Proceedings of Midwest Artificial Intelligence and Cognitive Science Society Conference, Dayton, 1998, pp. 124-131.

RELATIONSHIP BETWEEN TUTORIAL GOALS AND SENTENCE STRUCTURE IN A CORPUS OF TUTORING TRANSCRIPTS

Jung Hee Kim
Department of CSAM
Illinois Inst. of Technology
10 W. 31st Street 236–SB
Chicago, IL 60616
janice@steve.csam.iit.edu

Reva Freedman†
LRDC #819
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 15260
freedrk+@pitt.edu

Martha W. Evens
Department of CSAM
Illinois Inst. of Technology
10 W. 31st Street 236–SB
Chicago, IL 60616
csevens@minna.cns.iit.edu

ABSTRACT

An intelligent tutoring system, CIRCSIM-Tutor tutors first-year medical students on blood pressure regulation based on the dialogue patterns of human tutors. To obtain data about the language and conversation patterns of human tutors, we analyzed transcripts of human tutors working over a modem, then annotated them to show tutorial goal structure. In this paper we analyze clusters of sentences serving the same tutorial goal. We attempt to determine the information content required by each group and possible sources of these content elements. We show potential surface structures which could be generated from these elements. We discuss the influence on our work of the theories of Michael Halliday and Deborah Schiffrin. The results of this work will assist us in building a text generation system for CIRCSIM-Tutor v. 3 which will mimic some of the natural qualities of the speech of human tutors in a simple and efficient manner.

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.

†This work was performed while Reva Freedman was at the Illinois Institute of Technology.

INTRODUCTION

CIRCSIM-Tutor is an intelligent tutoring system designed to tutor first-year medical students on blood pressure regulation. The students are requested to predict the qualitative change (increase, decrease, no change) in seven core parameters according to the response of the baroreceptor reflex to a perturbation. The tutor analyzes these predictions and conducts a dialogue with the student to correct the errors.

As is the usual practice in text generation studies [Reiter and Dale, 1997], we analyzed a corpus of dialogues between students and human tutors to obtain data about the nature of their tutoring language. We annotated the transcripts with the tutorial goal structures which are the basis for our plan-based text generation [Freedman and Evens, 1996]. This analysis produced nested annotations showing global goals for tutoring and additional local goals for immediate response to the student.

We extracted groups of sentences serving the same tutorial goal and analyzed each group to determine a set of content elements which could be used to build each of the sentences in the group. Then we attempted to determine which pieces of knowledge in the planning environment could be used to determine the values of these elements. Finally, we used

this information to sketch potential surface structures which our generator could produce.

Breadth of coverage, cost of implementation and response time are all relevant issues in the implementation of CIRCSIM-Tutor v. 3. For these reasons we are experimenting with a sophisticated content planning process followed by a simple template-filling process for surface generation. According to our initial results, much of the tutoring language is stylized enough that we can assemble the sentences directly. But whether or not this module is used for surface generation in the final CIRCSIM-Tutor v. 3, the analysis described in this paper, which shows the relationship between tutorial goals and surface structure, can be used to determine the output which any eventual surface generation component must produce.

ANNOTATING TUTORIAL GOALS

The transcripts, of which we have approximately fifty, were made by physiology professors and medical students. To simulate the CIRCSIM-Tutor environment as closely as possible, they communicated with each other keyboard-to-keyboard from different rooms. Our current analysis is based on about 270 turns of dialogue, including approximately 350 instances of global tutoring goals and 50 instances of local goals.

Figure 1 shows an example of our SGML-based markup. Where possible, we form the names of tutorial goals by combining a predicate with the value of the ***info=*** argument which identifies the low-level content chunks in our representation. Thus ***T-tutors-value*** should be considered an abbreviation for ***T-tutors info=value***.

As Figure 1 shows, the tutorial goals are expanded hierarchically. For each variable which the student did not predict correctly, two sections of dialogue are generated:

```
<T-introduces-variable>
tu: Let's talk about TPR.
</T-introduces-variable>

<T-tutors-variable>
  <T-does-neural-DLR>
    <T-tutors-mechanism>
      <T-elicits>
tu: Can you tell me how TPR is controlled?
<S-answer catg=near-miss>
st: Sympathetic vasoconstriction.
</S-answer>
<T-ack type=positive>
tu: Right.
</T-ack>
      </T-elicits>
      <T-informs>
tu: TPR is primarily under neural control.
      </T-informs>
    </T-tutors-mechanism>

    <T-tutors-DR-info>
      <T-informs>
tu: We're talking about what happens before
    there are any neural changes.
      </T-informs>
    </T-tutors-DR-info>

    <T-tutors-value>
      <T-elicits>
tu: So what about TPR?
<S-ans catg=correct>
st: No change.
</S-ans>
<T-ack type=positive>
tu: Good.
</T-ack>
      </T-elicits>
    </T-tutors-value>
  </T-does-neural-DLR>
</T-tutors-variable>
```

Figure 1. Example of Annotated Transcript

T-introduces-variable introduces the variable that should be corrected and ***T-tutors-variable*** contains the actual tutoring. (The

goals in italics are not part of the hierarchy.)

Tutoring requires at least three levels of goals below the variable level: the *method* level, the *topic* level, and the *primitive* level. Although our planner, a general-purpose planner which we have adapted for dialogue generation, can handle an arbitrary number of levels, the text produced by our expert human tutors can be modeled with a restricted number of levels.

The method level shows how to teach about a variable. Within each method, a sequence of topics represent the items to be taught. The primitive level shows how this information is communicated to the student.

Text relating to each variable is generated by refining the goal ***T-tutors-variable***. The choice of tutorial method is determined by the tutorial agenda, the tutorial history, domain knowledge, and the student model. In Figure 1, the tutor uses domain information—the fact that TPR is a neurally controlled variable—to choose the most appropriate method, in this case the question-and-answer style method ***T-does-neural-DLR***. (Here "DLR" stands for 'directed line of reasoning', a form of Socratic dialogue.)

Each method consists of a series of topic-level plan operators. In this example, ***T-tutors-variable*** is decomposed into three topics, ***T-tutors-mechanism***, ***T-tutors-DR-info***, and ***T-tutors-value***.

Each topic operator represents one item to be taught to the student. These topic operators share the use of the standard text generation primitives ***T-informs*** and ***T-elicits***. The ***T-informs*** operator is used to give information to the student. ***T-elicits*** is used when the tutor wants the student to provide an answer. When expanding a plan operator, the planner has access to the arguments of all of the logic forms hierarchically above it, so we do not need to write those arguments explicitly. Combined with the fact that many topic-level arguments contain an implicit ***info=*** argument, we note that most primitives will therefore have this argument available.

Although ***T-informs*** is usually realized as a declarative sentence and ***T-elicits*** with an interrogative, other alternatives are possible. For example, ***T-elicits*** could be realized as an imperative: "Please tell me ..." [Freedman, 1996].

With the student's correct answer the tutor moves to the next goal. A clearly incorrect answer usually causes the tutor to add a corrective topic or change to a new method. When the student's answer is "I don't know," the tutor often gives a hint in the form of an additional topic before continuing with the method. The tutorial goals and their use in flexibly responding to student errors are described in more detail in [Kim, Freedman and Evens, to appear].

USE OF HALLIDAY'S THEORY

According to Halliday [1985], language is used to express three kinds of meaning simultaneously: *experiential meaning,* the propositional content of an utterance, *interpersonal meaning*, which represents the attitude of the speaker, and *textual meaning*, which represents the contribution of the utterance to the narrative coherence of the conversation.

In our planner, the content axis is represented by the plan operators themselves in addition to content-based arguments, e.g. ***info=***. We add additional arguments when it is desirable to represent the interpersonal and narrative axes. In this way we can considerably enrich the range of concepts which we can express. This ability sets v. 3 of CIRCSIM-Tutor apart from earlier versions as well as from question-answering systems.

The ***attitude=*** feature is used to express the

tutor's personal stance with respect to the material being uttered. For example, consider the differences between the following sentences:

(1) CO increased.
(2) But remember that CO increases. ***(attitude=remind)***
(3) CO certainly does increase. ***(attitude=support)***

The latter might be used, for example, in place of the more common *CO increases* to reply to a student who has made this assertion along with a number of incorrect assertions.

Similarly, the ***narrative-mode=*** argument is used to annotate aspects of text which relate to structural coherence of the dialogue. The following examples show some of the distinctions which can be made using this argument.

(4) You predicted that CO increased. ***(narrative-mode=reference)***
(5) So, CO increases. ***(narrative-mode=summary)***

When used in running conversation, ***narrative-mode=summary*** lets ***T-informs*** act as a transition element ("So, CO increases. Now, ..."). It is distinct from the tutorial method *summary* which generates summaries of domain reasoning ("So, in DR HR is up, CO is up, but SV is down.").

DATA ANALYSIS

From our annotated transcripts we extracted groups of sentences where the tutor was expressing the same meaning. For purposes of generation, we would like to understand the variation among the sentences in each set. Although it is axiomatic that different sentences can never have identical meanings, for text generation purposes we only need to represent differences which the tutoring system needs to make. In other words, if two sentences would serve the same purpose for our tutor, then we can consider them as different ways of expressing the same thought and generate them from the same logical form.

In many cases, the available research has not yet given us the tools to make well-grounded distinctions among these sentences. If two sentences can be used in the same slot, we prefer to consider them as equivalent rather than make arbitrary distinctions.

For example, consider the sentences in Figure 2. These sentences have been printed to show how they are built from common elements:

- *Main predicate* (elicit)
- *Information content* (mechanism of control)
- *Name of variable*

The main predicate and the value of ***info=***, which are realized together in this example, are printed in boldface with slots showing

(6) **How is** TPR **controlled**?
(7) **How is** TPR **determined**?
(8) **What is the primary mechanism of control of** TPR?
(9) Can you tell me **how** TPR **is controlled**?
(10) Do you know **what determines the value of** TPR?
(11) AND **what is the primary mechanism by which** arteriolar radius **is controlled**?

Figure 2. Examples of ***T-elicits info=mechanism***

where arguments like the variable name are to be inserted.

- *Softener (optional)*

We are using the term "softener" to describe expressions like *can you tell me* and *do you know*, which are underlined in the examples below. Although the exact meaning of these expressions is an oft-debated issue, it is not necessary for our purposes; we simply observe that these expressions can be used or omitted.

It is interesting to note that when one of these expressions is realized as a sentence-initial clause, the surface subject of the sentence is different from the deep subject of the predicate. Glass [1997] points out the same phenomenon in the student's side of the dialogue.

- *Discourse marker (optional)*

Discourse markers are printed in small caps, e.g. AND. In our dialogues, most of the discourse markers are sentence-initial.

Figure 3 shows a similar decomposition of sentences encoded by ***T-elicits-value***. In addition to the previous elements, a few new elements are present:

- *Time qualifier (optional)*

Time qualifiers are printed in sans-serif italics, e.g. *in DR*.

- *Context-setting expression (optional)*

We are using the term "context-setting expression" to describe expressions like *if CC is under neural control* or *that being the case*, which set the context for the main clause. Note that some context-setting expressions are constant while others contain slots.

- *Pointing expression (optional)*

Figure 3 also contains the expression *as you predicted*, which points to something which happened earlier in the dialogue. As Figure 4 shows, most pointing expressions in our dialogues are sentence-initial.

Figure 4 contains a selection of sentences from the ***T-informs-value*** group. The value of our approach can be seen in the fact that no new elements are required for these sentences. Occasionally we need to add an argument to further qualify an element. For example, note that (25) simply refers to a prediction ("... would change"), whereas the other examples in Figure 4 give a specific value for the prediction. When we need to differentiate between these two cases, we add

(12) SO **what about** TPR?

(13) SO **what's your prediction of** CC *in the DR*?

(14) NOW **what do you say about** TPR?

(15) BUT *if CC is under neural control*, **how would** it **be affected** *in the DR period*?

(16) SO **what's your prediction about** CC?

(17) SO **what would happen to** RAP?

(18) *That being the case*, **how would** RAP **change** *in DR*?

(19) SO, *in the DR* **will there be any change in** TPR?

(20) AND *if RAP increases* **what would happen to** RAP-*DR*?

(21) *That being the case*, **what will happen to** RAP-*DR* in this situation?

(22) *If cardiac output decreased* (*DR*) *as you predicted*, **what would happen to** RAP?

Figure 3. Examples of ***T-elicits info=value***

the argument ***specific-value=*** to the logic form.

Generalizing from these examples, we can see that our sentences are constructed from four kinds of elements:

- *Main predicate and required arguments*

In the examples above, the primary example of a required argument is the variable name.

- *Optional arguments*

Time qualifiers (e.g. *in DR*) are an example of an optional argument. If location qualifiers were used, they would also fall in this category.

- *Interpersonal modifiers*

Softening expressions are an example of this category.

- *Narrative modifiers*

Discourse markers, pointing expressions and context-setting expressions are included in this category.

For each element, we must determine which features in the planning environment are needed to determine its value (and, if it is optional, whether it should be included at all). The tutorial history, the domain model and the student model are among the data structures which can be consulted. However, in this paper, we are most interested in features which come from the current planning agenda, which includes the current planning goal, the goals above it in the current hierarchy, and the arguments of these goals. Of course, in different rule bases an element may be computed in different ways and using different inputs.

For example, consider the sentences in Figure 3 again. In these sentences, the main predicate can be determined from the current goal, i.e. ***T-elicits***. The content to be elicited is carried down from the ***info=*** argument of the parent goal. All of these sentences are derived from topic goals which contain either an explicit or implicit ***info=*** argument.

When we started this research, we assumed that discourse markers would have to be generated based on tutorial history in order to have a coherent conversation. However, the majority of the discourse markers in our corpus can be determined based only on the method they belong to and their position in it. The situation is similar for context-setting expressions.

Comparing the different sets of examples, we notice many regularities in the surface syntax. A large percent of the sentences can be generated from the following BNF:

Discourse particle (optional)
    Source: Discourse marker
Front clause (optional)
    Source: Pointing expression, context-setting expression, or softener
Main clause
    Source: Main predicate with required arguments in slots
Additional arguments (optional)
    Source: Time qualifiers and other potential optional arguments

A full-scale surface generation component would permit greater variety in the top-level

(23) *You predicted that* CC **would go up**.
(24) BUT *remember that you said that* MAP **decreases** *in DR*.
(25) WELL, *you made predictions about* **how** RAP and CC **would change**.
(26) *You predicted that* CO *in DR* **would go up**.

Figure 4. Examples of ***T-informs info=value***

syntax of the sentence. More importantly, it would also permit more complex rules for generating components, including rules which use more than one input element to determine a surface component and allow input elements to be realized in more than one part of the syntax, e.g. by more than one part of speech. However, this approach to surface generation may well be useful as an intermediate step.

DISCOURSE MARKER THEORIES

In the preceding examples, we have seen a number of discourse markers which perform a variety of functions. According to Schiffrin [1987], *well* is a way of providing conversational coherence when the speaker isn't satisfying coherence in an expected way. For example, *well* can begin a response where the tutor contradicts the student, without being confrontational. It can also signal that a previous question is being asked again, because it wasn't answered. The following example seems to illustrate both uses simultaneously:

tu: What do you think will happen to SV?
st: No change.
tu: *Well*, you predicted that RAP would in fact go down and you predicted that CC would not change.
tu: *So*, what happens to SV?

The above example also illustrates a use of the discourse marker *so.* According to Schiffrin's theory, *so* indicates that what follows, i.e. what happens to SV, is a result of the previous facts, i.e. that RAP went down and CC did not change.

Although Schiffrin's analysis may give one a feel for the use of discourse markers, it was not intended for text generation and is not well-suited for that purpose.

In our transcripts, the following clearly implementable rules suffice for generating one common category of discourse markers.

- *First* is used on the first topic of a multi-topic method.
- *And* is used on the intermediate topics of a method.
- *So* is used on the last (concluding) topic of a method.

BENEFITS OF THIS APPROACH

The value of this approach to text generation lies in the fact that it separates the pedagogical goals of a sentence from the text generation goal used to derive it. The rules required to realize a given text generation goal need only be provided once no matter how many contexts the resulting sentence can occur in or how many purposes it can serve.

For example, consider the following utterances:

(27) You predicted that CC would go up. But remember that we are dealing with the period before there can be any neural changes. How can CC go up if it's under neural control?
(28) You predicted that CC would go up. What does this tell you about the value of SV?

Both of these excerpts contain the sentence *you predicted that CC would go up*. In (27), the sentence is part of a realization of the method ***T-shows-contradiction***, where it is used to echo the student's prediction as part of a demonstration of a student error. In (28), it is derived from the method ***T-moves-forward***, where it is used to help the student follow a causal chain. But both sentences are derived from identical instances of ***T-elicits***.

Note that in (27), the variable name in the sentence is the one the tutor is teaching about, whereas in (28) the tutor points to the value of CC as part of tutoring about SV. Again, the derivation of the sentence is independent of the source of the information provided in

the arguments.

Since CIRCSIM-Tutor v. 3 is a rule-driven unification-based system, we do not need to specify at what grain size the translation from logic form to text takes place. In other words, all forms of ***T-elicit*** which can be converted to surface text in the same way can be handled with one rule. When specific combinations of arguments, e.g. ***T-elicit narrative-mode=reference*** or ***T-elicit info= value***, generate significantly different sentences, then new rules can be added which match on the required arguments.

CONCLUSION

In this paper we showed how to identify the semantic and pragmatic content of tutorial goal structures collected from transcripts of human tutors. These elements could serve as input to a surface generator. We showed how they could be correlated with components of a simple syntactic form in order to build a tiny surface generator which might be a useful intermediate step while data is collected for a more sophisticated surface generation module.

Analysis of the semantic and pragmatic content of tutorial goals is a convenient way to organize the data necessary to build a broad-coverage text generation system like CIRCSIM-Tutor v. 3. Our goal is to generate fluent and varied text, similar in structure to that generated by human tutors, without a corresponding degree of complexity in the generator.

ACKNOWLEDGMENTS

Joel A. Michael and Allen A. Rovick, professors of physiology at Rush Medical College, are responsible for the pedagogical and domain knowledge in CIRCSIM-Tutor, and served as expert tutors for the transcripts. We thank Michael Glass for his contributions to many aspects of this paper.

REFERENCES

Freedman, R. 1996. "Using a Text Planner to Model the Behavior of Human Tutors in an ITS," Seventh Midwest AI and Cognitive Science Conference (MAICS '96), Bloomington, IN. http://www.cs.indiana.edu/event/maics96/Proceedings/Freedman/freedman.{html, ps}.

Freedman, R. and Evens, M. W. 1996. "Generating and Revising Hierarchical Multi-Turn Text Plans in an ITS," Intelligent Tutoring Systems: Third International Conference (ITS '96), Montreal, pp. 632–640.

Glass, M. 1997. "Some Phenomena Handled by the CIRCSIM-Tutor Version 3 Input Understander," Proceedings of the Tenth Florida Artificial Intelligence Research Symposium, Daytona Beach, FL, pp. 21–25.

Halliday, M. A. K. 1985. Functional Grammar. London: Edward Arnold.

Kim, J., Freedman, R., and Evens, M. W. To appear. "Responding to Unexpected Student Utterances in CIRCSIM-Tutor v. 3: Analysis of Transcripts," Proceedings of the Eleventh Florida Artificial Intelligence Research Symposium (FLAIRS '98), Sanibel Island, FL.

Reiter, E. and Dale, R. (1997). "Building Applied Natural Language Generation Systems." Journal of Natural Language Engineering, 3: 57–87.

Schiffrin, D. 1987. Discourse Markers. Cambridge: Cambridge University Press.