Far East Journal of Experimental and Theoretical Artificial Intelligence
Volume 1, Issue 2, 2008, Pages 87-125
Published online: August 12, 2008
This paper is available online at http://www.pphmj.com
© 2008 Pushpa Publishing House

EXTRACTING INFORMATION FROM NATURAL LANGUAGE INPUT TO AN INTELLIGENT TUTORING SYSTEM

MICHAEL S. GLASS and MARTHA W. EVENS

Department of Mathematics and Computer Science
Valparaiso University
Valparaiso, IN 46383, U. S. A.
e-mail: michael.glass@valpo.edu

Computer Science Department
Illinois Institute of Technology
Chicago, IL 60616, U. S. A.
e-mail: evens@iit.edu

Abstract

We have constructed a new module to process student natural language input to CIRCSIM-Tutor, an intelligent tutoring system designed to help medical students learn to solve problems involving the negative feedback process that regulates blood pressure in the human body. CIRCSIM-Tutor spends most of its time engaging the student in a natural language-based dialogue. The new input understander uses an information extraction approach that is robust enough to handle free-form student input. We describe an evaluation of CIRCSIM-Tutor by forty-two students at Rush Medical College, with particular emphasis on the performance of the input understander.

Keywords and phrases: natural language understanding, intelligent tutoring system, information extraction, finite state parsing.

This research was partially supported by the Cognitive Science Program of the United States Office of Naval Research under Grant No. N00014-94-1-0338 to Illinois Institute of Technology and by the National Science Foundation under award No. 0634049 to Valparaiso University. The content does not reflect the position or policy of the government and no official endorsement should be inferred.

Communicated by Shun-Feng Su

Received January 25, 2008; Revised May 2, 2008

Introduction

We have constructed a new module designed to understand natural language student input to CIRCSIM-Tutor, an intelligent tutoring system that helps medical students learn to solve problems involving the negative feedback loop that regulates blood pressure in the human body. It has been tested on forty-two students in a standard physiology laboratory at Rush Medical College.

CIRCSIM-Tutor is a language-based tutor. In fact, the designers of this system were motivated by the belief that the use of language enhances learning, that learning new scientific concepts is inextricably enmeshed with learning the language of that science, and that trying to explain new phenomena in words is the best way to understand them. Thus the system primarily engages the student in dialogue. It begins by describing a disturbance to the cardiovascular system, such as a hemorrhage or a pacemaker malfunction; it asks the student to enter qualitative predictions about the responses of seven physiological variables to this perturbation; and then it launches a tutorial dialogue.

The new input understander was designed to be robust enough to handle free-form student input and fast enough that the system can respond to the student in under two seconds. Thus we chose an information extraction approach using a cascade of finite state machines. The new module was one of several improvements so that this version of CIRCSIM-Tutor became the first to be used with large classes of students. Later in this paper we describe the evaluation of our system in a regularly scheduled class laboratory exercise in November, 1999.

Interactive dialogue-based tutoring systems are less common and harder to build than other kinds of computer-aided instruction. But there is scientific evidence for endeavoring to create a dialogue-based system. There is evidence in the education literature that human tutoring is highly effective compared to other forms of instruction, and there are studies that show that putting theories into words and giving explanations aids learning.

Studies demonstrating the value of tutoring

There are a number of experiments that demonstrate the value of

one-on-one tutoring and make it clear that it is an effective kind of teaching. Cohen et al. [14] conducted a metastudy of 65 controlled evaluations of school tutoring programs, concluding that tutored students outperform control subjects to a high degree of statistical significance. Bloom [5] reported good results comparing classes of elementary school students taught in a normal classroom using mastery learning methods against students tutored individually or in groups of two or three together. Bloom found that the average tutored student performed two standard deviations above the average normal classroom student. Michael and Rovick [53] demonstrated that one-on-one tutoring is effective even for bright and motivated adults like those at Rush Medical College. Students tutored for one hour showed significantly more improvement between pre-test and post-test than a control group that spent the same time on task reading carefully chosen relevant text. A recent paper by VanLehn et al. [69] shows that tutoring produces greater improvement than reading under a variety of circumstances.

Studies demonstrating the importance of language in learning

Several recent studies have demonstrated the importance of natural language in learning. There is reason to believe that merely making students talk or write has value. That is, independent of whether the system understands what they say, making students articulate ideas improves retention and understanding. Chi et al. [13] demonstrated experimentally what they call the "self-explanation effect", where learners are prompted to explain back what they just learned to a neutral listener. Based on this notion, Aleven and Koedinger modified a geometry tutor to prompt for student explanations of their proofs [1, 2]. Their experiments showed that encouraging self-explanation by students improved learning outcomes. However without two-party dialogue---students did not receive feedback on their explanations from the geometry tutor---students left the majority of the explanation boxes empty. Aleven et al. later produced a tutor [3] that attempts to understand student input and to provide feedback to improve the quality of that input. In the realm of tutors that talk but do not listen, Di Eugenio et al. [18, 19] have demonstrated that more sophisticated natural language tutorial utterances produce better outcomes than the same knowledge expressed in template-constructed unsophisticated language.

Graesser [34, 36] dissected 66 one-hour tutoring sessions with untrained human tutors. Analysis of pre-tests and post-tests showed that tutoring was effective despite the fact that these tutors rarely used sophisticated tutoring strategies. The tutors asked many deep questions; the students also asked deep questions at the rate of eight per hour, which is much more often than in ordinary classroom teaching. Graesser et al. [37] concluded that it is these deep questions and the attempts to state them and answer them that account for much of the learning.

There is also evidence that dialogue itself is an important factor in learning. In an ingenious series of experiments, Fox Tree [22] has shown that people learn more from overhearing a dialogue than from overhearing a monologue with the same content.

Other experiments in adding natural language interaction to tutoring systems

From the beginning of research on intelligent tutoring systems, many researchers believed that natural language was essential to tutoring. Carbonell's [9] SCHOLAR geography tutoring system asked questions and produced the first mixed-initiative dialogue with a computer. Collins and his collaborators [15, 16] continued to stress natural language interaction as this project grew. Brown and Burton's SOPHIE system [7, 8], which tutored students in basic concepts of electricity, was also based on natural language interaction. Wilensky's Unix Consultant [70], which is more of a coach than a tutor, also used natural language interaction extensively, based on a semantic grammar for Unix concepts. The most pronounced difficulty with SCHOLAR and other early dialogue-based tutoring systems was a lack of multi-turn coherence to the dialogue. Typically a question would not be related to the previous or following question.

Cawsey [10] studied tutoring in the domain of electrical circuits and focused on the interactive nature of tutor explanations. In building a system to emulate this kind of tutoring she created a number of dialogue plans for tutoring. Cawsey's perception of the interactive nature of explanations in tutoring helped us to recognize the same kind of behavior in transcripts of human tutors, prompting the development of multiturn planning in CIRCSIM-Tutor [64].

Interest in natural language based tutoring systems has increased in the last decade. While ITS'96 had no papers on tutoring systems based on natural language dialogue except Freedman and Evens [26] from the CIRCSIM-Tutor project, the situation has since changed. Both the Conference on Intelligent Tutoring Systems and the Conference on Artificial Intelligence in Education regularly feature multiple sessions and ancillary workshops on natural language processing in tutoring systems. The center of much of this new activity was the NSF sponsored Circle Project, joint between the University of Pittsburgh and Carnegie Mellon University. VanLehn directed development of the Atlas Project [27, 59, 60, 68]. Atlas-Andes [41] is a physics tutor that combines a model tracing approach with natural language dialogue with the goal of improving the student's ability to solve problems in qualitative reasoning. Atlas uses Freedman's Atlas Planning Environment (APE) [23, 24, 25], a hierarchical, opportunistic, reactive planner, to plan natural language dialogue. Rosé's CARMEL parser [59, 60], which serves as the input understanding component of Atlas-Andes and other Circle projects, is intended to handle a wide range of student input, especially student-generated explanations. Lane and VanLehn [46] have built a dialogue-based tutor for program design. Moore's emphasis on the role of planning in generation, especially in the generation of dialogue, has been of great importance in text generation in general [55], with significant contributions to tutorial dialogue generation especially. Her Beetle system [17] illustrates what good planning can contribute to natural-sounding dialogue. Heffernan and others [38] have done significant work building a framework for more easily creating dialogue-based tutoring.

Graesser and others have produced another set of dialogue based tutoring systems [35]. The input understanding in his AutoTutor uses latent semantic analysis [45] for robust processing of complex student utterances. Student answers that are not easily recognized by more conventional means (usually the longer ones) are evaluated by their similarity to a collection of archetypal answers to the question. Some of the archetypes contain known misconceptions or partially correct answers. AutoTutor was originally designed as a computer literacy tutor, but it has now been ported to other domains such as physics [69].

There are now dialogue-based computer tutors that use spoken input and output, for example SCoT at Stanford [56] and ITSPOKE at Pittsburgh [51]. Litman's system, ITSPOKE, which is essentially Atlas-Andes with spoken input and output, has been shown to produce better and faster learning outcomes than Atlas-Andes. These systems still need the kind of language understanding and generation mechanisms developed by CIRCSIM-Tutor and other typed-language systems, as well as the kinds of dialogue strategies used in human tutoring.

How the CIRCSIM-Tutor project got started

Our goal from the beginning was to emulate a subset of expert human tutoring. Our primary source of data on both tutor and student behavior is a set of human-tutored keyboard-to-keyboard tutoring transcripts that the CIRCSIM-Tutor project has accumulated. These tutoring sessions were held with the student and the tutor in different rooms communicating only through the computer keyboard and screen in order to match the conditions under which the students would interact with the computer tutor; this process also simplified the collection of the session transcripts. We have 75 transcripts from expert tutor sessions, mostly one hour in length [33, 42, 43, 44]. In this paper we use raw transcript extracts, so that the reader can understand the kind of input that the system encounters. We have sometimes cleaned up the portion typed by the human tutor to make it easier to read. References to transcript abstracts are of the form "K51-tu-17-3" meaning transcript 51, turn 17, sentence 3. It is the tutor's turn. In all 75 transcripts the tutors are Joel Michael and Allen Rovick, professors of physiology at Rush Medical College; the students are first-year medical students. In addition to the data from human tutoring we also use examples from logs of medical students using various revisions of CIRCSIM-Tutor Version 2.

Before this project started, in addition to tutoring students face-to-face themselves, Michael and Rovick had already built several CAI systems for students of physiology: CIRCSIM, ABASE, and GASP [50, 61, 62, 63]. These systems have been very successful and are still being widely used, but Michael and Rovick decided that natural language input and output are essential to making better systems.

This paper outlines the problems we set out to solve, explains our solution, and describes an evaluation with a class of medical students. In

the next section we discuss the operation of CIRCSIM-Tutor. Then we describe the operation of the new input understanding component, showing the types of phenomena it is designed to address. We describe the results of evaluating CIRCSIM-Tutor with regard to learning gains and with regard to the performance of the input understander. Finally we outline further input understanding phenomena we might address in the future.

The Operation of CIRCSIM-Tutor

CIRCSIM-Tutor engages in problem-solving dialogues. Each problem is posed as a "procedure" in medical parlance, an influence on the human organism that will affect blood pressure. The student predicts changes to physiological parameters that are involved in blood pressure regulation. Figure 1 contains a view of the main screen with a description of the procedure that the student is working on, the predictions that the student has made so far, and the start of the remedial dialogue from Session 39 in November, 1999. Figure 2 contains just the dialogue itself.

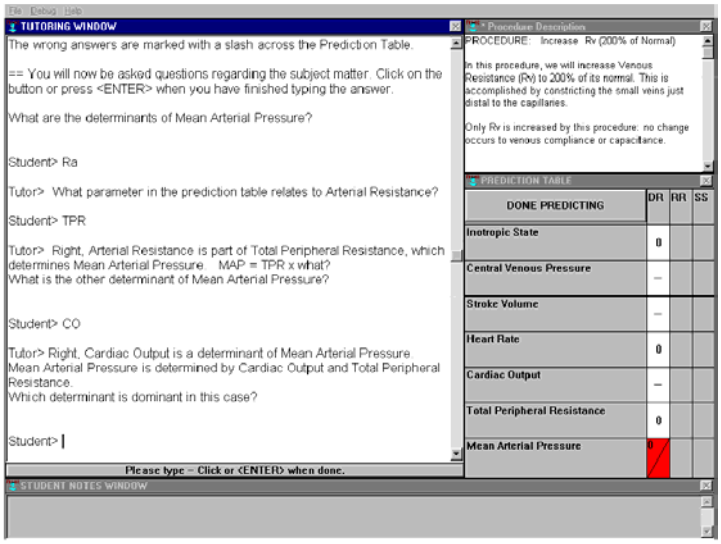


Figure 1. CIRCSIM-Tutor screen containing a fragment of dialogue from Session 39.

CIRCSIM-Tutor presupposes that students have already been exposed to the relevant physiology. Our system is designed to help

students learn to solve problems by asking questions and giving hints, as opposed to giving the student a lecture. So the dialogue in Figure 2 begins with a question.

T: You will now be asked questions regarding the subject matter.

Click on the button or press <ENTER> when you have finished typing the answer.

T: What are the determinants of Mean Arterial Pressure?

S: Ra

T: What parameter in the prediction table relates to Arterial Resistance?

S: TPR

T: Right, Arterial Resistance is part of Total Peripheral Resistance, which determines Mean Arterial Pressure.

T: MAP = TPR x what?

T: What is the other determinant of Mean Arterial Pressure?

S: CO

T: Right, Cardiac Output is one determinant of Mean Arterial Pressure.

T: Mean Arterial Pressure is determined by Cardiac Output and Total Peripheral Resistance.

T: Which determinant is dominant in this case?

S: CO

Figure 2. Dialogue fragment from Session 39.

CIRCSIM-Tutor's input understander and its operation is the focus of this paper, but we will now describe each of the other major modules very briefly in order to help the reader understand the system context. Woo et al. [71] and Evens and Michael [21] provided a more detailed description of the system architecture.

The instructional planner is the central controller of the system [71]. It fetches and interprets plans from the Knowledge Base, and calls on the other major modules to carry out the steps in those plans. These plans are organized at four different levels. The highest level provides the overall structure for a tutoring session with an individual student. Each session is organized as an introduction followed by a series of problems to solve. Next is the problem level; the rules at this level provide the overall organization for tutoring a problem, beginning with the presentation of

the problem, three separate stages of physiological response (the Direct Response Stage, the Reflex Response Stage, and the Steady State Stage), and the problem summary. Next comes the strategy level, with strategies for the problem-solving process and for teaching each topic within the problem. The bottom level is the tactical level, which provides tactics for teaching each concept in each topic.

The student modeler evaluates the student input and reports to the instructional planner about the student's progress. It categorizes student responses as to whether they are correct, are near misses that the instructional planner can respond to, or exhibit known misconceptions. Zhou et al. [73, 74] described some of these functions.

The text generator [11, 12] receives instructions from the instructional planner about the communicative goal (whether it is delivering an explanation, a hint, a question, a request, an acknowledgment, etc.) and the semantic content (represented as a logic form) of each sentence. It generates a sentence for display. The text generator uses a Lexical Functional Grammar and a lexicon for some utterances and a library of canned text and templates for others, providing a tradeoff between convenience and expressive power for the system developers. Generating a sentence at a time simplifies the task, but it occasionally produces an incoherent turn. Yang [72] has developed a turn planner to avoid this difficulty, but it was not available at the time of the experiment described here.

Figure 2 illustrates what happens when CIRCSIM-Tutor asks the student the question "What are the determinants of Mean Arterial Pressure?" and the student answers "Ra". The instructional planner passes the question (expressed as a logic form) and the student's input (a string) to the input understander. The logic form for the question is:

(QUESTION (AFFECTED-BY *MAP*)).

The input understander tries to interpret the input as an answer to the tutor's most recent question. The lexicon contains "Ra" as an abbreviation for "Arterial Resistance". There are several possible synonyms in the lexicon which are all abbreviated as Ra internally. The input understander verifies with an ontology of neurological concepts that this answer is conceptually within the domain of possible answers that the

system will be able to respond to. (In other words, it verifies that the student did not type "purple cow".) It returns a logic form to the instructional planner:

(ANSWER (AFFECTED-BY *MAP* ((*Ra*)))).

The student modeler classifies this particular answer as a "near-miss", i.e., it is a correct answer, although not the expected answer. The instructional planner decides that the next step is to hint, to push the student toward the expected answer. The planner produces the logic form for this hint, which is realized as:

T: What parameter in the prediction table relates to Arterial Resistance?

This hint succeeds and the student names a correct parameter. However the original question should have elicited two determinants of MAP; the student has now produced only one. The system then delivers another hint "MAP = TPR * what?" to prompt the student to come up with the other part of the answer.

The New Input Understander

The task of the input understander is to correct spelling errors, if any, expand abbreviations and ellipses so "don't" becomes "do not", parse the input, produce the appropriate logic form, determine whether it is really within the domain of things that might constitute an answer or a question from the student, and return the result to the instructional planner. Robustness on the part of the input understander is vital to conducting an interactive dialogue of this type. Earlier versions of the system confused the student by rejecting unexpected but physiologically correct answers as wrong. The system often failed to recognize valid elliptical answers. Not all these issues can be solved by the input understander by itself, but the new software is an essential part of the solution.

The new input understander is based on cascaded finite state machines [58], a robust approach to language processing developed by the information extraction [49] community. Information extraction does not attempt to build comprehensive models of the text, but rather to seek out and extract from the text specific bits of knowledge.

An alternative to finite state information extracting machines is symbolic parsing of messy text, as exemplified by Lavie's [47] GLR* parser, based on Tomita's Generalized LR Parsing Algorithm [67]. Tomita's algorithm replaced the basic stack used in context-free parsing with a more complicated data structure, one that holds the results of many alternative parses at once and combines their common parts. Lavie deals with extragrammatical input by selectively omitting words that do not fit. The parse that succeeds after dropping the fewest words is chosen to represent the probable meaning.

Input understanding starts with the question most recently asked by the system in combination with the student's response. Table 1 shows example questions and the logic form for each, along with each question's expected answer. Some of these questions are expressed in different words if the system asks them again, but the content stays the same.

Table 1. Some CIRCSIM-Tutor Questions, Logic Forms, and Answers

Here <var> is a physiological parameter, <varlist> is a list of parameters,
<val> is +, –, or 0, for values that increased, decreased, or did not change
<mech> is a mechanism of control, i.e., NEURAL or PHYSICAL,
<rel> is POSITIVE or NEGATIVE, the direction of a relationship between parameters,
<y-or-n> is YES or No

Which determinant is dominant in this case?
(QUESTION (ACTUAL-DETERMINANT <var>))
(ANSWER (ACTUAL-DETERMINANT <var>))

Which variables are changed by the reflex?
(QUESTION (AFFECT REFLEX <varlist>))
(ANSWER (AFFECT REFLEX <varlist>))

Which of the variables in the prediction table are determinants of <var>?
(QUESTION (AFFECTED-BY <var>))
(ANSWER (AFFECTED-BY <var> ((<varlist>))))

Will the reflex compensate for the change in Mean Arterial Pressure in DR?
(QUESTION (COMPENSATE REFLEX CHANGE *MAP*))
(ANSWER (<y-or-n>))

Will the reflex overcompensate for the change in Mean Arterial Pressure in DR?

(QUESTION (OVERCOMPENSATE REFLEX CHANGE *MAP*))

(ANSWER (<y-or-n>))

By what mechanism is <var> controlled?

(QUESTION (MECHANISM <var>))

(ANSWER (MECHANISM (<mech>) <var>))

Is the relationship from <var-1> to <var-2> direct or is it inverse?

(QUESTION (RELATION <var-1> <var-2>))

(ANSWER ((<rel>) <var-1> <var-2>))

What stage must the value of <var> follow in SS?

(QUESTION (FOLLOW <var>))

(ANSWER (<stage>))

Which variable is regulated by the baroreceptor reflex?

(QUESTION (REGULATE BARORECEPTOR-REFLEX <var>))

(ANSWER (<var>))

What is the correct value of <var>?

(QUESTION (VALUE <var>))

(ANSWER (VALUE <var> <val>))

What is the value of <var> in DR?

(QUESTION (VALUE-DR <var>))

(ANSWER (VALUE-DR <var> <val>))

Note: *There are two more logic forms for RR and SS similar to VALUE-DR.*

The input understander carries out the following stages of processing for each new piece of student input, as illustrated in the block diagram in Figure 3:

- Look up words in the lexicon
- Correct spelling (which is combined with lexical lookup)
- Recognize student initiatives and hedges using a finite state transducer
- Recognize possible answers to the question at hand with a cascade of finite state transducers
- Produce a logic form and check for errors

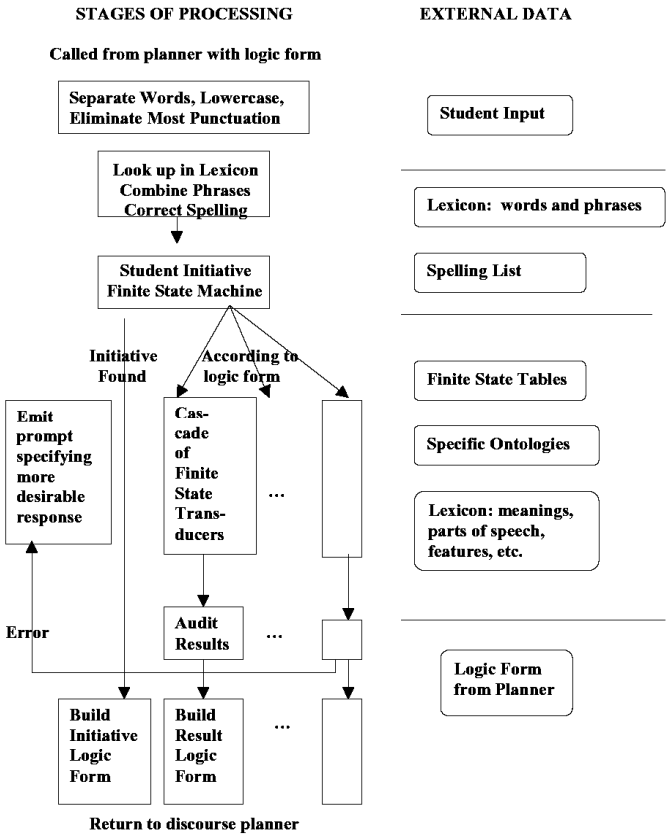


Figure 3. Organization of the input understander.

Lexical lookup and spelling correction

Spelling correction is an essential function in a system that accepts free text from users. In a dialogue situation, coming back to the user with a list of alternatives to a misspelled word is distracting. Our spelling correction is a process of quickly finding the closest match in the lexicon to the unrecognized word (Lee and Evens [48], and Elmi and Evens [20]). A two-letter sliding window compares the i-th and $(i+1)$-th letters of the unknown word with the j-th and $(j+1)$-th letters of a candidate word in the lexicon. The match process detects elided, added, and changed characters, as well as character reversals, which are common in keyboard entry. Weights are assigned to each mismatch in a scoring scheme informed by published data and our own experience. The result is a list of

words from the lexicon with the lowest weighted error score. To increase the chance of correctly recognizing a word, most common abbreviations in our domain are included in the lexicon along with some common mistypings like “hte” for “the”. Given that we have observed users abbreviating words by shortening them, letters that have been dropped from the end of a word are weighted less than letters dropped from the middle.

There is a certain amount of polysemy built into our domain. Much of the tutoring revolves around qualitative change in cardiovascular parameters, and the major verbs of change in English are consistently spelled the same way as associated verbal nouns, like *increase*, *decrease*, *rise*, *fall*, and *change* itself. More polysemy is introduced by systematic abbreviation used in the physiology sessions we are automating. Students use the letter *D* to stand for “direct” as well as “decrease” and “down”; similarly, the letter *I* indicates “inverse” or “indirect” as well as “increase”. The plus sign is used for both “increase” and “direct”. The minus sign (which also doubles as a hyphen) is used to signify both “decrease” and “indirect”.

Students in our experiments rarely use the string “*I*” to indicate a personal pronoun in an answer to a system question. The personal pronoun does appear in hedges and student initiatives, as described below.

The handling of phrases in the lexical lookup phase is based on the “maximal munch” strategy. The system takes the longest phrase in the lexicon that matches the next segment of input words. To speed up this strategy we built a table of all proper prefixes of phrases in the lexicon. A proper prefix is a sequence of words up to but not including the whole phrase. For example “right” and “right atrial” are the proper prefixes of “right atrial pressure”.

Recognizing student initiatives and processing hedges

Before the system attempts to extract an answer from the student utterance, two confounding factors are disposed of. The utterance is checked to see if it is a student initiative instead of an answer. If it is not an initiative, student hedges are recognized and stripped out.

Student initiatives are points where the student attempts to take control of the conversation. The new input understander can handle two simple kinds of initiative. One is where the student essentially refuses to answer the question by pressing the enter key without typing anything or by typing a phrase such as "I don't know" or "I don't understand". In these cases the input understander returns to the planner with an indication that no answer was received and the planner produces a hint or gives the student the correct answer. The other type of initiative that the input understander can handle is phrases such as "I don't understand RAP", "I am confused about RAP", and "What is RAP?" The input understander translates these initiatives to a logic form that requests a definition:

(QUESTION (EXPLAIN *RAP*))

A hedged answer may express uncertainty. The input understander recognizes that the student is hedging when the input contains an adverb like "probably" or "maybe", or ends with one or more question marks as in "CO???" Another frequently attested hedge in our human tutoring sessions is variations of "I think", as in the examples below.

K20-st-35-5: SO I THINK THAT SV GOES UP.

K20-st-37-1: SINCE CC IS NOT CHANGING THEN I WOULD THINK NO CHANGE IN SV.

K20-st-37-4: D THEN RAPI I THINK SV I.

In transcripts of human tutoring, "how about" might be ambiguous between a marker for hedged answer and a marker for an initiative. In this example the student uses "How about" to hedge an answer:

K25-tu-52-2: But what determines the volume of blood in the central venous compartment?

K25-st-53-1: How about co?

But our expert tutors often use "How about" to ask a question:

K11-tu-65-2: How about the influence of a change in CO on RAP?

K11-st-66-1: Ico->Ivenous pressure.

Hedge language is stripped out before the sentence is analyzed further. Michael and Rovick decided after the first eight transcribed human tutoring sessions that they should stop responding to hedges, but in fact they are observed to sometimes give more explicit explanations and more enthusiastic positive acknowledgments when students hedge [6]. Recently Pon-Barry et al. [57] have shown that responding to student uncertainty in the SCoT tutor improves learning outcomes.

Beyond recognizing hedges as an input phenomenon, CIRCSIM-Tutor does not take hedges into account in its planning, but this might be an opportunity for improvement. In fact, we have seen very few hedges and student initiatives in our trials with CIRCSIM-Tutor so far [4], but we may possibly see more as the system's natural language capabilities improve.

Processing with a cascade of finite state transducers

Cascaded finite state transducers have often been used in information extraction tasks; a good example is FASTUS [39]. Finite state machines are popular because they are fast and modular. Their running time is linear in the length of the input, while most algorithms for context-free grammars are slower. When transducers are cascaded, each machine produces an output that is some modification of the input. For example, a common model is to construct a transducer that takes input with the parts of speech marked, looks for noun phrases, and outputs the same string with markers at the beginning and end of each noun phrase. This output might then be input to a transducer that looks for coreference between the phrases found by the previous machine. The output of the last machine in the sequence is used for constructing the input understander's result, which is passed back to the planner.

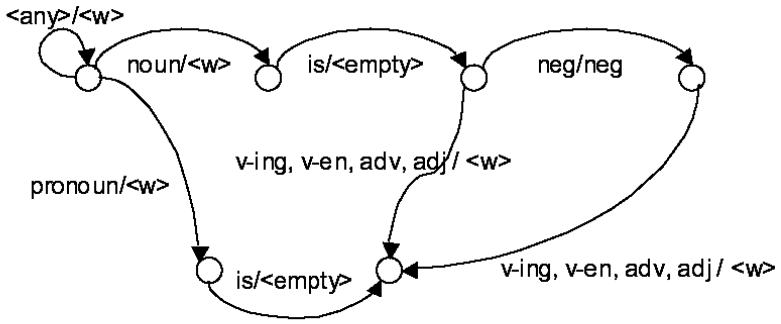
Figure 4 is an example of a non-deterministic Mealy machine [40] finite state transducer, a graph of state nodes connected by arcs. An arc consists of a label, which is matched against a symbol from the input string, and a string of output symbols to emit when the label is matched. For example, in Figure 4 one arc matches any noun from the input and emits the same word, the next arc matches "is" and emits nothing, and

another arc matches verb participles (designated "*v*-ing" and "*v*-en"). This fragment of transducer will, for example, transform "pressure is decreasing" to "pressure decreasing". Each machine has a comparison function that is used for matching input symbols against arc labels. The interpreter for these machines, a LISP function, iterates over the input string, making state transitions, and collecting emitted symbols until the input string has been consumed. The machines are non-deterministic since it is possible for more than one arc or even none to match in a given state. The interpreter mechanism maintains a set of all possible routes, where each route contains a possible current state together with the list of symbols emitted en route to the state.

The question at hand partly determines which finite state transducers process the student's utterance, thus extracting the information needed for answering the question. For example, to process the answer to a question about the relationship between two variables, machines for recognizing variable names and relationships are employed. If the question was about the qualitative change in a variable, the variable name machine is used in conjunction with a qualitative change machine.

CIRCSIM-Tutor's language understanding issues can be enumerated by looking at the finite state machines we built to handle each one. We describe the function of many of these here.

The finite state machine in Figure 4 deletes finite forms of the verb "to be" in student inputs like "SV is not changed" and "it is SV". We need a special machine to perform copula deletion in CIRCSIM-Tutor because one of the important domain parameters is named "Inotropic State", usually abbreviated "IS". We cannot rely on case to distinguish the abbreviation "IS" from the copula "is". This machine will not alter the input "is increased" or "sv increased", but "sv is increased" will be changed to "sv increased" and "is is increased" becomes "is increased". This machine, the first machine in many of the cascades, illustrates the style of the finite state transducer. After it runs, succeeding machines can assume that any occurrence of "is" is a reference to Inotropic State, not a finite verb.



X/Y = match symbol X, emit symbol Y

<any> = match any word

<w> = word that matched

Figure 4. Finite state machine for copula deletion.

Since CIRCSIM-Tutor is designed to teach qualitative causal reasoning it spends much of its time discussing qualitative changes in parameters. Several finite state machines in concert detect whether the student is talking about such a change. One machine looks for parameter names, another for qualitative changes (up, down, or no change), another for combinations of the two. The meaning attributes are emitted as shown in Table 2. For example, the word "afterload" is translated into "MAP" (short for Mean Arterial Pressure). Any word not selected in this process is dropped from the string and ignored in further steps of the translation process.

The negation machine looks for a negation followed by a qualitative change and combines the two so "doesn't change" is transformed into "neg + change" in the parameter extraction step and then into "nochange" in the next stage. Another machine tries to recognize whether a relationship between two parameters is being described as direct or inverse. Sometimes students use "*D*" and "+" to indicate that a relationship is direct, sometimes these same symbols are used for "decrease" and "increase" qualitative changes respectively. The polysemy is resolved because the question at hand determines which transducer is applied to the student's utterance---the relationship or the qualitative change recognizer.

Table 2. Meaning attributes of a sample of words in the lexicon

MEAN	Meaning	MEAN	Meaning
ABV	arterial blood volume	F-S	Frank-Starling effect
ANS	autonomic nervous system	HEDGE	<hedge>
BP	blood pressure	HR	heart rate
BR	baroreceptor	INVPROP	inversely
BRP	baroreceptor pressure	IS	inotropic state
BRR	baroreceptor reflex	KNOW	know/understand
BRRATE	baroreceptor firing rate	NEG	negation of verb (not, isn't, etc.)
BRSIZE	baroreceptor size	NO	no
BV	blood volume	NOCHANGE	unchanged
CBV	central blood volume	NS	nervous system
CC	cardiac contractility	PIT	intrathoracic pressure
CHANGE	change	PHYSICAL	hemodynamic/non-neural
CNS	central nervous system	PNS	parasympathetic nervous system
CNSR	central nervous system response	PRELOAD	preload
CO	cardiac output	Ra	arteriolar resistance
CVP	central venous pressure	AP	right atrial pressure
DIRPROP	directly	REFLEX	reflex
DN	decreased	RR	reflex response
DR	direct response	RV	venous resistance
DUNNO	doesn't know	SNS	sympathetic nervous system
EDP	end diastolic pressure	SS	steady state
EDV	end diastolic volume	SV	stroke volume
ESV	end systolic volume	TPR	total peripheral resistance
FILLING	ventricular filling	UP	increased
FL	fiber length	VR	venous resistance

The topic of neural control of variables is the central issue in the problems that CIRCSIM-Tutor's students solve. It is important that the system recognize a fairly fine grained variety of student inputs here, since in human tutoring we observe a lot of attention to the details of neural control. The neural mechanism finite state machine recognizes answers to the question "By what mechanism is *X* controlled?" It looks for an optional parameter name plus anything that can be matched to the mechanism ontology shown in Figure 5. So it recognizes "sympathetics" and "TPR neural" and "TPR controlled by nervous system" as meaning that total peripheral resistance is a neural variable.

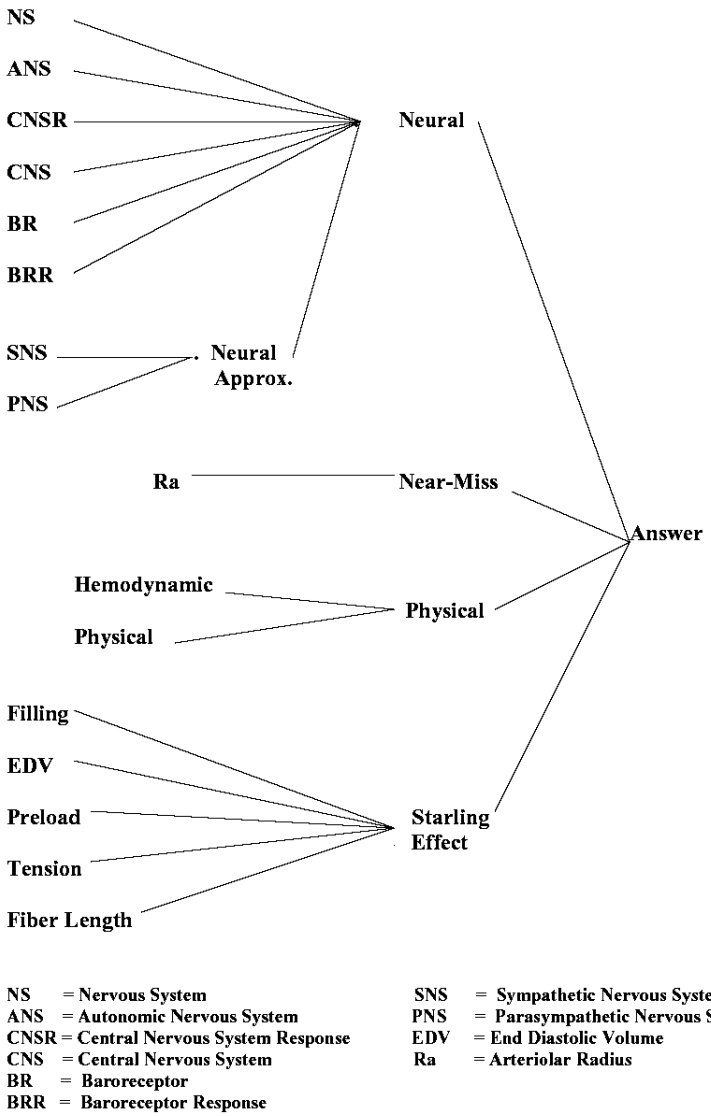


Figure 5. Ontology of mechanism answers.

Another purpose of the neural control ontology is to recognize student answers that are correct, and thus should not be contradicted, but nevertheless the human tutors respond to fine distinctions in the student's physiological language [32]. In this example the tutor subtly changes the student's answer "sympathetic vasoconstriction" to "neural":

K11-tu-49-3: How is TPR controlled?

K11-st-50-1: Sympathetic vasoconstriction->

K11-tu-51-1: Right. TPR is primarily under neural control.

Although the input understander's ontology is capable of identifying these linguistic near misses, the planner has no response for them at this time.

Producing the logic form and checking for errors

Unless a student initiative was recognized and reported back to the planner, the result of the input understanding process is typically a representation of an answer to the question most recently asked. For example the question represented by:

(QUESTION (AFFECTED-BY *MAP*))

causes the input understand to emit back to the planner a form:

(ANSWER (AFFECTED-BY <var> ((<varlist>)))),

where <var> and <varlist> contain information extracted from the student's answer. The output of the final transducer is assembled into the required logic form, using a different subroutine for each question.

Otherwise, if the input understander did not extract an answer to the question at hand, it attempts conversational repair with a message that explains what kind of input the system is expecting [30, 31]. The earlier version of CIRCSIM-Tutor responded to unrecognized input with "I am sorry. I did not understand you. Please rephrase". This response did nothing to repair the problem. In the current regime, dialogues such as the following are common:

T: Is the relationship from Stroke Volume to Cardiac Output direct or is it inverse?

S: HR and SV.

T: Didn't recognize directly or inversely related.

S: directl.

A complete list of these messages is given in Table 6 below along with the frequency of occurrence of each message in the trials with students.

Evaluation with a Class of Students

CIRCSIM-Tutor was evaluated by having a class of medical students use it. In this paper we evaluate the performance of the new input understanding component by illustrating the types of student input phenomena encountered and seeing how well it responded. We also briefly note learning gains, which are more fully evaluated elsewhere [30, 54].

Experimental setup

Michael and Rovick arranged to test the system in a regularly scheduled laboratory in their first-year physiology class at Rush Medical College in November, 1999. The students first took a paper pre-test for thirty minutes. Then they worked with CIRCSIM-Tutor for a full hour. Afterwards, they took a post-test and filled out a questionnaire about their reactions to the experience, also on paper. To validate our two test instruments used in pre- and post-testing, half the students used test A as a pre-test and half used test B as pre-test, with the tests switched for post-testing. Of 42 students who participated 14 people worked together in pairs on the machines but took the pre- and post-test individually, resulting in 35 computer sessions and 42 pre-tests and post-tests.

Evidence of learning gain

In the majority of the sessions (21 out of the 35) all 8 procedures supported by CIRCSIM-Tutor were completed. All but 4 sessions involve at least 5 completed procedures. The distribution is shown in Table 3.

Table 3. Number of procedures completed out of eight available in CIRCSIM-Tutor

No. Procedures Completed	No. Students or Pairs of Stu.
1	0
2	0
3	2
4	2
5	4
6	2
7	4
8	21

Comparison of the pre-test and post-test results in Table 4 shows that the students indeed learned some physiology from their experience with CIRCSIM-Tutor. The first sections of the pre-test and post-test ask the student to recall all twelve physiological relationships and whether they are directly or inversely related, yielding a top score of 24. The second sections of each test asked for 21 predictions (for 7 variables at 3 different stages) as a result of a perturbation. This part of the test is closest to the problem-solving task in the tutoring sessions. The maximum number of correct predictions is 21.

In the entire group of 42 students, there is a significant improvement from a mean score of 13.52 to 16.33 on the relation task (Section 1), with a one-tailed t-test result of $p = 0.024$. The group shows an improvement from a mean score of 13.40 to 17.00 on the prediction task (Section 2), with a one-tailed t-test result $p = 0.000036$.

To check that pre- and post-testing were comparable, students who used test A as a pre-test are shown above the line Table 4 and those who used the same test as a post-test are shown below. The pre-A, post-B group showed significant improvement in both the relation and prediction tasks, with means increasing from 13.52 to 17.4 ($p = 0.02$) and 13.17 to 17.29 ($p = 0.004$), respectively. The pre-B, post-A groups showed significant improvement on the prediction task with mean rising from 13.56 to 16.8 ($p = 0.002$), but improvement on the relation task was not significant with mean rising from 13.52 to 15.6 ($p = 0.15$). Despite the possibility that the two tests were not perfectly matched, we can conclude that CIRCSIM-Tutor is an effective learning tool.

These students learned more than a control group of students who read a carefully chosen chapter of text focusing on the same physiology and problem-solving tasks. The pre-test, the post-test, the results of the questionnaire, the control group and the learning gains achieved by these students are described in detail in [54].

Table 4. Pre-test and post-test scores for individual students using CIRCSIM-Tutor (T-tests show that the improvement on Section 1 (DiffRel) and the improvement on Section 2 (DiffCorr) are both significant at the .02 level. The two groups of students above and below the line had exactly the same experience except that the pre-test and post-test were switched.)

RelPtsBefore	CorrBefore	RelPtsAfter	Corrafter	DiffRel	DiffCorr
8	16	14	17	6	1
9	4	16	14	9	10
2	9	12	13	10	4
12	13	14	14	2	1
6	16	11	18	5	2
16	1	21	14	5	13
5	12	15	18	10	6
10	10	14	18	4	8
18	16	22	18	4	2
19	16	22	18	3	2
22	16	20	18	-2	2
8	15	14	18	6	3
18	16	16	18	-2	2
24	20	24	20	0	0
18	19	18	20	0	1
22	19	24	20	2	1
13	6	17	18	4	12
16	6	13	18	-3	12
2	4	2	18	0	14
8	13	6	18	-2	5
13	19	7	17	-6	-2
13	14	19	18	6	4
4	8	13	18	9	10
12	13	20	19	8	6
15	7	14	14	-1	7
8	11	0	16	-8	5
14	14	22	15	8	1
22	14	22	18	0	4
14	17	22	11	8	-6
21	19	24	20	3	1
18	11	20	12	2	1
24	18	24	19	0	1
16	18	18	19	2	1
18	18	20	18	2	0
4	14	4	14	0	1
18	15	18	17	0	2
1	15	22	18	21	3
15	16	17	8	2	-8
21	10	14	18	-7	8
5	19	5	20	0	1
20	18	24	18	4	9
16	8	20	19	4	11
Mean 13.52	Mean 13.40	Mean 16.33	Mean 17.00	Mean 2.80	Mean 3.59

Student input statistics

The 35 sessions contain 1642 student dialogue turns. A session-by-session summary is shown in Table 5. The number of student dialogue

turns per session ranges from 9 to 94. Examining these results we see that the students who made few errors in their predictions were asked very few questions. This is different from human tutoring, as in our transcripts of human tutoring the tutors probe for common misconceptions when students produce perfectly correct predictions [30, 44]. Later versions of CIRCSIM-Tutor included questions for use in these circumstances.

Table 5. Input understander results session by session

#PROCEDURES	#STUDENT TURNS	#ERROR MESSAGES	#SPELLING ERRORS	#SPELLING CORRECTIONS	#NON-ALPHABETIC TURNS
8	33	1	5	5	0
8	31	0	0	0	12
3	48	6	1	1	0
7	37	4	1	0	12
8	9	0	0	0	4
8	19	0	0	0	9
8	48	0	0	0	2
8	19	3	7	7	0
4	56	3	1	1	28
8	21	0	8	8	0
8	82	4	3	3	39
8	9	1	0	0	0
8	52	1	3	3	20
8	29	2	13	12	1
8	54	1	1	1	21
8	79	3	0	0	36
8	49	1	1	1	18
8	59	0	6	6	0
8	38	3	1	1	12
7	75	7	2	2	28
7	69	4	4	4	25
8	94	5	2	2	34
6	45	5	0	0	17
8	79	5	8	6	0
5	21	1	1	1	0
5	55	6	0	0	26
3	32	2	10	9	0
5	69	4	0	0	30
8	74	2	3	3	1
8	38	3	1	1	17
4	16	1	0	0	0
5	37	3	0	0	1
8	62	3	1	1	27
6	59	2	0	0	27
7	45	0	13	13	2
Totals 242	1642	86	96	91	449

Non-alphabetic turns

The 1642 student turns include 449 that were not alphabetic. The majority of these were plus signs, minus signs, or zeros in response to questions about values of variables. One student typed the letter "o", which the spelling correction module correctly mapped into a "0" (zero). There were a few student utterances consisting of question marks, for which the input understander issued a message explaining the kind of input it wanted to see. These messages are shown in Table 6. There are also three other instances of apparently random punctuation marks, to which the system responded by issuing a new prompt. In all three, the student then came up with the right answer, so we judged that simply issuing a new prompt was an appropriate response. However we are considering making the system produce an error message specifying the desired input in these cases also.

Spelling correction

The spelling correction module made 96 attempts at spelling correction and succeeded in all but five. Two of the failures were errors in correcting phrases and in two of the remaining ones the system corrected "in" to "increase" when "inverse" would have been preferable. In one case the system failed to correct "cop", an unusual abbreviation for cardiac output, to the standard abbreviation "co". This omission has been corrected.

Most of the successful corrections involved handling of abbreviations. For example, the system corrected "cvb" to "cbv" for central blood volume and "vp" to "cvp" for central venous pressure. It restored the initial "n" to "ormal" and "eural". It successfully corrected the phrases: "stroke voulme", "cardiac ouptut", "mean adreterial pressure", and "total periferal resistanse". It correctly changed "ionotropic state" to "inotropic state" and "contractibility" to "contractility".

The following example shows one of two instances where a misspelled phrase was mishandled by spelling correction. The system corrected "venour" to "veins", found the phrase "central veins", did not even look for "central venous pressure", and eventually extracted the word "pressure" as an answer to the question.

T: Stroke Volume is determined by Central Venous Pressure and Inotropic State. Which determinant is dominant in this case?

S: central venour pressure (*the system picks up the word pressure and decides that this means MAP*)

T: Nope, the important determinant of Stroke Volume in this case is not Mean Arterial Pressure. The important determinant of Stroke Volume in this case is Central Venous Pressure.

These errors were caused by the interaction between our method for processing phrases and our information extraction approach.

Missing vocabulary

In examining these sessions we found two additions to the lexicon that could have improved system understanding of student input. In one case the system asked the student to name a stage and the student typed "initial" instead of the expected "DR" or "Direct Response". Since the DR stage is indeed the initial one, we decided to add "initial" to the lexicon as a synonym for "DR". The other addition involves the above-mentioned polysemous abbreviation "in." for "inverse" and "increase", depending on context.

Error messages specifying desired input

The 35 one-hour sessions contained 86 error messages, and four-fifths of the sessions contained at least one of these messages, so it is clear that they impact on users. We believe that they are responsible for increased user satisfaction with the system, since we saw very few of the unrepaired repeated misunderstanding loops that we often observed in use of the previous input understander. These messages and their frequencies observed during this trial are listed in Table 6.

Table 6. Messages Explaining the Type of Answer Expected
(number given is actual frequency among 86 messages issued in 35 sessions in November, 1999)

Please respond with prediction table parameters.	28
Is the mechanism of control neural or physical?	17
Didn't recognize a stage: DR, RR or SS.	14
It should be increased, decreased, or unchanged.	13
Did not recognize directly or inversely related.	13
This is not the variable under discussion, please try again	1
You can't have it both ways.	0
(Used when two alternative answers are detected in one response)	

Most of the time one of these messages was enough to get the student back on track. There were seven sequences of two or more of these turns, the longest sequence of this kind had four error messages. These messages are issued by the input understander, which does not inform the instructional planner that the student is in trouble. The system has since been changed so that after two error messages the student receives the correct answer.

Problem dialogues and student frustration

There were only a handful of dialogue segments that exhibited serious problems. We exhibit a few of them here.

Bozena Thompson, who wrote one of the first interactive interfaces to database systems, discovered almost 30 years ago [66] that any interactive system that accepts natural language input must expect remarks from frustrated users. At this point such remarks from the student are not understood by the CIRCSIM-Tutor. In November, 1999, the system received only three such inputs which could be interpreted this way, and they were pretty mild. Two were strings of random letters, they could have been either expressions of frustration or accidents with the keyboard. The last is illustrated in this exchange:

T: What is the other determinant of Mean Arterial Pressure?

S: Reflex

T: Please respond with prediction table parameters

S: -

T: Please respond with prediction table parameters

S: clueless

T: Please respond with prediction table parameters

S: heart rate

It is possible that "clueless" was an insult directed at the system, or it may have been intended as "I don't know" student initiative that the system should have recognized. In trials with fifty students of an earlier version of CIRCSIM-Tutor in Fall, 1998 we had some clearly negative comments not fit to print. We did not see any in this trial. We would like to believe that improvements in the system averted these extremes of student frustration. It is also possible that the students were inhibited by the fact that we had team members patrolling the room to help in case of system crashes (there were none) and to make sure that no one left without taking a post-test and filling out a questionnaire. Since we do not have a cogent plan for responding to expressions of frustration we have not put any effort into recognizing them.

In this next example the student seems to be doggedly answering some other question rather than paying attention to the error messages.

T: What stage must the value of Cardiac Output follow in SS?

S: no change

T: Didn't recognize a stage: DR, RR or SS.

S: decrease

T: Didn't recognize a stage: DR, RR or SS.

S: 0

T: Didn't recognize a stage: DR, RR or SS.

S: no change in ss

T: Nope, the value of Cardiac Output doesn't follow the value of Cardiac Output in SS.

We have fixed this problem by giving the student the correct answer when the second failed attempt is not of the proper category, since a second category error indicates that the student did not understand the error message. This shows that the input understander should perhaps alert the planner when an attempt at repair fails.

Future Work

There are several areas where improvements in the capabilities of the input understander seem particularly germane to the future of CIRCSIM-Tutor. As noted, there are still problems in the spelling correction phase of the system. When the misspelled word is in the context of a phrase, the system sometimes fails. It seems also necessary to use the dialogue context to weight candidate words, as is the case where "in" could mean "increase" or "inverse", are still not using phrases and context effectively in spelling correction spelling correction is triggered only when a word in a string bounded by white space and punctuation does not appear in the lexicon. However this is inconsistent with correcting errors obtained by garbling one word into another, the error that misspells "form" as "from". Furthermore in the human tutoring transcripts we see conjoined terms such as "svi" for "sv i", a phenomenon that is inconsistent with our algorithm of finding the closest single word in the lexicon.

Prepositional phrases

From our study of the human tutoring sessions we can see that possible new tutoring plans will require the students to utter physiological concepts that commonly are expressed as variety of prepositional phrases. We already observed several such instances in November 1999 experiment. For example, the phrase "volume of blood" is sometimes used to denote the volume of blood in the central venous compartment. Our first thought was to include it in the lexicon as a synonym for the parameter "central blood volume", or CBV as it is usually abbreviated in the tutoring sessions. The problem is two-fold: there is a certain amount of linguistic creativity in describing parameters, and some phrases can be used in other contexts to denote different parameters. For example, "volume of blood in the ventricle" denotes ventricular filling and "volume of blood ejected by the heart" is a roundabout way of describing "stroke volume", so it would be infelicitous

to recognize “volume of blood” as a synonym for CBV. Similarly, the word “pressure” is also used in the range of contexts of the form

“pressure” + [“of blood”] + <preposition> + <anatomical structure>

Generally “pressure” when unqualified denotes mean arterial pressure but “pressure in the veins” means central venous pressure. We can add finite state machines to handle these phrases by looking for words such as “volume”, “pressure”, “blood”, and “length” combined with prepositions such as “of ” and “in” followed by noun phrases. We would also need a small ontology of anatomical terms.

Equations in the input

Examination of student utterances in human tutoring sessions shows many student answers containing algebraic expressions, even when the tutor did not explicitly ask for an equation. So a question asking for the determinants of cardiac output may be answered with an equation:

K40-tu-108-6: Can you tell me what determines the value of CO?

K40-st-109-1: CO=SVxHR.

These equations are invoked for their descriptive power; students are not solving problems quantitatively. Glass [29] catalogs these answers, discusses the possibility of adding a special purpose expression grammar, and discards that plan in favor of a grammar combining algebra and English because there are so many examples in which algebra and English are combined into one sentence, as in:

K7-st-100-1: But isn’t CO X TPR =MAP ?

K13-st-52-1: Sv times hr =co.

Understanding student input involving equations and making appropriate responses is known to be a tricky problem [28, 52].

We believe that the reason we see such language in human tutoring transcripts but not often in student use of CIRCSIM-Tutor is that unlike the human tutors, the computer does not invoke equations in its own language. Human tutors spontaneously produce combinations like this one:

K14-tu-49-2: It is true that co=sv x hr.

We have added to CIRCSIM-Tutor hints like this one shown in Figure 2:

T: MAP = TPR x what?

Because of its power and ubiquity in the human tutoring transcripts, we believe it will be useful to incorporate equation language into CIRCSIM-Tutor's dialogues as well.

Answering "Why" questions

Another concern is that the system does not ask as many deep questions such as "why" questions as we would like, because it cannot understand the answers well enough to respond to them. In order to collect more student answers to "why" questions and to discover whether the students benefit from trying to answer these questions even without much feedback, we added at least one "why" question to each procedure during the next set of experiments and then provided the student with a good answer based on our most recent set of human tutoring sessions. This change also addresses the fact that the computer tutor does not elicit any student language when the student correctly solves the problem with no errors. We also observe that in sessions with human tutors the students periodically ask substantive questions. Even more often they state a mini-theory and ask the tutor to comment [65]. We would like to try to provide this experience for the students. ITS architectures such as AutoTutor [35] have been evolved toward a two-tiered approach, one for short (several word) student utterances and another using technologies such as robust parsing or latent semantic analysis for longer student utterances.

Conclusion

The CIRCSIM-Tutor dialogue-based intelligent tutoring system incorporates an approach toward processing student utterances that is designed to be simple and robust, using finite state transducers to extract the answer to the tutor's question from the student's utterance. The tutoring system with this input understanding component was successfully tested with 35 tutoring sessions using first year medical students at Rush Medical College in a regular physiology laboratory.

The input understanding approach proved to be robust and successful at recognizing student utterances and responding appropriately. In addition, the users of this system exhibited significant improvements in learning outcomes. A detailed study of the educational results is presented elsewhere [54].

Table 7 summarizes the 35 sessions and the performance results described earlier. There were 1642 student dialogue turns in the 35 sessions, with the number of student turns ranging from 9 to 94 with a mean of 46.91. Most of these 1642 turns were very short. In fact, 449 of these turns consisted of only a single non-alphabetic symbol, most often a plus or a minus sign. The system corrected the student spelling appropriately in 91 cases, an average of 2.6 spelling errors per session, and failed with five words.

Table 7. Summary of input understander results

	Number	Average Number per Session	Percentage of Inputs
Hour-Long Sessions	35		
Student Turns (vary from 9-94 per session)	1642	46.9	100%
Non-Alphabetic Turns	449	12.8	27.9%
Error Messages (explaining expectations)	86	5.3	2.5%
Missing Vocabulary	2	0.06	0.12%
Valid Spelling Corrections	91	2.6	5.5%
Erroneous Corrections	5	0.14	0.31%

The input understanding component made six content errors in these 1642 turns (less than 0.4%); three of those led to serious misunderstanding. In the other three the student figured out what was going on and came up with an appropriate response. All of these errors involved erroneous spelling corrections and missing vocabulary items. There were 86 inputs (5.3%) where the system decided it could not interpret the student input correctly and issued an error message explaining what kind of input it was looking for. In a few of these situations an expert human tutor could have made productive use of the student answer, given a more appropriate response, or changed to another tutorial strategy.

In summary, a simple, robust, finite-state, information extraction approach to processing student input performed well in this experiment, effectively detecting what the student was trying to tell us or else providing the appropriate error message over 99% of the time.

References

[1] V. Aleven and K. Koedinger, The need for natural language dialogue to support explanation, C. P. Rosé and R. Freedman, eds., Proceedings of the AAAI Fall Symposium on Building Dialogue Systems for Tutorial Applications, AAAI Press, Menlo Park, CA, 2000, pp. 65-73.

[2] V. Aleven, K. Koedinger and K. Cross, Tutoring answer explanation fosters learning with understanding, Proceedings of Artificial Intelligence in Education, AI-ED'99, IOS Press, Amsterdam, 1999, pp. 199-206

[3] V. Aleven, O. Popescu and K. Koedinger, Towards tutorial dialogue to support self-explanation: Adding natural language understanding to a cognitive tutor, J. D. Moore, C. L. Redfield and W. L. Johnson, eds., Proceedings of Artificial Intelligence in Education, AI-ED2001, IOS Press, Amsterdam, 2001, pp. 246-255.

[4] K. S. Bhatt, M. W. Evens and S. Argamon, Hedged responses and expressions of affect in human/human and human/computer tutorial interactions, Proceedings of the Cognitive Science Society, COGSCI2004, Lawrence Erlbaum Associates, Mahwah, NJ, 2004, pp. 114-119.

[5] B. Bloom, The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring, Educational Researcher 136 (1984), 4-16.

[6] S. S. Brandle, Using joint actions to explain acknowledgments in tutorial discourse: application to intelligent tutoring systems, Unpublished Doctoral Dissertation, Illinois Institute of Technology, Chicago, 1998.

[7] J. S. Brown and R. R. Burton, Toward a natural language capability for computer-assisted instruction, D. G. Bobrow and A. Collins, eds., Representation and Understanding: Studies in Cognitive Science, Academic Press, New York, 1975, pp. 311-349.

[8] R. R. Burton and J. S. Brown, Toward a natural language capability for computer-assisted instruction, H. O'Neill, ed., Procedures for Instructional Systems Development, Academic Press, New York, 1979, pp. 272-313.

[9] J. R. Carbonell, AI in CAI: An artificial intelligence approach to computer-aided instruction, IEEE Transactions on Man-Machine Systems 11(4) (1970), 190-202.

[10] A. Cawsey, Explanation and Interaction: The Computer Generation of Explanatory Dialogues, MIT Press, Cambridge, MA, 1992.

[11] R. C. Chang, Surface level generation of tutorial dialogue using a specially developed lexical functional grammar and lexicon, Unpublished Doctoral Dissertation, Illinois Institute of Technology, Chicago, IL, 1992.

[12] R. C. Chang, M. W. Evens, J. A. Michael and A. A. Rovick, Surface generation in tutorial dialogues based on a sublanguage study, Y. F. Chang, ed., International Conference on Advanced Science and Technology, ICAST'94, CAPAMA, Chicago, 1994, pp. 113-119.

[13] M. Chi, N. de Leeuw, M. H. Chiu and C. LaVancher, Eliciting self-explanations improves understanding, Cognitive Science 18 (1994), 439-477.

[14] P. Cohen, J. Kulik and C. Kulik, Educational outcomes of tutoring, American Educational Research Journal 19 (1982), 237-248.

[15] A. Collins and A. L. Stevens, Goals and strategies of inquiry teachers, R. Glaser, ed., Advances in Instructional Psychology, 2, Lawrence Erlbaum Associates, Hillsdale, NJ, 1982, pp. 65-119.

[16] A. Collins, J. S. Brown and S. E. Newman, Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics, L. B. Resnick, ed., Knowing, Learning, and Instruction, Lawrence Erlbaum Associates, Hillsdale, NJ, 1989, pp. 453-494.

[17] M. G. Core and J. D. Moore, Robustness vs. fidelity in natural language understanding, Proceedings of the Workshop on Scalable Natural Language Understanding, HLT-NAACL 2004, ACL, Stroudsburg, PA, 2004, pp. 1-8.

[18] B. Di Eugenio, D. Fossati, D. Yu, S. Haller and M. S. Glass, Natural language generation for intelligent tutoring systems: A case study, Proceedings of AIED 2005, the 12th International Conference on Artificial Intelligence in Education, Amsterdam, 2005, pp. 217-224.

[19] B. Di Eugenio, D. Fossati, D. Yu, S. Haller and M. S. Glass, Aggregation improves learning: Experiments in natural language generation for intelligent tutoring systems, Proceedings of the Association for Computational Linguistics, ACL, Stroudsburg, PA, 2005, pp. 50-57.

[20] M. Elmi and M. W. Evens, Spelling correction using context, Proceedings of COLING 98, ACL, Stroudsburg, PA, 1998, pp. 360-364.

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

[22] J. E. Fox Tree, Listening in on monologues and dialogues, Discourse Processes, 27(1) (1999), 35-53.

[23] R. Freedman, Plan-based dialogue management in a physics tutor, Proceedings of the Sixth Applied Natural Language Processing Conference ANLP 2000, ACL, Stroudsburg, PA, 2000, pp. 52-59.

[24] R. Freedman, Using a reactive planner as the basis for a dialogue agent, Proceedings of the Florida Artificial Intelligence Research Symposium, FLAIRS 2000, AAAI Press, Menlo Park, CA, 2000, pp. 203-208.

[25] R. Freedman, An approach to increasing programming efficiency in plan-based dialogue systems, J. D. Moore, C. L. Redfield and W. L. Johnson, eds., Proceedings of Artificial Intelligence in Education, AI-ED2001, IOS Press, Amsterdam, 2001, pp. 200-209.

[26] R. Freedman and M. W. Evens, Generating and revising hierarchical multi-turn text plans in an ITS, C. Frasson, G. Gauthier and A. Lesgold, eds., Proceedings of Third International Conference on Intelligent Tutoring Systems, ITS'96, Springer-Verlag, Berlin, 1996, pp. 632-640.

[27] R. Freedman, C. P. Rosé, M. Ringenberg and K. VanLehn, ITS tools for natural language dialogue: A domain-independent parser and planner, G. Gauthier, C. Frasson and K. VanLehn, eds., Proceedings of Fifth International Conference on Intelligent Tutoring Systems, ITS2000, Springer-Verlag, Berlin, 2000, pp. 433-442.

[28] A. Gertner, Providing feedback to equation entries in an intelligent tutoring system for physics, B. Goettl, H. Halff, C. Redfield and V. Shue, eds., Proceedings of Fourth International Conference on Intelligent Tutoring Systems, ITS'98, Springer-Verlag, Berlin, 1998, pp. 254-263.

[29] M. S. Glass, Some phenomena handled by the CIRCSIM-Tutor Version 3 input understander, Proceedings of the Tenth International Florida Artificial Intelligence Research Symposium FLAIRS-97, AAAI Press, Menlo Park, CA, 1997, pp. 21-25.

[30] M. S. Glass, Broadening input understanding in a language-based intelligent tutoring system, Unpublished Ph.D. Dissertation, Computer Science Department, Illinois Institute of Technology, Chicago, IL, 1999.

[31] M. S. Glass, Processing language input in the CIRCSIM-Tutor intelligent tutoring system, C. P. Rosé and R. Freedman, eds., Proceedings of the AAAI Fall Symposium on Building Dialogue Systems for Tutorial Applications, AAAI Press, Menlo Park CA, 2000, pp. 74-79.

[32] M. S. Glass, Processing language input for an intelligent tutoring system, J. D. Moore, C. L. Redfield and W. L. Johnson, eds., Proceedings of Artificial Intelligence in Education, AI-ED 2001, IOS Press, Amsterdam, 2001, pp. 210-221.

[33] M. S. Glass, J. H. Kim, M. W. Evens, J. A. Michael and A. A. Rovick, Novice vs. expert tutors: A comparison of style, Proceedings of the Midwest Artificial Intelligence and Cognitive Science Conference, MAICS 99, AAAI Press, Menlo Park, CA, 1999, pp. 43-49.

[34] A. C. Graesser, Dialogue patterns and feedback mechanisms during naturalistic tutoring, Proceedings of the Fifteenth Annual Meeting of the Cognitive Science Society, Lawrence Erlbaum Associates, Hillsdale, NJ, 1993, pp. 127-130.

[35] A. C. Graesser, A. Olney, B. C. Haynes and P. Chipman, AutoTutor: a cognitive system that simulates a tutor that facilitates learning through mixed-initiative dialogue, C. Forsythe, M. L. Bernard and T. E. Goldsmith, eds., Cognitive Systems: Human Cognitive Models in Systems Design, Lawrence Erlbaum Associates, Mahway, NJ, 2005.

[36] A. C. Graesser, N. K. Person and J. Huber, Question asking during tutoring and in the design of educational software, M. Rabinowitz, ed., Cognitive Science Foundations of Instruction, Lawrence Erlbaum Associates, Hillsdale, NJ, 1993, pp. 149-172.

[37] A. C. Graesser, N. K. Person and J. P. Magliano, Collaborative dialogue patterns in naturalistic one-on-one tutoring, Applied Cognitive Psychology 9 (1995), 495-522.

[38] N. Heffernan, T. E. Turner, A. L. N. Lourenco, M. A. Macasek, G. Nuzzo-Jones and K. Koedinger, The ASSISTment builder: Towards an analysis of cost effectiveness of ITS creation, Proceedings of the 19th International FLAIRS Conference, AAAI Press, Menlo Park, CA, 2006.

[39] J. R. Hobbs, D. Appelt, J. Bear, D. Israel, M. Kameyama, M. Stickel and M. Tyson, FASTUS: A cascaded finite-state transducer for extracting information from natural-language text, E. Roche and Y. Schabes, eds., Finite-State Language Processing, MIT Press, Cambridge, MA, 1997, pp. 383-406.

[40] J. E. Hopcroft and J. Ullman, Introduction to Automata Theory, Languages, and Computation, Addison-Wesley, Reading, MA, 1979.

[41] P. Jordan, C. P. Rosé and K. VanLehn, Tools for authoring tutorial dialogue knowledge, J. D. Moore, C. L. Redfield and W. L. Johnson, eds., Proceedings of Artificial Intelligence in Education, AI-ED2001, IOS Press, Amsterdam 2001, pp. 222-233.

[42] J. H. Kim, R. Freedman and M. W. Evens, Responding to unexpected student utterances in CIRCSIM-Tutor v.3: Analysis of transcripts, Proceedings of the Florida Artificial Intelligence Research Symposium, FLAIRS'98, AAAI Press, Menlo Park, CA, 1998, pp. 153-157.

[43] J. H. Kim, M. S. Glass, R. Freedman and M. W. Evens, Learning the use of discourse markers in tutorial dialogue for an intelligent tutoring system, L. R. Gleitman and A. K. Joshi, eds., Proceedings of Cognitive Science 2000, Lawrence Erlbaum Associates, Mahwah, NJ, 2000, pp. 262-267.

[44] J. H. Kim, R. Freedman, M. S. Glass and M. W. Evens, Annotation of tutorial goals for natural language generation, Discourse Processes 42(1) (2006), 37-74.

[45] T. K. P. Landauer, P. W. Folz and D. Laham, An introduction to latent semantic analysis, Discourse Processes 25 (1998), 259-284.

[46] H. C. Lane and K. VanLehn, Teaching program planning skills to novices with natural language tutoring, Computer Science Education 15(3) (2005), 183-201.

[47] A. Lavie, GLR*: A grammar based robust parser for spontaneous speech, Unpublished Ph.D. Dissertation, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1996.

[48] Y. H. Lee and M. W. Evens, Natural language interface for an expert system, Expert Systems: International Journal of Knowledge Engineering 15(4) (1998), 233-239.

[49] W. G. Lehnert, Information extraction: What have we learned? Discourse Processes 25(3) (1997), 441-470.

[50] J. Li, A. A. Rovick and J. A. Michael, ABASE - a computer program that teaches physiological acid base regulation, I. Tomek, ed., Computer Assisted Learning: Proceedings of the 4th International Conference on Computer-Assisted Learning, ICCAL'92, Springer-Verlag, Berlin, 1992, pp. 380-390.

[51] D. J. Litman, C. Rosé, K. Forbes-Riley, K. VanLehn, D. Bhembe and S. Silliman, Spoken versus typed human and computer dialogue tutoring, J. C. Lester, R. M. Vicari and F. Paraguaçu, eds., Proceedings of Intelligent Tutoring Systems, ITS 2004, Springer-Verlag, Berlin, 2004, pp. 368-379.

[52] M. A. Mark, K. R. Koedinger and W. S. Hadley, Elaborating models of algebraic expression-writing, B. Goettl, H. Halff, C. Redfield and V. Shute, eds., Proceedings of ITS98, Springer-Verlag, Berlin, 1998, pp. 524-533.

[53] J. A. Michael and A. A. Rovick, The learning effect produced by expert tutoring, Presented at Experimental Biology 95, 1995.

[54] J. A. Michael, A. A. Rovick, M. S. Glass, Y. Zhou and M. W. Evens, Learning from a computer tutor with natural language capabilities, Interactive Learning Environments 11(3) (2003), 233-262.

[55] J. D. Moore, Participating in Explanatory Dialogues, MIT Press, Cambridge, MA, 1995.

[56] H. Pon-Barry, B. Z. Clark, K. Schultz, E. O. Bratt and S. Peters, Evaluating the effectiveness of SCoT, a spoken conversational tutor, N. Heffernan and P. Wiemer-Hastings, eds., Proceedings of the Workshop on Dialogue-Based Intelligent Tutoring Systems, ITS 2004, Jhongli, Taiwan, 2004, pp. 23-32.

[57] H. Pon-Barry, K. Schultz, E. O. Bratt, B. Clark and S. Peters, Responding to student uncertainty in spoken tutorial dialogue systems, International Journal of Artificial Intelligence in Education 16 (2006), 171-194.

[58] E. Roche and Y. Schabes, Finite-State Language Processing, MIT Press, Cambridge, MA, 1997.

[59] C. P. Rosé, Facilitating the rapid development of language understanding interfaces for tutoring systems, C. P. Rosé and R. Freedman, eds., Proceedings of the AAAI Fall Symposium on Building Dialogue Systems for Tutorial Applications, AAAI Press, Menlo Park, CA, 2000, pp. 87-95.

[60] C. P. Rosé, P. Jordan, M. Ringenberg, S. Siler, K. VanLehn and A. Weinstein, Interactive conceptual tutoring in Atlas-Andes, J. D. Moore, C. L. Redfield and W. L. Johnson, eds., Proceedings of Artificial Intelligence in Education, IOS Press, Amsterdam, 2001, pp. 256-266.

[61] A. A. Rovick and J. A. Michael, CIRCSIM: An IBM PC Computer Teaching Exercise on Blood Pressure Regulation, Paper Presented at the XXX IUPS Congress, Vancouver, BC, Canada, 1986.

[62] A. A. Rovick and J. A. Michael, GASP: A computer program for teaching the chemical control of ventilation, Advances in Physiology Education, 13, S98.

[63] A. A. Rovick and J. A. Michael, The prediction table: A tool for assessing students' knowledge, American Journal of Physiology, 2636, Pt. 3, (1992), S33-S36, Also available as Advances in Physiology Education, 81, S33-S36.

[64] G. Sanders, Generation of explanations and multi-turn discourse structures in tutorial dialogue based on transcript analysis, Unpublished Ph.D. Dissertation, Computer Science Department, Illinois Institute of Technology, 1995.

[65] F. Shah, M. Evens, J. A. Michael and A. A. Rovick, Classifying student initiatives and tutor responses in human keyboard-to-keyboard tutoring sessions, Discourse Processes 33(1) (2002), 23-52.

[66] B. H. Thompson, Linguistic Analysis of Natural Language Communication with Computers, COLING 80, ACL Press, Stroudsburg, PA, 1980, pp. 190-201.

[67] M. Tomita, An efficient context-free parsing algorithm, Computational Linguistics 13(1-2) (1987), 31-46.

[68] K. VanLehn, R. Freedman, P. Jordan, C. Murray, R. Osan, M. Ringenberg, C. P. Rosé, K. Schulze, R. Shelby, D. Treacy, A. Weinstein and M. Wintersgill, Fading and deepening: The next steps for Andes and other model-tracing tutors, G. Gauthier, C. Frasson and K. VanLehn, eds., Proceedings of ITS 2000, Springer-Verlag, Berlin, 2000, pp. 474-483.

[69] K. VanLehn, A. C. Graesser, G. T. Jackson, P. Jordan, A. Olney and C. P. Rosé, When are tutorial dialogues more effective than reading? Cognitive Science 35 (2007), 3-62.

[70] R. Wilensky, D. Chin, M. Luria, J. Martin, J. Mayfield and D. Wu, The Berkeley unix consultant project, Computational Linguistics 14(1) (1988), 35-84.

[71] C. W. Woo, M. W. Evens, R. Freedman, M. Glass, L. S. Shim, Y. Zhang, Y. Zhou and J. A. Michael, An intelligent tutoring system that generates a natural language dialogue using dynamic multi-level planning, Artificial Intelligence in Medicine 38(1) (2006), 25-46.

[72] F. J. Yang, J. H. Kim, M. S. Glass and M. W. Evens, Turn Planning in CIRCSIM-Tutor, Proceedings of the Florida Artificial Intelligence Research Symposium, FLAIRS 2000, AAAI Press, Menlo Park, CA, 2000, pp. 60-64.

[73] Y. Zhou, R. Freedman, M. Glass, J. A. Michael, A. A. Rovick and M. Evens, What should the tutor do when the student cannot answer a question? Proceedings of the 12th Florida Artificial Intelligence Symposium FLAIRS-99, AAAI Press, Menlo Park, CA, 1999, pp. 187-191.

[74] Y. Zhou, R. Freedman, M. Glass, J. A. Michael, A. A. Rovick and M. Evens, Delivering hints in a dialogue-based intelligent tutoring system, Proceedings of the American Association for Artificial Intelligence, AAAI99, AAAI Press, Menlo Park, CA, 1999, pp. 128-134.