed for various reasons. One way is that the local network (the only available communication channel) is busy at the time that the computer wishes to send the acknowledgment. If so, it must wait until a chance to communicate is arises, so that it can issue the acknowledgment. In certain network topologies and using certain protocols, it is also possible that the acknowledgment got routed through a roundabout path that results in the acknowledgment arriving at the destination computer much later that would have been anticipated. A third category of possibility is that the acknowledgment reached a relay node and got stuck there for a while because of a high traffic load.

- **Lost Acknowledgment**. Another category of problem occurs when the receiving system sends an acknowledgment – either positive or negative – but this packet gets lost. The sending system times out on the acknowledgment after a while and determines that since it should have received an acknowledgment by then, but has not. It then needs to treat the lack of a positive (or negative) acknowledgment as an implicit negative acknowledgment and begin retransmitting, starting with the data right after the last positively acknowledged packet.

- **Acknowledgment Windows.** It turns out that sending a packet and then waiting for the acknowledgment before sending the next packet is very inefficient. The sending system cannot transmit anything from the time that the last bit of the packet leaves the system, while it traverses the signal carrier to the other system, while the other system processes the result, while the acknowledgment travels back, and until the received acknowledgment packet has been processed enough to determine that it is indeed the sought-after acknowledgment. So data communications programmers soon developed ways of allowing the sending system to keep on sending without having to stop and wait, unless there were serious problems with a breakdown in the communication channel. The idea is that the sending system may send up to *N* packets past the last acknowledged packet before it has to declare a problem situation and go back to deal with the unacknowledged packet. In this situation, an implicit positive acknowledgment is assumed pending arrival of the explicit acknowledgment. If the positive acknowledgment arrives before the sending system has sent it's "window" of *N* packets, it just increments the identifier of the last acknowledged packet and keeps transmitting. If the transmitting system sends *N* packets past the last acknowledged packet, then it has a couple of choices: 1) it could just stall for a bit and hope that the acknowledgment will show up, or 2) it can just assume an implicit negative acknowledgment and resend the packet. If the goal is to maximize throughput, option two is the better choice in most situations, and that is what the standard windowing protocols choose to follow. Depending on the size of the packets, the signaling rate, the transmission time, and other constraints, the number N will be adjusted to maximize the throughput.

- **Subsumed Acknowledgment.** In communications environments where the channels are largely error-free, further assumptions can be made to further reduce the metacommunicative overhead on the data transfers. One approach is to decide not to acknowledge every received packet, but to send out an acknowledgment for the most recently received packet every so many packets. For instance, the system might send an acknowledgment packet acknowledging every tenth packet. The underlying assumption is that when the transmitting system receives an acknowledgment for a particular packet, that means that all packets with a lower sequence number were safely received and can be considered acknowledged. The matching protocol rule states that the receiving station should send out a negative acknowledgment on the lowest numbered packet for which there is a problem. Thus when the transmitting system receives the negative acknowledgment, it knows that all packets were good prior to the one in question. A whole book could be written just on all the variations and considerations when designing windowing protocols and acknowledgment subsumption rules, but this is enough to give the reader a general understanding of the idea.

- **Link Keep-Alive Signal.** Certain protocols depend on knowing whether the link to another system is available at any given time, and also on keeping it available. If the other system has crashed, been turned off, or been otherwise rendered incapable of communication because the intermediate link has failed or been shut down, that can have important consequences. In particular, maintaining a link consumes resources and can be expensive, so the communication channels are liable to shut the link down if there is no observed traffic after a specified time interval. The solution is to send occasional packets through the link that have the dual purpose of keeping the link active, thus exempt from being shut down, and letting the receiving system know that the sender is still available for communication.

- **Request for Acknowledgment.** When a system is transmitting information and fails to receive expected acknowledgments, one action possibility is to send the other system a specific request for acknowledgment. Such a request can also be sent at the beginning of a transmission as part of establishing that a connection is possible, but this isn't usually done, because any response to a connection-request is implicitly an acknowledgment.

- **Flow Control.** Flow control is the regulation of the rate of information transmission. Failing to regulate transmission rates effectively results in a breakdown in communications due to overloading the channel or the recipient. It can also result in inefficient use of the communication channel due to wasted signaling capacity.

- **Protocol Negotiation.** Some combinations of communications channels and communicating devices result in many possible protocol combinations. The devices must determine both whether they can communicate, and if so, what is the most effective way of communicating. Modems are a good example of protocol negotiation. After a called modem answers, the two go through a protocol negotiation phase during which they try to establish a highest common denominator with respect to the available protocols.

- **Dynamic Link Adjustment.** It is not unusual for the communication link characteristics to change during the course of a communication session. If the link (a combination of the communicating devices and the information channel) quality worsens, the devices must either cope with higher error rates or else perform fallback to some lower common denominator. This could just mean a change in signaling rate, but could also entail changes in the encoding mechanism used. The goal is to maximize the throughput given the situation. On the other hand, the link quality could improve. If the communicating devices did not adjust the common denominator upwards, this would also represent inefficiency. Most modems have traditionally not performed much dynamic link adjustment, and of those that did, most would only adjust downward to cope with decreased link quality.

- **Error Correction.** This is critical for communicative success. It is possible to choose a protocol such that a combination of the information in a message unit, and possibly the information in the preceding and following message units, is sufficient to detect and correct errors. More commonly, error correction consists of issuing a negative acknowledgment and waiting for a new copy of the packet to be sent again. This represents the simplest protocol and the least error correction overhead in relatively error-free environments. In minimally acceptable contexts, however, it can become impossible to send packets through without errors. In those cases, error correction can provide the optimal efficiency in communication.

- **Checkpointing.** We can transmit a whole file, and when done, then send information that permits verifying whether the transfer was successful. The problem with this is that if it is determined that there was a problem, it is impossible to know where and why the problem occurred, so the whole file must be retransmitted. The probability of at least one error in the message is a function of the length of the message and the probability of channel signaling errors. As the length of the message increases and the probability of channel signaling errors increases, the probability of a whole message getting through without errors converges to zero. The idea behind checkpointing is to send smaller chunks of the message that are each verified independently, where the chunk size is chosen as a compromise between maximizing the throughput (which entails trying to minimize protocol overhead) and maximizing the probability of an error-free message chunk transmission. Each of these points where the communication success is verified is a checkpoint. If a problem is detected, instead of having to restart the transmission, all that happens is that the systems restart from the previous checkpoint.

- **Full Duplex Signaling.** Yet another efficiency mechanism is to perform signaling simultaneously in both directions (send and receive). Many early channels were half duplex. That is, only one system could be signaling at a time. This is something that we see even today on citizen band radio where only one participant may be transmitting at a time. To avoid collisions, protocols are invented such as the practice of saying "over" at the end of the speaker's turn. Most ethernet devices today still function in half-duplex mode. In half duplex, the transmitting device must stop to permit acknowledgments and other information to be sent back. In full duplex, on a clean channel, the transmitting unit might never have to stop transmitting because it stalled to wait for the necessary acknowledgments; all the necessary acknowledgments can keep coming in on the receive channel. In some cases, the send and receive channels are two different modalities. As an example, a device at an isolated, expensive-to-reach location might have a slow modem channel out to a data source (such as its Internet provider), while the return channel is a high bandwidth satellite channel. Since most of what has to be sent are brief requests for information, and the replies are typically larger by one or more orders of magnitude, this asymmetric architecture is a very efficient solution for the context. Human usage of back channel communications is similar. Because speech works best in half-duplex mode, the recipient can send simple positive or negative acknowledgments through a separate, lower bandwidth channel/modality without restricting the speaker.

This list is not exhaustive, but it is sufficient to demonstrate the viability of studying artificial communications in order to supplement the study of natural language. Given these categories, I then turned to finding instances of those acknowledgment categories in human communication.

3.4 Application to Natural Language

3.4.1 Applying Data Communications to Human Communication. I was pleasantly surprised to discover that I could find examples of essentially all these data communication mechanisms and techniques in human communication. Positive acknowledgments are as easy to spot as recognizing "Yes" and "OK". Negative acknowledgments are equally easy; "No", "I don't understand" and similar utterances are clearly negative acknowledgments. Delayed acknowledgments are seen in the fact that humans do not acknowledge everything right away, but can delay in reacting long enough to cause the communicator to recognize a lack of the expected response, and perhaps get triggered into repeating or rephrasing. Lost acknowledgments can arise because of distraction of the speaker--external or internal interference--and can also require communication repair with sentences such as "I'm sorry, did you say you wanted milk in your coffee?" Subsumed acknowledgment is seen in the fact that humans don't typically try to acknowledge every bit of the communication flow, but will periodically issue an acknowledgment that is intended to indicate an acknowledgment of all the communication since the previous acknowledgment was issued. In normal circumstances we expect the other party to understand what we are attempting to communicate, and do not require immediate and constant acknowledgment, but as the amount of time since the last acknowledgment increases, the communicator gets more and more worried. This is partially alleviated by the link keep-alive signal, like those in telephone conversations when one party will be heard issuing a stream of "Yes", "Right", "Uh-huh" and such. (I have timed various people who issue these types of back-channel acknowledgments at the rate of 20 or more times per minute). The request for acknowledgment shows up in sentences like "Did you hear me, Johnny?" or "Do you follow?" Flow control is observed in sentences such as "Slow down, you're not making sense." and "Look, I'm in a hurry. Would you please get to the point?" Protocol negotiation arises when there is a need for accurate communication, as expressed by "Tell me when you don't understand, OK?". Dynamic link quality adjustment occurs naturally as the listener increases the frequency of positive acknowledgments and thus encourages the speaker to progress both faster and with more confidence. Lastly, error correction is simple; and speakers say "I said 'Jane has a buyer', not 'a tire'" or repeat back a serial number to ensure that the information was received correctly.

- **Communication Between Peer Processing Layers**. The study of another aspect of computer network acknowledgments proved equally illuminating. Computer communication is typically modeled in terms of a hierarchy of seven layers. The acknowledgments take place between a specific layer on one machine and the corresponding layer on the other machine. All communication is done by packaging the information and handing it to the next lower layer, which encapsulates it and hands it down to the next lower layer, until it reaches the bottom or physical layer. From there it is transmitted to the physical layer of the destination machine, and the information moves up through the layers of processing until it reaches the layer that is a peer of the originating layer. Each layer has a different sphere of communication responsibility and ensures communication with its peer layers on other machines. As part of the communications protocol, there is no confusion about the intended destination layer since that is explicitly indicated as part of the contents of each data unit.

- **Communication Between Processing Layers in Human Communication**. Part of what makes this study so interesting is that human communication is characterized by a similar set of layers of communication processing. At the lowest layers we have auditory or visual stimuli that require processing and categorization into low-level units of meaning, perhaps phonemes or roman characters. A higher level combines these elementary units into higher-level lexical units, or words. Above this the words are structured into phrasal units through syntactic parsing mechanisms, and the results are passed on for semantic processing. These results are then sent on for pragmatic and other processing. Unlike communication layers within computers, these layers in humans do not necessarily hand received information up from layer to layer like a bucket brigade –they may run in parallel – but the similarities may shed more light on human communication.

The parallelism in multi-layer processing between the computer and human communication is striking. It is not surprising that one also finds parallel inter-layer acknowledgment mechanisms. The lowest layer acknowledgments are primarily various forms of negative acknowledgment; humans – and to a lesser degree, computers too – do not spend their time explicitly notifying the communicator that phonemes or other low-level units are being clearly and accurately identified, though there is a certain amount of back-channel "keep-it-coming" communication. Humans and computers do, however, quickly notify the other party when these low-level units are not being correctly received or are not capable of being processed. At a higher level, humans are more likely to generate positive acknowledgments such as "Yes, you're absolutely right" or "Please tell me more." As the targeted acknowledgment layer gets higher, there is a more even mix of negative and positive acknowledgments.

One marked difference between computer and human communication in general – and acknowledgments in particular – is that, in computers, communicating layers are unambiguously identified and layers communicate only with peers on other machines, while in human communication the layers are not usually explicitly identified and confusion can arise over which layer was intended as the recipient of a particular communication. For instance, a student is asked to make a prediction, makes the prediction, is told "OK, continue", continues in the belief that the prediction was correct, and is later told that the statement was in fact incorrect. Drawing on the power of the analogical model, an explanation is clear: the "Ok, continue" was not intended as a judgment that the statement in question was accurate, but as an indication that the prediction had been understood and that the student was to proceed to develop the line of reasoning. The student understood the acknowledgment to be a judgment that the prediction was in fact correct, as opposed to what the acknowledgment was actually intended to convey. It is my belief that many examples of human miscommunication can be accounted for by this phenomenon of incorrectly concluding which processing layer a particular acknowledgment was targeted at.

3.4.2 Conclusions. The purpose of this chapter was to explore the sources of information on human communication. This work in developing a taxonomy of acknowledgments is obviously not final; much more study of computer communications, the corpus, and other related areas is needed before a firmer taxonomy can be proposed.

These intermediate results and the subsequent application to language are reported in "Understanding Human Communication Acknowledgment Protocols by Studying Artificial Communication Protocols" [Brandle, 1996]. Halsall [1988, Chapters 4-6, 9-10] and Tanenbaum [1989, Chapters 1-9] provide a good introduction to relevant aspects of data communication. (Tanenbaum, in particular, is extremely lucid.) Another source of information specifically about computer protocols is the thousands of documents, the RFCs (Request for Comments), that are the basis of the current Internet communication. These are available at many locations on the Internet.

3.5 Relevance of the Behaviorist Literature to Acknowledments

One of the committee members, Dr. Charles Merbitz of the Illinois Institute of Technology's Institute for Psychology, pointed out an interesting set of parallels between the ideas about the use of language that Clark presents, the principles of cooperation and adaptive behavior in data communications, and the principles presented by the behaviorists in the social sciences. B.F. Skinner's Verbal Behavior [Skinner, 1957] is the classic behaviorist document that discusses how humans use language is used to achieve goals.

One the very first page of his book, Skinner states that our interaction with the environment is the indirect result of using language to induce others to engage in behavior that results in achieving the "desired" effect. The behaviorist literature contains a great amount of information on how communication behavior is learned and should be useful in studying why coordination is needed and how it is learned. One of the implications of this research is that one should study coordination by studying the behavior of the listener as well as the behavior of the speaker, that it is not sufficient to study the behavior of the speaker. The efficiency of the communication can be measured by studying the behavior of the listener. Karen Pryor [1985] is a well-known animal trainer who discusses communication channels, bandwidth constraints, and how trainers study animal behavior to learn what does and doesn't work in communication. The study of communication between humans and animals resembles the study of data communication in that it has the similar property that we must have a methodology that is both explicit and precise if we wish to succeed in our communication. This brief discussion is not intended to be a substantial discussion of verbal behavior. It is intended as a "pointer" to the behaviorist body of literature as a source of information that is indeed relevant to the study of cooperation and coordination in general, and that is particularly relevant to the study of acknowledgments.

CHAPTER IV

CLASSIFICATION EXPERIMENTS

One of the primary purposes of the CIRCSIM-Tutor project is to develop an intelligent tutoring system capable of replicating the tutoring behavior of an expert human tutor. In particular, it includes the ability to perform discourse-based tutoring. In turn, that implies a need to be able to document, describe, and explain the linguistic behavior observed during tutoring sessions. This requires reliable ways of recognizing and categorizing these various types of observed linguistic behavior.

In Chapter III, I developed what I believe is a better understanding of acknowledgments, based on communication theory. It appeared important to try to verify whether this theory of acknowledgments was something that others would be able to understand and agree upon. I performed three experiments that permitted me to start evaluating how well my theories and reality match up. I also did some preliminary tests in using machine learning to automatically build rules that describe the association between the values of various tutoring variables and the acknowledgments issued by the tutors.

4.1 Previous Work

4.1.1 Evens and Spitkovsky. Evens et al. [1993] presented a study of negative acknowledgments and hinting. Their motivation was Susan Chipman's observation, reported in [Spitkovsky & Evens, 1993] that Barbara Fox's tutors essentially never said "Wrong" or "No", whereas our system said that constantly. Evens et al. showed that the expert tutors used explicit negative keywords, but only about 25% of the time; 75% of the negative acknowledgments were of a different form. They developed an ad hoc set of ten categories of negative acknowledgment which were ranked according to severity, from most severe (direct negative response) to least severe (minor clarification by tutor).

The study had a problem in that there was no serious reference theory of acknowledgments to guide the development and ranking of the categories, or even to support deciding what utterances were negative acknowledgments. This lack of a strong theory also posed a problem for attempts to rank negative acknowledgment categories on a severity scale. The authors recognized weaknesses and stated "There may also be multiple error response categories contained in one continuous response by the tutor. This is certainly not a true one-dimensional scale" (p. 138). It is our belief that for many of those negative acknowledgments the differences are more qualitative than quantitative, and that the attempt to rank them on a severity scale is misguided. Granted, it is true that some negative acknowledgments are perceived as more severe than others (e.g., K25-tu-114-1: "No, I think you are reasoning backwards" is more severe than K25-st-62-1/2: "So, in dr hr is up, co is up, but sv is down. How is this possible?"), but this doesn't demonstrate that these categories of negative acknowledgment can be organized on a “severity spectrum”. In general, the perceived "negativity" of a negative acknowledgment depends more on the context and the role that the acknowledgment plays, than on the surface form of the acknowledgment.

An analysis of the previously-proposed ten categories suggests that they describe common techniques for issuing negative acknowledgments, but that 1) the categorization classes do not appear to belong properly in one categorization dimension, and 2) provides limited usefulness to a text-generation system which must decide what category of acknowledgment to issue. One of them, Direct Negative Response, is categorized strictly on the basis of whether an explicitly negative keyword is present in the utterance. Another, "Indirect" Direct Negative Response, covers all explicit negative utterances in the communication or metacommunication tracks (channels), levels one through four, which do not contain a negative marking keyword. A more useful categorization from the perspective of text generation would provide better guidance in determining what utterance to issue. For instance, the system knows that the tutee made a mistake in predicting that blood pressure would rise; it would not change. This indicates a negative acknowledgment at level 4 in track 1, but there remain the questions of whether this should be explicit or implicit, have an explicitly negative keyword, and deciding which of the categories meeting the (negative acknowledgment, level 4, track 1) criterion n-tuple should be used. We can not, unfortunately, just match features to choose a category of acknowledgment, because some of the ten categories differ on the basis of pedagogical purpose and technique, not just syntactic and semantic features.

Implicit negative acknowledgments may be better pedagogically in that they force the tutee to work at a higher cognitive level in order to decode the tutor's utterances. Another case where implicit acknowledgments are preferred is the matter of what counts as sufficient evidence from the student for the tutor to conclude that the student knows the material. Our expert medical tutors avoid asking yes/no questions such as "Do you understand?" on the basis that the self-evaluation returned by the students is often faulty. This position is reinforced by the findings in [Graesser, 1993] that "the most reliable information source for inferring student knowledge was the students' answers to questions" (p. 128). Our tutors choose to ask questions so that the student cannot get away with issuing a positive acknowledgment, but must instead furnish evidence which can be construed as sufficient proof of understanding to achieve closure on the joint action at hand. Graesser's conclusions fit in with Clark's four classes of signals that are positive evidence of understanding: 1) assertions of understanding, 2) presuppositions of understanding, 3) displays of understanding, and 4) exemplifications of understanding [Clark, 1996, pp. 228-229]. He states that "displays and exemplifications tend to be more valid evidence than assertions and presuppositions."

My reaction to this research was to propose characterizing acknowledgments based on a multi-dimensional characteristic space, with the following dimensions: 1) positive vs. negative, 2) explicit vs. implicit, 3) track one (normal communication) vs. track two (metacommunication), 4) the level at which it is used (levels one through four) and 5) the presence of a keyword that functions as a linguistic marker for acknowledgments. These are the values that I believe a text-generation system would have to consider in choosing what type of acknowledgment to issue.

4.1.2 Graesser. Graesser [1993b] presents useful data on face-to-face tutoring by untrained tutors. He notes "There were virtually no occurrences of sophisticated tutoring strategies, such as the Socratic method, inquiry learning, and the reciprocal training method" [p. 127] (70% of the tutor's utterances were scripted, based on the list of material to cover) and remarks that there is almost no "tailoring to the student's knowledge and misconceptions" [p. 127] (only 8% of tutoring turns were spent on student bugs and misconceptions). He says that ITSs have the potential to contribute a lot since the systems' techniques should be better than those of untrained tutors.

Graesser documents a "five-step dialogue frame" that showed up frequently in the untrained tutoring [p. 127]:

1. Tutor asks question
2. Student answers question
3. Tutor gives feedback on answer
4. Tutor improves quality of answer
5. Tutor assesses student's understanding of answer

Acknowledgments may be expected to occur in steps 2, 3 and 5.

Of more direct relevance to my study, he provides statistical data that relate to the use of both positive and negative acknowledgments. He says [p. 128]:

> We segregated student answer contributions into four quality levels: error-ridden, vague (or none), partially correct, and completely correct. The likelihood of a tutor giving positive feedback increased with answer quality, .31, .45, .51, and .62, respectively. The likelihood of giving negative feedback was extremely low, but sensitive to answer quality, .15, .04, .02, and .01, respectively. The tutors' use of hesitations or pauses in their feedback was not related to the quality of student answers, .08, .13, .15, and .13, respectively.

Graesser also states that asking "Do you understand?" questions is not worthwhile. He notes a "positive correlation between student achievement and the likelihood of a student answering 'No' (I don't understand)" [p. 128], and that both weak and good students were less likely to answer "yes" than medium strength students. This lines up well with the instructions given to us by our cooperating expert tutors. They have told that they try to never ask “Do you understand?” questions, and they wished to be certain that Circsim-Tutor would never ask any similarly weak questions.

4.2 Data Collection

Our cooperating researchers at Rush Medical College, Dr. Allen A. Rovick (A.A.R.) and Dr. Joel A. Michael (J.A.R.), have performed a number of tutoring experiments. These include a set of face-to-face tutoring sessions, many keyboard-to-keyboard sessions with expert tutors (A.A.R. and J.A.R.), and a number of keyboard-to-keyboard sessions with novice tutors. Transcripts of the face-to-face tutoring sessions were made by listening to recordings. Transcripts of the keyboard-to-keyboard sessions were generated by the program used to perform the keyboard-to-keyboard sessions, CDS [Li et al., 1992]. I am using transcripts from both the expert tutor and novice tutor keyboard-to-keyboard experiments. The signaling channels, available bandwidth, and protocols imposed by the keyboard-to-keyboard environment (e.g., half-duplex, no backchannel, communication) make it more like the environment of CIRCSIM-Tutor than is the environment of the face-to-face sessions, so I do not use any of the face-to-face session transcripts. In cases where I am trying to make normative – not just descriptive – statements, I use the expert tutor keyboard-to-keyboard transcripts.

Three protocols were used over the course of the tutoring experiments. Protocol 1 was used in transcripts K1-8, Protocol 2 was used in K9-25, and Protocol 3 was used in K30-46 and N1-31. For more details about the three protocols see [Khuwaja, 1994].

4.3 Experiments in Classifying Acknowledgments

In an effort to start validating the application of my ideas about the role of acknowledgments – that they coordinate closure of joint actions – I performed three experiments. They were as follows: 1) a study of the agreement on categorizing acknowledgments, 2) a study of agreement on identifying acknowledgments, and 3) another experiment in categorizing acknowledgments, but with a new categorization scheme.

4.3.1 Experiment 1 – Categorization. The first experiment was developed in order to test my acknowledgment categorization scheme. Specifically, I wished to verify that researchers could establish a significant level of agreement on the categorization of acknowledgments.

There are three common tests of reliability: 1) Stability: intra-rater reliability, a measure of coding variance by the same rater over time, also known as test-retest reliability. 2) Reproducibility: inter-rater reliability, a measure of coding variance between different informants (raters). 3) Accuracy: a measure of coding variance from an established "standard" [Carletta et al, 1997, p. 24; Dawson-Saunders and Trapp, 1994, p. 57]. In this experiment I tested inter-rater reliability (reproducibility) and coding variance (accuracy). The inter-rater reliability test was performed by comparing the agreement between informants. Accuracy was determined by comparing the agreement between each and the "standard" categorization. The reference categorization was jointly developed by Brandle and Evens. No stability (intra-rater reliability) tests were performed. It seemed more important to start by measuring lateral reliability, and if that proved acceptable, then to perform the longitudinal reliability tests. (The results of the stability and reproducibility tests did not appear to justify continuing this experiment by performing any intra-rater reliability tests.)

4.3.1.1 Experimental Design

• **Informants.** I had two informants: BV and BM. They are graduate students in computer science. One informant is female, one is male. One (the female) is a non-native speaker of English, but she is fluent and has a superior grasp of the English language, in my estimation.

• **Materials and Instructions.** The data used for this experiment is based on the keyboard-to-keyboard transcripts K25, K26, K27, and K28. These particular transcripts were chosen, in part, because I had already spent some time analyzing them, but more specifically, because they constituted the set of transcripts used as the data for a previous study of acknowledgments by Evens et al. [Evens et al., 1993; Spitkovsky and Evens, 1993].

These transcripts contain a total of 813 sentences in 597 speaking turns. Of these, I determined that 308 of the sentences (286 of the speaking turns) contained acknowledgments worth studying. (By worth studying, I mean that although all utterances play some role in acknowledgment; many of these utterances are not currently of interest to me.) See Table 3 for more detailed counts of the sentences and turns in transcripts K25 through K28.

I developed a simple classification scheme for acknowledgment-related utterances. The classification categories were polarity, markedness, and role. Role had three

Table 3. Sentences and Turns in K25-K28

Transcript	Sentences	Turns	Sentences Studied	Turns Studied
k25	235	164	84	75
k26	191	155	73	72
k27	217	167	91	84
k28	170	111	60	55
Total	813	597	308	286

subcategories: judgment, repair, and joint action closure (see Table 4, Acknowledgment Categories).

Table 4. Acknowledgment Categories

Categories	Values
Polarity	Positive (+), Neutral (0), Negative(–)
Explicitness	Explicit-marked (EM), Explicit (E), Implicit (I)
Role:	Judgment (J)
	Repair (R)
	Closure: Closure (C), Deferred Closure (DC)

Polarity: This category is a measure of whether an acknowledgment is positive (+), neutral (0), or negative (-). This is quite straightforward, although there was dispute as to whether an utterance could be properly classified as neutral. I discouraged marking acknowledgments as neutral because of my doubts concerning the existence of neutral acknowledgments (as discussed in Chapter III); the informant was instructed that, unless an acknowledgment was judged to be clearly neutral, is should be marked as either positive or negative.

Sample positive acknowledgment (+):

Stu> CO increases. [correct answer]
Tu> Right. [clearly positive]

Sample negative acknowledgment (–):

Stu> CO increases. [incorrect answer]
Tu> Sorry. [clearly negative]

Sample [possibly] neutral acknowledgment (0):

Stu> CO increases. [Either right or wrong]
Tu> Next prediction.
[The tutor doesn't comment on correctness]

Markedness: The categories are 1) explicit with a clear keyword/marker (EM), 2) explicit (E), and 3) implicit (I). The informant was asked to decide in this case whether the utterance functioned explicitly or implicitly as an acknowledgment, and if explicitly, whether there was a keyword or marker to clearly identify this as an acknowledgment.

Sample explicit acknowledgment with marker (EM):

Stu> CO increases.
Tu> Absolutely.
[or "Right", "Correct", "Wrong", "Nope", etc.]

Sample explicit acknowledgment without marker (E):

Stu> CO increases.
Tu> CO does increase.

Sample implicit acknowledgment (I):

Stu> CO increases.
Tu> So what happens next?
[Tutor implicitly accepts the prediction]

Role: Each acknowledgment can play multiple simultaneous roles. I wished to study the following three roles: 1) Judgment (J) – does the utterance play a role in judgment/correction of truth, accuracy, or completeness, 2) Closure (C, DC) – does the utterance play a role in agreeing on/marking closure (C) or in deferring/denying closure (DC), and 3) Repair (R) – does the utterance play a role in communication repair. This is specifically not repair of cognitive constructs, but has to do with detecting and repairing communication failures.

Sample judgment and closure (J,C):

Stu> CO increases.
Tu> Try again. [Incorrect and deny closure]

Sample repair, which defers closure (R, DC):

Stu> CP omcreases. [Correct but hand misplaced on keyboard]
Tu> Come again? [Tutor initiates repair]

<u>Compound Acknowledgments</u>: I was uncertain about how best to handle compound acknowledgments. These are sentences that contain multiple acknowledgments. What makes it particularly hard is the fact that these acknowledgments can have different polarities and roles. This makes it quite difficult to code the sentence with a single correct categorization. Sherri Condon [Condon and Cech, 1992] recommends that sentences like this be separated into separate pieces of text for classification purposes. Consequently, in the transcripts given to the informants, I separated all sentences with compound acknowledgments into their component acknowledgment clauses.

Sample compound acknowledgment (one positive and one negative judgment):

Stu> CO increases.
[Student is correct, but prediction is out of sequence.]
Tu> That's true, but you must first predict the determinants of HR.

where the tutor's response gets separated into:

Tu> That's true,

but you must first predict the determinants of HR.

Sample compound acknowledgment:

Stu> CO and RAP increase.
[CO does increase, but RAP decreases.]
Tu> CO is right, but RAP doesn't increase.

where the tutor's response gets separated into:

Tu> CO is right,

but RAP doesn't increase.

• **Samples of the Marked-up Transcripts.** I used SGML-like syntax to mark the beginning and end of each acknowledgment. The line that marks the beginning of each acknowledgment also has blank spaces for the informants to enter categorization information.

The following is a transcript sample with the markup template and spaces for categorization information:

K28-tu-6-2: What variable in our list reflects this?
K28-st-7-1: RAP
<ACK Pol= ; Mark= ; Role= >
K28-tu-8-1: Right.
</ACK>

Here is the same sample with categorization information filled in:

K28-tu-6-2: What variable in our list reflects this?
K28-st-7-1: RAP
<ACK Pol= + ; Mark= EM ; Role= J, C >
K28-tu-8-1: Right.
</ACK>

• **Training the Informants.** I wrote a guide on how to code the acknowledgments in tutoring transcripts. The document orients the informants with a brief explanation of communication channels, metacommunication, acknowledgments, and my theoretical foundation for acknowledgments. It then presents a categorization scheme to be applied to the acknowledgments. For each category, it provides examples of sequences of dialogue turns and explains how they are coded and why the particular categorization is appropriate.

I trained each informant individually. After they read the markup guide, they were given a sample marked-up transcript (K1) in which all acknowledgments and the categorizations for each acknowledgment were filled in. Together we studied the marked instances of acknowledgments and discussed how each acknowledgment was categorized and why.

Next, I gave the informants three transcripts (K2-K4) in which the acknowledgments were marked, but not categorized. After the informants performed the categorization task, they returned to review the results with me. It was intended that this training stage would clear up any misconceptions about the task. The informants were then each given the same four transcripts (K25-K28). Like the training set (K2-K4), the acknowledgments marked, but categorization information was not filled in. The purpose in working on this set was to determine how consistently the informants could apply the categorization scheme. The informants were left completely alone while working on these transcripts.

4.3.1.2 Methodology. I used the kappa statistic [Dawson-Saunders and Trapp, 1994; Georgakis et al., 1988; Shah and Evens, 1997] to measure classification reliability. The kappa coefficient (*K*) measures pairwise agreement among classified items. The following are the possible *K* values:

$K = 1$	complete agreement
$K = 0$	no more than chance agreement
$K < 0$	less agreement than would be expected by chance alone

The goal of the kappa statistic is to provide a measurement of the agreement between the coders that removes the agreement that can be accounted for by nothing more than chance. Thus it measures the degree of agreement beyond that which would be expected due to chance.

The kappa statistic is defined as follows:

$$K = \frac{P^o - P^c}{1 - P^c}$$

P^o represents the observed proportion of agreement. P^c represents the proportion attributable to chance alone.

Chance agreement is determined as follows. 1) The degree of positive agreement attributable to chance is determined by calculating the proportions that each coder marked as positive, and then multiplying these proportions and the total cases using the following formula:

$$P^c_{\text{positive agreement}} = (\text{coder1 positive/Total}) * (\text{coder2 positive/Total}) * \text{Total}$$

2) The degree of negative agreement attributable to chance is determined by calculating the proportions that each coder marked as negative, and then multiplying these proportions and the total cases using the following formula:

$$P^c_{\text{negative agreement}} = (\text{coder1 negative/Total}) * (\text{coder2 negative/Total}) * \text{Total}$$

3) The positive and negative chance agreement values are added, and the results divided by the total number of cases. This is the proportion for chance agreement.

$$P^c = (P^c_{\text{positive agreement}} + P^c_{\text{negative agreement}}) / \text{Total}$$

The observed agreement is determined by adding the values measuring agreement (the matrix diagonal) and dividing the result by the total cases:

$$P^o = \text{Sum(diagonal)} / \text{Total}$$

Here’s an example with the sample data from Table 5.

Table 5. Data Used to Calculate the Sample Kappa Value.

		BV	BV	SB
		Yes	No	Totals
SB	Yes	131	34	165
SB	No	30	287	317
BV	Totals	161	321	482

$P^c_{\text{positive agreement}}$ = (coder1 positive/total) * (coder2 positive/total) * total
= (165 / 482) * (161 / 482) * 482 = 55

$P^c_{\text{negative agreement}}$ = (coder1 negative/total) * (coder2 positive/total) * total
= (317 / 482) * (321 / 482) * 482 = 211

$P^c = (P^c_{\text{positive agreement}} + P^c_{\text{negative agreement}})$ / total = (55 + 211) / 482 = 0.55

P^o = Sum(diagonal) / total = (131 + 287) / 482 = 0.87

$K = (P^o - P^c) / (1 - P^c) = (0.87 - 0.55) / (1 - 0.55) = 0.71$

4.3.1.3 Results. Two of the informants performed the full classification task and are reported on. There are a total of 308 sentences or clauses marked for classification. BV failed to mark part or all of 4 items. BM failed to mark part or all of 12 items. Table 6 summarizes the results.

Table 6. Summary of Acknowledgment Categorizations

Coder	Polarity		Explicitness			Judgment		Repair		Closure		
	+	–	EM	E	I	J	0	R	0	C	DC	0
"Standard"	253	53	133	37	136	183	122	6	299	240	52	12
BV	261	45	118	36	152	263	42	1	304	244	57	3
BM	261	41	85	17	194	288	8	8	288	19	1	276

Table 7. Accuracy Statistics

Kappa Measure of Agreement and Significance Levels			
		BV	BM
"Standard" (SB & MWE)	Polarity	0.830, p<0.001	0.872, p<0.001
	Marking	0.710, p<0.001	0.511, p<0.001
	Judgment	0.294, p<0.001	0.014, p<0.553
	Closure	0.665, p<0.001	0.008, p<0.289
	Repair	0.282, p<0.001	0.491, p<0.001

All the classification data was entered into a spreadsheet and recoded with numeric values. The resulting data was then read into SPSS, coded for missing values, and the crosstabulation figures and the kappa data reliability values were then generated. A summary of the accuracy statistics is presented in Table 7. A summary of the inter-rater reliability (reproducibility) statistics is presented in Table 8.

4.3.1.4 Discussion. Dawson-Saunders and Trapp [1994] discuss how to evaluate the level of inter-rater agreement derived from the kappa statistic (pp. 57-58). They make the point that the level of agreement can vary widely based on the task. They cite studies with results "ranging from 57% agreement with a *K* of 0.3 for two cardiologists examining the same set of electrocardiograms, to 97% agreement with a *K* of 0.67 for two radiologists examining the same set of mammograms" (p. 58).

Carletta et al., working in the coding (classification) of discourse moves, also discuss inter-rater reliability using the kappa statistic. They state that "reliability in essence measures the amount of noise in the data; whether or not that will interfere with results depends on where the noise is and the strength of the relationship being measured" (p. 25). They say that Krippendorff "warns against taking overall reliability figures too seriously, in favor of always calculating reliability with respect to the particular hypothesis under test" (p. 25). They say that he recommends, as a "rule of thumb," when studying "associations between two variables that both rely on subjective distinctions", that $K > 0.8$

Table 8. Reproducibility Statistics

Kappa Measure of Agreement and Significance Levels			
	Category	BM	
	Polarity	0.818	$p < 0.001$
	Marking	0.585	$p < 0.001$
BV	Judgment	0.042	$p = 0.324$
	Closure	0.001	$p = 0.845$
	Repair	0.217	$p < 0.001$

is required to declare agreement, while $0.67 < K < 0.8$ permits drawing tentative conclusions. Carletta refers to language-oriented experiments that achieved a K of only 0.44. It is pointed out that medical researchers use much less strict guidelines, with the following descriptions for different kappa values: $K < 0$ means "poor" agreement; $0 <= K <= 0.2$ is "slight"; $0.2 < K <= 0.40$ is "fair"; $0.4 < K <= 0.6$ is "moderate"; $0.6 < K <= 0.8$ is "substantial"; and $0.8 < K <= 1.0$ is "near perfect". It seems to me that a $K > 0.8$ is worth striving for, even if it is hard to reach.

- **Reproducibility.** BV and BM demonstrated near perfect agreement in categorizing polarity, and moderate agreement in categorizing marking. The higher agreement can be attributed to the less ambiguous nature of the choice – particularly for determining polarity – and the fact that these were mandatory categories. Unlike the role categories, which were optional, polarity and marking had no default value and thus could not be left empty.

The K value for repair indicates fair agreement, but I am uncertain if it is even that good. The crosstabulation information showed that BM coded eight items as instances of repair, but that BV only coded one item as an instance of repair. They did agree on that one item, disagreed on the other seven items selected by BM, and agreed that all remaining items were not instances of repair.

Table 9. Closure Crosstabulation BV x BM

Closure		BM			
		None	DC	C	Total
BV	None	2	0	1	3
	DC	50	0	1	51
	C	221	1	16	238
Total		273	1	18	292

The *K* values for closure and judgment were "slight". Table 9 shows that BM coded a mere 18 cases of closure (C) and one of deferred closure (DC), out of a total of 292 items categorized. BM's total cases of closure and deferred closure amount to only about 6.5% of the total items. In contrast, BV coded 238 cases of closure and 51 cases of deferred closure, a total of approximately 99% of the identified items. In perfect agreement on coding, all the data values fall along the crosstabulation matrix diagonal. Reasonable levels of disagreement result in some values off the diagonal, but the sum of the diagonal should still contain a high percentage of the total of the matrix sum. In this case, the majority of the values are not contained in the diagonal, indicating major disagreement. BM's coding was inconsistent with the coding of both the other rater, BV, and the "standard". There was clearly a major misunderstanding about the role of closure.

Although the categorization of judgment (see Table 10) was less extremely polarized, it also demonstrated a poor agreement about what constituted judgment. BM coded all but eight out of 293 cases as instances of judgment, while BM coded all but 39 cases as instances of judgment.

Table 10. Judgment Crosstabulation BV x BM

Judgment		BM		
		None	Judgment	Total
BV	None	2	37	39
	Judgment	6	248	254
Total		8	285	293

• **Accuracy**. Referring back to Table 7, the informants showed near perfect agreement on polarity and BV showed substantial agreement on marking and closure. BM displayed moderate agreement on categorizing marking and repair. Everything else was quite weak.

• **General Discussion.** As we coded the categories in the transcripts, trying to reach a "standard" consensus categorization for the acknowledgments, it became apparent that 1) some utterances are extremely difficult to classify, 2) the training was apparently ineffective in some areas, and 3) the categorization scheme probably needed to be revised. It was no surprise to discover sentences that we have difficulty classifying. Consequently, disagreement is to be expected, particularly when dealing with natural language dialogue issues. It had also been expected that the taxonomy, instructions, and examples would need revision. I had deliberately tried to start with a simple taxonomy rather than a complex one, and I had simplified by throwing out various category classification choices – even whole categories – in the taxonomy that was given to the informants.

I believe that I did not spend enough time assuring myself that the informants really understood the underlying theory. This resulted in a failure to reach the appropriate level of accuracy.

Another problem that became apparent was that the informants didn't understand the implications of the protocol that controlled these specific tutoring sessions (Protocol 2). This is particularly important for the identification of instances of judgment. Specifically, during the predictions phases of the first and second stages of the tutoring session, the tutor required that the predictions be submitted as variable/value pairs, and the variable and the value had to be predicted during separate turns. The reason was that the protocol required the student to make predictions in a sequence that demonstrated following a valid causal reasoning chain, and the tutor did not wish the student predicting possibly incorrect values if that student had not yet determined the value of the determinants. The consequence of this protocol is that when the tutee presents a variable as a suitable candidate variable, but before the student predicts the value, the tutor has to agree that, based on the already predicted variables, it is permissible. That is, the tutor must at least implicitly issue a judgment.

The training transcripts (K1-K4) were tutored under a different protocol (protocol one) which had no prediction sequencing, other than requiring that the "primary variable" (the first variable in the predictions table, that is affected by the physiological perturbation) be predicted correctly in the first stage, before any other predictions were made. It was a mistake to train with transcripts of tutoring session that were performed under a different tutoring protocol.

My conclusion was that the classification scheme needed rewriting. There are two primary reasons why I concluded that rewriting was needed: 1) there were utterances that I believed needed to be included in our study because they were important to successful tutorial dialogue, but for which I didn't have an appropriate categorization, and 2) the inter-rater reliability scores for some of the categories indicated serious problems.

Problems in Classification

- **Sample Problem 1: Neutral acknowledgments.** In the following sentence, the student predicted that the value of cardiac output will increase ("I"). The tutor replied according to protocol 2 (Khuwaja et al., pp. 115) whereby the tutor collects and ensures a correct value for the primary prediction, tutoring the primary prediction if necessary, and then proceeds to collect the rest of the predictions, giving hints about the correct prediction sequence, but without tutoring. It turns out that the prediction that cardiac output will increase is correct, but the tutor would have responded in the same manner if the prediction were incorrect. Tutoring is performed once all the predictions have been received.

K25-st-19-1: I
<ACK Pol=+; Mark=I; Role=C>
K25-tu-20-1: Next?
</ACK>

In this case, the informant marks the utterance "Next?" as 1) positive polarity, 2) implicitly marked, 3) no particular role in judgment, 4) role in closure (on getting the predicted value for cardiac output), and 5) no particular role in repair. The informant could have chosen to mark this as a neutral acknowledgment. There are similar cases where the tutor said "Ok. Next?" Some members of the CIRCSIM-Tutor project have argued that these are neutral acknowledgments. Others argued that these were positive acknowledgments, that there is no such a thing as a neutral acknowledgment. We have had cases where the tutors claim to have intended an utterance as a neutral acknowledgment, but where the student interpreted it as positive and was later surprised during tutoring to discover that the "correct prediction" was in fact incorrect.

There are several relevant issues. A problem with categorizing utterances is the question of perspective. There are multiple perspectives on the meaning of each utterance. There is, at a minimum, the tutor's intended meaning and the tutee's interpreted meaning. Even though the tutor may intend an utterance as a neutral acknowledgment, it is not necessarily perceived that way. This implies that a thorough categorization would include information on meaning from both the tutor's and the student's perspective. A study of where the two differ would be illuminating.

Another problem is that our categorization scheme only partially supports the "multiple payloads" concept. A more accurate way of specifying the interpretation is that "Next?" was intended to signal different polarities in different dimensions. It was to convey a limited – possibly neutral – judgment value per the protocol 2 stipulation that the tutor does not tutor the value of most predictions until all predictions have been collected.

It was also intended as a positive closure on the most local (smallest scope) joint action, that of collecting the predicted value for the change in cardiac output. I believe that this is a case where I was not being precise enough in my categorization scheme. A more accurate and useful scheme would drop the general utterance polarity categorization in favor of a combination of each possible role with the polarity value, and perhaps even the markedness value. Leaving markedness alone for now, this modification in categorization would change the markup information from the following:

<ACK Pol=+; Mark=I; Role=C>

to something like

<ACK Judge=0; Closure=+; Repair=0; Mark=I; Role=C>.

This scheme more clearly represents the utterance "multiple payload content". In turn, that offers a better framework for comparing the tutor's intended meaning to the tutee's interpreted meaning. The scheme also provides more meaningful input to the discourse management subsystem to assist it in directing text generation.

- **Sample Problem 2: The tutor appears to make a judgment, but does not intend to make one.**

K25-st-61-1: Rap decreases, and SV must follow suit.
<ACK Pol= –; Mark= E; Role= J, DC>
K25-tu-62-1: So, in dr hr is up, co is up, but sv is down.
K25-tu-62-2: How is this possible?
</ACK>

In this case, the student presented an acceptable answer to the previous question. The tutor is now testing the student to determine how thoroughly the student understands the relevant physiological causal relationships. We refer to this as "fishing or trolling for problems." There are several problems here: 1) Should the two tutor's utterances be bundled together for classification purposes? (Classification granularity) 2) How should the utterances be classified?

Let me expand briefly on the idea of classification granularity. I believe that there are not only different "payloads" conveyed in parallel by each utterance, but that there are different payloads at different levels of joint action granularity. A quick list of granularity levels includes at least the following: 1) the whole tutoring session, 2) a specific phase of the session, 3) the collection of a prediction or the tutoring of a prediction, 4) the turn level, 5) the utterance level, 6) and the intra-sentence level.

4.3.2 Experiment 2 – Identifying and Delimiting Acknowledgments. I had two questions regarding which parts of the text in the transcripts should be considered acknowledgments appropriate for my study. Specifically, I wished to determine 1) how well people could agree on which utterances in the transcripts should be included in my study of acknowledgments and 2) how well people could agree on where the acknowledgments started and stopped, i.e., delimiting the acknowledgments.

The question of what utterances should be classified as acknowledgments was of interest because in the first study the informants were asked to categorize the acknowledgments, but I had made the decisions about which utterances to study. It is quite possible that some of the agreement problems in that study arose because the informants were not convinced that some of what they had been asked to categorize were in fact acknowledgments. I had invited comments on anything they disagreed with, but received negligible input.

Discovering how well people could agree on delimiting acknowledgments was also of interest because delimiting speech acts in general is not clear cut and tends to result in a fair amount of disagreement. For instance, I participated earlier in a study of student initiatives [Shah, 1997] where one of the serious concerns related to how the initiatives could be classified differently depending on where one thought the initiatives started and stopped. The problems tended to be of two types: 1) agreeing on where initiatives started and stopped, and 2) agreeing on whether initiatives should be merged or split; i.e., should multiple adjacent initiatives be counted as one instance of a particular category of initiative and merged into one initiative, or should an initiative be split up into a set of separate initiatives of one or more categories.

I decided that it was worth performing a study where the informants were trained in how to identify acknowledgments and categorize them, but were told nothing about which parts of the transcripts are acknowledgments, or where the acknowledgments begin and end.

4.3.2.1 Experimental Design

- **Informants.** I had two informants: BV and BM. Both are graduate students in computer science. One informant is female, one is male. Both are fluent English speakers, although one of them, BV, is not a native speaker of English. Both had previous experience with studying acknowledgments because they both participated in experiment 1.

- **Materials and Instructions.** The informants had the markup document (version 1) that was used in experiment 1. They were given a set of four transcripts (k17-k20) that had no special markings at all. They were told that the purpose of this experiment was to determine both consistency of agreement on what constituted acknowledgments and consistency of agreement in categorizing acknowledgments. The categorization of acknowledgments was not the primary focus of the experiment, but was included because it might be useful in explaining where and why disagreements in identifying acknowledgments arose.

- **Training.** The informants were provided no special training. Their participation in the previous experiment was considered sufficient background and training in identifying acknowledgments. Furthermore, unlike in the previous experiment, I was not trying to determine how reliably people could follow a set of coding instructions. I really wanted to discover where informants might disagree; carefully training the informants to produce the same results would have been counterproductive.

4.3.2.2 Methodology. I used the kappa statistic described in Experiment 1 to measure the pairwise agreement between the classified items. This statistic measures the degree to which the observed agreement is greater than would be expected from chance alone. I also performed two measurements. The first indicates a sentence-level agreement about the occurrence of acknowledgments. In the second, I combined the data to indicate turn-level agreement on the occurrence of acknowledgments. This helps eliminate the disagreements over the boundaries of acknowledgments and helps point out where there is agreement on the occurrence of acknowledgments.

4.3.2.3 Results. Both informants performed the full classification task and are reported on. There were a total of 476 sentences, 340 turns, available to be marked as containing acknowledgments. All the classification data was entered into a spreadsheet and recoded with numerical values. The resulting data was then imported into SPSS, where the crosstabulation figures and the Kappa data reliability values were generated. A summary of the sentence-level and turn-level agreement between SB and BV, and SB and BM is presented in Table 11. All the Kappa values are greater than 0.73, with $p < 0.001$.

Table 11. Sentence- and Turn-Level Agreement on Occurrence of Acknowledgments

		BV	BM
SB	sentence	0.733, p<0.001	0.756, p<0.001
	turn	0.805, p<0.001	0.739, p<0.001

A summary of the sentence-level and turn-level agreement between BV and BM is presented in Table 12.

Table 12. Sentence- and Turn-Level Agreement for BV and BM

	BM	
	sentence	turn
BV	0.718, p<0.001	0.743, p<0.001

4.3.2.4 Discussion. This experiment produced good agreement between all the coders. A Kappa (K) value in the range $0.6 < K <= 0.8$ are considered substantial, and $K > 0.8$ is considered near perfect. This experiment demonstrated very good agreement for both the sentence-level agreement and the turn-level agreement. Both the reproducibility (inter-rater agreement) and accuracy (agreement to the standard) are substantial.

4.3.2.5 Conclusions. This experiment suggests that we can reach substantial agreement on the occurrence of acknowledgments in tutoring transcripts. It would be interesting to identify all the cases of disagreement in the transcripts and try to determine the basis for disagreement. It might turn out to be a consistent disagreement that can be resolved. If so, that would support even stronger agreement on the occurrence of acknowledgments in the transcripts. In turn, this should produce greater confidence in the results of categorizing acknowledgments.

4.3.3 Experiment 3 – Second Attempt at Categorizing Acknowledgments. I decided that part of the reason for the poor showing in the first categorization experiment (experiment 1) was that I was trying to fit too much complexity into too simple a scheme. This made it hard to reach agreement on categorizing the acknowledgments. The result was that I rewrote the classification scheme and tried the experiment again. The full classification scheme and supporting material are presented in Appendix C.

4.3.3.1 Experimental Design

• **Informants**. I had two informants: BM and FS. One is a graduate student in computer science, the other is a recent graduate. One informant is female, one is male. The female is a non-native speaker of English, but she is fluent and has a good mastery of the English language.

• **Materials and Instructions**. The data used for this experiment is based on the keyboard-to-keyboard transcripts N24, N26, N30, and N31. These are transcripts of tutoring sessions by novice tutors (medical students who have been trained in the physiology domain, but are untrained in the tutoring domain).

These transcripts contain a total of 153 acknowledgments that were marked for categorization. The acknowledgments are categorized according to each of the following three classes:

• Judgment – is the speaker signaling judgment?

• Closure – does the utterance relate to closure on a joint action?

• Repair – does the utterance initiate discourse repair?

For each selected utterance and for each of the three classes of acknowledgment, the utterance is categorized for polarity and explicitness. Polarity is a measure of whether the acknowledgment is positive, negative, or neutral and is represented by the values "+", "–" and "0". Explicitness measures whether the acknowledgment explicitly plays a role in that class or whether it only plays its role implicitly, and is represented by the values "E" and "I". Explicitness is marked only if there is a polarity marking for that category in the acknowledgment being considered. See below for more detailed descriptions and examples.

The utterances are also classified by whether or not they contain one or more keywords that mark (signal, indicate) the categorization of the utterance. The values for this class are yes and no, and are represented by “Y” and “N”.

Each marked acknowledgment must have an entry for each category listed below. Only one classification per category is permitted.

- Judgment
  Values are “+”, “–” or “0”. If other than “0”, also mark explicitness using “E” or “I”.
- Closure
  Values are “+”, “–” or “0”. If other than “0”, also mark explicitness using “E” or “I”.
- Repair
  Values are “+”, “–” or “0”. If other than “0”, also mark explicitness using “E” or “I”.
- Marking
  Values are “Y” or “N”.

Example – Closure, no judgment or repair, marked:

N22-tu-2-2: Please give me your full name, sex, age, and social security number.

N22-st-3-1: K.A., female, 22, XXX-XX-XXXX

<ACK Judgment= 0 ,Explicit= ; Closure= + ,Explicit= E ; Repair= 0 ,Explicit= ; Marked= Y >

N22-tu-4-1: very good.

</ACK>

Here the student has provided some administrative information (needed for the study and to pay the student). When the tutor says “very good”, the tutor is not really making a judgment about whether or not the information provided is correct, so we put a “0” by “Judgment:” to indicate that judgment is not significantly active in this acknowledgment. The utterance does indicate closure, so we put a “+” by closure and then mark an “E” to show that this is an explicit instance of closure. Repair is not relevant here, so we mark repair “0”. Finally, the word “good” does carry special meaning in the acknowledgment, so we mark “Y” to show that there are keywords in the sentence and then circle all such keywords; in this case, the word “good”.

• **Training the Informants**. I wrote a new guide on how to code the acknowledgments in tutoring transcripts. The document orients the informants with a brief overview of my view of the communication theory underlying acknowledgments, and then presents a categorization scheme to be applied to the acknowledgments. For each category, it provides examples of sequences of dialogue and explains how they are coded and why the particular categorization is appropriate.

I trained each informant individually. One of them, BM, had participated in the two previous experiments. Consequently, not much training was needed, other than to explain the new categorization scheme and explain how to apply it. BM was given a sample transcript, N22, which was marked up by way of an example of how to perform the classifying. FS was familiar with the transcripts, but had not participated in categorizing acknowledgments. We reviewed the categorization guide, reviewed the examples, and then went through parts of a transcript performing the classification. The informants were left completely alone while working on these transcripts.

4.3.3.2 Methodology. I used the kappa statistic described in experiment 1 to measure classification reliability. The goal of the kappa statistic is to provide a measurement of the agreement between the coders that removes the agreement that can be accounted for by nothing more than chance. Thus it measures the degree of agreement beyond that which would be expected due to chance.

4.3.3.3 Results. The informants coded all the 153 acknowledgments that were marked for classification. The classification data was entered into a spreadsheet and recoded with numeric values. The resulting data was then read into SPSS, coded for missing values, and the crosstabulation information and the kappa values were then generated. A summary of the accuracy statistics is presented in Table 13. A summary of the inter-rater reliability (reproducibility) statistics is presented in Table 14.

Table 13. Kappa Measure of Accuracy in Classification

		BM	FS
SB	Judgment	0.750, p<0.001	0.668, p<0.001
	Judgment explicitness	0.704, p<0.001	0.359, p<0.001
	Closure	0.322, p<0.001	0.368, p<0.001
	Closure explicitness	0.571, p<0.001	0.582, p<0.001
	Repair	0.468, p<0.001	0.508, p<0.001
	Repair explicitness	-0.179, p<0.198	0.657, p<0.001
	Marking keyword	0.504, p<0.001	0.450, p<0.001

Table 14. Kappa Measure of Reproducibility

		FS
BM	Judgment	0.697, p<0.001
	Judgment explicitness	0.292, p<0.001
	Closure	0.483, p<0.001
	Closure explicitness	0.668, p<0.001
	Repair	0.193, p<0.001
	Repair explicitness	-0.085, p<0.500
	Marking keyword	0.219, p<0.001

4.3.3.4 Discussion. The results are disappointing. I had heard that reaching agreement on categorizing meaningful discourse is difficult, but thought that the new classification scheme was significantly better than the one used in experiment 1, and that this would translate into a greater degree of agreement on the acknowledgments. This will apparently require further work before I can come up with results that indicate solid agreement. I believe that the problem lies in some combination of a lack of clarity in the categorization scheme, poor choices in determining which sections of discourse constitute acknowledgments, and in having coders work alone. With something this complicated, it is likely that better results will be obtained if the experimenter works through all the preparatory decisions – such a selecting text to be classified – with at least one other well-informed person, and the coders work in pairs while performing the classification. This should result in a much higher quality data and be reflected in a greater degree of agreement.

CHAPTER V

A MODEL FOR GENERATING ACKNOWLEDGMENTS

In this chapter I propose and discuss a computer-implementable model to guide what acknowledgments should be generated. I also discuss using machine learning to analyze transcripts and derive a set of rules that describe how the tutors issue acknowledgments. There is also a brief discussion of how my understanding of acknowledgments provides suggestions for processing input.

5.1 The Proposed Model for Generating Acknowledgments

5.1.1 The Model. The following is my proposed model for the generation of acknowledgments.

1. If the student assessment is good
    - and a statement is correct
      issue an implicit positive acknowledgment
    - and a statement is incorrect
      issue an implicit negative acknowledgment

2. If the student assessment is fair
    - and a statement is correct
      issue an implicit positive acknowledgment
    - and a statement is incorrect
      issue an explicit marked negative acknowledgment

3. If the student assessment is poor
    - and a statement is correct
      issue an explicit marked positive acknowledgment
    - and a statement is incorrect
      issue an explicit marked negative acknowledgment

5.1.2. Discussion of the Model. A model that guides the issuing of acknowledgments needs to achieve a balance among the following goals.

• **Maximizing Communication Efficiency**. The need to maximize efficiency is directly proportional to the cost of communication. The cost of communication is a function of available bandwidth, available signaling rate, and cost of signal generation and decoding. Keyboard-to-keyboard tutoring is a reduced bandwidth, low signaling rate, high signaling effort situation. The bandwidth is reduced because a number of signaling channels – such as body language and prosody – have been eliminated. The low signaling rate arises because of the limited speed at which humans can type. Typing is also more effort than speaking. Consequently, there is an incentive to reduce the amount of language generated.

These factors favor implicit acknowledgments over explicit ones. I also think that they would also favor explicit marked acknowledgments over explicit acknowledgments, because marking words, or keywords, seem to carry a greater "information density" than other words. For instance, in the sentence "xxx xxx xxxxxxxxxx correct", we do not need to know that the first three words are "You are absolutely" in order to get the gist of the sentence. If maximizing efficiency were the primary goal, then we would probably simplify the sentence to "Correct". At an extreme, this leads to the style of communication used earlier with telegraphs. The bandwidth, channel cost, and signaling rate dominated to the point that every word was in a message only because it was absolutely vital. Or to put it another way, the signal content to signal words ratio had to be much higher than for face-to-face speech.

• **Coordinating Closure**. Given the importance of closure in communications, clearly marking closure is a significant goal. This goal would prejudice the acknowledgments in favor of explicit marking, because explicit marked acknowledgments are more likely to be correctly recognized as marking closure.

• **Maximizing Pedagogical Goals**. Maximizing communication efficiency is not the goal of tutoring; tutoring is performed in order to instruct a student. The following are among the tutor's pedagogical goals: 1) to detect and eliminate factual errors, 2) to detect and eliminate misconceptions, and 3) to impart new domain information, 4) to teach the student the way of thinking appropriate to the domain, 5) to teach the student the language of the domain, and 6) to build up the student's confidence. Some of these goals have clear implications for the types of acknowledgment issued. For instance, building up the student's confidence might imply that the tutor be less explicit when issuing negative acknowledgments. If the student had made one or more errors, and then made a correct statement, it seems reasonable that the tutor would give an explicit positive acknowledgment by way of reinforcement. (Please note that I am not trying to make statements about what acknowledgment protocols work best; that is another topic. Here, I am just trying to make the point that the tutor's perceptions of the tutoring goals and the effectiveness of various protocols, will influence what acknowledgments issued.) On the other hand, trying to teach effective problem solving methods, or trying to enhance the student's understanding of the domain, might cause the tutor to issue implicit, rather than explicit, acknowledgments. Evens et al. [1993] discuss how hints are combined with negative acknowledgments. If the tutor believes that it will enhance the student's grasp of the domain, the tutor may choose not to use explicit marked or explicit acknowledgments at certain times. The tutor will instead issue implicit acknowledgments that force the student to use domain knowledge in order to infer whether an acknowledgment is positive or negative.

I have not reached the point where I can really say how to balance these various goals. It seems to me that the pedagogical goals are the primary concern when deciding what acknowledgments to generate. I consider communication efficiency and closure to be of secondary importance in an intelligent tutoring system.

The proposed model is not very sophisticated, but it is a starting point. I still have questions, such as "How does a change in the student assessment (falling or rising) affect the acknowledgment?" The next section, studying the acknowledgment protocols actually used by tutors, presents further insight into these questions and suggests what other rules should be in the model.

5.2 Machine Learning to Find Rules

5.2.1 Introduction to Machine Learning. I used the C4.5 system developed by J. Ross Quinlan [1993]. This system works by analyzing numerous pre-categorized data records, in order to inductively construct a model for categorizing the data. Each input record consists of a set of values for variables that might or might not be relevant to the classification, and the correct classification for that record. The system builds decision trees that can be used to classify the data. Then it tests them to determine which one would give the classification set closest to the "correct" values. The system chooses the best decision tree and builds rules that describe the tree. These rules are then tested against the data to determine how accurately the rules would classify the data. The system then supplies the decision tree, the rules, and a cost/benefit analysis of using the rules.

5.2.2 Data Preparation. I returned to the transcripts used in Experiment 1. The "standard" classification by S.B. and M.W.E. was used as the correct classification. The information was entered into a spreadsheet with columns for the following variables:

- Transcript Type = face-to-face (F), expert keyboard-to-keyboard (K: expert tutor), novice keyboard-to-keyboard (N: novice tutor).
- Transcript identifier = 25, 26, 27, 28.
- Speaker = tutor, student (TU, ST).
- Turn in transcript = turn number (e.g., 58; sentence numbers were not used).
- Discourse Phase = administrative (A: getting student name, student ID, etc.), giving instructions (I: e.g., "Read page 3"), predictions (P: acquiring predictions from the student), and tutoring (T).
- Was the previous turn's answer acceptable = yes (1), no (-1), neutral (0).

- Student Assessment (how well has the student been doing) = 1, 0.75, 0.5, 0.25, 0. This decreases by 0.25 after each unacceptable answer, increases by 0.25 after each acceptable answer, and doesn't change after neutral answers.

- Student Assessment Delta = up (1), down (-1), no change (0).

- Acknowledgment Polarity = positive (1), negative (-1) , neutral (0).

- Marking = explicit marked (EM), explicit (E), implicit (I).

- Judgment = judgment (J), no judgment (NJ).

- Closes joint action = closes (C), defers closure (DC), not relevant (0).

- Repair = repair (R), no repair (NR).

The information from the spreadsheet was then copied to other pages on the spreadsheet and the columns were rearranged to build different data sets; the last column of data must contain the variable to be classified, and some of the columns were omitted for certain tests. These pages were saved as comma separated variable documents and transferred to a UNIX computer system for analysis by C4.5.

5.2.3 Results. I performed a number of experiments to build classification models for different variables. Certain experiments were performed multiple times, using different combinations of input data each time. For instance, the early tests indicated that the types of acknowledgments issued were different in the different phases of a session, or even between different tutoring sessions. Because C4.5 has no idea of the meaning of the variables, it can get sidetracked by differences between records. For instance, in an earlier test, one of the generated rules stated that if the turn occurred before a certain turn number in the session, then the acknowledgment type would have a certain classification. This certainly has some descriptive power – it might imply that tutors change their acknowledgment behavior as a tutoring session progresses – and can be useful in terms of detecting associations, but when I am trying to obtain classifications based on the acceptability of the student answer and the change in the student assessment, including the turn information can cause C4.5 to build less helpful trees that obscure the associations I wish to study. Following are the reports on rules for classifying acknowledgment polarity and acknowledgment explicitness.

5.2.2.1 Predicting Polarity. The purpose of this test was to see what model C4.5 would build in order to describe the observed polarity of the acknowledgments. The data variables provided in the input were transcript type, speaker, tutoring phase, answer acceptability, student assessment, student assessment delta, explicitness, judgment, joint action closes, and repair. The classification as either positive, negative, or neutral, was also provided. There were a total of 252 cases (the entire data set).

The resulting classification rules are as follows. It includes a specification of the percent of cases in the input data that would be correctly classified by using the rule.

1. If the previous answer was unacceptable

    issue a negative acknowledgment [91% accurate]

2. If the previous answer was acceptable

    issue a positive acknowledgment [98.9% accurate]

3. If the previous answer was of neutral acceptability

    issue a positive acknowledgment [94.2% accurate]

The default rule is “issue a positive acknowledgment”.

The results were not particularly surprising. It makes sense that acceptable answers are followed by positive acknowledgments, and that unacceptable answers are followed by negative acknowledgments. It is less clear that answers of neutral acceptability should be followed by positive acknowledgments, instead of neutral acknowledgments, but this result was biased by my bias against classifying acknowledgments (in the Experiment 1 data used for this) as neutral. Given the tutoring context, it makes sense that the tutor would not issue negative acknowledgments for neutrally acceptable responses.

5.2.2.2 Predicting the Degree of Explicitness. The purpose of this test was to find out what model C4.5 would build to describe whether an acknowledgment will be explicit marked, explicit, or implicit. The result of the first test (with the full data set) produced interesting rules relating to the tutoring phase, the student assessment, the change in student assessment, and the transcript number. I decided to rerun the test as two separate tests, one for the predictions phase, and one for the tutoring phase. I also eliminated what I believe to be data that has descriptive value, such as the transcript number, but which does not assist in building a useful model. The "student assessment delta" below refers to whether the student assessment had improved (1), had showed no change (0), or had become worse (-1).

Rules for the Predictions Phase

1. If the student assessment >= 0.75 and the student assessment delta = -1

    issue an explicit acknowledgment [32.3%]

2. If the student assessment < 0.75

    issue an explicit marked acknowledgment [63.0%]

3. If the student assessment delta = 1

    issue an explicit marked acknowledgment [36.2%]

4. If the student assessment delta = 0

    issue an implicit acknowledgment [84.9%]

The default acknowledgment is implicit.

These rules do make sense. If the student has been doing well (student assessment > 0.75 means one correct answer away from 1.0), then issue an explicit acknowledgment, but the acknowledgment doesn't have to be marked. If the student is doing less well (student assessment < 0.75 means at least two sequential errors), then issue an explicit marked acknowledgment to ensure that the tutor is communicating clearly. If the student assessment has been climbing, then issue an explicit marked acknowledgment, presumably to emphasize that the student's answers are acceptable. If there are no changes in the student assessment, then issue an implicit acknowledgment. This rule can only be reached if the student is doing well and has an assessment of 1.0.

Rules for the Tutoring Phase

1. If student's answer was neutrally acceptable and the student assessment < 0.75

    issue an explicit acknowledgment [50.0%]

2. If student's answer was acceptable

    issue an explicit marked acknowledgment [79.0%]

The default rule is issue an explicit marked acknowledgment

The tutoring phase rules are different from the predictions phase rules. Part of the reason is that the tutors issued many more explicit marked acknowledgments (67/95) than during the predictions phase (26/151). Because tutoring happens about predictions where the student made errors, it makes sense that the tutor would issue more explicit marked acknowledgments, especially after the student provides a correct answer. The students do often get the correct answer after the first question, so there are more instances of the tutor issuing explicit marked positive acknowledgments, than anything else.

5.2.3 Discussion. The results of these tests were interesting. There are various problems in working with the results. My biggest concern is the quality of the data that is input to C4.5. As I was entering the data into the spreadsheet, I encountered a number of instances where I was not happy with the data classification. The main concern arose from using the data from Experiment 1. I believe that the Experiment 1 data detail merges too many categories, and that this produces some classifications that are too generic to be useful. The rule-building exercise should be performed again, but with data from Experiment 3.

The rules generated by C4.5 seem to generally fit the simple model that I proposed at the beginning of this chapter. The rules that C4.5 issued are relatively simple, but it is possible to build a more complicated rule set based on the full decision tree. I intend these rules to serve as a guide for issuing acknowledgments in CIRCSIM-Tutor. I believe that a more complex model would not necessarily be helpful; we already have more complexity that we can deal with and be sure of our results. Also, as I mentioned earlier, the quality of the input data is probably not good enough to support building of more sophisticated rules.

5.3 Rules for Processing Textual Input

This is just a quick side note about ideas for processing textual input. The underlying observation is that any time the student types a sentence, the student is either participating in a joint action that was established by the tutor, or the student may be starting a separate (though possibly related) initiative [Shah, 1997]. If the student is participating, then the tutor should try parsing the answer with the joint action context in mind. If a question was asked about the heart rate, then the answer should somehow relate to the heart rate. The parsing should be easier and more productive if a "heart rate context" is used for parsing the input. The second suggestion set relates to how repair is handled. The current version of CIRCSIM-Tutor just states that the answer could not be understood, but does not provide any information that would guide the user during the next attempt at communication. When there are errors, the system should give as much information as possible to the student. If the impasse cannot be solved quickly, it is probably better to recover by bringing closure to the current joint action, and then moving on to the next item in the agenda. The current version of CIRCSIM-Tutor will get stuck in "I don't understand. Please rephrase" loops indefinitely.

1. If able to perform a sentence-level parsing of the input:
    - First try to match the input against a set of expected answers.
    - If that fails, try matching the input against a set of student initiatives, including metacommunication templates.
2. If unable to parse the input:
    - If it is a lexical problem, issue an explicit marked negative acknowledgment as part of a repair utterance that specifies the lexical item that caused the problem.
    - If several attempts at repair fail, the system should decide that a communications impasse has occurred, generate text explaining the desired answer, mark closure on the joint action within which the breakdown occurred, and then proceed to the next joint action.

CHAPTER VI

ACKNOWLEDGMENTS AND THE STUDENT MODEL

The student model in an intelligent tutoring system is a set of data structures that contain all the information that the system is keeping about the student. Student modeling is the process of providing the student model with this information [Hume, 1995, pp. 3, 39-40]. One item maintained in the student model is an assessment of the student. This assessment is used to guide the tutor in choosing tutoring tactics. Shim [1991] represented the student assessment as a real number in the range of 0 to 1.0, where 1.0 represents the highest possible assessment of the student, and 0 represents the lowest possible assessment of the student. Shim designed a simple algorithm for determining the value of the assessment [pp. 48-63]. The algorithm takes the last N answers in the response history and returns a value that varies directly as a function of the correctness of the responses and the recency of each response. He used $N = 3$ in his actual calculations. Hume [1995, pp. 56-72] argued that the tutors use a more coarse-grained assessment than Shim's continuous value assessment, and proposed a function similar to Shim's, but with only five possible values (a rank value). Hume also argued for the use of both a local assessment using only recent response history, and a global, overall assessment.

In order to build a model for issuing acknowledgments (Chapter V), I needed an assessment value for my machine learning program. I determined the local assessment by using an easily calculated simplification of Hume's five-value function. The possible values are 1.0, 0.75, 0.5, 0.25, and 0. If the student gives an unacceptable answers, the assessment drops by 0.25, and when the student gives an acceptable answer, the assessment rises by 0.25. If the student's response is not considered categorizable as either correct or incorrect, the assessment is unchanged.

The results of the machine learning tests demonstrate an association between the local student assessment and the acknowledgments issued by the tutor. The polarity of the acknowledgments appears to be determined only by the acceptability of the student's answer, and thus relies on an extreme simplification of the student assessment – one which takes into account only the student's last response. Determining the explicitness of the acknowledgment to be issued uses more past history as described in my simplified student assessment algorithm.

As far as I know, my machine learning tests are the first time that an association between the student assessment and the type of acknowledgment has actually been demonstrated empirically, rather than anecdotally. These initial tests also demonstrate that the tutor uses the assessment differently in the prediction and tutoring phases. It appears that machine learning can be used to test various student assessment functions to determine which ones show the highest association with the observed tutor behavior.

CHAPTER VII

THE SCREEN MANAGER

My main programming contributions to the project have been 1) the porting of the original version 2 program from the Procyon Common Lisp environment on the Macintosh to the Macintosh Common Lisp (MCL) environment, 2) the subsequent porting of the program to the Allegro Common Lisp environment under Windows 3.1, Windows 95, and Windows NT, 3) the elimination of a number of coding flaws that caused incorrect behavior and relatively frequent crashes, 4) the redesign of the interface, and 5) some prototype work on developing parts of CST V3 that did not exist.

The redesign of the interface started when I realized that almost none of the Procyon interface code could work in the MCL environment because essentially all the interface routines were different. This precipitated an interface "spring cleaning" in which I decided that since I had to do a lot of rewriting anyway, I might as well try to improve the program's interface so as to make it easier to use. If the ideal interface is one that the user doesn't even have to deliberately think about, then the next best interface is one that minimizes the user's conscious effort. As I considered my changes, I realized that many of them reflected the same sorts of principles about closure and acknowledgments that I had discovered in discourse. My goal here is to briefly consider interface theory, the changes between three CST interfaces, and how the changes that I have made are relevant to enhancing communication and metacommunication.

7.1 Computer Interfaces

I now wish to take a quick look at computer interfaces and how my study of acknowledgments relates to computer interfaces.

7.1.1 Human-Computer Interfaces. There is a field of study devoted to human-computer interfaces (HCI). The ideal computer interface would be one where I could just sit down and use the computer. If the interface demands a minimum of attention, I could instead focus on what I was trying to achieve. Although every program has different interaction operations that need to be learned, there would be kept to a minimum and require little effort. Unfortunately though, it is much easier to build a clumsy and confusing interface that demands lots of attention than to build the ideal, productive, low-overhead interface.

I am not about to try to cover the whole field here, but HCI is extremely relevant to the CIRCSIM-Tutor project and intelligent tutoring systems in general. As has already been discussed, one of the challenges in keyboard-based communication is the reduction in communication channels and the limited bandwidth. In human discourse, much of the information is conveyed by signal channels such as prosody and body language. Current intelligent tutoring systems like ours are not aware of these modalities and are unable to use them. A reasonable question is "What can we do to try to compensate for these losses in communicative ability?"

Donald Norman is well known for his study of interaction with products developed by humans. His book, <u>The Design of Everyday Things</u> [Norman, 1990], discusses how we interact with doors, stoves, and other devices. Many of the things that he points out appear self-evident once we have been made aware of them, but we might not have thought about them by ourselves.

By way of example, Norman studies the problems that can arise when we do something simple like trying to open a door. Doors typically have hinges on one side and cannot be pushed open on the side with the hinges. One can always try doing that, as I have on occasion, but pushing on the hinge side of the door is not a very effective technique for opening doors. We do not generally have to stop and consider which side to push because there is a handle, a door knob, or a pushbar on the appropriate side and we all know what to do with handles, door knobs, and pushbars. These are clues that tell us where to push. The designer could, however, confuse a trained user by poor placement of these devices. For instance, the handle could be placed in the middle of the door, or even over by the hinges. Operating the handle would correctly cause the door lock mechanism to open, but the user would then be frustrated if he tried pushing the handle. A less extreme example would be to take a door with the handle correctly located with respect to the hinges, but to install that door so that the door must be pulled open instead of pushed open. This is typical of the dual door systems installed at most gas stations. There is often a sign on the glass that read “PUSH” or “PULL”, but it is still quite common to make mistakes. An enhancement would be to build a new door that only provides a pullable handle on one side of the door, and to install a pushable panel on the reverse side of the door. Because we are most likely to try pulling on a door based on whether or not there is a protruding device, this new door is far more likely to be used correctly on the first attempt.

This type of analysis can be productively applied to the study and development of interfaces for computer programs. A program that presents the user with fifty pushable, clickable, or dragable items is inherently more confusing for the user than one with only five or ten. Reducing the number of elements a user – in particular, a new user – has to deal with should make the program much easier to use effectively because of the reduced effort required to know what to do at each step.

An even better move would be to have the interface signal the user about what to do next. If there are twenty buttons, but at any given time one is displayed differently from the others (e.g., change of color, texture, etc.), then the user is more likely to do the correct thing and to do so after minimal consideration of the interface.

In his essay, “Turn Signals Are the Facial Expressions of Automobiles” [Norman, 1992], Norman considers the visible signals that animals use to provide an external sign of internal states. He writes that these external signs "tell the person about the internal states of the machine" [p. 126]. With respect to tools that we use, he says "One of the special kinds of signals that this relationship requires is feedback about the operation itself. It is difficult to use a machine that does not provide feedback to the user" [p. 126]. Electronic devices, in particularly, tend not to "interact gracefully" and can be easily constructed so as to give us minimal feedback.

Let me extend one of Norman's examples. We activate turn signals to notify other drivers of our intentions, but there is also the problem of letting the driver know whether the turn signals are on. The user might get into trouble by thinking that the lever had been moved far enough when in fact it hadn't been moved enough to engage the mechanism. The user might also unknowingly deactivate the mechanism or activate the mechanism to signal a turn in the wrong direction. Similarly, the user might not realize that the signal was on following accidental activation or failure to deactivate the mechanism. We have all been behind drivers where one of these conditions held. Without sufficient feedback, it is much more likely that there will be an unrecognized discrepancy between the user's perception of the state of the device and its actual internal state. So the designers of cars provide visual and audio feedback to enhance the probability that the car and the driver are "on the same page."

This consideration of appropriate feedback and hints about how to use a device is especially applicable to the design of computer interfaces.

7.1.2 Acknowledgments and Computer Interfaces. Programmers need to design user interfaces so that they issue useful acknowledgments and avoid issuing superfluous acknowledgments. Fortunately, we have seen enough reasonable computer interfaces, and some implementation rules are self-evident enough, that we tend to intuitively do a passable job in many areas of feedback. For instance, when we move the mouse over an object such as a clickable button or field that can be typed in, the object may change appearance to let us know that the object is within focus and would be activated if we were to click. Pulldown menus often have check marks by the items that are active, making it unnecessary for us to remember exactly what was activated by default or activated later, and what was inactive. When we type in a text box, we see the letters displayed on the screen, thus providing feedback on what the system thinks we typed. If we press the wrong key, or a fault in the keyboard activated the wrong key, or nothing got activated, or a key was multiply activated, we have visual feedback to inform us of what happened. We can then take corrective action. This provides a simple error detection and correction system.

How the Computer Detects Closure from the User. The computer has no natural way of deciding whether the user has finished his turn and means to relinquish the communication channel. It could wait until the user has stopped typing (or entering predictions), and then take its turn. The problem is that the user might be pausing to think, might be talking to someone else, or might even have left the area for a while. They are accustomed to having the computer wait on them for as long as necessary, and are not used to the thought that the computer might be "sitting there" trying to determine whether or not they are finished. This evokes the famous computer science "Halting Problem"; the computer has to wait "until the user doesn't enter any more data", but the problem is that for any time period T that the computer waits, the user might have been otherwise occupied, and then decide to renew input at some time T + 1.

There are two basic approaches to solving this problem: 1) the system can include some sort of handshake that the user performs to explicitly indicate to the system that no more input is intended, and 2) the system can have time-outs that allow implicit indications that input has ended. The first can be implemented by having the user enter some special value at the keyboard – such as pressing the <ENTER> key – or by having the user click the pointing device (most likely, a mouse) in a specially designated area. The user clearly indicates to the system that input is done, then the system does something to acknowledge that signal. It is possible for the user to accidentally signal closure (e.g., accidentally click on the “Done Predicting” button), but the system often recognizes that the input is incomplete and instructs the user to complete the task first (e.g., “You must first finish entering all predictions.”). Even if the system does not treat the signal as premature closure, the system will probably signal an error of some sort and the user will repair the error by performing proper input. This should also train the user to be careful about signaling closure prematurely.

The second approach can be implemented by setting a timer that activates a response routine once the timer counter reaches a specified value. This timer is reset each time there is input, so as long as the user continues to perform input, and the timer trigger threshold is high enough to exclude most false alarms, this should usually detect input termination.

Of these two approaches, the more powerful one is the explicit handshake. There is no doubt that a termination of input has been signaled, and the acknowledgment from the system permits the user to reach closure on the input. The second approach has the problem it depends on an implicit signal, or in this case, the agreement that the lack of further input constitutes a signal. Thus it has to wait the full time-out each and every time before it can decide that closure has been reached. Since the time-out has to be set high in order to minimize accidental triggering, this means that under normal interaction (i.e., the user didn’t suspend interaction with the system) there will be this wasted time between when the user ceases input and the time-out is triggered. This high time-out pause will also contribute to perceived poor system response times and become aggravating to the user.

The best solution is to combine the two approaches. The explicit handshake results in clear agreement on closure and is much more efficient because the system can instantly start working on the input, and doesn’t have to waste time waiting for implicit closure.

Since the user will normally be signaling explicit closure, the problem of determining implicit closure will only arise on the relatively rare occasions when, for whatever reason, the user leaves the area for an extended period of time. This situation permits a much higher time-out value, which will greatly reduce the number of false alarms and the attendant aggravation to the user. It also permits the system to use a multistage time-out strategy to determine that the user is no longer paying attention. After the first time-out, the system can temporarily take over the initiative by issuing text to ask whether the user is still present and interested in participating in the session. Then the system returns the initiative to the user and waits for a second time-out period. When the system determines that the user is no longer involved, it can save the session and perform any other necessary housekeeping, and then cycle around to the initial screen and wait for another user to show up. If the same user returns at a later point, there would be the option of resuming the session from the point of departure.

Presenting the closure issue as some combination of using explicit and implicit closure detection is a simplification. There are variations on the theme, such as the one where the system starts processing the input as it is entered. The problem of determining final closure still remains, but the relevant insight is that the system doesn't have to wait for complete input before starting to process the input. The problem is that this leads to very complicated processing. Among other things, it assumes that the user is not going to try to back up and make changes to any of the input from earlier in the turn. If the user does change earlier input, then the system needs to throw away part or all of its preliminary input processing and restart. Although there are indications that humans do something like this in many tasks (e.g., processing linguistic input), I believe this to be well beyond the current capabilities of intelligent tutoring systems. We can, however, allow people to restart a procedure.

How the User Detects Closure from the Computer. In the previous section I addressed the question of how the computer detects closure from the user. Equally interesting is the question of how the user detects closure. There are two categories of closure signals: 1) an explicit closure message from the computer, and 2) implicit closure from the computer.

In the <u>explicit closure</u> category, there are various types of text output. These could be displayed in a text dialogue window, or at some other visible location on the computer screen, such as a button that the user might wish to click on in order to perform an operation. In the text window, it might display the following message (see Figure 1):

"Ok, you now have the correct value of RAP. Please predict the other variables."

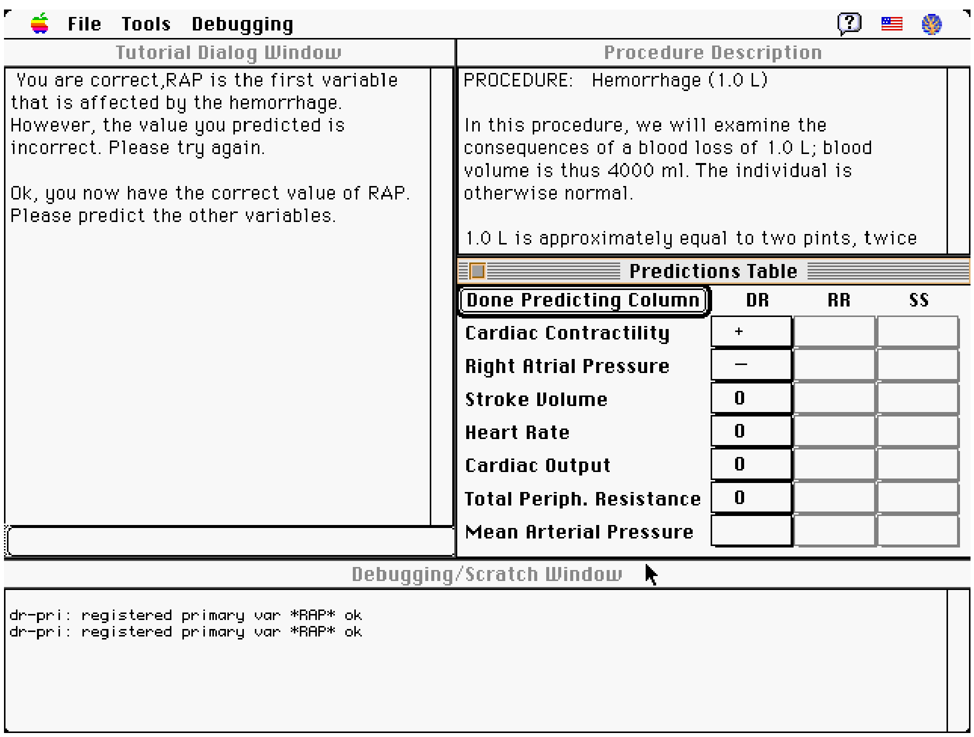

Figure 1. CST Generating Explicit Closure Message

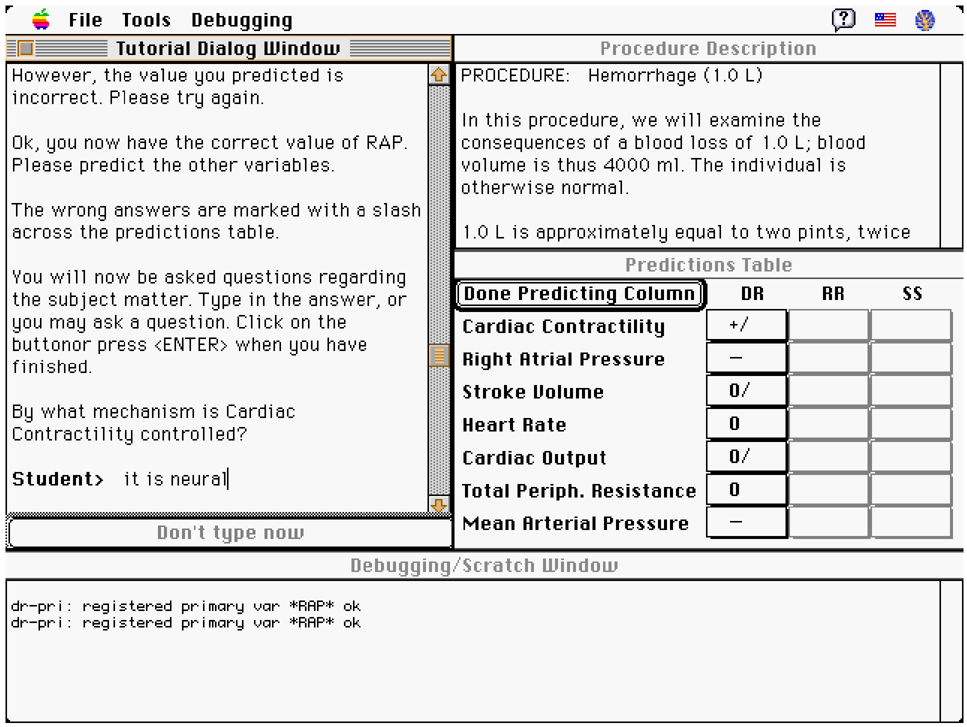

Figure 2. Signaling Closure by Changing Message on a Button

An example of signaling closure on some activity by changing the message on an important button is seen in the example in Figure 2, where the wording on the button at the bottom of the Tutorial Dialogue Window changes from “Please type — Click or <ENTER> when done.” to “Don’t type now” after the user signaled the he is done typing. It occurs again when the system has finished printing its message on the Tutorial Dialogue Window and changes the button message from “Don’t type now” back to “Please type — Click or <ENTER> when done.” and issues a **“**Student> **”** prompt to indicate that the system has stopped generating text and is waiting for the student to enter a response (Figure 3).

The system also uses implicit signals to communicate closure. One hard-to-miss signal is by markedly changing the color or other display attributes of a window (e.g., the title bar at the top of a window) to show that it has become either active or inactive. By guiding the user to change focus from one part of the screen to another, the system is implicitly telling the immediate work the user was engaged in has reached closure. See Figure 1 and Figure 4 to find the results of the system shifting focus from the Predictions Table to the Tutorial Dialogue Window. Another example of changing highlighting to indicate closure is seen in Figure 5 where the system has just directed the user to start making predictions in the "RR" column in the Predictions Table. The "DR" column had been the active column and was drawn in white, but it is now redrawn in the window background color, and the new active column, "RR", is drawn in white.

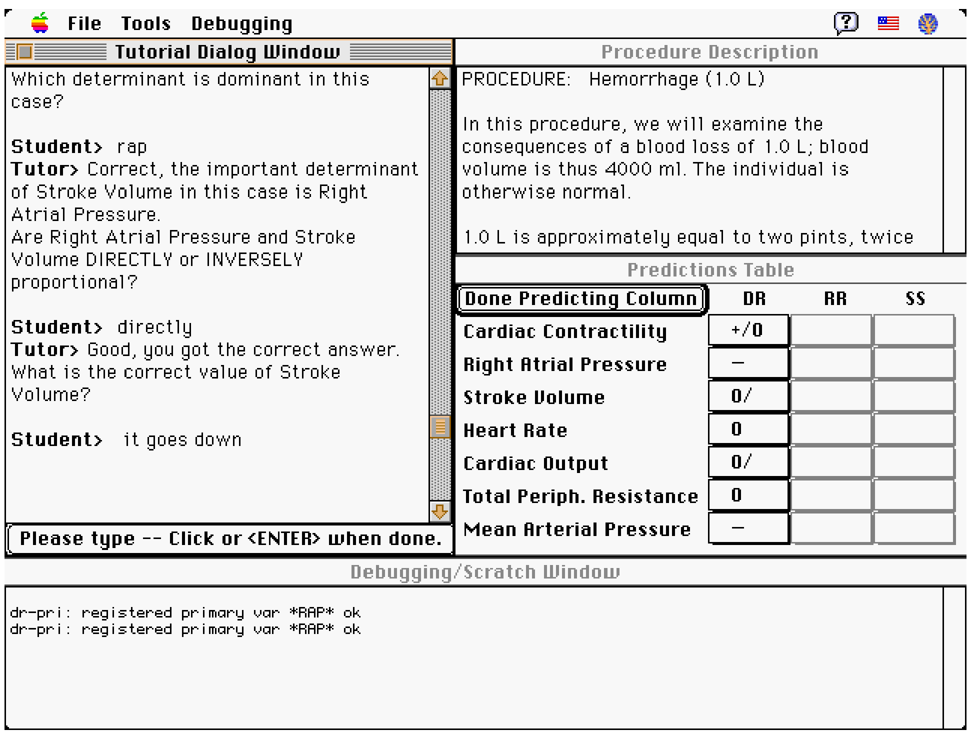

Figure 3. Issuing Signals to Coordinate Turn-taking

Another way that the system can signal implicit closure is by making further input in a window physically impossible by disabling input. The prediction boxes in the Predictions Table become inactive, so that the user is unable to enter any predictions. The system signals closure on typing in the Tutoring Window by locking the window; any attempts to type are ignored and a bell is sounded.

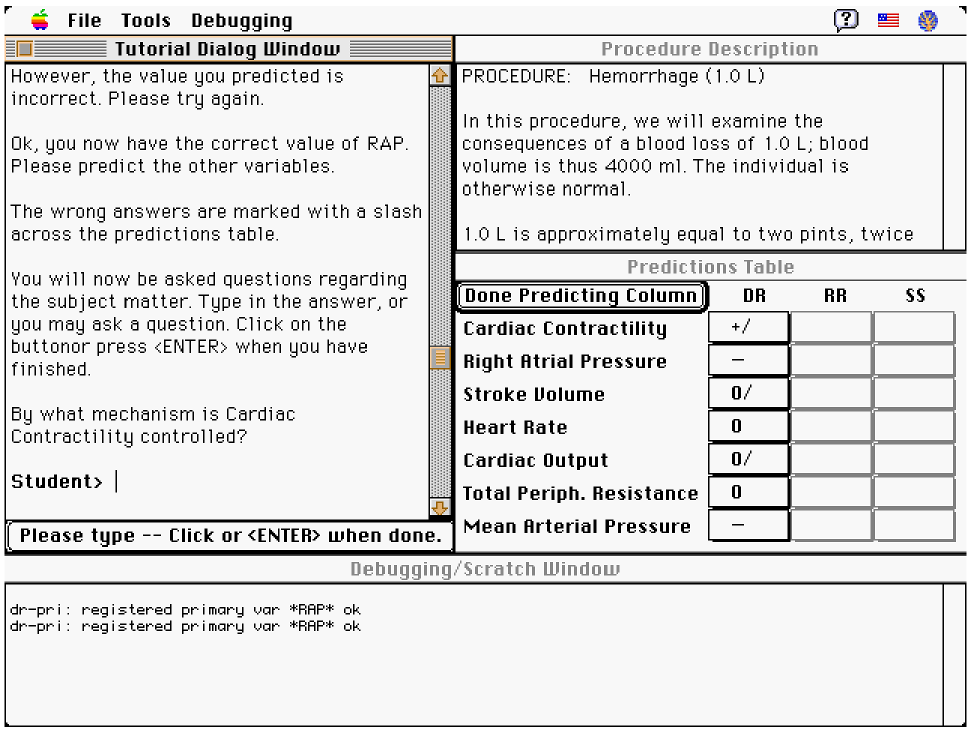

Figure 4. Focus Shifted from Predictions Table to Dialog Window

A simple way of signaling a type of low-level, tight-granularity closure is to display the user's input on the screen. When the user chooses a prediction value from the popup menu associated with each prediction box, the system displays that value in the prediction box. Similarly, when the user types in the Tutoring Window, every letter typed is automatically drawn on the screen. Although we do not expect to comment on this today, it was not always the case that systems displayed for the user what the system had recorded the user as having typed. This switch to immediate feedback allows users to know about problems, such as pressing the wrong key on the keyboard or selecting the wrong prediction choice from the menu, right away. This forms a simple, but quite effective, built-in error detection and correction system. It also has the virtue of providing error correction even in cases where the errors occurred within the system. The system somehow misreads the input and displays the incorrect information to the user. The user realizes that the displayed value does not match what was expected, and so the user corrects the value by backspacing over the incorrect input, or by reentering the prediction.

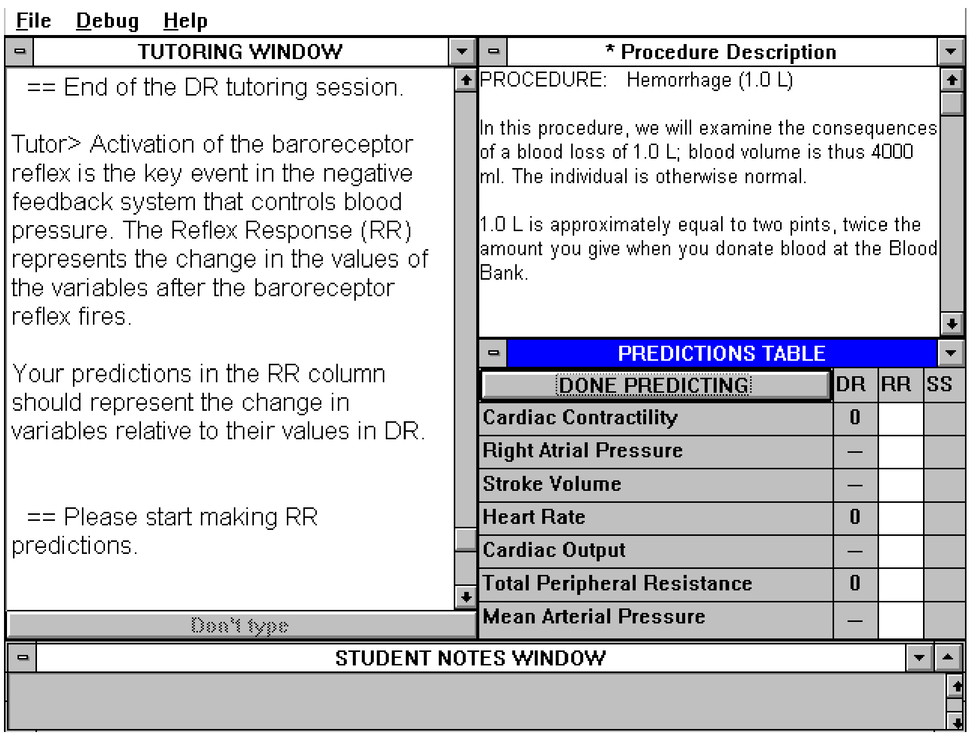

Figure 5. Changing Prediction Table Highlighting to Guide User

It is also possible that the user could detect some sort of implicit closure by way of “time-outs”. This is much trickier in that I find it hard to imagine the system designer deliberately choosing to invoke “user time-outs”, when it is extremely easy to generate an explicit closure signal such as the combination used in the Tutoring Window; writing

"Student> " on a new line after the end of the tutor output, and changing the message on the button at the bottom of the window to "Please type — Click or <ENTER> when done." The only three examples of invoking a user time-out – that I can think of – are 1) when the operating system or runtime environment administrative functions, such as the LISP garbage collection routines, get activated and produce a marked delay in system response (this is not as common or noticeable on the newer, faster systems), 2) when the tutoring system takes an unusually long period of time to finish some computation, and 3) when the operating system or program performs a serious test of user time-out sensitivity by crashing! The first two of these do not really indicate closure, but the impatient user might treat the situations as a variation on category number three. A system crash plays the role of an indisputable closure on the tutoring session.

7.2 Previous CIRCSIM-Tutor Interface

The developer of the previous version of CIRCSIM-Tutor (Figure 6) put a lot of effort into ensuring coordination with the student, but the resulting system was confusing to users and needed changes. The way to enter a prediction for the value of a variable is to go to the predictions table, find the row for the variable, move over to the column representing the current stage (DR, RR, or SS), and then start clicking in that box. The user can cycle through the possible prediction values by clicking: the first click results in the system displaying a plus ("+", the variable increases) in the box, the next click changes the displayed value to a minus ("–", the variable decreases), the third click produces a zero ("0", no change), and another click brings the cycle back to the plus sign.

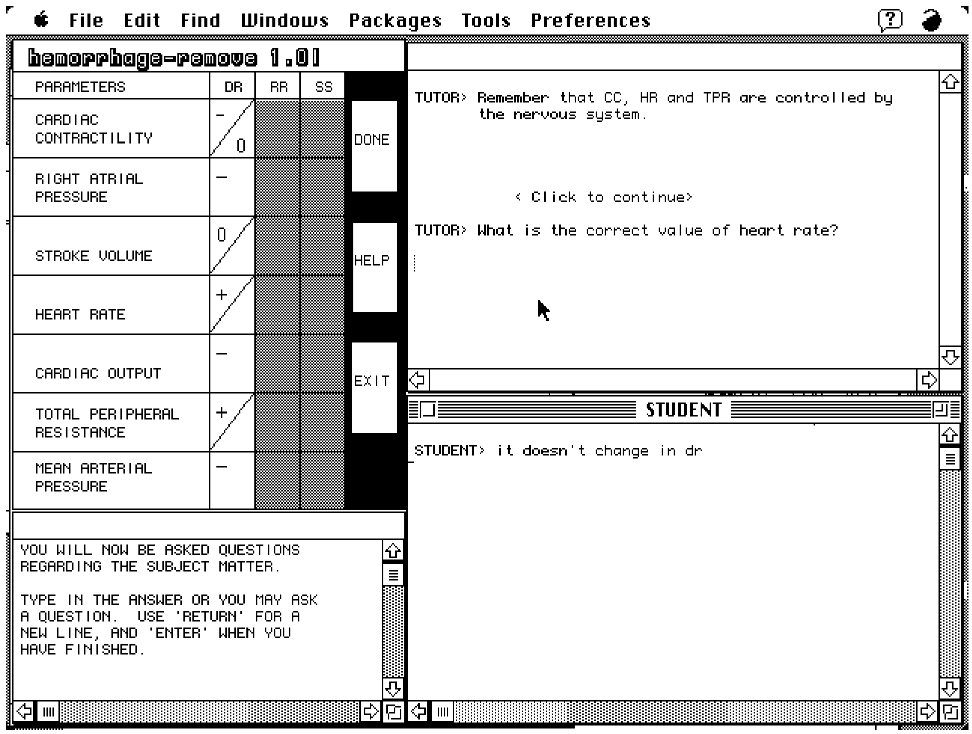

Figure 6. Early CST Version 2 Screen Image

When the system generates messages, it typically prints more than one message at a time. These messages are individually delimited by issuing the text "Click to continue", and requiring the user to click in the window where the message was printed. The system then proceeds to issue the next piece of dialogue. When the system has issued all the queued up messages, it then instructs the user on what to do next.

When the user is allowed to type in text, the text input window switches from read-only to typing mode to permit input. The user then types in some text. As the typing reaches the end of the line, the text just wraps around to the next line. The user can also choose to press the <RETURN> key to start typing on a new line. On the Macintosh, with the Procyon compiler used for the early CST V2, this is distinguished from the <RETURN> key, which is what most people think of when one speaks of the <ENTER> key. If the user wishes to signal to the system that the input is done, the user must press the <ENTER> key on the numeric key pad. This peculiar quirk has frequently caused confusion, particularly during the user's first session.

7.3 Problems with The Previous CST Interface

The original version 2 of CIRCSIM-Tutor was quite confusing for members of the project, let alone people who had never seen the program before. One of the most obvious problems was the need to do frequent clicking in boxes to indicate closure on actions. It was not always clear which boxes needed to be clicked, nor in which order. Some of these problems arose from restrictions imposed by the compiler and development environment, some of them arose from a lack of experience in interface design, and some of the problems were those that all early versions of programs are prone to have before the developers come up with better ways of interacting with the user.

The decision to make predictions by clicking in a box to cycle through prediction choices is not a terrible one, but it seems to me that using it requires extra time and effort on the part of the user. This is especially true if the user overshoots the desired prediction value and has to cycle back again through the other predictions. The time and effort spent performing extra clicks in various boxes and windows just adds to the interface overhead; if the user can perform the prediction with one click, that is better. It is rather like having to choose a card from among a set of cards stacked on a deck – the user has to pick cards and say, "Not that one", "Not this one either", "That's the one I want" – versus laying the cards out on the table and saying "That's the one I want".

I would also like to argue that this is not a familiar computer interface metaphor for choosing from among a set of options. Most computer-trained users are accustomed to being presented with a menu of options and choosing one, or even none, of those options.

This early version of CST V2 tried to offer full-duplex communication by providing two half-duplex communications channels. The idea was that the user would type text in the "user window", and CST would output text to the "system window". Two windows are needed to avoid collisions and the confusion that they bring in when they occur in a full-duplex, simultaneous transmission environment. It is a form of implementing collision avoidance in the communications. The idea behind the full-duplex communication was that human tutors do use full-duplex communication with their tutees; there is a lot of backchannel signaling, especially for acknowledgments and interrupting initiatives. However, this did not work very well for CST. Probably the main problem was that it is complicated enough trying to get a system that works in a turn-taking manner; I do not think that we really know how to build a system that can handle backchannel traffic at the same time. Students are actually using CIRCSIM-Tutor now, but there are both architecture and technical programming challenges to overcome here before we can hope to implement a mature system. I believe that full-duplex communication is important, but that it is premature for our current level of technology.

7.4 Enhancements to the CST Interface

On the very first window, the user is required to enter his name. Part of the feedback results from displaying the typed characters so that the user can perform self-monitoring and correct any typing mistakes. If the user fails to type in a name (for whatever reason) and presses <ENTER> or clicks on the button labelled “Continue” to signal closure on that screen, the system issues a negative acknowledgment and defers closure by popping up a small window that informs the user that a name must be provided before continuing. Throughout all the introductory screens (all but the tutoring window itself), closure is indicated in exactly the same way, by pressing <ENTER> or by clicking on the “Continue” button at the bottom of the screen. This is a low effort and extremely consistent way for the user to indicate closure. Except for the first and screen, all the system is doing is presenting information, the system has no reason to deny closure, and consequently just accepts the indication of closure. In each of these cases, its acknowledgment to the user comes instantly when it advances to the next screen.

When we get to the tutoring screen, there is more opportunity for acknowledgment. This is where the predictions are made, where the tutoring instructions are presented and where the tutorial dialogue takes place. The user's attention is guided by controlling which window in the screen is active; the currently active window has a highlighted title bar and all others are white. Changing the color on the title bars is particularly useful in guiding focus. The system also changes the text on the button at the bottom of the tutorial discourse window to remind the user of the possible acknowledgments that can be sent to the system (clicking on a button or pressing the <ENTER> key). The system also changes the tutorial discourse window between read/write and read-only mode to ensure that the user is not typing input when the correct activity is making predictions in the predictions table. There is a change in colors/shading on prediction boxes to indicate whether a prediction may be placed in that box, and later to indicate whether a prediction was correct (white background) or incorrect (red background).

CHAPTER VIII

CONCLUSIONS

ITSs are amazingly complex. As far as I can tell, although there are functional ITSs, their primary current use is as concept systems that permit researchers to explore ideas – and model and implement those ideas – in AI, communications, education, linguistics, pedagogy, philosophy, psychology and other social sciences.

There remains the dream that ITSs will some day be proficient enough at tutoring, that they will provide large-scale viable alternatives to tutoring by humans. How well this can be achieved, remains to be determined. A relevant comment comes from Paul Churchland (a philosopher at UCSD) who presented the closing address at the 1996 Cognitive Science Society meeting. In his talk on the likelihood that sophisticated neural networks will be capable of successfully modeling, explaining, and performing various complex processes such as vision and language, he suggested that this was a question that would be solved, not from the armchair, but in the laboratory.

8.1 Summary

I have presented a study of the mechanisms inherent in coordinating communication. This included the following: 1) a review of coordination theory as seen from the perspectives of multiple disciplines, 2) an analysis of how these mechanisms are used during tutoring sessions between humans, and 3) an analysis of the relevance and implementation of these mechanisms to the linguistic and graphical interfaces of intelligent tutoring systems.

8.2 Significance

At the 1995 meeting of the Cognitive Science Society in Pittsburgh, I met George Miller. When I asked what I – as someone working within a computer science framework

– could do to help improve learning, he recommended intelligent tutoring systems. ITSs represent one of the meeting points of many disciplines and an opportunity to test various theories by actually implementing them and examining the results.

My personal work is significant because of the added understanding of communication coordination mechanisms that it brings to the study of tutoring and the implementation of intelligent tutoring systems. The parallelism drawn between the acknowledgments used in computer communication and the acknowledgments used in human communication is, to the best of my knowledge, original. I believe that studying communication mechanisms in artificial communication systems provides a better understanding of human communication mechanisms than could obtained from the study of human communication mechanisms alone. This analogy has both predictive and explanatory power for discovering and explaining the coordination mechanisms used in human communication. It does not appear to be a forced comparison, but seems instead natural and reasonable.

The development of a taxonomy of acknowledgments contributes to our efforts to document the linguistic behavior evidenced in our corpus of human tutoring transcripts. A better understanding of acknowledgments also provides deeper insight into the tutor's strategies as recorded in the transcripts. A better understanding of these strategies is important to enhancing many parts of CIRCSIM-Tutor, including the instructional planner and the text generation and input understanding subsystems. Last, my work assists in the process of discourse management and dialogue generation by providing a deeper understanding of what acknowledgments CIRCSIM-Tutor's users might expect and benefit from receiving.

8.3 Future Work

8.3.1 Complete the Categorizing and Coding Reliability Analysis. The analysis of the acknowledgment categorization experiments needs to be extended to verify stability (test-retest reliability), reproducibility (inter-rater reliability), and accuracy (comparing the coding to an agreed-upon standard coded transcript set).

8.3.2 Perform Further Machine Learning Tests. I am interested in continuing to use machine learning to enhance the acknowledgment generation model. My test using C4.5 [Quinlan, 1993] provided some promising results, but much more complete data must be gathered and prepared for use in machine learning. The great benefit of gathering this data is that, once we have the data, we can use machine learning to rapidly analyze many models of both tutorial discourse and tactics. For instance, we could test a number of possible student assessment calculation functions to determine which model best describes the observed acknowledgments.

We need far more contextual and linguistic information about every sentence and turn. This includes information such as the following: 1) position of the previous, current, and next sentences within the closest joint action, 2) the current tutoring tactic in use, 3) student assessments – both local and global – based on various student assessment models, 4) student initiative classification, 5) judgment value (especially regarding correctness) of the previous sentences, 6) tutoring phases of current sentences (e.g., administrative, acquiring predictions, tutoring), 7) surface form of previous and current sentences, location of this sentence within the current turn, 8) metacommunicative value of previous, current and next sentences by both speakers, 9) current tutoring protocol, 10) time in the previous turn and sentences until the speaker initiated communication, 11) total time for the previous, current, and next turns, 12) sex of the student, 13) marking category (explicit marked, explicit, implicit), 14) closure classification, and 15) polarity classification.

8.3.3 Correlate Joint Actions and Acknowledgment Categories. I went through a sample transcript (K25) marking the beginning and end of each joint action as well as the acknowledgments. The hypothesis was that the type of acknowledgment observed shows a meaningful correlation with the beginning and end of joint actions. Specifically, I expected to see explicit negative acknowledgments at the beginnings of newly introduced joint actions because the tutor might be starting a new joint action in response to a student error or problem. I expected to see explicit positive acknowledgments at the end of a joint action that was precipitated by a student error, especially if the student made several mistakes in a row and required extensive tutoring in order to reach the point of giving a correct answer. The preliminary results support these hypotheses, but I need to code the transcripts to identify the association between joint actions and the turns and sentences.

8.3.4 Enhance CIRCSIM-Tutor's Acknowledgments. CIRCSIM-Tutor does issue acknowledgments, but they are more ad hoc than model-based. Once the work to verify the acknowledgment coding is complete, a new model for generating acknowledgments can be built. This model then needs to be implemented in CIRCSIM-Tutor.

8.3.5 Combine the Separate Coded Transcript Information. There is a need to combine my coding of acknowledgments with the work of other researchers in our group. For instance, it should prove useful to study the association between acknowledgments [Brandle, 1998; Evens et al. 1993], student initiatives [Sanders, 1995; Shah, 1997], hinting [Hume, 1995], and tutoring tactics [Freedman, 1996; Kim, 1998]. The coding styles for these studies are different. Not all of the coded transcripts are easily available. Our group would greatly benefit if all this information could be brought together and combined in some easy to use format.

APPENDIX A

DATA FROM EXPERIMENT 1

A.1 Instructions Document Given to Coders

Guide to Marking Acknowledgments in Transcripts

0 – Introduction

When communicating, humans don't just run around magically tossing out utterances or parsing utterances from others. Many things go into communication; I can summarize by saying that humans construct and depend on a very complex model in order to communicate successfully. It is my belief (and hope!) that this is a computationally tractable modeling problem, particularly for microdomains where we can hope to develop a model that is sufficiently complete with respect to domain and communication knowledge needed to support competent communication. The study of acknowledgments is a part of the work necessary for developing more powerful and usable computational models of communication.

The study of language is made harder by the fact that each utterance has more than just one purpose or role. Communicating parties always have multiple issues that they deal with in parallel, sometimes producing a very complex agenda. These agenda items generate multiple information streams that must be encoded onto the available communication modalities (communication channels: e.g., written, spoken, body language) with consideration to the signal bandwidth of each modality. My theory of communication holds that maximizing efficiency in communication is an important goal of communication. Maximizing efficiency is even more important in bandwidth-constrained situations, such as a keyboard-to-keyboard dialogue where the signaling modality is restricted largely to the typed text. Consequently, we modulate multiple streams of information – each with different goals – onto a restricted bandwidth. Because each utterance plays multiple roles, the question becomes "Which roles does this utterance play", not "Does this utterance have a role to play." A simple utterance like "OK" can indicate that the speaker understood the prior utterance, judges the prior utterance to be true, and is moving to close the current topic. In our study of acknowledgments, we are focusing on the roles of acknowledgment in rendering judgment on the truth, accuracy, and *sufficiency* of the student's knowledge, and in performing metacommunication functions. Metacommunication is communication about communication; it is used to signal attention, to correct misunderstanding (repair), to control the flow of discourse (e.g., starting or ending a topic), and for a few other lesser functions.

1 – The Purpose of Studying Acknowledgments

Acknowledgments are necessary for communicating successfully. At an abstract level, studying acknowledgments enhances our understanding of natural language (NL) and our competence in NL. At a more concrete level, our study enhances the ability of tutoring systems to issue and process acknowledgments, thus making the systems more powerful and useful. Most specifically, this study will guide development of CIRCSIM-Tutor as we seek to improve its linguistic and tutoring competence.

2 – Theoretical Foundation

When communicating, we strive to ensure success. We do this by tailoring our utterances to maximize relevance, clarity, precision, and efficiency (see Grice's Maxims) by controlling word choice and syntax, by ensuring a shared context, and by setting an appropriate information transmission rate.

I have chosen to define acknowledgments in the light of Clark's presentation of discourse as a joint activity composed of joint actions (*Using Language*, 1996). He focuses on the mutual coordination of their individual actions by the participants in linguistic activity. Clark says: "There is coordination of both *content*, what the participants intend to do, and *processes*, the physical and mental systems they recruit in carrying out those intentions" (p. 59). Similar ideas are expressed by Barbara Fox (1993) who states as a major finding of her book, *The Human Tutorial Dialogue Project*, that "tutoring involves constant, and local, management. This requires a pervasive mutual orientation between tutor and student, such that every session (indeed, every utterance) is a thoroughly interactional achievement, produced by both tutor and student" (p. 3).

Clark says that joint actions can be divided into phases – entry, body, and exit – and these phases are what get coordinated. Entries and exits are coordinated by syntactic, morphological, and intonational markers. A joint action is complete when there is a mutual recognition of closure on that action. "It is a fundamental principle of intentional action that people look for evidence that they have done what they intended to do" (p. 222). He restates Norman's *Principle of Closure*: "Agents performing an action require evidence, sufficient for current purposes, that they have succeeded in performing it." (p. 222) and then introduces the *Principle of Joint Closure*: "The participants in a joint action try to establish the mutual belief that they have succeeded well enough for current purposes" (p. 226).

The goal of every discourse segment is to successfully reach closure. Consequently, each utterance will in some way deal with whether we are communicating successfully and whether we have reached closure.

3 – Definition of an Acknowledgment

An acknowledgment is any utterance that indicates agreement or lack of agreement in regular communication or metacommunication. More specifically, it marks closure or a lack of closure on a joint action. (THIS NEEDS A BUNCH MORE WORK).

The famous German philosopher Hegel once wrote that the rest of the study of philosophy would be merely a set of footnotes to his writings; I am somewhat more modest. Specifically, I still consider my understanding of acknowledgments to be blurred and incomplete in many aspects. (Indeed, much of the state of the art in natural language processing can be described that way.) It is my hope that readers will promote growth in our understanding of acknowledgments by contributing their observations, questions, and challenges.

4 – Categories of Acknowledgment

I am asking you to categorize acknowledgments for each of the following three dimensions: 1) polarity, 2) explicitness and 3) role. Polarity is a measure of whether the acknowledgment is positive, negative, or perhaps could be considered neutral. Explicitness indicates whether the utterance is explicitly an acknowledgment – possibly with a clear keyword or discourse marker – or whether it is an implicit acknowledgment, that is, it must be inferred that the utterance is an acknowledgment. The role refers to which of two primary role types and multiple subtypes the particular utterance plays in its context.

Following is a discussion of the acknowledgment categories, their possible values, and a set of examples – and non-examples as deemed relevant – for each category.

A) Polarity:

- Positive
- Neutral
- Negative

This category is the mostly straightforward, although neutral acknowledgments are rare and open to dispute. The acknowledgment will either indicate agreement, closure, success, or it will indicate a failure to agree, to reach closure, to declare the communication successful. Simple examples would be exchanges such as the following:

Example 1 — positive acknowledgment:

Stu> CO increases. *[Correct answer]*
Tu> Right.

Example 2 — negative acknowledgment:

Stu> CO increases. *[Incorrect answer]*
Tu> Sorry.

Example 3 — possible neutral acknowledgment:

Stu> CO increases. *[Maybe correct, maybe incorrect answer]*
Tu> Next. *[Tutor don't say anything about correctness]*

The existence of neutral acknowledgments is debatable. Part of the problem lies in the need for the speaker and the listener to reach a mutually shared interpretation of the acknowledgment. Our medical tutors would try to issue neutral acknowledgments (e.g., deferring judgment on what the student had uttered) by saying things like "OK" or "Next" after the prediction was made. The tutor intended to acknowledge that the prediction had been understood and recorded, but not to issue a judgment/evaluation. The students tend to interpret this as indicating understanding, recording, and positive judgment of the predictions. It seems that unless there was a prior discussion of the rules of the game such that it was clear that predictions would be recorded without immediate judgment, the student will interpret these as implicit positive acknowledgments. Unless you are absolutely convinced that an utterance is a neutral acknowledgment, you should code it as something else. (Is there a default value for these cases?)

Coding utterances as positive or negative is complicated by the fact that utterances containing acknowledgments are not uniquely positive or negative. Consider the following example:

Example 4 — compound acknowledgment:

Stu> CO increases. *[Student is correct but out of sequence.]*
Tu> That's true, but you should first predict HR.

Example 5 — compound acknowledgment:

Stu> CO and RAP increase.
[CO does increase, but RAP actually decreases.]
Tu> CO is right, but RAP doesn't increase.

Example 6 — compound acknowledgment?:

Stu> CBV increases. *[student performs a "near miss".]*
Tu> So what does that do to RAP?

The tutor issues mixed acknowledgments here. In example 4, there is a positive acknowledgment judging the student's prediction as correct, but at the same time there is a negative acknowledgment about the causal reasoning order that the student is following in making the predictions. In example 5, the tutor indicates that the first prediction is correct, but that the second one is incorrect. In example 6, the student is basically right (CBV and RAP are closely related), but the tutor wanted the answer to be expressed in terms of RAP, not CBV.

Example 7 — partial acknowledgment:

Stu> CO increases. *[student is correct, but answer was incomplete]*.
Tu>and ... *[tutor prompts for more]*

The tutor issues a partial acknowledgment. What the student stated was true, but this might have been in response to a question such as "What happens to CO and RAP?"; the answer was incomplete.

The examples above illustrate compound acknowledgments. It is my belief (inspired by the work of Sherri Condon) that the best way to handle these situations are to treat them as separate acknowledgments. Rhetorical Structure Theory would argue that “CO and RAP increase.” could have started as two separate utterances at a deep linguistic level and then get combined into one utterance on the way to the surface form. The “and ...” could be interpreted as “CO does increase” and “You are wrong in your belief that you have fully answered my question.” I propose that these cases be dealt with as two (or more) separate acknowledgments and marked on that basis. It is a simpler system, and in our communication model we must treat simplicity as a virtue.

There is also the question of degrees of positiveness or negativeness; the polarity is not really the product of a discrete function with two or three values (-1, 0?, 1), but is instead the product of a continuous function. However, I’m going to repeat the *simplicity is a virtue* mantra and request that you perform the markup on a discrete basis with the values of positive, negative, and maybe neutral if you really need it.

B) Markedness:

- Explicit with a clear keyword/marker
- Explicit
- Implicit

In this category we indicate whether the utterance is explicitly functioning as an acknowledgment, or whether it is an implicit acknowledgment. A keyword or a marker is a word that, even standing alone, is associated with being an explicit acknowledgment.

Example 8 — explicit acknowledgment with marker:

Stu> CO increases.
Tu> *[E.g., “You're correct”, “Absolutely”, “Wrong” or “Nope”.]*

If the acknowledgment contains a word like this, it should be categorized as explicit with a marker. If it is explicit, but appears to lack the marker, you should code it as explicit.

Example 9 — explicit acknowledgment without marker:

Stu> CO increases.
Tu> CO does increase.

Implicit acknowledgments are utterances where the acknowledgment aspect isn’t obvious and must be inferred. The tutor intends that the student recognize the utterance as an acknowledgment, but the student must perform an inference in order to recognize it as an acknowledgment.

Example 10 — implicit acknowledgment:

Stu> CO increases.
Tu> What happens to MAP? *[Positive implicit ack.]*

Example 11 — implicit acknowledgment:

Stu> CO increases.
Tu> That completes this stage. Please read page 6.

The fact that the tutor makes no apparent judgment on the sufficiency of the answer – in particular, that there was no negative acknowledgment – but instead continues with examining some other part of the causal reasoning chain or moves on to a new topic, leads the student to believe this to be a positive implicit acknowledgment. This would be the normal conclusion unless there were special discourse rules in effect and both parties were aware of mutual cognizance of the rules.

There can be confusion over what appears to be an explicit marked acknowledgment, but which is not primarily an acknowledgment.

Non-example 12 — explicit acknowledgment:

Tu> Have you used HEARTSIM or CIRCSIM?

Stu> No.

On the surface, this looks like an explicit marked negative acknowledgment; that's what "No" might be thought to mean. In this context, the "No" is really the propositional content of the student's reply. The student is saying something like "It is the case that I have not used HEARTSIM or CIRCSIM." The student is not really saying anything about metacommunication or trying to make a judgment about some previous value. I suppose that one could construe the tutor's question as a statement that "It is the case that you have previously used HEARTSIM or CIRSIM." and that the student is judging that statement to be incorrect, but this seems to be really stretching the model. It does play a minor role in marking attention and comprehension of the tutor's question, perhaps even in proposing closure, but its main purpose is to communicate that the student has not used either of the two named systems. It is a coincidence that the surface form looks like an explicit negative acknolwedgment. We will *not* mark this as an acknowledgment of any kind.

Non-example 13 — implicit acknowledgment:

Stu> CO increases.

Tu> You did your reading. *[Explicit positive ack.]*

It's tempting to call this an implicit acknowledgment. The student must make a logical inference to recognize that this is a positive acknowledgment, but the primary purpose of this utterance is to issue an acknowledgment. Implicit acknowledgments are not intended primarily as acknowledgments and are characterized by the absence of anything that looks like an acknowledgment.

Implicit acknowledgments are the default case. Any utterance that does not function as an explicit acknowledgment, will function in some way as an implicit acknowledgment. We're not going to try to mark every single one of these; many implicit acknowledgments are not, at this time, worth spending the time marking.

C) Role:

- Judgment/correction of truth, accuracy, and completeness, i.e. acceptability of statement.
- Metacommunication: controlling discourse
  * Repair
  * Coordinating closure: proposing, deferring/denying, or marking closure

Judging the role of an acknowledgment is not an either/or proposition. All acknowledgments may play a role in both judgment and metacommunication, so you should mark whatever applies. However, only choose one subcategory each within judgment and metacommunication. This is partly to keep things simple, and partly because one subcategory subsumes or is mutually exclusive of another category. For instance, repair is initiated when there is apparently some miscommunication. This automatically implies deferring/denying closure.

Example 14 — marking judgment and metacommunication:

Stu> CO increases.

Tu> Try again. *[Incorrect and deny closure]*

Example 15 — marking only judgment/correction:

Stu> CO increases.

Tu> *[I can't come up with an example. Can you?]*

Example 16 — marking only metacommunication:

Stu> CP omcreases. *[Correct, but hand misplaced on keyboard]*

Tu>Come again? *[Tutor initiates repair]*

Example 17 — marking only metacommunication:

Stu> CO increases. *[Correct, but prediction out of sequence]*

Tu> You must first predict the determinants of CO before predicting CO.

5 – Coding Instructions

The Coding Process

The acknowledgments have already been marked for you. All that you need to do is fill in the values for each acknowledgment category in the markup header in your text. If there are multiple acknowledgments in an utterance, each acknowledgment will be listed as a separate acknowledgment and must be categorized as a separate acknowledgment.

Markup Instructions

Each acknowledgment is marked by preceding the utterance with one or more lines, starting at the beginning of a line in the following format:

<ACK Pol=X; Mark=Y; Role=comma-delimited roles [;optional comments]>

and followed by a line containing

</ACK>

by itself. The start and end SGML-style markup blocks will be indented one tab space (0.5 inches) relative to the text that is being referenced. E.g.,

<ACK Pol=+; Mark=EM; R=J,C>
K1-tu-20-1: Super.
</ACK>

Categories of acknowledgment

Each marked acknowledgment must have an entry for each category listed below. Only one type per category is permitted unless otherwise specified.

Polarity

Pol=+	Positive
Pol=0	Neutral
Pol=–	Negative

Explicitness

Mark=EM	Explicit acknowledgment with a clear marker/keyword. Not only is it explicit, but there is a word that can be pointed to as a keyword marking the utterance as an acknowledgment.
Mark=E	Explicit acknowledgment without a clear keyword/marker. It is explicit if a primary function of the utterance was to indicate acknowledgment.
Mark=I	Implicit acknowledgment. Any utterance that is not an explicit acknowledgment is in some way an implicit acknowledgment.

There is a gray area between explicit and implicit. At a certain point, it's a judgment call. If you think that one of the primary functions of the utterance was to convey an acknolwedgment, then you should mark it as explicit.

Roles

Role=J	Judgment that OK/acceptance of previous utterance
Role=R	Repair
Role=PC	Propose closure
Role=DC	Defer/deny closure
Role=C	Closure of joint action

Annotate roles by typing Role= followed by any of J, R, and PC/DC/C that apply. J and R are mutually exclusive. So are PC, DC and C. At least one of the roles must be filled. E.g., *R=J,DC*.

A.2 Coding Results Table

```
01/31/98 ,SB & MWE,,,,,BP,,,,,BM,,,,
Utterance,Pol,Mark,Judge,Closure,Repair,Pol,Mark,Judge,Closure,Repair,Po
l,Mark,Judge,Closure,Repair
k25-st-7-1,pos,em,0 ,c,0 ,pos,em,0 ,c,0 ,pos,99 ,j,0 ,0
k25-st-9-1,pos,em,0 ,c,0 ,pos,em,0 ,c,0 ,pos,99 ,j,0 ,0
k25-st-11-1,pos,em,0 ,c,0 ,pos,em,0 ,c,0 ,pos,99 ,j,0 ,0
k25-st-13-1,pos,em,0 ,c,0 ,pos,em,0 ,c,0 ,pos,99 ,j,0 ,0
k25-st-15-1,pos,i,0 ,c,0 ,pos,em,0 ,c,0 ,pos,99 ,j,0 ,0
k25-st-16-1,pos,em,j,c,0 ,pos,em,0 ,c,0 ,pos,99 ,j,0 ,0
k25-tu-16-2,pos,i,j,c,0 ,pos,e,j,c,0 ,pos,i,j,0 ,0
k25-tu-18-1,pos,i,j,c,0 ,pos,i,j,dc,0 ,pos,i,j,0 ,0
k25-tu-20-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k25-tu-22-1,neg,i,j,dc,0 ,neg,i,j,dc,0 ,neg,i,j,0 ,0
k25-tu-22-2,neg,em,j,dc,0 ,pos,e,j,dc,0 ,neg,e,j,0 ,0
k25-tu-24-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k25-tu-26-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k25-tu-28-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k25-tu-30-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k25-tu-32-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,r,0
k25-tu-34-1,neg,em,j,dc,0 ,neg,i,j,dc,0 ,neg,i,j,0 ,0
k25-tu-36-1,pos,i,j,c,0 ,pos,i,j,dc,0 ,pos,i,j,0 ,0
k25-tu-38-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k25-tu-40-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k25-tu-42-1,pos,em,0 ,c,0 ,pos,e,j,c,0 ,pos,em,j,0 ,0
k25-tu-42-2/3,neg,em,j,dc,0 ,neg,e,j,dc,0 ,neg,i,j,0 ,0
k25-tu-44-1,pos,em,j,0 ,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k25-ti-46-1,neg,e,j,dc,0 ,neg,e,j,dc,0 ,neg,i,j,0 ,0
k25-tu-48-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k25-tu-48-2,neg,e,j,0 ,0 ,neg,i,j,dc,0 ,neg,i,j,0 ,0
k25-tu-50-1,pos,i,j,dc,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k25-tu-52-1,pos,em,j,0 ,0 ,pos,em,j,c,0 ,pos,em,j,c,0
k25-tu-54-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k25-tu-56-1,neg,em,j,dc,0 ,neg,em,j,dc,0 ,neg,em,j,0 ,0
k25-tu-58-1,neg,em,j,dc,0 ,neg,em,j,dc,0 ,neg,em,j,0 ,0
k25-tu-60-1,pos,i,j,0 ,0 ,pos,e,j,dc,0 ,99 ,99 ,99 ,99 ,99
k25-tu-62-1/2,neg,e,j,dc,0 ,neg,e,j,dc,0 ,neg,e,j,0 ,0
k25-st-63-1,neg,e,0 ,0 ,r,pos,e,j,dc,0 ,neg,i,99 ,99 ,99
k25-tu-66-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k25-tu-68-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k25-tu-68-2,pos,em,0 ,c,0 ,pos,em,j,c,0 ,pos,e,j,c,0
k25-st-69-1,pos,em,0 ,c,0 ,pos,em,0 ,c,0 ,pos,em,j,c,0
k25-tu-72-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k25-tu-74-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k25-tu-76-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,99 ,99 ,99
k25-tu-78-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k25-tu-80-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k25-tu-82-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k25-tu-84-1,neg,e,j,dc,0 ,neg,i,j,dc,0 ,neg,i,j,0 ,0
k25-tu-86-1,neg,em,j,dc,0 ,neg,i,j,dc,0 ,neg,i,j,0 ,0
k25-tu-88-1,pos,em,j,0 ,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
```

```
k25-tu-98-1,pos,em,0 ,c,0 ,pos,em,j,c,0 ,pos,em,99 ,99 ,99
k25-tu-100-1,neg,em,j,dc,0 ,neg,e,j,dc,0 ,neg,e,99 ,99 ,99
k25-tu-102-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k25-tu-104-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k25-tu-106-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k25-tu-108-1a,pos,em,0 ,c,0 ,pos,em,j,c,0 ,pos,em,j,c,0
k25-tu-108-1b,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,c,0
k25-st-109-1,neg,e,0 ,0 ,r,pos,e,0 ,dc,0 ,99 ,99 ,99 ,99 ,99
k25-tu-112-1,pos,em,j,0 ,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k25-tu-112-2,neg,em,j,0 ,0 ,neg,em,j,c,0 ,pos,i,j,0 ,0
k25-tu-114-1,neg,em,j,dc,0 ,neg,em,j,dc,0 ,neg,em,j,0 ,0
k25-tu-116-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k25-tu-116-2,pos,i,0 ,c,0 ,pos,e,0 ,dc,0 ,pos,i,j,c,0
k25-st-117-1,pos,em,0 ,c,0 ,pos,em,0 ,c,0 ,pos,i,99 ,99 ,99
k25-tu-120-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,99 ,99 ,99
k25-tu-122-1,neg,i,j,dc,0 ,neg,i,j,dc,0 ,neg,i,0 ,0 ,r
k25-st-123-1,pos,em,0 ,c,0 ,pos,em,j,c,0 ,pos,e,0 ,0 ,r
k25-tu-126-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k25-tu-128-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k25-tu-130-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k25-tu-132-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k25-tu-134-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k25-tu-136-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k25-tu-138-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k25-tu-140-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k25-tu-142-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k25-tu-144-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k25-tu-146-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k25-tu-148-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k25-tu-150-1,neg,e,j,dc,0 ,neg,e,j,dc,0 ,neg,i,j,0 ,0
k25-tu-152-1,neg,i,j,dc,0 ,pos,e,j,dc,0 ,neg,i,j,0 ,0
k25-tu-154-1/2,pos,em,j,dc,0 ,pos,em,j,dc,0 ,pos,em,j,0 ,0
k25-tu-156-1,pos,em,j,dc,0 ,pos,enm,j,dc,0 ,pos,em,j,0 ,0
k25-tu-160-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,c,0
k25-st-163-1,pos,em,0 ,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k25-tu-164-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k25-tu-164-2,pos,i,0 ,c,0 ,pos,e,99 ,99 ,99 ,pos,em,j,c,0
,,,,,,,,,,,,,,,
k26-tu-6-1,pos,em,0 ,c,0 ,pos,em,j,c,0 ,pos,i,j,0 ,0
k26-st-7-1,pos,em,0 ,c,0 ,pos,em,0 ,c,0 ,pos,i,j,0 ,0
k26-st-9-1,pos,em,0 ,c,0 ,pos,em,0 ,c,0 ,pos,i,j,0 ,0
k26-st-11-1,pos,em,0 ,c,0 ,pos,em,0 ,c,0 ,pos,i,j,0 ,0
k26-st-13-1,pos,em,0 ,c,0 ,pos,em,0 ,c,0 ,pos,i,j,0 ,0
k26-st-15-1,pos,em,0 ,c,0 ,pos,em,0 ,c,0 ,pos,i,j,0 ,0
k26-tu-18-1,pos,em,0 ,c,0 ,pos,em,0 ,c,0 ,pos,i,j,0 ,0
k26-tu-20-1,pos,i,j,c,0 ,pos,i,0 ,c,0 ,pos,i,j,0 ,0
k26-tu-22-1,neg,i,j,dc,0 ,neg,i,j,dc,0 ,neg,i,0 ,0 ,r
k26-tu-24-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-26-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-28-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-30-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-32-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-34-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-36-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-38-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
```

```
k26-tu-40-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-42-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-44-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-46-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-48-1,pos,em,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-50-1,pos,em,0 ,c,0 ,pos,em,j,dc,0 ,pos,i,j,0 ,0
k26-tu-52-1,neg,e,j,dc,0 ,neg,i,j,dc,0 ,neg,i,j,0 ,0
k26-tu-54-1,neg,em,j,dc,0 ,neg,em,j,dc,0 ,neg,em,j,0 ,0
k26-tu-56-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k26-tu-58-1/2,neg,em,j,dc,0 ,neg,e,j,dc,0 ,pos,em,j,0 ,0
k26-tu-60-1,pos,e,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-62-1,neg,e,j,dc,0 ,pos,e,j,dc,0 ,99 ,99 ,99 ,99 ,99
k26-tu-64-1,neg,e,j,dc,0 ,neg,i,j,dc,0 ,pos,i,j,0 ,0
k26-tu-68-1,neg,em,j,dc,0 ,neg,em,j,dc,0 ,neg,em,j,0 ,0
k26-tu-72-1,neg,em,j,dc,0 ,neg,em,j,dc,0 ,neg,em,j,0 ,0
k26-tu-74-1,pos,e,j,c,0 ,pos,i,j,dc,0 ,pos,i,j,0 ,0
k26-tu-76-1,pos,em,j,c,0 ,pos,e,j,0 ,0 ,pos,em,j,0 ,0
k26-tu-78-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k26-st-79-1,pos,em,0 ,c,0 ,pos,em,0 ,c,0 ,pos,e,j,c,0
k26-tu-82-1,pos,e,j,c,0 ,pos,i,j,c,0 ,pos,e,j,0 ,0
k26-tu-84-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-86-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-88-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-90-1,neg,e,j,dc,0 ,neg,i,j,dc,0 ,neg,i,j,0 ,0
k26-tu-94-1,neg,em,j,dc,0 ,neg,e,j,dc,0 ,neg,i,j,0 ,0
k26-tu-96-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,i,j,0 ,0
k26-tu-98-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-100-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-102-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-104-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-106-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-108-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-110-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-112-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-114-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k26-tu-116-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,i,j,0 ,0
k26-tu-118-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,c,0
k26-st-119-1,pos,em,0 ,c,0 ,pos,em,0 ,c,0 ,pos,i,j,0 ,0
k26-tu-122-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-124-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-126-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-128-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-130-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-132-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-134-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-136-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-138-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-140-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-142-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-144-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-146-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k26-tu-148-1,pos,em,j,c,0 ,pos,i,j,c,0 ,pos,em,j,c,0
k26-tu-150-1,neg,e,j,dc,0 ,neg,i,j,dc,0 ,99 ,99 ,99 ,99 ,99
k26-tu-152-1,pos,em,j,c,0 ,pos,e,j,c,0 ,pos,em,j,0 ,0
k26-tu-152-2,neg,em,j,c,0 ,neg,em,j,c,0 ,neg,em,j,0 ,0
```

```
k26-tu-154-4,pos,e,0 ,c,0 ,pos,em,0 ,c,0 ,pos,e,j,c,0
,,,,,,,,,,,,,,,
k27-st-7-1,pos,em,0 ,c,0 ,pos,em,0 ,c,0 ,pos,i,j,0 ,0
k27-st-9-1,pos,em,0 ,c,0 ,pos,em,0 ,c,0 ,pos,i,j,0 ,0
k27-st-11-1,pos,em,0 ,c,0 ,pos,em,0 ,c,0 ,pos,i,j,0 ,0
k27-st-13-1,neg,e,0 ,dc,r,pos,i,0 ,c,0 ,pos,i,0 ,0 ,r
k27-st-15-1,pos,em,0 ,c,0 ,pos,em,0 ,c,0 ,pos,i,j,0 ,0
k27-tu-16-1,pos,em,0 ,c,0 ,pos,em,j,c,0 ,pos,i,j,0 ,0
k27-tu-18-1,neg,e,j,dc,0 ,neg,i,j,dc,0 ,neg,em,j,0 ,0
k27-tu-19-1,pos,e,0 ,dc,r,pos,e,j,dc,0 ,pos,em,0 ,0 ,r
k27-tu-20-1,pos,em,0 ,c,0 ,pos,em,j,c,0 ,99 ,em,j,0 ,0
k27-tu-22-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-24-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k27-tu-24-2,pos,i,0 ,c,0 ,pos,e,0 ,0 ,0 ,pos,e,j,0 ,0
k27-tu-26-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-28-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-30-1,neg,e,j,dc,0 ,neg,e,j,dc,0 ,neg,em,j,0 ,0
k27-tu-30-2,neg,e,0 ,dc,0 ,neg,i,j,dc,0 ,neg,em,j,0 ,0
k27-tu-32-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-34-1,pos,em,0 ,c,0 ,pos,em,j,c,0 ,pos,i,j,0 ,0
k27-tu-36-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-38-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-40-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-42-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-44-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-46-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-48-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-50-1,pos,em,0 ,c,0 ,pos,em,j,c,0 ,pos,i,j,0 ,0
k27-tu-52--1,neg,e,j,dc,0 ,99 ,99 ,99 ,99 ,99 ,neg,i,j,0 ,0
k27-tu-54-1,pos,e,j,dc,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-56-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k27-tu-58-1,pos,e,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-60-1,pos,e,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-62-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k27-tu-64-1,pos,em,j,0 ,0 ,pos,e,j,c,0 ,pos,e,0 ,0 ,r
k27-tu-64-2,neg,em,j,c,0 ,neg,em,j,c,0 ,pos,e,0 ,0 ,r
k27-tu-66-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k27-tu-68-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k27-tu-70-1,neg,em,j,dc,0 ,neg,em,j,dc,0 ,neg,em,j,0 ,0
k27-st-71-1,pos,e,0 ,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k27-tu-74-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k27-tu-74-2,pos,e,0 ,c,0 ,pos,e,j,0 ,0 ,pos,e,j,c,0
k27-st-75-1,pos,em,0 ,c,0 ,pos,em,0 ,c,0 ,pos,i,j,0 ,0
k27-tu-78-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-80-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-82-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-84-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-86-1,neg,e,j,dc,0 ,neg,i,j,dc,0 ,pos,i,j,0 ,0
k27-tu-88-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,i,j,0 ,0
k27-tu-90-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-92-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-94-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-96-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-98-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-100-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
```

```
k27-tu-102-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-104-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-106-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,c,0
k27-tu-108-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k27-tu-110-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k27-tu-112-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-114-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-116-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-118-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-120-1,neg,em,0 ,dc,r,neg,e,0 ,dc,r,neg,e,0 ,0 ,r
k27-tu-122-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-124-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-126-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-128-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-130-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-132-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-134-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-136-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-138-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-140-1,pos,em,0 ,c,0 ,pos,em,j,c,0 ,pos,i,j,0 ,0
k27-tu-140-2,neg,e,j,dc,0 ,neg,e,j,dc,0 ,neg,i,j,0 ,0
k27-tu-142-1,neg,e,j,c,0 ,neg,i,j,dc,0 ,neg,em,j,0 ,0
k27-tu-144-1a,neg,em,j,dc,0 ,neg,e,j,dc,0 ,neg,e,j,0 ,0
k27-tu-144-1b,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k27-tu-146-1,neg,e,0 ,dc,r,neg,i,j,dc,0 ,neg,i,j,0 ,0
k27-tu-148-1,neg,e,j,dc,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k27-tu-150-1,pos,em,j,c,0 ,pos,e,j,c,0 ,pos,em,j,0 ,0
k27-tu-152-1,neg,em,j,dc,0 ,neg,e,j,dc,0 ,neg,e,j,0 ,0
k27-tu-152-2,neg,e,j,c,0 ,pos,e,0 ,c,0 ,pos,i,j,0 ,0
k27-tu-154-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k27-st-155-1,pos,em,0 ,c,0 ,pos,em,0 ,c,0 ,pos,em,j,0 ,0
k27-tu-156-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k27-tu-158-1,pos,em,j,0 ,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k27-tu-160-1,pos,em,j,c,0 ,pos,em,0 ,c,0 ,pos,em,j,0 ,0
k27-tu-162-1,pos,em,j,c,0 ,pos,em,0 ,c,0 ,pos,em,j,0 ,0
k27-tu-164-1,neg,em,j,c,0 ,pos,i,j,c,0 ,pos,e,j,0 ,0
k27-st-165-1,pos,em,j,c,0 ,pos,em,0 ,c,0 ,pos,em,j,0 ,0
k27-tu-166-1,pos,em,0 ,c,0 ,pos,em,0 ,c,0 ,pos,i,j,c,0
,,,,,,,,,,,,,,,
k28-tu-6-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k28-tu-8-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k28-tu-10-1,pos,e,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k28-tu-12-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k28-st-13-1,pos,em,0 ,c,0 ,neg,em,0 ,c,0 ,pos,em,j,0 ,0
k28-tu-14-1,pos,em,0 ,c,0 ,pos,em,j,c,0 ,pos,i,j,0 ,0
k28-st-15-1,pos,em,0 ,c,0 ,pos,em,0 ,c,0 ,pos,i,j,c,0
k28-tu-18-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k28-tu-20-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k28-tu-22-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k28-tu-24-1,neg,em,j,dc,0 ,neg,em,j,dc,0 ,neg,em,j,0 ,0
k28-tu-28-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k82-tu-28-2,pos,i,0 ,c,0 ,pos,e,0 ,dc,0 ,pos,em,j,0 ,0
k28-tu-30-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k28-tu-32-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k28-tu-34-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
```

```
k28-tu-36-2,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k28-tu-38-1,pos,em,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k28-tu-38-2,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k28-tu-40-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k28-tu-42-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k28-tu-44-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,i,j,0 ,0
k28-tu-44-2,pos,i,0 ,c,0 ,pos,e,0 ,dc,0 ,pos,i,j,0 ,0
k28-tu-46-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k28-tu-48-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k28-tu-50-1a,pos,i,0 ,c,0 ,pos,em,j,c,0 ,pos,i,j,0 ,0
k28-tu-50-1b,pos,e,0 ,c,0 ,pos,e,j,c,0 ,pos,i,j,c,0
k28-tu-52-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k28-st-53-1,neg,em,j,0 ,0 ,neg,em,j,c,0 ,pos,em,j,0 ,0
k28-tu-54-1,pos,em,j,c,0 ,pos,em,j,dc,0 ,pos,i,j,0 ,0
k28-tu-56-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k28-tu-58-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k28-tu-60-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k28-tu-62-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k28-st-63-1,pos,em,0 ,c,0 ,neg,em,0 ,c,0 ,pos,99 ,99 ,99 ,99
k28-st-65-1,pos,em,0 ,c,0 ,pos,em,0 ,c,0 ,pos,i,j,c,0
k28-tu-68-1,pos,i,j,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k28-tu-70-1,pos,em,0 ,c,0 ,pos,em,j,c,0 ,pos,i,j,0 ,0
k28-tu-72-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k28-tu-74-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k28-tu-76-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k28-tu-78-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k28-tu-80-1,pos,i,0 ,c,0 ,pos,i,j,c,0 ,pos,i,j,0 ,0
k28-tu-82-1,pos,i,0 ,c,0 ,neg,i,j,dc,0 ,pos,i,j,0 ,0
k28-tu-84-1,neg,em,j,dc,0 ,neg,em,j,dc,0 ,neg,em,j,0 ,0
k28-tu-86-1,neg,em,j,dc,0 ,neg,em,j,dc,0 ,neg,em,j,0 ,0
k28-tu-88-1,pos,e,j,dc,0 ,pos,i,j,dc,0 ,pos,i,j,0 ,0
k28-tu-90-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k28-tu-92-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k28-tu-92-2,neg,e,j,dc,0 ,pos,e,j,c,0 ,99 ,99 ,99 ,99 ,99
k28-st-93-1,pos,em,0 ,c,0 ,pos,em,0 ,c,0 ,pos,i,j,0 ,0
k28-tu-94-1/2,neg,e,j,dc,0 ,99 ,99 ,99 ,99 ,99 ,neg,i,j,0 ,0
k28-tu-95-1,pos,em,0 ,c,0 ,pos,em,j,c,0 ,pos,i,j,0 ,0
k28-tu-96-1,neg,i,j,dc,0 ,pos,e,j,c,0 ,neg,i,j,0 ,0
k28-tu-98-1,pos,em,j,c,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k28-tu-102-1,pos,em,j,dc,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k28-tu-104-1,pos,em,j,dc,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k28-tu-106-1,pos,em,j,dc,0 ,pos,em,j,c,0 ,pos,em,j,0 ,0
k28-st-107-1,pos,e,0 ,c,0 ,pos,e,j,99 ,0 ,pos,e,j,0 ,0
k28-tu-108-1,pos,em,0 ,c,0 ,pos,em,0 ,c,0 ,pos,i,j,c,0
```

APPENDIX B

DATA FROM EXPERIMENT 2

Appendix B.1 Raw Data for Experiment 2

```
K17-K20 Ack Agreement Test,,,,,,,,,,
Nov. 1997,,,,,,,,,,
,,,,,,,,,,
,Sentence,,Turn,,Turn,,,Sentence,,
,Has Ack,,Has Ack,,Agreement,,,Agreement,,
Reference,SB,BV,SB2,BV2,SB&BV,SB-BV,BV-SB,SB&BV,SB-BV,BV-SB
k17-tu-8-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k17-st-9-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k17-st-11-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k17-st-13-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k17-st-15-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k17-st-17-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k17-tu-20-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k17-tu-22-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k17-tu-24-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k17-tu-26-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k17-tu-28-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k17-tu-28-2,1 ,0 ,,,,,,0 ,1 ,0
k17-st-29-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k17-tu-32-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k17-tu-32-2,0 ,1 ,,,,,,0 ,0 ,1
k17-tu-34-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k17-tu-36-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k17-tu-38-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k17-tu-38-2,0 ,1 ,,,,,,0 ,0 ,1
k17-tu-40-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k17-tu-42-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k17-tu-44-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k17-tu-46-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k17-tu-48-1,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1 ,0
k17-st-49-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k17-tu-52-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k17-tu-54-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k17-tu-56-1,1 ,0 ,1 ,1 ,1 ,0 ,0 ,0 ,1 ,0
k17-tu-56-2,0 ,1 ,,,,,,0 ,0 ,1
k17-st-57-2,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1 ,0
k17-tu-58-1to3,0 ,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1
k17-st-59-1,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1 ,0
k17-tu-60-1,1 ,0 ,1 ,1 ,1 ,0 ,0 ,0 ,1 ,0
k17-tu-60-1to2,0 ,1 ,,,,,,0 ,0 ,1
k17-tu-62-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k17-tu-64-1,1 ,0 ,1 ,1 ,1 ,0 ,0 ,0 ,1 ,0
k17-tu-64-2,1 ,0 ,,,,,,0 ,1 ,0
k17-tu-64-1to2,0 ,1 ,,,,,,0 ,0 ,1
k17-tu-st-65-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k17-st-67-1,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1 ,0
k17-tu-68-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
```

```
,,,,,,,,,,
k18-tu-8-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k18-st-9-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k18-st-11-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k18-st-13-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k18-st-15-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k18-st-17-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k18-st-19-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k18-tu-22-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k18-tu-22-2,0 ,1 ,,,,,,0 ,0 ,1
k18-tu-24-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k18-tu-26-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k18-tu-28-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k18-tu-30-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k18-tu-32-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k18-st-33-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k18-st-35-1,0 ,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1
k18-tu-38-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k18-tu-38-2,0 ,1 ,,,,,,0 ,0 ,1
k18-tu-40-1,1 ,0 ,1 ,1 ,1 ,0 ,0 ,0 ,1 ,0
k18-tu-40-1to2,0 ,1 ,,,,,,0 ,0 ,1
k18-tu-42-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k18-st-43-1,0 ,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1
k18-tu-44-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k18-tu-46-1,1 ,0 ,1 ,1 ,1 ,0 ,0 ,0 ,1 ,0
k18-tu-46-1to2,0 ,1 ,,,,,,0 ,0 ,1
k18-tu-48-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k18-tu-50-1,0 ,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1
k18-tu-52-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k18-tu-54-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k18-tu-56-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k18-tu-58-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k18-st-59-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k18-tu-62-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k18-tu-66-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k18-tu-68-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k18-tu-70-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k18-tu-72-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k18-tu-74-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k18-tu-76-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k18-st-77-1,0 ,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1
k18-tu-78-2,0 ,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1
,,,,,,,,,,
k19-tu-8-1,0 ,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1
k19-st-9-1,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1 ,0
k19-st-11-1,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1 ,0
k19-st-13-1,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1 ,0
k19-st-15-1,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1 ,0
k19-st-17-1,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1 ,0
k19-tu-20-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k19-tu-22-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k19-tu-24-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k19-tu-26-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k19-tu-28-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k19-tu-30-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
```

```
k19-tu-30-2,0 ,1 ,,,,,,0 ,0 ,1
k19-tu-32-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k19-tu-34-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k19-tu-36-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k19-st-37-1,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1 ,0
k19-tu-38-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k19-tu-40-1,0 ,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1
k19-tu-40-2,0 ,1 ,,,,,,0 ,0 ,1
k19-tu-42-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k19-st-43-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k19-tu-44-1,0 ,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1
k19-tu-46-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k19-tu-46-2,0 ,1 ,,,,,,0 ,0 ,1
k19-tu-48-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k19-tu-48-2,0 ,1 ,,,,,,0 ,0 ,1
k-19-tu-50-1,1 ,0 ,1 ,1 ,1 ,0 ,0 ,0 ,1 ,0
k19-tu-50-1to2,0 ,1 ,,,,,,0 ,0 ,1
k19-tu-52-1,0 ,1 ,1 ,1 ,1 ,0 ,0 ,0 ,0 ,1
k19-tu-52-1to3,1 ,0 ,,,,,,0 ,1 ,0
k19-tu-54-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k19-tu-56-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k19-tu-58-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k19-tu-60-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k19-tu-62-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k19-tu-64-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k19-tu-66-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k19-tu-68-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k19-st-69-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k19-tu-70-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k19-tu-72-1,0 ,1 ,1 ,1 ,1 ,0 ,0 ,0 ,0 ,1
k19-tu-72-1to2,1 ,0 ,,,,,,0 ,1 ,0
k19-tu-74-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k19-tu-76-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k19-tu-78-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k19-tu-80-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k19-tu-80-2,0 ,1 ,,,,,,0 ,0 ,1
k19-tu-82-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k19-tu-84-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k19-tu-86-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k19-st-87-1,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1 ,0
k19-tu-88-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k19-st-89-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k19-st-91-1,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1 ,0
k19-tu-92-1,0 ,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1
,,,,,,,,,,
k20-tu-6-1,0 ,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1
k20-st-7-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-st-9-1,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1 ,0
k20-st-11-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-st-13-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-st-15-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-st-17-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-tu-20-2,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-tu-22-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-tu-24-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
```

```
k20-tu-26-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-tu-28-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-tu-30-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-tu-32-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-tu-34-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-tu-36-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-tu-38-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-st-39-1,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1 ,0
k20-tu-40-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-st-41-1,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1 ,0
k20-tu-42-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-st-43-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-tu-46-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-tu-46-2,1 ,0 ,,,,,,0 ,1 ,0
k20-st-47-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-tu-48-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-tu-48-2,1 ,0 ,,,,,,0 ,1 ,0
k20-tu-48-3,0 ,1 ,,,,,,0 ,0 ,1
k20-st-49-1,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1 ,0
k20-tu-50-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-tu-52-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-tu-54-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-tu-56-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-tu-60-1,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1 ,0
k20-st-61-1,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1 ,0
k20-tu-62-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-tu-64-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-tu-66-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-tu-68-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-st-69-1,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1 ,0
k20-tu-72-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-st-73-1,0 ,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1
k20-st-75-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-tu-78-1,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1 ,0
k20-tu-80-1,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1 ,0
k20-tu-82-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-tu-84-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-tu-86-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-tu-88-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-tu-90-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-st-91-1,1 ,0 ,1 ,0 ,0 ,1 ,0 ,0 ,1 ,0
k20-tu-92-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-tu-94-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-tu-96-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-st-97-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-st-99-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
k20-tu-100-1,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,0
```

APPENDIX C

DATA FROM EXPERIMENT 3

C.1 Instructions Document Given to Coders

Guide to Categorizing Acknowledgments in Transcripts

0 – Introduction

When communicating, humans don't just run around magically tossing out utterances or parsing utterances from others. Many things go into communication; I can summarize by saying that humans construct and depend on a very complex model in order to communicate successfully. It is my belief (and hope!) that this is a computationally tractable modeling problem, particularly for microdomains where we can hope to develop a model that is sufficiently complete with respect to domain and communication knowledge needed to support competent communication. The study of acknowledgments is a part of the work necessary for developing more powerful and usable computational models of communication.

The study of language is made harder by the fact that each utterance has more than just one purpose or role. Communicating parties always have multiple issues that they deal with in parallel, sometimes producing a very complex agenda. These agenda items generate multiple information streams that must be encoded onto the available communication modalities (communication channels: e.g., written, spoken, body language) with consideration to the signal bandwidth of each modality. My theory of communication holds that maximizing efficiency in communication is an important goal of communication. Maximizing efficiency is even more important in bandwidth-constrained situations, such as a keyboard-to-keyboard dialogue where the signaling modality is restricted largely to the typed text. Consequently, we modulate multiple streams of information – each with different goals – onto a restricted bandwidth. Because each utterance plays multiple roles, the question becomes "Which roles does this utterance play", not "Does this utterance have a role to play." A simple utterance like "OK" can indicate that the speaker understood the prior utterance, judges the prior utterance to be true, and is moving to close the current topic. In our study of acknowledgments, we are focusing on the roles of acknowledgment in rendering judgment on the truth, accuracy, and *sufficiency* of the student's knowledge, and in performing metacommunication functions. Metacommunication is communication about communication; it is used to signal attention, to correct misunderstanding (repair), to control the flow of discourse (e.g., starting or ending a topic), and for a few other lesser functions.

1 – Definition of an Acknowledgment

An acknowledgment is an indication that a message has been received. An utterance that signals acknowledgment often also carries an indication of agreement or the lack of agreement, closure or deferral of closure, or initiation of repair.

2 – The Purpose of Studying Acknowledgments

Acknowledgments are necessary for communicating successfully. At an abstract level, studying acknowledgments enhances our understanding of natural language (NL) and our competence in NL. At a more concrete level, our study enhances the ability of tutoring systems to issue and process acknowledgments, thus making the systems more powerful and useful. Most specifically, this study will guide development of CIRCSIM-Tutor as we seek to improve its linguistic and tutoring competence.

Developing a formal model of acknowledgment use will guide our development of CIRCSIM-Tutor by giving us a model for knowing what kinds of acknowledgment to generate and when to generate them. This could also be useful for input understanding.

3 – Theoretical Foundation

When communicating, we strive to ensure success. We do this by tailoring our utterances to maximize relevance, clarity, precision, and efficiency (see Grice's Maxims) by controlling word choice and syntax, by ensuring a shared context, and by setting an appropriate information transmission rate.

I have chosen to define acknowledgments in the light of Clark's presentation of discourse as a joint activity composed of joint actions (*Using Language*, 1996). He focuses on the mutual coordination of their individual actions by the participants in linguistic activity. Clark says: "There is coordination of both *content*, what the participants intend to do, and *processes*, the physical and mental systems they recruit in carrying out those intentions" (p. 59). Similar ideas are expressed by Barbara Fox (1993) who states as a major finding of her book, *The Human Tutorial Dialogue Project*, that "tutoring involves constant, and local, management. This requires a pervasive mutual orientation between tutor and student, such that every session (indeed, every utterance) is a thoroughly interactional achievement, produced by both tutor and student" (p. 3).

Clark says that joint actions can be divided into phases – entry, body, and exit – and these phases are what get coordinated. Entries and exits are coordinated by syntactic, morphological, and intonational markers. A joint action is complete when there is a mutual recognition of closure on that action. "It is a fundamental principle of intentional action that people look for evidence that they have done what they intended to do" (p. 222). He restates Norman's *Principle of Closure*: "Agents performing an action require evidence, sufficient for current purposes, that they have succeeded in performing it." (p. 222) and then introduces the *Principle of Joint Closure*: "The participants in a joint action try to establish the mutual belief that they have succeeded well enough for current purposes" (p. 226).

The goal of every discourse segment is to successfully reach closure. Consequently, each utterance will in some way deal with whether we are communicating successfully and whether we have reached closure.

4 – Context Information You Should Know

Knowledge of context is critical when studying acknowledgments and related phenomena. Among the chunks of knowledge that you need to understand when categorizing acknowledgments are domain/subject context and tutoring context. See (Khuwaja, 1994) for detail about these contexts (domains of expertise) in tutoring.

4.1 – Information about Physiology

The goal of the tutoring is that the students develop a useable understanding of blood pressure regulation by the barroreceptor reflex. This reflex gets activated as blood pressure increases or decreases. The central nervous system is notified and commands changes in parts of the blood circulatory system in order to maintain steady blood pressure. Unless the blood pressure perturbing factor overwhelms the system, this mechanism will maintain an approximately steady blood pressure.

The students are presented with a model of the blood pressure system that abstracts the response mechanism as the changes in seven variables measured during three phases: 1) Direct Response (DR), the change in the variables induced by the perturbation, 2) Reflex Response (RR), the change induced by the central nervous system intervention, and 3) Steady State (SS), the change in the variables relative to their values before the perturbation. The seven variables are as follows:

1. Inotropic State (IS) – the ion state (ion density) in the heart muscle. It the heart muscle contraction force.
2. Central Venous Pressure (CVP) – the pressure of the blood returning to the heart.
3. Stroke Volume (SV) – the volume of blood pumped out of the heart per stroke. SV is determined by CVP and IS.
4. Heart Rate (HR) – the number of heart beats per minute.
5. Cardiac Output (CO) – the volume of blood pumped out of the heart per minute. CO=SV*HR.
6. Total Peripheral Resistance (TPR) – a measure of resistance to blood flow.
7. Mean Arterial Pressure (MAP) – blood pressure in the arteries. MAP=CO*TPR.

4.2 – Information about the Tutoring Protocols

A tutoring protocol is a set of rules about how tutoring is performed. The protocol our tutors currently use is as follows:

1. Solve the Direct Response (DR) stage

   Collect & tutor the primary variable
   - First collect student's prediction for primary variable and its value
   - Then tutor the primary variable and/or its value as needed
   - *[Student must get this right before proceeding]*

   Collect & tutor the rest of the prediction table variables for DR
   - First collect all the student's predictions for the rest of the variables
   - After all the predictions are collected, tutor those predictions as needed

2. Solve the Reflex Response (RR) stage

   Collect & tutor all the prediction table variables for RR
   - First collect the student's predictions for all prediction table variables
   - After all the predictions are collected, tutor those predictions as needed

3. Solve the Steady State (SS) stage

   Collect & tutor all the prediction table variables for SS
   - First collect the student's predictions for all prediction table variables
   - After all the predictions are collected, tutor those predictions as needed

It is important to understand is that under the current protocol the tutor carefully avoids making any judgment about the predictions until all predictions have been collected; the exception is the primary variable in DR. Consequently, the tutor's utterances are intended as judgments only when collecting the primary variable in DR and during the tutoring phase in each stage after collecting all the predictions. By contrast, closure and repair can and do occur at any point during the tutoring session. This restriction on judgment is one of the main ways in which the current protocol (Protocol 3) differs from the previous protocols. During the tutoring phases, the tutor may also go fishing for possible problems in the student's understanding.

For your reference, Protocol 1 is used in transcripts K1-8, Protocol 2 in K9-25, and Protocol 3 in K30-46 and N1-31. For more details about the three protocols see (Khuwaja, 1995).

5 – Categories of Acknowledgment

We are not studying all acknowledgments. We are studying a subset of acknowledgment, those that special roles in tutorial discourse. Consequently, we are interested in categorizing acknowledgments according to each of the following three classes:

1. Judgment – is the speaker signaling judgment?
2. Closure – does the utterance indicate closure on a joint action, or perhaps deny closure?
3. Repair – does the utterance initiate discourse repair?

For each selected utterance and for each of the three classes of acknowledgment, the utterance is categorized for polarity and explicitness. Polarity is a measure of whether the acknowledgment is positive, negative, or neutral and is represented by the values "+", "–" and "0". Explicitness measures whether the acknowledgment explicitly plays a role in that class or whether it only plays its role implicitly, and is represented by the values "E" and "I". See below for more detailed descriptions and examples.

The utterances are also classified by whether or not they contain one or more keywords that mark (signal, indicate) the categorization of the utterance. The values for this class are yes and no, and are represented by "Y" and "N".

Following is a presentation of the acknowledgment classes and values within each class, a set of examples – and non-examples as deemed relevant – for each category, and a discussion of problem areas or issues requiring cognizance by the person performing categorization. Most of the examples are simple contrived ones. I try to minimize the differences between the examples so as to make it easier to understand the features that characterize each categorization. It is also worth noting that a majority of the acknowledgments that we will look at are uttered by the tutor, but we are equally interested in acknowledgments from the student. Studying both is important to a description of tutoring, and for CIRCSIM-Tutor we need both; studying acknowledgments from the tutor provides knowledge to help us with text generation and acknowledgments from the student guide development of input understanding.

5.1 – Judgment

When a person provides an evaluation, we call that judgment. It is possibly the case that almost any utterance could be construed as a judgment of some sort, but we are primarily interested in judgments that are tied in with acknowledgments. Positive judgments are those which signal correctness, providing positive feedback. A negative judgment signals incorrectness and provides negative feedback. There is the question of whether there are neutral judgments. These would be cases where the tutor would like to defer judgment until the student has clarified or developed the information further, or else where the tutor wishes to withhold expressing a judgment. For my purposes, I am calling a neutral judgment anything that is not a positive or negative judgment. It is not quite the same as "no judgment" – which occurs when there is simply no judgment involved – but I suggest that you should mark them the same way. Judgments can also be explicit or implicit. An explicit judgment is one where the judgment signal is clearly part of the focus of the utterance. An utterance that signals an implicit judgment is one that does not directly express a judgment, but from which it is intended that a judgment be inferred.

Example 1 – positive judgment:

Stu>	CO increases.	*[Correct answer]*
Tu>	Right.	

Example 2 – negative judgment:

Stu>	CO increases.	*[Incorrect answer]*
Tu>	Wrong.	

Example 3 – no judgment:

Stu>	CO increases.	*[Either correct or incorrect]*
Tu>	Next prediction.	*[Tutor is only collecting predictions, not judging them now]*

Example 4 – explicit judgment:

Stu>	CO increases.	*[Correct answer]*
Tu>	Right.	*[Tutor explicitly signals judgment]*

Example 5 – implicit judgment:

Stu>	CO increases.	*[Correct answer]*
Tu>	Did HR or SV increase?	*[Tutor points out contradiction]*

Discussion of problems in classifying judgment:

Not all judgments come neatly packaged. There are compound judgments where an utterance provides two or more judgments that may be of the same or opposite polarity. There are also judgments where part of the judgment is explicit and the other part is implicit. Some of the student's explanations are judged close to being correct, maybe true but not the desired answer. This is a "close miss"; we will consider these judgments to be negative judgments for now. We try to divide utterances with compound judgments into separate parts that can be categorized separately. If the utterance is not divided into separate parts and only one category can be recorded, you should record the judgment that you consider had the greatest influence in controlling the consequent flow of discourse.

Part of the problem lies in the need for the speaker and the listener to reach a mutually shared interpretation of the acknowledgment. Our medical tutors would try to issue neutral acknowledgments (e.g., deferring judgment on what the student had uttered) by saying things like "OK" or "Next" after the prediction was made. The tutor intended to acknowledge that the prediction had been understood and recorded, but not to issue a judgment/evaluation. The students tend to interpret this as indicating understanding, recording, and positive judgment of the predictions. It seems that unless there was a prior discussion of the rules of the game such that it was clear that predictions would be recorded without immediate judgment, the student will interpret these as implicit positive acknowledgments. For our purposes, we are interested in the intention/interpretation of the tutor, not the student's perspective; the student's perspective is important also, but we are currently focusing on the tutor with the intention of applying this information to CIRCSIM-Tutor.

Example 6 – compound judgment (same polarity):

Stu> CO and MAP increase. *[Both correct]*

Tu> Correct. *[Tutor judges both predictions to be true]*

Example 7 – compound judgment (different polarities):

Stu> CO and RAP increase. *[CO is correct, RAP is incorrect]*

Tu> CO is right. Let's talk about RAP. *[Tutor signals different judgments]*

Example 8 – compound judgment (different polarities and explicitness):

Stu> CO and RAP increase. *[CO is correct, RAP is incorrect]*

Tu> CO is right. *[Explicit judgment of CO. Then turns control back over to student as an implicit negative judgment.]*

Example 9 – messy negative judgment:

Stu> [Student makes statement about myocardial cells]

Tu> They might and then again they might not. We're assuming in this case that they don't. *[Tutor judges against the student]*

Example 10 – messy split compound judgment:

Stu> [Makes a set of statements]

Tu> You are right, but for all the wrong reasons.

In example 10, the tutor says that the student is right, but that is not a cause for celebration. The problem is that the student used invalid reasoning. Because correct reasoning is the key to consistent accuracy, the tutor considers the students answer to be more wrong than right. In cases like this, where you are asked to categorize a split judgment (positive and negative), decide what the overall judgment is and categorize on that basis. In this case, it would really be a negative judgment; the "You are right" is not the focus of the utterance.

There can be confusion over what appears to be an explicit marked acknowledgment, but which is not primarily an acknowledgment (example 11).

Example 11 — looks like judgment but isn't:

Tu> Have you used HEARTSIM or CIRCSIM?

Stu> No.

On the surface, example 11 might look like a negative explicit judgment with a marker; the "No" might be thought to mean that. However, in this context, the "No" is really the propositional content of the student's reply. The student is saying something like "It is the case that I have not used HEARTSIM or CIRCSIM." The student is not really saying anything about metacommunication or trying to make a judgment about some previous value. I suppose that one could construe the tutor's question as a statement that "It is the case that you have previously used HEARTSIM or CIRCSIM." and that the student is judging that statement to be incorrect, but this seems to be really stretching the model. It does play a minor role in marking attention and comprehension of the tutor's question, perhaps even in proposing closure, but its main purpose is to communicate that the student has not used either of the two named systems. We will *not* mark this as a judgment.

5.2 – Closure

Utterances in this category signal whether the current joint action has reached closure. Remember that Joint Action theory views all discourse as a tree of joint actions. I believe that tutoring sessions fit into the model particularly well. At the highest level, the participants have started a joint action whose goal is to perform a tutoring session. This is then decomposed into subsidiary joint actions: 1) perform the administrative tasks such as obtaining the student's name and establishing the "rules of the game", 2) do the DR stage, 3) do the RR stage, and 4) do the SS stage. When these subsidiary joint actions are completed, the participants then establish closure on the top-level joint action, the tutoring session. For each of these subsidiary joint actions, the same sort of joint action decomposition occurs; they are each broken down into a set of joint actions and after those joint actions are carried out, closure is established over the joint action. So closure is any signal the is recognized by the recipient as an indication that the joint action is done. When the student delivers an answer that is believed to be correct and sufficiently complete, it's the student's expectation that this answer is sufficient to bring the lowest level joint action to an end. The tutor could then explicitly confirm this, or could just continue on to another topic, thus providing implicit closure. Because one of the goals of communication is to maximize efficiency, especially in bandwidth constrained settings, it makes sense that implicit closure would be the default action and that explicit closure would be provided only when necessary.

Of course, closure requires acceptance by all parties, and it's always possible that one of the participants does not accept closure. Thus the tutor could signal closure on a topic and move to another topic, only to have the student return to the same joint action when the student next gets a chance to signal. Or the student might believe that an answer was correct and sufficiently complete, but get a explicit or implicit signal from the tutor that the answer was not sufficient for closure. In these cases, the utterance signals a denial of closure. So our closure category polarities are positive (signal closure), negative (signal closure failure/deny closure) or neutral (the utterance does not contribute significantly to closure). For each utterance that does signal something about closure, the signal can be explicit or implicit.

<u>Example 12 – positive closure:</u>

Stu> CO increases. *[Student makes a prediction]*

Tu> Ok. Next prediction.
[Tutor signals closure and initiates next joint action. Under protocol 3, the tutor doesn't judge most predictions while collecting them; each joint action requires collecting a prediction about a variable and its value, no more.]

<u>Example 13 – negative closure:</u>

Stu> CO increases. *[Student makes a prediction that violates protocol 3]*

Tu> CO is not the first variable to change.
[Tutor denies closure on collecting the first prediction. Under protocol 3, the student must correctly predict the primary variable to change in DR before going on to make any other predictions.]

<u>Example 14 – neutral closure</u> (this is tricky – I can't find any examples in our transcripts of an acknowledgment that is not either positively or negatively related to closure. I'll include one if I find one.):

Example 15 – explicit closure:

Stu> CO increases.

Tu> Ok. Next.

["Ok" provides the explicit closure. Here, "Next" acts as an introducer to start the next joint action by eliciting the next prediction; "Next" is not really about closure, though it does make it clear that the "Ok" really was intended to provide closure.]

Example 16 – implicit closure:

Stu> CO increases.

Tu> Next.

["Next" provides implicit closure. It's primary purpose is still to introduce the next joint action (collect another prediction), but since nothing is explicitly marking closure, the default rule is that the introducer implicitly provides closure on the previous joint action.]

Discussion of problems

Just as every utterance could in a sense be interpreted as an acknowledgment, so it might be possible to interpret every utterance as having meaning for closure. We'll try not to overdo this and only mark closure when the utterance has a significant role relating to closure.

Initiation of repair (discussed in the next section) seems to always signal deferral of closure. The focus of repair is not on closure, so it seems reasonable to categorize repair as negative implicit (implicit deferral) of closure. As a note, it is possible that some combinations of polarity and explicitness of closure do not occur in the transcripts, or even occur in human dialogue.

5.3 – Repair

Utterances in this category signal a communication failure. It is important to understand that we are not speaking of cognitive repair (focusing on misconceptions and such) but on communication failures. Negative repair focuses on initiating repair, but doesn't provide the actual repair. Positive repair is the utterance that actually repairs the communication failure. Most utterances are repair-neutral in that they don't refer to repair other than in a weak sense by signaling that no repair is needed. The repair can be performed explicitly or implicitly. Explicit repair involves reference to the problem, but utterances with implicit repair just show signs of the repair without actually commenting on the repair.

Example 17 – negative repair:

Stu> VO increases. *[Meant to type "CO"]*

Tu> Do you mean CO? *[Tutor could skip explicit repair, but wants to make sure]*

Example 18 – positive repair:

Tu> Do you mean CO? *[Same utterance as above]*

Stu> Oops. Yes, I meant CO. *[Student signals acceptance of repair]*

Example 19 – explicitrepair:

Stu> VO increases. *[Meant to type "CO"]*

Tu> Do you mean CO? *[Tutor clearly initiates repair]*

Example 20 – implicit repair:

Stu> VO increases. *[Meant to type "CO"]*

Tu> Yes, CO increases. *[Tutor performs implicit positive repair]*

Problems and extended discussion

Associated with Clark's joint action theory is a hierarchy of levels of communication. The listener performs at the following four levels: 1) paying attention to the signal, 2) recording and decoding the parts of the signal, 3) understanding the meaning and purpose of the signal, 4) performing uptake on the signal. To give examples of these in keyboard-to-keyboard tutoring, at level 1 the listener looks at the screen and pays attention to the typing. At level 2, the listener is performing character and word recognition. At level three, the listener is deciding the meaning and purpose of the utterance (e.g., to signal closure or to signal judgment). And at level four, the listener is performing response/uptake on the utterance. I believe that repair can occur at the lower three levels. For instance, I might have to initiate level 1 repair because I got side-tracked, wasn't watching the screen and the other person typed so much that the critical first part of the utterance has vanished irretrievably from the computer screen. (I don't believe we have any instances of level one repair in our transcripts.) Level two repair tends to involve critical typing errors (e.g., hands misplaced on keyboard or a strategic typo) that the listener must get corrected before being able to proceed. At level three, the words might be completely intelligible, but the listener is unable to determine the meaning/purpose of the signal. This does start to edge towards cognitive repair, but we're not going to worry about that here.

5.4 – Marking

There are words that carry an extra load of meaning in acknowledgments. For instance, if the student makes a prediction, the tutor could reply "You are absolutely correct." If we were to choose one word in that sentence as the key word, it would be "correct". We could actually replace all letters in the other words in the sentence and the overall meaning would still be communicated ("xxx xxx xxxxxxxxxx correct."). If the student made a statement and the tutor replied with "But what about Y?", the keyword would be "But"; it is a clear indication that the answer was somehow not completely acceptable. These keywords are important in the study of effective tutoring and are particularly important for both text generation and input understanding. Linguists speak of "discourse markers". I'm not speaking of exactly the same thing, but rather of a set of words that is a partial subset of the set of discourse markers. Our interest in this area is to identify any keywords that carry extra meaning in acknowledgments.

<u>Example 21 – utterance with a keyword:</u>

Stu> Increased CO decreases SV. *[Correct]*
Tu> That's **right.**

<u>Example 22 – utterance with a keyword:</u>

Stu> Increased CO decreases SV. *[Correct]*
Tu> **But** you're missing RAP. *[Tutor refuses to accept it]*

<u>Example 23 – utterance without a keyword:</u>

Stu> Increased CO decreases SV. *[Correct]*
Tu> What would that imply about RAP?
[Tutor accepts SV, but wants to collect RAP too.]

6 – Coding Instructions

6.1 – The Coding Process

The acknowledgments have already been marked for you. All that you need to do is fill in the values for each acknowledgment category in the markup header in your text. If there are multiple acknowledgments in an utterance, each acknowledgment will be listed as a separate acknowledgment and must be categorized as a separate acknowledgment. If you believe that the acknowledgments are incorrectly delimited, please do your best with what is marked and then put in a comment and use pen or pencil to mark what you believe the correct delimiters should have been. If there is a particular instance that you found extremely difficult or interesting, I would appreciate your written comment in the margin.

6.2 – Markup Instructions

Each acknowledgment is marked by preceding the utterance with one or more lines, starting at the beginning of a line in the following format:

<ACK Judgment=   ,Explicit=   ; Closure=   ,Explicit=   ; Repair=   ,Explicit=   ; Marked=   >
N22-tu-4-1: very good.
</ACK>

The start and end SGML-style markup are italicized to make them stand out.

Categories of acknowledgment
Each marked acknowledgment must have an entry for each category listed below. Only one classification per category is permitted.

1. Judgment
   Values are "+", "–" or "0". If other than "0", also mark explicitness using "E" or "I".
2. Closure
   Values are "+", "–" or "0". If other than "0", also mark explicitness using "E" or "I".
3. Repair
   Values are "+", "–" or "0". If other than "0", also mark explicitness using "E" or "I".
4. Marking
   Values are "Y" or "N".

Example 24 – Closure, no judgment or repair, marked:

N22-tu-2-2: Please give me your full name, sex, age, and social security number.
N22-st-3-1: K.A., female, 22, XXX-XX-XXXX
<ACK Judgment= 0 ,Explicit= ; Closure= + ,Explicit= E ; Repair= 0 ,Explicit= ; Marked= Y >
N22-tu-4-1: very good.
</ACK>

Here the student has provided some administrative information (needed for the study and to pay the student). When the tutor says "very good", the tutor is not really making a judgment about whether or not the information provided is correct, so we put a "0" but judgment to indicate that judgment is not significantly active in this acknowledgment. The utterance does indicate closure, so we put a "+" by closure and then mark an "E" to show that this is an explicit instance of closure. Repair is not relevant here, so we mark repair "0". Finally, the word "good" does carry special meaning in the acknowledgment, so we mark "Y" to show that there are keywords in the sentence and then circle all such keywords; in this case, the word "good".

Example 25 – Judgment and closure, nor repair, not marked:

N22-tu-18-1: Please start by looking over the seven variables and predicting which will change first, given the problem stated.
<Right prediction>
N22-st-19-1: hr
<ACK Judgment= + ,Explicit= I ; Closure= + ,Explicit= I ; Repair= 0 ,Explicit= ; Marked= N >
N22-tu-20-1: in which direction will it change?
</ACK>

In this example the tutor judges that the prediction is correct in accordance with the rules of protocol 3. If it were not correct, the tutor would have refused to accept the prediction. The tutor does not explicitly say that it is correct, but implies correctness by asking for the change; this is an implicit judgment. Similarly, the tutor is performing implicit closure of the joint action to predict the variable that will change first. There is no repair, and there are no keywords that carry special meaning regarding judgment, closure, or repair.

Example 26 – No judgment, closure, negative and positive repair, marked:

K2-tu-28-3: This phenomenon gives rise to the vascular function (curce) relationship.
K2-tu-28-4: Do you remember it?
<ACK Judgment= 0 ,Explicit= ; Closure= – ,Explicit= I ; Repair= – ,Explicit= E ; Marked= Y >
K2-st-29-1: I am not familiar with a "curce" relationship
</ACK>
<ACK Judgment= 0 ,Explicit= ; Closure= + ,Explicit= I ; Repair= + ,Explicit= E ; Marked= Y >
K2-tu-30-1: Sorry I mistyped it should be "curve", the vascular function curve.
</ACK>

This exchange contains an instance of repair. This is relatively rare in our transcripts. There is no judgment. There is first a move away from closure in K2-st-29-1 (negative closure). Then the tutor produces closure on the repair in K2-tu-30-1 (positive closure). Negative repair occurs in K2-st-29-1 when the student points out that there is a communication problem. Positive repair follows in K2-tu-30-1 when the tutor fixes the communication problem. Both utterances have a marking keyword; "not" in the student's utterance is a bit weak, but "sorry" is fairly strong at indicating positive repair.

Please go through the sample marked up transcript and evaluate the categorizations before proceeding to the uncategorized transcripts. If you have general questions about what you are supposed to do, feel free to ask me. In order not to influence the results, I will not answer questions about how to categorize a particular acknowledgment.

7 – References:

Fox, Barbara, A.. 1993. The Human Tutorial Dialogue Project: Issues in the Design of Instructional Systems. Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Grice, H.P. 1975. Logic and Conversation. In Cole, P., and Morgan, J.L., eds, Syntax and Semantics. III. Speech Acts. Academic Press, New York, New York: pp.41-58.

Khuwaja, Ramzan. 1994. A Model of Tutoring: Facilitating Knowledge Integration Using Multiple Models of the Domain. Ph.D., Illinois Institute of Technology, Chicago, IL.

Khuwaja, Ramzan, Rovick, Allen A., Michael, Joel A., and Evens, Martha W. 1995. A Tale of Three Tutoring Protocols: The Implications for Intelligent Tutoring Systems. In Yfantis, E. A., Ed. Intelligent Systems. Kluwer Academic Publishers, Netherlands:pp. 109-118.

APPENDIX D

MACHINE LEARNING RESULTS

D.1 C4.5 Polarity Raw Input Data

1, -1, 0.

Transcript: 25, 26, 27, 28.
Speaker: tu, st.
Phase: a, i, p, t.
Answer-accept: 1, -1, 0.
Stu-Asmt: continuous.
Delta-Stu-Asmt: 1, 0, -1.
Explicitness: em, e, i.
Judgement: j, nj.
Closes-JA: c, dc, 0.
Repair: r, nr.

```
25,tu,p,1,1,0,em,j,c,nr,1.
25,tu,p,1,1,0,i,j,c,nr,1.
25,tu,p,0,1,0,i,nj,c,nr,1.
25,tu,p,-1,0.75,-1,i,j,dc,nr,-1.
25,tu,p,1,1,1,em,j,c,nr,1.
25,tu,p,0,1,0,i,nj,c,nr,1.
25,tu,p,1,1,0,i,j,c,nr,1.
25,tu,p,0,1,0,i,nj,c,nr,1.
25,tu,p,1,1,0,em,j,c,nr,1.
25,tu,p,-1,0.75,-1,em,j,dc,nr,-1.
25,tu,p,1,1,1,i,j,c,nr,1.
25,tu,p,0,1,0,i,nj,c,nr,1.
25,tu,p,0,1,0,i,nj,c,nr,1.
25,tu,p,0,1,0,em,nj,c,nr,1.
25,tu,t,0,1,0,em,j,0 ,nr,1.
25,tu,t,-1,0.75,-1,e,j,dc,nr,-1.
25,tu,t,1,1,1,em,j,c,nr,1.
25,tu,t,0,1,0,i,j,dc,nr,1.
25,tu,t,0,1,0,em,j,0 ,nr,1.
25,tu,t,1,1,0,em,j,c,nr,1.
25,tu,t,-1,0.75,-1,em,j,dc,nr,-1.
25,tu,t,-1,0.5,-1,em,j,dc,nr,-1.
25,tu,t,1,0.75,1,em,j,c,nr,1.
25,tu,t,1,1,1,em,j,c,nr,1.
25,tu,i,0,1,0,em,nj,c,nr,1.
25,tu,p,1,1,0,i,j,c,nr,1.
25,tu,p,0,1,0,i,nj,c,nr,1.
25,tu,p,1,1,0,i,j,c,nr,1.
25,tu,p,0,1,0,i,nj,c,nr,1.
25,tu,p,1,1,0,i,j,c,nr,1.
25,tu,p,0,1,0,i,nj,c,nr,1.
25,tu,p,-1,0.75,-1,e,j,dc,nr,-1.
25,tu,p,-1,0.5,-1,em,j,dc,nr,-1.
25,tu,p,1,0.75,1,em,nj,c,nr,1.
25,tu,p,-1,0.5,-1,em,j,dc,nr,-1.
25,tu,p,1,0.75,1,i,j,c,nr,1.
25,tu,p,0,0.75,0,i,nj,c,nr,1.
25,tu,p,1,1,1,i,nj,c,nr,1.
25,tu,p,0,1,0,em,nj,c,nr,1.
25,tu,t,0,1,0,e,nj,0 ,r,-1.
25,tu,t,-1,0.75,-1,em,j,0 ,nr,-1.
25,tu,t,-1,0.5,-1,em,j,dc,nr,-1.
25,tu,t,1,0.75,1,em,j,c,nr,1.
25,tu,p,1,1,1,i,j,c,nr,1.
25,tu,p,-1,0.75,-1,i,j,dc,nr,-1.
25,tu,p,1,1,1,em,j,c,nr,1.
25,tu,p,1,1,0,i,j,c,nr,1.
25,tu,p,0,1,0,i,nj,c,nr,1.
25,tu,p,1,1,0,i,j,c,nr,1.
25,tu,p,0,1,0,i,nj,c,nr,1.
25,tu,p,1,1,0,i,j,c,nr,1.
25,tu,p,0,1,0,i,nj,c,nr,1.
25,tu,p,1,1,0,i,j,c,nr,1.
25,tu,p,0,1,0,i,nj,c,nr,1.
25,tu,p,1,1,0,i,j,c,nr,1.
25,tu,p,0,1,0,i,nj,c,nr,1.
25,tu,p,1,1,0,i,j,c,nr,1.
25,tu,t,-1,0.75,-1,e,j,dc,nr,-1.
25,tu,t,-1,0.5,-1,i,j,dc,nr,-1.
25,tu,t,1,0.75,1,em,j,dc,nr,1.
25,tu,t,0,0.75,0,em,j,dc,nr,1.
25,tu,t,1,1,1,em,j,c,nr,1.
25,tu,t,1,1,0,em,j,c,nr,1.
26,tu,a,0,1,0,em,nj,c,nr,1.
26,tu,i,0,1,0,em,nj,c,nr,1.
26,tu,t,1,1,0,i,j,c,nr,1.
26,tu,t,-1,0.75,-1,i,j,dc,nr,-1.
26,tu,t,1,1,1,i,j,c,nr,1.
26,tu,p,1,1,0,i,j,c,nr,1.
26,tu,p,0,1,0,i,nj,c,nr,1.
26,tu,p,1,1,0,i,j,c,nr,1.
26,tu,p,0,1,0,i,nj,c,nr,1.
26,tu,p,1,1,0,i,j,c,nr,1.
26,tu,p,0,1,0,i,nj,c,nr,1.
```

```
26,tu,p,1,1,0,i,j,c,nr,1.
26,tu,p,0,1,0,i,nj,c,nr,1.
26,tu,p,1,1,0,i,j,c,nr,1.
26,tu,p,0,1,0,i,nj,c,nr,1.
26,tu,p,1,1,0,i,j,c,nr,1.
26,tu,p,0,1,0,em,nj,c,nr,1.
26,tu,t,1,1,0,em,nj,c,nr,1.
26,tu,t,-1,0.75,-1,e,j,dc,nr,-1.
26,tu,t,-1,0.5,-1,em,j,dc,nr,-1.
26,tu,t,1,0.75,1,em,j,c,nr,1.
26,tu,t,-1,0.5,-1,em,j,dc,nr,-1.
26,tu,t,1,0.75,1,e,j,c,nr,1.
26,tu,t,-1,0.5,-1,e,j,dc,nr,-1.
26,tu,t,-1,0.25,-1,e,j,dc,nr,-1.
26,tu,t,-1,0,-1,em,j,dc,nr,-1.
26,tu,t,-1,0.25,1,em,j,dc,nr,-1.
26,tu,t,1,0.5,1,e,j,c,nr,1.
26,tu,t,1,0.75,1,em,j,c,nr,1.
26,tu,t,1,1,1,em,j,c,nr,1.
26,tu,p,1,1,0,e,j,c,nr,1.
26,tu,p,0,1,0,i,nj,c,nr,1.
26,tu,p,1,1,0,i,j,c,nr,1.
26,tu,p,0,1,0,i,nj,c,nr,1.
26,tu,p,-1,0.75,-1,e,j,dc,nr,-1.
26,tu,p,-1,0.5,-1,em,j,dc,nr,-1.
26,tu,p,1,0.75,1,em,j,c,nr,1.
26,tu,p,0,0.75,0,i,nj,c,nr,1.
26,tu,p,1,1,1,i,j,c,nr,1.
26,tu,p,0,1,0,i,nj,c,nr,1.
26,tu,p,1,1,0,i,j,c,nr,1.
26,tu,p,0,1,0,i,nj,c,nr,1.
26,tu,p,1,1,0,i,j,c,nr,1.
26,tu,p,0,1,0,i,nj,c,nr,1.
26,tu,p,1,1,0,i,j,c,nr,1.
26,tu,p,1,1,0,em,j,c,nr,1.
26,tu,t,1,1,0,em,j,c,nr,1.
26,tu,t,1,1,0,em,j,c,nr,1.
26,tu,p,0,1,0,i,nj,c,nr,1.
26,tu,p,0,1,0,i,nj,c,nr,1.
26,tu,p,0,1,0,i,nj,c,nr,1.
26,tu,p,0,1,0,i,nj,c,nr,1.
26,tu,p,0,1,0,i,nj,c,nr,1.
26,tu,p,0,1,0,i,nj,c,nr,1.
26,tu,p,0,1,0,i,nj,c,nr,1.
26,tu,p,0,1,0,i,nj,c,nr,1.
26,tu,p,0,1,0,i,nj,c,nr,1.
26,tu,p,0,1,0,i,nj,c,nr,1.
26,tu,p,0,1,0,i,nj,c,nr,1.
26,tu,p,0,1,0,i,nj,c,nr,1.
26,tu,p,0,1,0,i,nj,c,nr,1.
26,tu,p,1,1,0,em,j,c,nr,1.
26,tu,t,-1,0.75,-1,e,j,dc,nr,-1.
26,tu,t,-1,0.5,-1,em,j,c,nr,-1.
26,tu,a,0,0.5,0,e,nj,c,nr,1.
27,tu,i,0,1,0,em,nj,c,nr,1.
27,tu,t,-1,0.75,-1,e,j,dc,nr,-1.
27,tu,t,1,1,1,em,nj,c,nr,1.
27,tu,p,1,1,0,i,j,c,nr,1.
27,tu,p,1,1,0,em,j,c,nr,1.
27,tu,p,1,1,0,i,j,c,nr,1.
27,tu,p,1,1,0,i,nj,c,nr,1.
27,tu,p,-1,0.75,-1,e,j,dc,nr,-1.
27,tu,p,1,1,1,i,j,c,nr,1.
27,tu,p,1,1,0,em,nj,c,nr,1.
27,tu,p,1,1,0,i,j,c,nr,1.
27,tu,p,1,1,0,i,nj,c,nr,1.
27,tu,p,1,1,0,i,j,c,nr,1.
27,tu,p,1,1,0,i,nj,c,nr,1.
27,tu,p,1,1,0,i,j,c,nr,1.
27,tu,p,1,1,0,i,nj,c,nr,1.
27,tu,p,1,1,0,i,j,c,nr,1.
27,tu,p,1,1,0,em,nj,c,nr,1.
27,tu,t,-1,0.75,-1,e,j,dc,nr,1.
27,tu,t,1,1,1,e,j,dc,nr,1.
27,tu,t,1,1,0,em,j,c,nr,1.
27,tu,t,1,1,0,e,j,c,nr,1.
27,tu,t,1,1,0,e,j,c,nr,1.
27,tu,t,1,1,0,em,j,c,nr,1.
27,tu,t,-1,0.75,-1,em,j,dc,nr,-1.
27,tu,t,1,1,1,em,j,c,nr,1.
27,tu,t,1,1,0,em,j,c,nr,1.
27,tu,t,-1,0.75,-1,em,j,dc,nr,-1.
27,tu,t,1,1,1,em,j,c,nr,1.
27,tu,p,1,1,0,i,j,c,nr,1.
27,tu,p,0,1,0,i,nj,c,nr,1.
27,tu,p,1,1,0,i,j,c,nr,1.
27,tu,p,0,1,0,i,nj,c,nr,1.
27,tu,p,-1,0.75,-1,e,j,dc,nr,1.
27,tu,p,1,1,1,em,j,c,nr,1.
27,tu,p,0,1,0,i,nj,c,nr,1.
27,tu,p,1,1,0,i,j,c,nr,1.
27,tu,p,0,1,0,i,nj,c,nr,1.
27,tu,p,1,1,0,i,j,c,nr,1.
27,tu,p,0,1,0,i,nj,c,nr,1.
27,tu,p,1,1,0,i,j,c,nr,1.
27,tu,p,0,1,0,i,nj,c,nr,1.
27,tu,p,1,1,0,i,j,c,nr,1.
27,tu,p,1,1,0,em,j,c,nr,1.
27,tu,t,1,1,0,em,j,c,nr,1.
27,tu,t,1,1,0,em,j,c,nr,1.
27,tu,p,1,1,0,i,j,c,nr,1.
27,tu,p,0,1,0,i,nj,c,nr,1.
27,tu,p,1,1,0,i,j,c,nr,1.
27,tu,p,0,1,0,i,nj,c,nr,1.
27,tu,p,0,1,0,em,nj,dc,r,-1.
27,tu,p,1,1,0,i,j,c,nr,1.
27,tu,p,0,1,0,i,nj,c,nr,1.
27,tu,p,1,1,0,i,j,c,nr,1.
27,tu,p,0,1,0,i,nj,c,nr,1.
27,tu,p,1,1,0,i,j,c,nr,1.
```

```
27,tu,p,0,1,0,i,nj,c,nr,1.
27,tu,p,1,1,0,i,j,c,nr,1.
27,tu,p,0,1,0,i,nj,c,nr,1.
27,tu,p,1,1,0,i,j,c,nr,1.
27,tu,t,-1,0.75,-1,e,j,dc,nr,-1.
27,tu,t,-1,0.5,-1,e,j,dc,nr,-1.
27,tu,t,-1,0.25,-1,em,j,dc,nr,-1.
27,tu,t,0,0.25,0,e,nj,dc,r,-1.
27,tu,t,-1,0,-1,e,j,dc,nr,-1.
27,tu,t,1,0.25,1,em,j,c,nr,1.
27,tu,t,-1,0,-1,em,j,dc,nr,-1.
27,tu,t,1,0.25,1,em,j,c,nr,1.
27,tu,t,1,0.5,1,em,j,c,nr,1.
27,tu,t,1,0.75,1,em,j,c,nr,1.
27,tu,t,0,0.75,0,em,j,0,nr,1.
27,tu,t,1,1,1,em,j,c,nr,1.
27,tu,t,1,1,0,em,j,c,nr,1.
27,tu,t,-1,0.75,-1,em,j,dc,nr,-1.
27,tu,t,0,0.75,0,em,nj,c,nr,1.
28,tu,t,1,1,0,em,j,c,nr,1.
28,tu,t,1,1,0,em,j,c,nr,1.
28,tu,t,1,1,0,e,j,c,nr,1.
28,tu,t,1,1,0,em,j,c,nr,1.
28,tu,t,0,1,0,em,nj,c,nr,1.
28,tu,p,1,1,0,i,j,c,nr,1.
28,tu,p,0,1,0,i,nj,c,nr,1.
28,tu,p,1,1,0,i,j,c,nr,1.
28,tu,p,-1,0.75,-1,em,j,dc,nr,-1.
28,tu,p,1,1,1,em,j,c,nr,1.
28,tu,p,1,1,0,i,j,c,nr,1.
28,tu,p,0,1,0,i,nj,c,nr,1.
28,tu,p,1,1,0,i,j,c,nr,1.
28,tu,p,0,1,0,i,nj,c,nr,1.
28,tu,p,1,1,0,em,j,c,nr,1.
28,tu,p,0,1,0,i,nj,c,nr,1.
28,tu,p,1,1,0,i,j,c,nr,1.
28,tu,p,0,1,0,em,j,c,nr,1.
28,tu,p,1,1,0,i,j,c,nr,1.
28,tu,p,0,1,0,i,nj,c,nr,1.
28,tu,p,0,1,0,i,nj,c,nr,1.
28,tu,t,1,1,0,em,j,c,nr,1.
28,tu,t,1,1,1,em,j,c,nr,1.
28,tu,t,1,1,0,em,j,c,nr,1.
28,tu,t,1,1,0,em,j,c,nr,1.
28,tu,t,1,1,0,em,j,c,nr,1.
28,tu,t,1,1,0,em,j,c,nr,1.
28,tu,t,0,1,0,em,nj,c,nr,1.
28,tu,i,0,1,0,em,nj,c,nr,1.
28,tu,p,1,1,0,i,j,c,nr,1.
28,tu,p,0,1,0,em,nj,c,nr,1.
28,tu,p,0,1,0,i,nj,c,nr,1.
28,tu,p,0,1,0,i,nj,c,nr,1.
28,tu,p,0,1,0,i,nj,c,nr,1.
28,tu,p,0,1,0,i,nj,c,nr,1.
28,tu,p,0,1,0,i,nj,c,nr,1.
28,tu,t,0,1,0,i,nj,c,nr,1.
28,tu,t,-1,0.75,-1,em,j,dc,nr,-1.
28,tu,t,-1,0.5,-1,em,j,dc,nr,-1.
28,tu,t,0,0.5,0,e,j,dc,nr,1.
28,tu,t,1,0.75,1,em,j,c,nr,1.
28,tu,t,0,0.75,0,em,j,c,nr,1.
28,tu,t,-1,0.5,-1,e,j,dc,nr,-1.
28,tu,t,-1,0.25,-1,i,j,dc,nr,-1.
28,tu,t,1,0.5,1,em,j,c,nr,1.
28,tu,t,1,0.75,1,em,j,dc,nr,1.
28,tu,t,1,1,1,em,j,dc,nr,1.
28,tu,t,1,1,0,em,j,dc,nr,1.
28,tu,t,0,1,0,em,nj,c,nr,1.
```

D.2 C4.5 Polarity: Decision Tree Output

```
C4.5 [release 8] decision tree generator  Thu Apr  2 11:30:52 1998
----------------------------------------

    Options:
   File stem <polarity>

Read 252 cases (10 attributes) from polarity.data

Decision Tree:

Closes-JA = c: 1 (197.0/1.0)
Closes-JA = 0: 1 (5.0/2.0)
Closes-JA = dc:
|   Answer-accept = 1: 1 (5.0)
|   Answer-accept = -1: -1 (40.0/2.0)
|   Answer-accept = 0:
|   |   Judgement = j: 1 (3.0)
|   |   Judgement = nj: -1 (2.0)


Tree saved


Evaluation on training data (252 items):

    Before Pruning          After Pruning
   ----------------   ---------------------------
   Size      Errors   Size      Errors   Estimate

      9    5( 2.0%)      9    5( 2.0%)    ( 5.1%)   <<
```

D.3 C4.5 Polarity: Rules Output

```
C4.5 [release 8] rule generator  Thu Apr  2 11:47:21 1998
-------------------------------

    Options:
   File stem <polarity>

Read 252 cases (10 attributes) from polarity

------------------
Processing tree 0

Final rules from tree 0:

Rule 3:
      Answer-accept = -1
   ->  class -1  [91.0%]

Rule 2:
      Answer-accept = 1
   ->  class 1  [98.9%]

Rule 4:
      Answer-accept = 0
   ->  class 1  [94.2%]

Default class: 1


Evaluation on training data (252 items):

Rule  Size  Error  Used  Wrong             Advantage
----  ----  -----  ----  -----             ---------
   3     1   9.0%    42      2 (4.8%)      38 (40|2)   -1
   2     1   1.1%   123      0 (0.0%)       0 (0|0)    1
   4     1   5.8%    87      3 (3.4%)       0 (0|0)    1

Tested 252, errors 5 (2.0%)   <<


     (a)  (b)  (c)   <-classified as
    ---- ---- ----
     207    2         (a): class 1
       3   40         (b): class -1
                      (c): class 0
```

D.4 C4.5 Marking for Predictions Phase: Raw Data Input

em, e, i.

Answer-accept: 1, -1, 0.
Stu-Asmt: continuous.
Delta-Stu-Asmt: 1, 0, -1.

```
1,1,0,em.
1,1,0,i.
0,1,0,i.
-1,0.75,-1,i.
1,1,1,em.
0,1,0,i.
1,1,0,i.
0,1,0,i.
1,1,0,em.
-1,0.75,-
1,em.
1,1,1,i.
0,1,0,i.
0,1,0,i.
0,1,0,em.
1,1,0,i.
0,1,0,i.
1,1,0,i.
0,1,0,i.
1,1,0,i.
0,1,0,i.
-1,0.75,-1,e.
-1,0.5,-1,em.
1,0.75,1,em.
-1,0.5,-1,em.
1,0.75,1,i.
0,0.75,0,i.
1,1,1,i.
0,1,0,em.
1,1,1,i.
-1,0.75,-1,i.
1,1,1,em.
1,1,0,i.
0,1,0,i.
1,1,0,i.
0,1,0,i.
1,1,0,i.
0,1,0,i.
1,1,0,i.
```

```
0,1,0,i.
1,1,0,i.
0,1,0,i.
1,1,0,i.
1,1,0,i.
0,1,0,i.
1,1,0,i.
0,1,0,i.
1,1,0,i.
0,1,0,i.
1,1,0,i.
0,1,0,i.
1,1,0,i.
0,1,0,i.
1,1,0,i.
0,1,0,em.
1,1,0,e.
0,1,0,i.
1,1,0,i.
0,1,0,i.
-1,0.75,-1,e.
-1,0.5,-1,em.
1,0.75,1,em.
0,0.75,0,i.
1,1,1,i.
0,1,0,i.
1,1,0,i.
0,1,0,i.
1,1,0,i.
0,1,0,i.
1,1,0,i.
1,1,0,em.
0,1,0,i.
0,1,0,i.
0,1,0,i.
0,1,0,i.
0,1,0,i.
0,1,0,i.
0,1,0,i.
```

```
0,1,0,i.
0,1,0,i.
0,1,0,i.
0,1,0,i.
0,1,0,i.
0,1,0,i.
1,1,0,em.
1,1,0,i.
1,1,0,em.
1,1,0,i.
1,1,0,i.
-1,0.75,-1,e.
1,1,1,i.
1,1,0,em.
1,1,0,i.
1,1,0,i.
1,1,0,i.
1,1,0,i.
1,1,0,i.
1,1,0,i.
1,1,0,i.
1,1,0,em.
1,1,0,i.
0,1,0,i.
1,1,0,i.
0,1,0,i.
-1,0.75,-1,e.
1,1,1,em.
0,1,0,i.
1,1,0,i.
0,1,0,i.
1,1,0,i.
0,1,0,i.
1,1,0,i.
0,1,0,i.
1,1,0,i.
1,1,0,em.
1,1,0,i.
0,1,0,i.
```

```
1,1,0,i.
0,1,0,i.
0,1,0,em.
1,1,0,i.
0,1,0,i.
1,1,0,i.
0,1,0,i.
1,1,0,i.
0,1,0,i.
1,1,0,i.
0,1,0,i.
1,1,0,i.
1,1,0,i.
0,1,0,i.
1,1,0,i.
-1,0.75,-
1,em.
1,1,1,em.
1,1,0,i.
0,1,0,i.
1,1,0,i.
0,1,0,i.
1,1,0,em.
0,1,0,i.
1,1,0,i.
0,1,0,em.
1,1,0,i.
0,1,0,i.
0,1,0,i.
1,1,0,i.
0,1,0,em.
0,1,0,i.
0,1,0,i.
0,1,0,i.
0,1,0,i.
0,1,0,i.
```

D.5 C4.5 Marking for Predictions Phase: Decision Tree Output

```
C4.5 [release 8] decision tree generator  Thu Apr  2 15:33:42 1998
----------------------------------------

    Options:
   File stem <marking2p>

Read 151 cases (3 attributes) from marking2p.data

Decision Tree:

Delta-Stu-Asmt = 0: i (128.0/16.0)
Delta-Stu-Asmt = 1:
|   Stu-Asmt <= 0.75 : em (3.0/1.0)
|   Stu-Asmt > 0.75 : i (9.0/4.0)
Delta-Stu-Asmt = -1:
|   Stu-Asmt <= 0.5 : em (3.0)
|   Stu-Asmt > 0.5 : e (8.0/4.0)


Simplified Decision Tree:

Delta-Stu-Asmt = 1: em (12.0/7.7)
Delta-Stu-Asmt = 0: i (128.0/19.3)
Delta-Stu-Asmt = -1:
|   Stu-Asmt <= 0.5 : em (3.0/1.1)
|   Stu-Asmt > 0.5 : e (8.0/5.4)


Tree saved


Evaluation on training data (151 items):

    Before Pruning          After Pruning
   ----------------   ---------------------------
   Size      Errors   Size      Errors   Estimate

      8   25(16.6%)      6   26(17.2%)    (22.2%)   <<
```

D.6 C4.5 Marking for Predictions Phase: Rules Output

```
C4.5 [release 8] rule generator  Thu Apr  2 15:34:22 1998
-------------------------------

    Options:
   File stem <marking2p>

Read 151 cases (3 attributes) from marking2p

------------------
Processing tree 0

Final rules from tree 0:

Rule 5:
      Stu-Asmt > 0.5
      Delta-Stu-Asmt = -1
   ->  class e  [32.3%]

Rule 4:
      Stu-Asmt <= 0.5
   ->  class em  [63.0%]

Rule 1:
      Delta-Stu-Asmt = 1
   ->  class em  [36.2%]

Rule 3:
      Delta-Stu-Asmt = 0
   ->  class i  [84.9%]

Default class: i


Evaluation on training data (151 items):

Rule  Size  Error  Used  Wrong             Advantage
----  ----  -----  ----  -----             ---------
   5     2  67.7%     8      4 (50.0%)      2 (4|2)    e
   4     1  37.0%     3      0 (0.0%)       3 (3|0)    em
   1     1  63.8%    12      6 (50.0%)      0 (6|6)    em
   3     1  15.1%   128     16 (12.5%)      0 (0|0)    i

Tested 151, errors 26 (17.2%)   <<


     (a)   (b)   (c)    <-classified as
    ----  ----  ----
       9     2    15    (a): class em
             4     1    (b): class e
       6     2   112    (c): class i
```

D.7 C4.5 Marking for Tutoring Phase: Raw Data Input

em, e, i.

Answer-accept: 1, -1, 0.
Stu-Asmt: continuous.
Delta-Stu-Asmt: 1, 0, -1.

0,1,0,em.	-1,0.75,-1,e.	-1,0.75,-1,em.	1,1,0,e.
-1,0.75,-1,e.	-1,0.5,-1,em.	1,1,1,em.	1,1,0,em.
1,1,1,em.	1,0.75,1,em.	1,1,0,em.	0,1,0,em.
0,1,0,i.	-1,0.5,-1,em.	-1,0.75,-1,em.	1,1,0,em.
0,1,0,em.	1,0.75,1,e.	1,1,1,em.	1,1,1,em.
1,1,0,em.	-1,0.5,-1,e.	1,1,0,em.	1,1,0,em.
-1,0.75,-1,em.	-1,0.25,-1,e.	1,1,0,em.	1,1,0,em.
-1,0.5,-1,em.	-1,0,-1,em.	-1,0.75,-1,e.	1,1,0,em.
1,0.75,1,em.	-1,0.25,1,em.	-1,0.5,-1,e.	1,1,0,em.
1,1,1,em.	1,0.5,1,e.	-1,0.25,-1,em.	0,1,0,em.
0,1,0,e.	1,0.75,1,em.	0,0.25,0,e.	0,1,0,i.
-1,0.75,-1,em.	1,1,1,em.	-1,0,-1,e.	-1,0.75,-1,em.
-1,0.5,-1,em.	1,1,0,em.	1,0.25,1,em.	-1,0.5,-1,em.
1,0.75,1,em.	1,1,0,em.	-1,0,-1,em.	0,0.5,0,e.
-1,0.75,-1,e.	-1,0.75,-1,e.	1,0.25,1,em.	1,0.75,1,em.
-1,0.5,-1,i.	-1,0.5,-1,em.	1,0.5,1,em.	0,0.75,0,em.
1,0.75,1,em.	-1,0.75,-1,e.	1,0.75,1,em.	-1,0.5,-1,e.
0,0.75,0,em.	1,1,1,em.	0,0.75,0,em.	-1,0.25,-1,i.
1,1,1,em.	-1,0.75,-1,e.	1,1,1,em.	1,0.5,1,em.
1,1,0,em.	1,1,1,e.	1,1,0,em.	1,0.75,1,em.
1,1,0,i.	1,1,0,em.	-1,0.75,-1,em.	1,1,1,em.
-1,0.75,-1,i.	1,1,0,e.	0,0.75,0,em.	1,1,0,em.
1,1,1,i.	1,1,0,e.	1,1,0,em.	0,1,0,em.
1,1,0,em.	1,1,0,em.	1,1,0,em.	

D.8 C4.5 Marking for Tutoring Phase: Decision Tree Output

```
C4.5 [release 8] decision tree generator  Thu Apr  2 16:01:05 1998
----------------------------------------

    Options:
   File stem <marking2t>

Read 95 cases (3 attributes) from marking2t.data

Decision Tree:

Answer-accept = 1: em (50.0/8.0)
Answer-accept = -1: em (31.0/15.0)
Answer-accept = 0:
|   Stu-Asmt <= 0.5 : e (2.0)
|   Stu-Asmt > 0.5 : em (12.0/3.0)


Simplified Decision Tree:
 em (95.0/31.7)


Tree saved


Evaluation on training data (95 items):

    Before Pruning           After Pruning
   ----------------   ---------------------------
   Size      Errors   Size      Errors   Estimate

      6   26(27.4%)      1   28(29.5%)    (33.3%)   <<
```

D.9 C4.5 Marking for Tutoring Phase: Rules Output

```
C4.5 [release 8] rule generator  Thu Apr  2 16:01:43 1998
-------------------------------

    Options:
   File stem <marking2t>

Read 95 cases (3 attributes) from marking2t

------------------
Processing tree 0

Final rules from tree 0:

Rule 2:
      Answer-accept = 0
      Stu-Asmt <= 0.5
   ->  class e  [50.0%]

Rule 1:
      Answer-accept = 1
   ->  class em  [79.0%]

Default class: em


Evaluation on training data (95 items):

Rule  Size  Error  Used  Wrong             Advantage
----  ----  -----  ----  -----             ---------
   2     2  50.0%     2      0 (0.0%)       2 (2|0)    e
   1     1  21.0%    50      8 (16.0%)      0 (0|0)    em

Tested 95, errors 26 (27.4%)   <<


     (a)  (b)  (c)    <-classified as
    ---- ---- ----
      67              (a): class em
      19    2         (b): class e
       7              (c): class i
```

BIBLIOGRAPHY

Allen, James. 1995. Natural Language Understanding. 2nd edition, Benjamin/Cummings Publishing Company, Redwood City, California.

Allen, James, Miller, Bradford W., Ringger, Eric K., and Sikorski, Teresa. 1996. A Robust System for Natural Spoken Dialogue. Proceedings of the Annual Meeting of the Association for Computational Linguistics, Santa Cruz, New Mexico: 62-70.

Allen, Vernon L. 1976. The Helping Relationship and Socialization of Children: Some Perspectives on Tutoring. In Allen, Vernon L., Ed. Children as Teachers: Theory and Research on Tutoring. Academic Press. New York, New York: 9-26.

Anderson, Edward L. 1994. Education that Works: The Child is Always Right. The Cambridge Center for Behavioral Studies, Cambridge, Massachusetts.

Anderson, John R. 1988. The Expert Module. In Polson, Martha C., and Richardson, J. Jeffrey, Eds. Foundations of Intelligent Tutoring Systems. Lawrence Erlbaum Assoc., Hillsdale, New Jersey: 21-53.

Argyle, Michael. 1976. Social Skills Theory. In Allen, Vernon L., Ed. Children as Teachers: Theory and Research on Tutoring. Academic Press, New York, New York: 57-73.

Boyle, Patrick J., Evens, Martha W., and Spitkovsky, John A. 1993. Finding a Link Between Hinting and Negative Acknowledgments in Natural Language Tutoring Systems. Unpublished paper. Department of Computer Science and Applied Mathematics, Illinois Institute of Technology, Chicago, Illinois.

Brandle, Stefan. 1996. Understanding Human Communication Acknowledgment Protocols by Studying Artificial Communication Protocols. Proceedings of Annual MAICS Conference. http://www.cs.indiana.edu/event/maics96/ Proceedings/brandle.html.

Brandle, Stefan, and Evens, Martha. 1997. Acknowledgments in Tutorial Dialogue. In Eugene Santos, Ed., Eighth Midwest AI and Cognitive Science Conference, Dayton, OH, AAAI Press, AAAI Technical Report CF-97-01, Menlo Park, CA: pp. 13-17.

Brandle, Stefan, Evens, Martha W., Michael, Joel A., Rovick, Allen A. 1998. Categorizing Acknowledgments in Tutorial Dialogue. Unpublished paper. Department of Computer Science and Applied Mathematics, Illinois Institute of Technology, Chicago, Illinois.

Brandle, Stefan, Robergé, James, and Smith, George. 1996. The Realities of Teaching the Distance Learner. Proceedings of the 13th International Conference on Technology and Education, New Orleans, Louisiana: 416-418.

Burns, Hugh L., and Capps, Charles G. 1988. Foundations of Intelligent Tutoring Systems: An Introduction. In Polson, Martha C., and Richardson, J. Jeffrey, Eds. Foundations of Intelligent Tutoring Systems. Lawrence Erlbaum Assoc., Hillsdale, New Jersey: 1-19.

Carbonell, Jaime R. 1970. " AI in CAI: Artificial Intelligence Approach to Computer Assisted Instruction". IEEE Transactions on Man-Machine Systems, Vol. 11, No. 4: 190-202.

Carletta, Jean, Isard, Amy, Isard, Stephen, Kowtko, Jackqueline C., Doherty-Sneddon, Gwyneth, Anderson, Anne H. 1997. "The Reliability of a Dialogue Structure Coding Scheme". Computational Linguistics, March 1997, Vol. 23, No. 1: 13-31.

Cawsey, Alison. 1989a. The Structure of Tutorial Discourse. DAI Research Paper No. 425. Department of Artificial Intelligence, University of Edinburgh, Scotland.

Cawsey, Alison. 1989b. A Computational Model of Explanatory Discourse: Local Interactions in a Plan-Based Explanation. DAI Research Paper No. 442. Department of Artificial Intelligence, University of Edinburgh, Scotland.

Cawsey, Alison. 1990. Understandable Explanations: The FUDGE Discourse Generator. DAI Research Paper No. 473. Department of Artificial Intelligence, University of Edinburgh, Scotland.

Cawsey, Alison. 1992. Explanation and Interaction: The Computer Generation of Explanatory Dialogues. MIT Press, Cambridge, Massachusetts.

Cawsey, Alison, and Raudaskoski, Pirkko. 1990. Repair Work in Human-Computer Dialogue. DAI Research Paper No. 474. Department of Artificial Intelligence, University of Edinburgh, Scotland.

Chambers, Jack A., and Sprecher, Jerry W. 1984. Computer-Assisted Instruction: Current Trends and Critical Issues. In Walker, Decker F., and Hess, Robert D., Eds. Instructional Software: Principles and Perspectives for Design and Use. Wadsworth Publishing Company, Belmont, California: 6-20.

Cherryh, C. J. 1988. Cyteen. Warner Books, New York, New York.

Chi, Michelene T. H., de Leeuw, Nicholas, Chiu, Mei-Hung, and LaVancher, Christian. 1994. "Eliciting Self-Explanations Improves Understanding." Cognitive Science, Vol. 18, No. 3: 439-477.

Chiesa, Mecca. "Radical Behaviorism and Scientific Frameworks." American Psychologist, Vol. 47, No. 11, pp. 1287-1299, November 1992: 1287-1299.

Chipman, Susan F. 1993. Gazing Once More into the Silicon Chip: Who's Revolutionary Now? In Lajoie, Susanne P., and Derry, Sheron J. Computers as Cognitive Tools. Lawrence Earlbaum Associates, Hillsdale, New Jersey: 341-367.

Clark, Herbert H. 1996. Using Language. Cambridge University Press, Cambridge, England.

Condon, S., and Cech, C. 1992. Manual for Coding Decision-Making Interactions. Revised 1995. Unpublished manuscript available at the Discourse Resource Initiative web site at http://www.georgetown.edu/luperfoy/Discourse-Treebank/dri-home.html.

Dawson-Saunders, Beth, and Trapp, Robert G. 1994. Basic & Clinical Biostatistics. Appleton & Lange, Norwalk, Connecticut.

Ehly, Stewart W., and Larsen, Stephen C. 1980. Peer Tutoring for Individualized Instruction. Allyn and Bacon, Boston, Massachusetts.

Evens, Martha, Michael, Joel, Rovick, Allen, Markowitz, Judith, and Hume, Gregory. 1995. An Authoring System for Intelligent Tutoring Systems That Generate Spoken Natural Language Dialogue. Unpublished Paper. Department of Computer Science and Applied Mathematics, Illinois Institute of Technology, Chicago, Illinois.

Evens, Martha W., Woo, Chong W., Michael, Joel, and Rovick, Allen. 1995. Generating Natural Language Dialogue in an Intelligent Tutoring System for Cardiovascular Physiology. Unpublished paper. Department of Computer Science and Applied Mathematics, Illinois Institute of Technology, Chicago, Illinois.

Fox, Barbara. 1988. Repair as a Factor in Interface Design. Technical Report 88-4. Institute of Cognitive Science, University of Colorado, Boulder, Colorado.

Fox, Barbara. 1988. Robust Learning Environments: The Issue of Canned Text. Technical Report 88-5. Institute of Cognitive Science, University of Colorado, Boulder, Colorado.

Fox, Barbara A. 1993. The Human Tutorial Dialogue Project: Issues in the Design of Instructional Systems. Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Fraser, Jane M., and Smith, Philip J. 1992. A Catalog of Errors. International Journal Man-Machine Studies. Vol. 37, No. 3: 265-307.

Freedman, Reva K. 1996. Interaction of Discourse Planning, Instructional Planning and Dialogue Management in an Interactive Tutoring System. Ph.D. Dissertation. Electrical Engineering and Computer Science, Northwestern University, Evanston, Illinois.

Gaines, Brian R. 1990. Knowledge Acquisition Systems. In Adeli, Hojjat, Ed. Knowledge Engineering: Volume 1, Fundamentals. McGraw-Hill, New York, New York: 52-102.

Gardner, Howard. 1985. The Mind's New Science. Basic Books, Inc., Publishers, New York, New York.

Georgakis, D. C., Rosenthal, R., Trace, D., and Evens, M. W. 1988. "Measures of Performance of the MEDAS System." Proceedings of Fourth AI West, Long Beach, California: 50-58.

Graesser, Arthur C. 1993a. Questioning Mechanisms during Tutoring, Conversation, and Human-Computer Interaction. ONR Technical Report 93-1, Memphis, Tennessee.

Graesser, Arthur C. 1993b. Dialogue Patterns and Feedback Mechanisms during Naturalistic Tutoring. Proceedings of the Annual Conference of the Cognitive Science Society, Boulder, Colorado: 126-130.

Graesser, Arthur C. 1996a. Private communication regarding the need for more detailed empirical study of human tutoring sessions.

Graesser, Arthur C. 1996b. Private communication expressing support for my research and agreeing to make his transcripts of tutoring sessions available. Email on July 23, 1996.

Grice, H.P. 1975. Logic and Conversation. In Cole, P., and Morgan, J.L., Eds., Syntax and Semantics. III. Speech Acts. Academic Press, New York, New York: pp.41-58.

Halff, Henry M. 1988. Curriculum and Instruction in Automated Tutors. In Polson, Martha C., and Richardson, J. Jeffrey, Eds. Foundations of Intelligent Tutoring Systems. Lawrence Erlbaum Assoc., Hillsdale, New Jersey: 79-108.

Halsall, Fred. 1988. Data Communications, Computer Networks and OSI. 2nd Ed., Addison-Wesley Publishing Company, Wokingham, England.

Harrison, Grant Von. 1976. Structured Tutoring: Antidote for Low Achievement. In Allen, Vernon L., Ed. Children as Teachers: Theory and Research on Tutoring. Academic Press, New York, New York: 169-177.

Hartup, Willard W. 1976. Cross-Age versus Same-Age Peer Interaction: Ethological and Cross-Cultural Perspectives. In Allen, Vernon L., Ed. Children as Teachers: Theory and Research on Tutoring. Academic Press, New York, New York: 41-55.

Hirst, Graeme, McRoy, Susan, Heeman, Peter, Edmonds, Philip, and Horton, Diane. 1994. Repairing conversational misunderstandings and non-understandings. Speech Communication Vol. 15, Elsevier Science Publishers, New York, New York: 213-229.

Hoschka, Peter. 1996. Computers as Assistants–Introduction and Overview. In Computers as Assistants: A New Generation of Support Systems. Hoschka, Peter, Ed. Lawrence Erlbaum Associates, Mahwah, New Jersey: 1-16.

Hostetter, Gene H., Savant, Clement J., Jr., and Stefani, Raymond T.. 1982. Design of Feedback Control Systems. Holt, Rinehart and Winston, New York, New York.

Hovy, Eduard H. 1993. Automated discourse generation using discourse structure relations. Artificial Intelligence. Elsevier Science Publishers, New York, New York, Vol. 63: 341-385.

Hudson, Keith. 1984. Introducing CAL: A Practical Guide to Writing Computer-Assisted Learning Programs. Chapman and Hall Ltd, London, England.

Hume, Gregory D. 1995. Using Student Modelling to Determine When and How to Hint in an Intelligent Tutoring System. IIT Ph.D. Dissertation. Computer Science Department, Illinois Institute of Technology, Chicago, Illinois.

Hume, Gregory, Michael, Joel, Rovick, Allen, and Evens, Martha. 1996. Hinting as a Tactic in One-on-One Tutoring. The Journal of the Learning Sciences. Vol. 5, No. 1: 23-47.

Jacobs, O.L.R. 1974. Introduction to Control Theory. Oxford University Press, Oxford, England.

Johnson, Kent R., and Layng, T.V. Joe. 1994. The Morningside Model of Generative Instruction. In Gardner, Ralph, III et al., Eds. Behavior Analysis in Education: Focus on Measurably Superior Instruction. Brooks/Cole Publishing Company, Pacific Grove, California: 173-197.

Johnson, William B. 1988. Pragmatic Considerations in Research, Development, and Implementation of Intelligent Tutoring Systems. In Polson, Martha C., and Richardson, J. Jeffrey, Eds. Foundations of Intelligent Tutoring Systems. Lawrence Erlbaum Assoc., Hillsdale, New Jersey: 191-207.

Kellerman, Dana F. (Ed.). 1981. New Webster's Dictionary of the English Language. Delair Publishing Company, USA.

Khuwaja, Ramzan A., Evens, Martha W., Rovick, Allen A., and Michael, Joel A. 1994. Architecture of CIRCSIM-Tutor (v.3): A Smart Cardiovascular Physiology Tutor. Proceedings of the IEEE Seventh Symposium on Computer-Based Medical Systems: 158-163.

Khuwaja, Ramzan A., Rovick, Allen A., Michael, Joel A., and Evens, Martha W. 1994. A Tale of Three Tutoring Protocols: The Implications for Intelligent Tutoring Systems. In Yfantis, E.A., Ed. Intelligent Systems. Kluwer Academic Publishers: 109-118.

Khuwaja, Ramzan. 1994. A Model of Tutoring: Facilitating Knowledge Integration Using Multiple Models of the Domain. Ph.D. Dissertation, Department of Computer Science and Applied Mathematics, Illinois Institute of Technology, Chicago, Illinois.

Kim, Jung Hee, Freedman, Reva, and Evens, Martha W. 1998. Relationship between Tutorial Goals and Sentence Structure in a Corpus of Tutoring Transcripts. In Evens, Martha W., Ed., Proceedings of the Midwest Artificial Intelligence and Cognitive Science Society Conference, Dayton, Ohio: 124-131.

Kim, Nakhoon. 1991. CIRCSIM-Tutor: An Intelligent Tutoring System for Circulatory Physiology. Ph.D. Dissertation, Department of Computer Science and Applied Mathematics, Illinois Institute of Technology, Chicago, Illinois.

Lewis, Harry R., and Papadimitriou, Christos, H. 1981. Elements of the Theory of Computation. Prentice Hall, Englewood Cliffs, New Jersey: Chapter 3.

Li, Jun. 1991. Abase: A Computer Program That Teaches Physiological Acid Base Regulation. M.S. Thesis, Department of Computer Science and Applied Mathematics, Illinois Institute of Technology, Chicago, Illinois.

Li, Jun, Seu, Jai H., Evens, Martha, Michael, Joel A., and Rovick, Allen A. 1992. Computer dialogue system (CDS): A system for capturing computer-mediated dialogue. Behavior Research Methods, Instruments, & Computers. Vol. 24, No. 4: 535-540.

Lias, Edward J. 1982. Future Mind. Little, Brown and Company, Boston, Massachusetts.

Littman, David, and Soloway, Elliot. 1988. Evaluating ITSs: The Cognitive Science Perspective. In Polson, Martha C., and Richardson, J. Jeffrey, Eds. Foundations of Intelligent Tutoring Systems. Lawrence Erlbaum Assoc., Hillsdale, New Jersey: 209-242.

Luff, Paul, Gilbert, Nigel, and Frohlich, David, Eds. 1990. Computers and Conversation. Academic Press, London, England.

Maltz, Maxwell. 1960. Psycho-Cybernetics. Pocket Books, New York, New York.

McCaffrey, Anne. 1990. Pegasus in Flight. Ballantine Books, New York, New York.

McCarthy, Michael. 1993. Spoken Discourse Markers in Written Text. In Sinclair, John M., Hoey, Michael, and Fox, Gwyneth, Eds. Techniques of Description: Spoken and Written Discourse. Routledge, Inc., New York, New York: 170-182.

McKeown, Kathleen R. 1985. Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text. Cambridge University Press, Cambridge, England.

Michael, Joel A, and Rovick, Allen A. 1996. The Once and For All History of HEARTSIM/CIRCSIM. Web document. http://www.csam.iit.edu/~circsim/prehistory.html

Miller, James R. 1988. The Role of Human-Computer Interaction in Intelligent Tutoring Systems. In Polson, Martha C., and Richardson, J. Jeffrey, Eds. Foundations of Intelligent Tutoring Systems. Lawrence Erlbaum Assoc., Hillsdale, New Jersey: 143-189.

Norman, Donald A. 1990. The Design of Everyday Things. Doubleday, New York.

Norman, Donald A. 1992. Turn Signals Are the Facial Expressions of Automobiles. Addison-Wesley, Reading, Massachusetts.

Novick, David G., and Sutton, Stephen. 1994. An Empirical Model of Acknowledgment for Spoken-Language Systems. Proceedings of the Annual Conference of the ACL, Las Cruces, New Mexico: 96-101.

Palmer, David C., and Donahoe, John W. 1992. Essentialism and Selectionism in Cognitive Science and Behavior Analysis. American Psychologist. Vol. 47, No. 11: 1344-1358.

Pinker, Steven. 1994. The Language Instinct. W. Morrow and Co., New York, New York.

Pryor, Karen. 1985. Don’t Shoot the Dog. Bantam Books, New York, New York.

Quinlan, J. Ross. 1993. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, Inc., San Mateo, California.

Richardson, J. Jeffrey. 1988. Directions for Research and Applications. In Polson, Martha C., and Richardson, J. Jeffrey, Eds. Foundations of Intelligent Tutoring Systems. Lawrence Erlbaum Assoc., Hillsdale, New Jersey: 243-252.

Sanders, Gregory A. 1995. Generation of Explanations and Multi-Turn Discourse Structures in Tutorial Dialogue, Based on Transcript Analysis. Ph.D. Dissertation, Department of Computer Science and Applied Mathematics, Illinois Institute of Technology, Chicago, Illinois.

Sarbin, Theodore R. 1976. Cross-Age Tutoring and Social Identity. In Allen, Vernon L., Ed. Children as Teachers: Theory and Research on Tutoring. Academic Press, New York, New York: 27-40.

Schiffrin, Deborah. 1987. Discourse Markers. Cambridge University Press, New York.

Self, John A. 1988. Bypassing the Intractable Problem of Student Modeling. In Frasson, Claude, and Gauthier, Gilles, Eds. Intelligent Tutoring Systems: At the Crossroad of Artificial Intelligence and Education. Ablex Publishing Corporation, Norwood, New Jersey: 107-123.

Shah, Farhana. 1997. Recognizing and Responding to Student Plans in an Intelligent Tutoring System: CIRCSIM-Tutor. Ph.D. Dissertation, Department of Computer Science and Applied Mathematics, Illinois Institute of Technology, Chicago, Illinois.

Shah, Farhana, and Evens, Martha. 1997. Classifying Student Initiatives and Tutor Responses: A First Step toward Mixed-Initiative Dialogue. Unpublished document. Department of Computer Science and Applied Mathematics, Illinois Institute of Technology, Chicago, Illinois.

Shim, Leemseop. 1991. Student Modeling for an Intelligent Tutoring System: Based on the Analysis of Human Tutoring Sessions. Ph.D. Dissertation, Department of Computer Science and Applied Mathematics, Illinois Institute of Technology, Chicago, Illinois.

Shneiderman, Ben. 1984. Response Time and Display Rate in Human Performance with Computers. ACM Computing Surveys. Vol. 16, No. 3, September 1984: 265-285.

Sinclair, John M. 1993. Written Discourse Structure. In Sinclair, John M., Hoey, Michael, and Fox, Gwyneth, Eds. Techniques of Description: Spoken and Written Discourse. Routledge, London: 6-31.

Sinclair, John M., and Coulthard, R. Malcolm. 1975. Towards an Analysis of Discourse: The English Used by Teachers and Pupils. Oxford University Press, London, England.

Skinner, B. F. 1957. Verbal Behavior. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Spitkovsky, John A., and Evens, Martha W. 1993. Negative Acknowledgments in Natural Language Tutoring Systems. Proceedings of the Annual MAICS Conference: 41-45.

Stubbs, Michael. 1983. Discourse Analysis. The University of Chicago Press, Chicago, Illinois.

Stumpf, Samuel Enoch. 1982. Socrates to Sartre. 3rd edition. McGraw-Hill Book Company, New York, New York.

Tanenbaum, Andrew S. 1989. Computer Networks. 2nd edition. Prentice-Hall, Englewood Cliffs, New Jersey.

Taylor, R. Gregory. 1998. Models of Computation and Formal Languages. Oxford University Press, New York, New York.

Taylor, Robert. 1980. Introduction. In Taylor, Robert, Ed. The Computer in the School: Tutor, Tool, Tutee. Teachers College Press, Columbia University, New York, New York: 1-11.

van Lehn, Kurt. 1988. Student Modeling. In Polson, Martha C., and Richardson, J. Jeffrey, Eds. Foundations of Intelligent Tutoring Systems. Lawrence Erlbaum Assoc., Hillsdale, New Jersey: 55-78.

Waterman, Donald A. 1986. A Guide to Expert Systems. Addison-Wesley, Reading, Massachusetts.

Woo, Chong Woo. 1991. Instructional Planning in an Intelligent Tutoring System: Combining Global Lesson Plans with Local Discourse Control. Ph.D. Dissertation, Department of Computer Science and Applied Mathematics, Illinois Institute of Technology, Chicago, Illinois.

Woo, Chong W. 1992. A Multi-Level Dynamic Instructional Planner for an Intelligent Physiology Tutoring System. ONR Interim Report, April 1992.

Wooffitt, Robin. 1990. On the Analysis of Interaction: An Introduction to Conversation Analysis. In Luff, Paul, Gilbert, Nigel, and Frohlich, David, Eds. Computers and Conversation. Academic Press, London, England: 7-38.

Woolf, Beverly. 1990. 20 Years in the Trenches: What Have We Learned? In Frasson, Claude, and Gauthier, Gilles, Eds. Intelligent Tutoring Systems: At the Crossroad of Artificial Intelligence and Education. Ablex Publishing Corporation, Norwood, New Jersey: 234-250