Processing Language Input in the CIRCSIM-Tutor Intelligent Tutoring System

Michael Glass

Computer Science Department
Illinois Institute of Technology
Chicago, IL 60616
glass@steve.iit.edu

Abstract*

We have replaced the input understanding component of CIRCSIM-Tutor, an intelligent tutoring system that engages the student in Socratic dialogue. Students type free-text answers to the computer's questions. Even though the questions can be answered very simply, the variety of student responses prompted us to make the understanding component more robust. The new software also enables the tutor to recognize and classify a greater number of unexpected responses. In this paper we report on the motivation and design of the new software and our results in using it.

Introduction

CIRCSIM-Tutor (CST) is an intelligent tutoring system for teaching the baroreceptor reflex mechanism of blood pressure control to first-year medical students. What distinguishes CST from many other ITSs is its high reliance on natural language. Aside from a simple table that the student fills in, tutoring is accomplished exclusively by natural language dialogue. There are no diagrams, multiple-choice lists, or animated displays. The dialogue is under the tutor's control; the machine asks questions and the student answers with free text in imitation of the Socratic style of human tutoring.

Imitation of human Socratic tutoring was adopted by some of the earliest ITSs, for example SCHOLAR (Carbonell 1970), but they had limited capabilities to categorize student utterances beyond simply right/wrong, and limited ability to plan multi-turn dialogue acts. The resulting dialogue was not very human-like. More recent Socratic tutors, for example the AutoTutor computer literacy tutor (Graesser et al. 1998), incorporate extensive dialogue planning. With the ability to handle many tutorial contingencies comes the need for more fine-grained understanding the student utterances. AutoTutor employs Latent Semantic Analysis technology in novel ways toward this end (Wiemer-Hastings et al. 1999), most notably in an effort to categorize longer student utterances that are difficult to parse and understand.

Partly because of improvements in CIRCSIM-Tutor's ability to respond to more categories of unexpected student input (Zhou et al. 1999), and partly to be more robust in handling the vagaries of free-text input, the input understanding component of CIRCSIM-Tutor has been upgraded. In contradistinction to dialogues in domains such as computer literacy, the human tutorial dialogues that inform CIRCSIM-Tutor contain many instances of very short answers (Glass 1997), and the computer tutor is dependent on correctly categorizing such answers. This consideration drives the design of the input understander in the direction of robust extraction of simple concepts.

CIRCSIM-Tutor's Tutoring Dialogue

CIRCSIM-Tutor presents the student with a description of a perturbation that disturbs blood pressure in the human organism. For example, a patient might hemorrhage blood. The student is required to predict the direct effect on seven physiological variables, then predict how these variables will change. Variables include, for example, heart rate and central venous pressure. Predictions are qualitative: a variable increases, decreases, or does not change. Learning the operation and effects of the baroreceptor reflex, a control mechanism which regulates blood pressure, is the main point of the exercise. After each set of seven predictions the tutor engages the student in remedial dialogue with the goal of having the student correct the incorrect predictions. An example of a CIRCSIM-Tutor screen is shown in Figure 1.

In the dialogues, the tutor's questions always admit of short one- or two-word answers. The range of expected answers is limited, for example one or two of the seven variables in the prediction table. The students usually do not produce longer utterances than needed. The following is a sample of dialogue with the computer tutor, edited slightly for readability, showing a typical variety of short student answers:

T: What are the determinants of Stroke Volume?
S: inotropic state
T: Right. The other determinant represents preload. What is the other determinant of Stroke Volume?

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

S: cvp
T: Right, Stroke Volume is determined by Central Venous Pressure and Inotropic State.
T: Which determinant is dominant in this case?
S: cvp
T: Correct, the important determinant of Stroke Volume in this case is Central Venous Pressure. Is the relationship from Central Venous Pressure to Stroke Volume direct or is it inverse?
S: inverse
T: Sorry, [gives physiological explanation]. What is the correct value of Stroke Volume?
S: down

In this dialogue the tutor's question are closed, the expected answers are quite short, and indeed the student input is usually short. The fact that it is unconstrained input presents us with a number of challenges. Given that the range of acceptable student answers is small, it would be easier to have the student pick from a list of answers. Nevertheless we chose unconstrained free-text input for three reasons:

1. We want to rely on the student's recall memory instead of recognition memory if possible. Constraining the student's utterance, for example by presenting the student with alternative answers to pick from, gives away the answer to the student.
2. We have observed that experienced human tutors are more likely than inexperienced tutors to force the student to enunciate an answer, as opposed to enunciating it themselves (Glass 1999).
3. Human tutoring transcripts show that a number of unexpected simple answers stimulate useful tutoring. Free-text input gives the student opportunity to produce such unexpected answers.

Freedman (1997) makes the case that by virtue of free-text input CIRCSIM-Tutor can imitate the tutorial dialogue patterns of human tutors, even though the machine asks only short-answer questions while the human tutors often ask more open questions.

An implication of relying on recall memory is that even when the student responds with the correct expected concept, the student might express that concept in an unusual way. To take one concept that occurs frequently in the dialogues, there are a variety of ways that students express the notion that a variable is controlled by the nervous system. The students sometimes say "autonomics" or "neural system." Sometimes they utter a bare adverb "neurally" or a simple sentence "it is neurally controlled." Sometimes they refer to one part of the nervous system, as in "sympathetics." The students can be linguistically productive, even in a very small domain.

Aside from linguistic variation, there is considerable unexpected semantic variation that CIRCSIM-Tutor needs to understand and handle.

A common type of unexpected input is a "near miss," where the student's response is not what the tutor expected, but it is nevertheless close enough that the tutor can introduce extra dialogue steps to bring the student from the near-miss answer to the desired answer. In our dialogues, a near-miss answer often takes the form of the name of a variable that is causally related to one of the desired answers. The near-miss variable might be one that the student learned in class but is not part of the computer tutor's normal tutorial discussion.

Common misconceptions are also evident in free-text student answers to CIRCSIM-Tutor's questions. In human dialogues, the tutor often diagnoses these misconceptions based on extremely little data, only a word or two. The tutor seems not to engage in deep understanding of a student misconception, but rather relies on "stereotype" modeling of the student (Kay 2000).

Even when the student is simply wrong, there are a few hinting strategies that can be invoked in different contexts. This can mean that there are several varieties of wrong. In response to an answer that is wrong because it appears to be outside the domain of discourse the tutor can inform the student what kind of answer is expected. Otherwise, the tutor can deliver a physiological hint.

A complete classification of answers recognized by CIRCSIM-Tutor (Zhou et al. 1999) is as follows:

1. Correct
2. Partially correct answer, i.e. some part is correct and the rest is not.
3. Near miss answer, pedagogically useful but not the desired answer
4. "I don't know" answer
5. "Grain of truth" answer, incorrect but indicative of a partially correct understanding of the problem
6. Misconception, a common confusion or piece of false knowledge
7. Other incorrect answers
8. Mixed answers, a combination of answers from the other categories

It will be noted that all these varieties of unexpected student input are dependent on the student using recall memory and free-text input. They also provide opportunities for teaching, so it is important that CIRCSIM-Tutor handle them.

An additional important motivation for improving the computer tutor to recognize a wider variety of student answers is to reduce the number of times CIRCSIM-Tutor denies a true answer. When the student's answer is true in fact, but not what the tutor was looking for, the tutor should not respond with "wrong." However these kinds of answers are often misunderstood by the machine. Reducing these blunders was an important prerequisite for using CIRCSIM-Tutor with real classes of medical students.

Other Considerations for the Understander

In CIRCSIM-Tutor the input understander is responsible for matching the student's input to the question that the tutor just asked. If the student input cannot be recognized in this way, or seems to be a category error, the input understander responds with a useful error message

indicating the type of answer required. Further processing of the student answer, evaluating the correctness of the answer or classifying an unexpected answer, is performed by other modules.

The previous version of the input understander (Lee 1990; Seu 1992; Seu and Evens 1991) used an LFG parser to parse the input, then extracted from the f-structure the features that matched the question. There were three reasons we became dissatisfied with this approach:

1. Extragrammaticality was not handled very well. The grammar recognized a number of cases, and the code would insert some missing components (for example copula "be"), but it did not incorporate other common features to improve robustness such as skipping. As we used the input understander with more students and recognized a wider variety of semantic inputs, expanding the grammar to handle all the possible erroneous cases became a problem.
2. With the inclusion of a variety of unexpected student answers the range of possible f-structures expanded. Extracting the student answer from the f-structure is more involved when there is a larger variety of possible structures.
3. ·Since the desired answer can always be expressed in a few words, and students sometimes embedded those words in various sentences and constructions, it would be more robust to simply find and extract the key phrases.

It must be noted that there are often extra words and syntax in the student input that do not contribute to the answer, complicating both the parsing task and the interpretation of the resultant f-structures. An interesting example is when the student hedges an answer, indicating uncertainty. We see this frequently in transcripts of human tutoring, but it also occurs occasionally in the computer tutor sessions, as when one student typed "increase, evidently." This kind of impromptu transformation can profoundly change the syntax of the utterance. When "it increases" is expressed as "I think it increases" the main verb of the sentence changes from "increase" (the answer) to "think" (irrelevant). Examples such as this make the robust parsers that are now available (Rosé 2000) less attractive. There are aspects of student input that we would rather not parse, even though the technology might be available to parse them.

The fundamental idea in the new input understander is to be as permissive as possible. It extracts whatever is needed from the student's input and ignores the rest. Only a little parsing is done; mostly the understander searches for words and phrases that could be answers to the question that CIRCSIM-Tutor just asked. It is possible to "fool" the understander, viz.:

T: Which determinant is dominant in this case?
S: Anything except HR.
T: HR is correct.

In practice, we are not worried about such behavior. If the input understander manages to correctly interpret the utterances of a person who is not trying to fool the system, it is performing its job. In the more than 3600 student turns the understander has processed during normal class use with medical students, the software has never responded inappropriately because of this kind of misunderstanding.

Processing in the Upgraded Understander

Processing in the upgraded input understander described in this paper consists of the following steps in sequence: lexicon lookup, spelling correction, processing by finite state transducers, lookup in concept ontologies, and finally matching to the question. I will discuss these in turn.

Lexicon lookup largely retrieves a meaning token and a part of speech. The lexicon contains many of the standard abbreviations used by the students. Spelling correction is invoked when lexicon lookup fails. There are many spelling errors and impromptu abbreviations in the student utterances; we cannot ignore them. For example, among the 1860 student turns in one recent batch of tutoring sessions, we observed the word "increase" to have been variously abbreviated or misspelled as "i," "in," "inc," "incr," "increase," "increasw," and "ncrease." For this purpose, we have been using the spelling corrector written by Elmi (Elmi 1994; Elmi and Evens 1998). To a large extent, spelling correction is aided by the fact that the recognition step that follows uses only little syntax. If "increase" is spell-corrected to any form of "increase," regardless of part of speech, it will probably be recognized.

In the new input understander the recognition mechanism is a cascade of finite state transducers (Roche and Schabes 1997). This approach has been popular in information extraction applications because it is robust and can be designed to extract from the input sentence only the information which is being sought. Accordingly I coded a number of small machines for specific kinds of answers. One machine can recognize a variety of utterances about a neural mechanism, another recognizes utterances containing a parameter and a qualitative change, and so on.

The finite state transducers are often simple enough that they are just performing keyword lookup. This is appropriate, considering that often one- and two-word answers suffice. Nevertheless keyword lookup is not always sufficient, so the transducers do engage in a small amount of syntax recognition. For example there is a state machine to distinguish the copula "is" from its homograph, an abbreviation for the physiological parameter Inotropic State. Another transducer recognizes some simple prepositional phrases such as "volume of blood in the atrium"; the words "volume" and "blood" do not capture the concepts by themselves. Negation occurs in a few transducers, e.g., "not really changed" is converted to "unchanged."

The processing step where the student answer is compared to concept ontologies is motivated by the following kinds of cases.

One issue that arises while matching the answer to the question is that the meaning of an answer depends on the question that was asked. Here are the classifications of two

similar physiological parameters, each supplied as answers to two questions.

Answer	"How is CC controlled?"	"What determines CO?"
"CVP"	Misconception	Wrong answer
"SV"	Category error	Correct answer

The input understander tries to identify these category errors so the tutor can issue an appropriate message informing the student of the kind of answer that is expected. This determination comes *before* the determination of right and wrong answers; it is making a determination whether an answer is valid in the context of the question. The concept ontologies are specialized for particular questions, enabling the input understander to map a group of physiologically related answers to the same notion of "category error."

Another example where the ontology enhances matching the answer to the question arises because we have evidence that human tutors make small distinctions among the several different possible neural mechanism answers. If the student's utterance is sufficiently indirect, the tutor accepts it, but echoes back an answer using more desirable language. Freedman (1997) calls this a "linguistic near miss." Here is an example from a human tutoring session. The student's answer and the tutor's corrected answer in more desirable language are both underlined:

T: How is TPR controlled?
S: Sympathetic vasoconstriction.
T: Right, TPR is primarily under neural control. We're talking about what happens before there are any neural changes.

When the student responds with a more direct answer, e.g. "nervous system," the tutor does not echo back more desirable language. Although CIRCSIM-Tutor does not currently make a distinction between neural mechanism answers, the same ontological map which classifies answers as valid or category errors can be used to make fine classifications among various answers.

The classification issue is one of mapping the meaning tokens of the parsed input (which came from the lexicon) to a class of answer. This is partly accomplished by small ontological maps, one map for every question. In the map for the neural mechanism question, the sample cases we have discussed look like this:

- "Sympathetic vasoconstriction" is a kind of "approximate neural answer" is a kind of "neural answer" is a kind of "valid answer."
- CVP is a kind of "misconception" is a kind of "valid answer."
- SV does not appear, so it is a category error.

Student Input	Reason Not Recognized
"increaseed"	Spelling correction algorithm did not repair it.
"istpr"	Here "is" and "tpr" are joined, but joins are not currently recognized by spelling correction.
"neurological"	There was a defect in the lexicon entry for this word.
"central venous volume"	Concept was missing from the lexicon.
"kiss my ass"	Expressions of frustration are not explicitly recognized.
"in"	This is a too-drastic abbreviation for "inverse."
"help"	The word "help" was in the lexicon with no assigned meaning; it should have been mapped to an "I don't know" category of answer.
"metabolic factors"	This is a domain concept beyond CST's knowledge, so the words have no assigned meaning in the lexicon

Table 1. Representative Samples of Input Understander Failure in November 1998

The final step in the input understander is to match the result of the processing by transducers and maps to the question, producing a representation of the answer. This step is accomplished by *ad hoc* code for each type of question. This process could be more table-driven in the future. Much of the processing is devoted to error checking. As an example, one kind of error it checks for is an answer with logical inconsistencies, such as "it goes up and down," which is not a possible answer to any of our questions. On the other hand, "it goes up and increases" would not be flagged as erroneous.

Experience with the Input Understander

In 1998 CIRCSIM-Tutor was used with the revised understander with a class of first-year medical students in a physiology class at Rush Medical College. There were fifty students, twenty-four of whom used it in pairs while the rest used it individually, yielding thirty-eight sessions. We recorded a total of 1801 student dialogue turns, of which ten were garbled beyond human recognition. Of the remaining 1791 turns, only 19 were not handled properly by the input understander. In this accounting, the ten garbled unusable turns include any turns for which I could not divine an intention of the student. All were no more than few characters long, for example the number "93" and

the letter "h." If there was a recognizable answer plus a few garbled characters additionally, the turn was counted as usable, not garbled.

Of the usable turns, some were simply the plus, minus, and zero symbols used to indicate values of increased, decreased, and no change. The input understander recognized all of these. I included the occasions a student typed a single letter "o" instead of digit "0" in this category. Also there were three turns which consisted of a bare question mark. The question mark is recognized by the input understander, but it does nothing special with it, so an error message is emitted explaining what kind of answer is expected. I deemed this to be an appropriate response.

The remaining 1395 turns were almost entirely alphabetic characters. Of these, nineteen (1.4% of alphabetic turns, 1.1% of all usable turns) were not recognized. Table 1 shows examples of unrecognized input.

Conclusion

CIRCSIM-Tutor supports free-text natural language input in a Socratic intelligent tutoring system. It demands only short responses from the students, but we believe the very act of eliciting answers from the students is good tutoring practice. From these short free-text responses there are a number of interesting phenomena in the realm of unexpected student utterances. It is important to recognize these phenomena in the text understanding process, partly so the machine tutor doesn't deny perfectly true student answers, and partly because some of them provide teaching opportunities. Free-text input also makes demands on the robustness of the understander.

We have implemented a fairly robust input understanding mechanism in CIRCSIM-Tutor that works well on a physiology class of medical students.

Acknowledgments

The members of the CIRCSIM-Tutor group at the Illinois Institute of Technology all worked to improve the tutor to the quality where it could be used in the classroom on a non-experimental basis. Particular mention goes to Yujian Zhou, who was responsible for improving its response to unexpected student inputs, and Reva Freedman, who showed us how to think about all manner of issues.

References

Carbonell, Jaime R., 1970. AI in CAI: An Artificial Intelligence Approach to Computer Assisted Instruction, *IEEE Transactions on Man-Machine Systems*, vol. 11 no. 4, pp. 190–202.

Elmi, Mohammad Ali, 1994. A Natural Language Parser with Interleaved Spelling Correction Supporting Lexical Functional Grammar and Ill-Formed Input, Ph.D. diss., Department of Computer Science, Illinois Institute of Technology.

Elmi, Mohammad Ali and Martha W. Evens, 1998. Spelling Correction using Context. In *Proceedings of the 17th International Conference on Computational Linguistics (COLING–98), Montréal*, published together with *36th Annual Meeting of the Association for Computational Linguistics (ACL '98)*, pp. 360–364.

Freedman, Reva, 1997. Degrees of Mixed-Initiative Interaction in an Intelligent Tutoring System, AAAI 1997 Spring Symposium on Computational Models for Mixed-Initiative Interaction. Menlo Park, CA: AAAI Press.

Glass, Michael, 1997. Some Phenomena Handled by the CIRCSIM-Tutor Version 3 Input Understander. In *Proceedings of the Tenth Florida Artificial Intelligence Research Symposium (FLAIRS '97), Daytona Beach, FL*. Menlo Park, CA: AAAI Press.

Glass, Michael, 1999. Broadening Input Understanding in and Intelligent Tutoring System Ph.D. diss., Dept. of Computer Science, Illinois Institute of Technology.

Graesser, Art, Stan Franklin, Peter Wiemer-Hastings, and the Tutoring Research Group, 1998. Simulating Smooth Tutorial Dialogue with Pedagogical Value. In *Proceedings of the 11th International Florida Artificial Intelligence Research Symposium (FLAIRS '98), Sanibel Island, FL*. Menlo Park, CA: AAAI Press

Kay, Judy, 2000. Stereotypes, Student Models, and Scrutability. In Giles Gauthier, Claude Frasson, and Kurt VanLehn, eds., *Intelligent Tutoring Systems: 5th International Conference (ITS 2000), Montréal*, pp. 19–30. Berlin: Springer-Verlag.

Lee, Yoon-Hee. 1990. Handling Ill-Formed Natural Language Input for an Intelligent Tutoring System, Ph.D. diss., Illinois Institute of Technology.

Roche, Emmanuel and Yves Schabes, 1997. *Finite-State Language Processing*. Cambridge, MA: MIT Press.

Rosé, Carolyn Penstein, 2000. A Framework for Robust Semantic Interpretation. In Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP '00) Seattle.

Seu, Jai Hyun, 1992. The Development of an Input Understander for an Intelligent Tutoring System Based on a Sublanguage Study. Ph.D. diss., Illinois Institute of Technology.

Seu, Jai Hyun and Martha Evens, 1991. Understanding Ill-Formed Input to an Intelligent Tutoring System in an LFG Framework. In Proceedings of the Third Midwest Artificial Intelligence and Cognitive Science Society Conference, Southern Illinois University, Carbondale, pp. 36–40.

Wiemer-Hastings, Peter, Katja Wiemer-Hastings, and Art Graesser, 1999. Improving an Intelligent Tutor's Comprehension of Students with Latent Semantic Analysis. In *AI in Education 1999, Le Mans, France*, pp. 535–542. Amsterdam: IOS Press.

Zhou, Yujian, Reva Freedman, Michael Glass, Joel A. Michael, Allen A. Rovick and Martha W. Evens. 1999. "What Should the Tutor Do When the Student Cannot Answer a Question?" *Proceedings of the 12th International Florida Artificial Intelligence Symposium (FLAIRS '99)*, Orlando, FL.

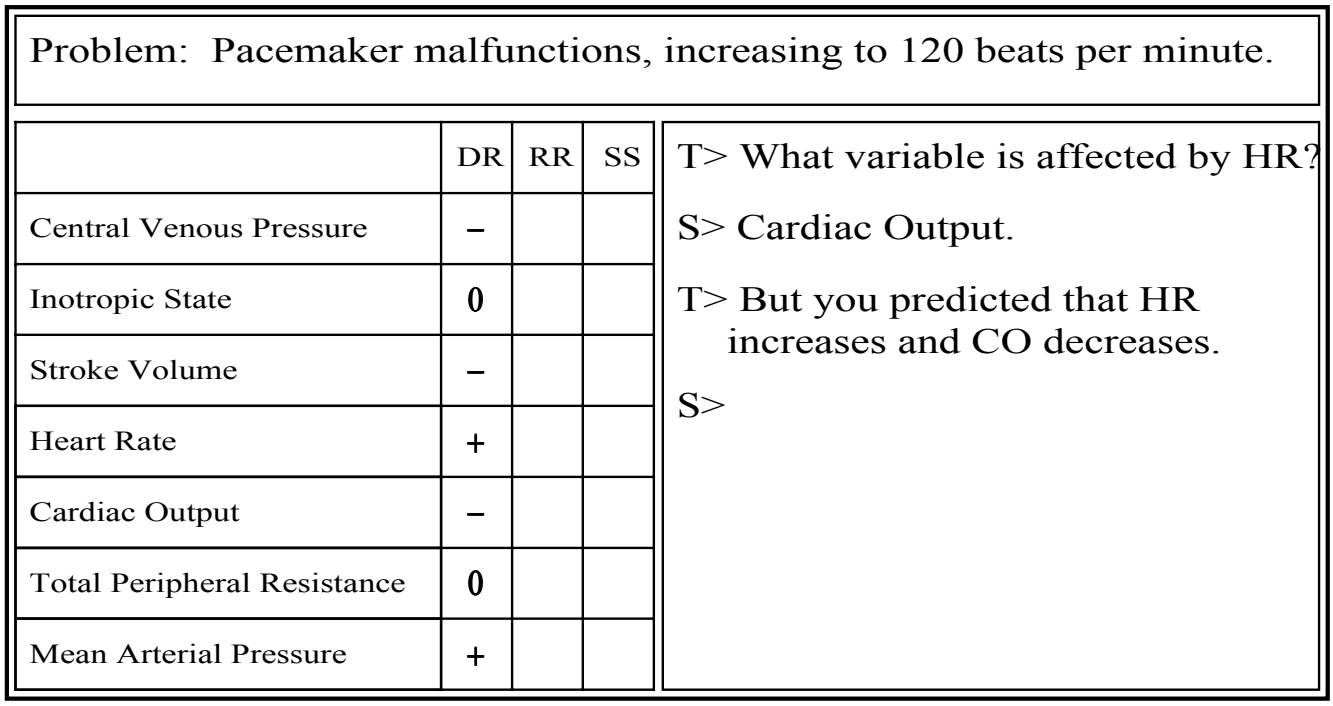

Figure 1. Simplified View of CIRCISM-Tutor Screen