

Tutorial Dialogue

GORDON I. McCALLA

ARIES Laboratory

Department of Computational Science

University of Saskatchewan, Saskatoon, Saskatchewan, S7N 0W0, Canada

Abstract: In this paper I will provide a short tutorial on the problem of modelling tutorial dialogue. First, a brief overview of traditional approaches to the computational modelling of language will be presented in order to form a context for the discussion of tutorial dialogue. Then, a comparison between these traditional approaches and tutorial dialogue will be made. Finally, a small selection of tutorial dialogue systems will be discussed in order to give a flavour for the kinds of issues that have been and still must be explored if tutoring systems are eventually going to engage their learners in natural language dialogues about the subject domain.

Introduction

Ideally, an intelligent tutoring system should be able to interact with the learner in the learner's own natural language. Modelling such *tutorial dialogue* is not at all straightforward. Tutorial dialogues are characterized by being mixed initiative, with either speaker being able to take control of the conversation, the learner to ask a question, the tutor to make a pedagogical intervention. A full range of natural language expressions can be used, including badly formed input, partial sentences, and inappropriate use of words and phrases, especially domain terminology the learner has not yet quite understood. In fact, the learner's level of understanding critically affects both what is said and how it is expressed. Moreover, this level of understanding is constantly evolving and changing as the learner makes progress in grappling with the domain. As in many dialogue situations, the speakers may each be following a plan, but in the instructional arena the tutor's plan may implement a subtle teaching strategy, and the learner's plan may be deliberately obstructionist, or unconnected to the tutor's plan, or virtually non-existent for all practical purposes.

These characteristics make it very difficult to adapt traditional natural language techniques holus bolus to the problem of modelling tutorial dialogue. Many traditional applications for natural language understanding assume that the user will be helpful or that the dialogue situation is task-oriented or that the user is more or less static in their understanding of the world or that the user will use fully formed correct expressions or that the domain is simple enough to constrain the possible ambiguities in what is said. In general, none of these assumptions holds true for tutorial dialogues. Nevertheless, methodologies developed for traditional natural language understanding systems can be adapted to some extent to the problem of modelling tutorial dialogues.

In this paper I will present very briefly some of the traditional techniques for natural language understanding. This will provide a context for a discussion of tutorial dialogues, what makes them different, and what work has been done to model them. I want to distinguish tutorial dialogue systems from tutoring systems that have natural language as their domain. There are many tutoring systems that help learners to gain facility in a second language (Schwarz & Yazdani, 1992), help them learn some feature of language better (e.g. English article usage, as in the Fawlt tutor (Kurup et al, 1992), etc. These systems are not the focus of this paper. Instead, the paper is attempting to look at the issue of how to incorporate dialogue capabilities into *any* tutoring system, regardless of domain, a much more general, and much more difficult problem. In fact, modelling tutorial dialogue is so difficult a problem that not a lot of work has been done in the area. Tutorial dialogue is still mostly a marvellous research testbed for inventing and testing new dialogue modelling ideas, rather than an area with lots of immediately practical spin-offs. Thus, for the foreseeable future, most intelligent tutoring systems will have to either have fairly crude natural language interfaces or will have to find end-runs around the need for natural language at all.

Overview of Standard Approaches to Natural Language Understanding

Language can be viewed as having three levels of analysis: syntax (structure), semantics (meaning), and pragmatics (context). The stereotypical approach to understanding natural language is to assume that the understanding process peels back each layer of analysis in sequence. First, the linguistic utterance is parsed to find its structure, a meaning representation is concocted, and then this meaning representation is refined according to the context of the linguistic utterance. After this, an appropriate action is taken by the natural language understanding system, and natural language output is generated. This general paradigm has been used in various applications, such as question answering, queries to databases, paraphrase and precis generation, providing help, and natural language translation. The assumption in most of these applications is that the system is neutral and that the system's main goal is to help the user to achieve his or her goals. The normal unit of discourse is the single sentence or single question.

Syntax

Techniques have been developed for each level of analysis. Most well understood is the syntactic level. The goal of syntactic analysis is to find the grammatical relationships in a sentence. This goal is made more difficult by a number of problems that can arise. One is syntactic ambiguity. Sentences such as "I saw Hong Kong on the way to Taipei." can have more than one syntactic structure (is the prepositional phrase "on the way to Taipei" attached to the verb "saw" or to the noun "Hong Kong"?). Another problem is ungrammatical input, for example "What is length with dog?". A third problem is partial input, for example the first sentence in the following pair of expressions: "Gadgets. Where are they?"

Recognizing the syntactic structure of a sentence is usually done using a grammar that captures the grammatical relationships in a subset of a natural language. A grammar uses rewrite rules to show how each grammatical category on the left hand side can be broken down into grammatical sub-categories or words on the right hand side. Standard parsing methodologies can be used to actually parse the input sentence using the grammar, by relating the words in the input to the categories into which they can be grouped according to the grammar. For example, consider this grammar for a minuscule subset of English:

S	\rightarrow	NP VP
NP	\rightarrow	Det N Npr
VP	\rightarrow	V NP
N	\rightarrow	box ball dog ...
Det	\rightarrow	the a
Npr	\rightarrow	Ruff Max ...
V	\rightarrow	chases hits ...

Using this grammar, a parser could parse the sentence "The dog chases Max.", producing the parse tree shown in Figure 1.

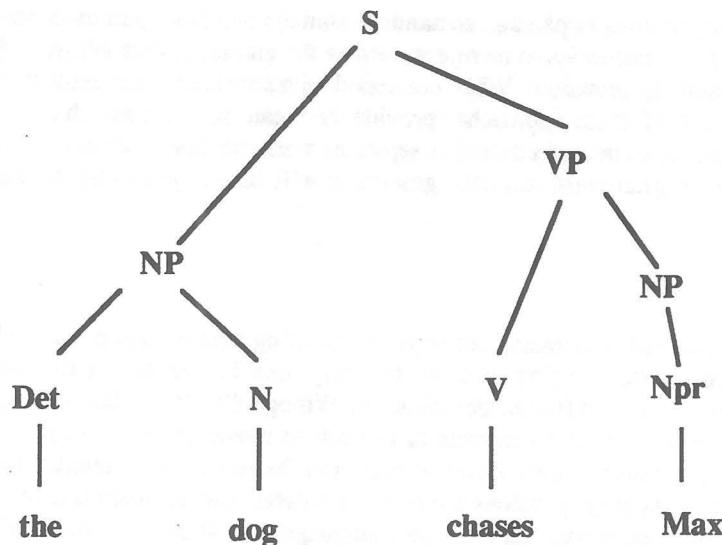


Figure 1. A parse tree

Standard parsing methodologies have difficulty handling a number of syntactic aspects of language. First, the grammars normally used by the parsers are so-called context free grammars, grammars that only allow one non-terminal symbol on the left hand side of a rule (such as the one above). There is considerable debate about whether or not context free grammars can capture all aspects of natural language. Second, certain regularities, such as agreement in number between verb and noun, are not easily captured in grammatical formalisms. Third, anomalous or meaningless input is allowed because of limitations in the grammatical categories. Thus, the sentence "A ball chases the box." cannot be ruled out on syntactic grounds unless some of the grammatical categories are specialized to include not just categories based on parts of speech of words but also categories based on attributes of the concepts that those words represent. To know that "ball" is not animate, and hence cannot be used in the example sentence, requires more attributes than traditionally allowed in a syntactic formalism. This points out the final problem with syntactic analysis as it is traditionally carried out: it really shouldn't be done separately from semantic processing. Not only is it useful to bring semantic-style categories like *animate* into the grammar, but it is also critical for efficient parsing to carry out a degree of semantic processing

while doing the syntax so as to reduce as quickly as possible the number of ambiguities being carried along in the parsing process.

A number of enhancements have been made to traditional parsing styles that help to overcome some of the shortcomings. In particular, the notion of allowing storage of temporary information during the parse can overcome many of the limitations of traditional approaches, for example by allowing storage of information about tense and person that can be used to enforce noun-verb agreement. Woods (1970) developed the "augmented transition network" (ATN) formalism that grafted registers and other programming techniques onto a context free grammar (represented as a finite state transition network) to allow fully powerful parsing (theoretically the ATN formalism has the power of a Turing machine, and hence can compute anything that can be computed).

Two other approaches to parsing, semantic grammars and case grammars, took the route of deepening the kinds of categories to be represented in the grammar, thus allowing the merging of semantic and syntactic information. When combined with sophisticated control structures used in the parsing process itself, these approaches provide the means to overcome the limitations of a strictly stratified approach to interpretation that separates syntactic from semantic processing. An elaboration of semantic grammars and case grammars will be delayed until the discussion of semantic processing.

Semantics

The idea of semantic processing is to create a meaning representation that can be used by the natural language understanding system to carry out its goals. Different meaning representations make sense in different domains. In Winograd's (1972) blocks world domain, where a simulated robot is given commands in English to move simulated blocks around, the natural meaning representation is procedures to carry out the robot's movements. In a database domain, where natural language questions are being translated into database queries, the natural meaning representation is an expression in a query language such as SQL. If the application is to translate one natural language to another or to summarize natural language discourse, then a specially designed formalism, such as the conceptual dependency formalism of Schank (1972), that captures the meaningful domain relationships is appropriate.

To generate this meaning representation requires the natural language understanding system to look through the sentence for linkages that are meaningful in the domain. Two approaches to doing this will be briefly synopsized here. The first approach, semantic grammars, takes the idea of a context free grammar, but deepens it so that semantic categories appear in the grammar. Consider the semantic grammar below representing a fragment of the kinds of expressions that might be used to query a database full of ship information (this is adapted from the LIFER/INLAND system of Hendrix et al, 1978):

<sentence>	-->	<present> the <attribute> of <ship>
<present>	-->	what is give me
<attribute>	-->	length beam class
<ship>	-->	the <shipname> <classname> class ships
<shipname>	-->	Bonaventure Intrepid
<classname>	-->	battleship cruiser

This semantic grammar can be used by a suitably augmented parsing system (such as LIFER) to parse sentences like "What is the class of the Bonaventure?" or "Give me the length of cruiser class ships." easily and directly into queries to a database, say:

```
(find ?x
  where (attribute ?x class)
        (ship ?x Bonaventure))
```

The semantic categories chosen must be relevant to the domain, in this case attributes in a relational database. While semantic grammars are extremely elegant, and nicely merge syntactic and semantic processing into one unified formalism, they do have a problem: coverage. The added depth that the semantic categories provide to the interpretation process is at the cost of breadth of linguistic coverage. A semantic grammar to handle a wide range of natural language inputs would be very large, and would probably still have gaps in its coverage.

Case grammars are an alternative approach to semantic processing that similarly merges syntactic and semantic processing, but using a more domain-independent theory of semantics. Various approaches have been proposed, including some by linguists such as Chafe (1970) and Fillmore (1968) and some by computational linguists such as Bruce (1975) and Schank (1972). The basic idea is that each verb has associated with it case frames (one for each sense of the verb) that describe both the syntactic and semantic relationships required by the verb. These can be either generic relationships or verb-specific conditions, and can incorporate syntactic or semantic information. Consider the case frame for the verb "chase":

```
chase
agent : <?x>
  (must-be animate physobj)
  (should-be subject position)
patient : <?y>
  (must-be moveable physobj)
  (should-be object position)
locative : <?z>
  (should-be place)
  (marked-by locative preposition)
```

During parsing, case frame slots (represented in angle brackets) are filled to produce the meaning representation. The slot fillers must satisfy the conditions attached to the slots. For example in the sentence "Max chased the car down the street.", the case frame would be instantiated as follows:

```
chase
agent : Max
patient : car
locative : street
```

This interpretation succeeds because "Max" is animate and is in the syntactic subject position of the sentence, so can be an agent; "car" is moveable and in the syntactic object position, so can be a patient; and "street" is a place and is marked syntactically in the sentence by the locative preposition "in", so can fill the locative slot. However, if some inanimate agent, for example

"The table" was in the subject position, then the case semantics would reject an interpretation of the table as the agent.

Since slot restrictions can be relaxed if there is a failure to understand, since extraneous words can be ignored, and since words that are out of order can still be sought out to fill unfilled slots, it is obvious that case-based parsers are very useful for handling ill-formed input. They also, in general, allow broader coverage than semantic grammars because they are based on more domain-independent theories of semantics. However, no truly general semantic theory yet exists, so the coverage will still be haphazard, and the case frame output is not always the most natural form from which to generate a meaning representation.

Pragmatics

Pragmatics is the level of linguistic analysis concerned with context. Some pragmatics issues that arise include

- ellipsis, as in
 - "Give me the beam of the Intrepid."
 - "the length."
- anaphora, e.g.
 - "It chases that down there."
- importance of roles and goals of speakers; e.g.
 - clerk : "Mangos?"
 - stock-boy : "Aisle 3"

vs

 - clerk : "Mangos?"
 - other clerk : "2 dollars"
- importance of what is going on in the environment; e.g.
 - ticket-seller: "Yes?"
 - ticket-buyer: "Two on the aisle."

In each of these pragmatics issues the unifying theme is the need to refer to a context beyond the linguistic input itself in order to understand what is going on. It should also be noted that the style of syntactic and semantic processing can make the resolution of these pragmatics issues easier or harder. For example, if semantic grammars are being used, ellipsis can be readily dealt with because the complete parse tree, including semantic relationships, of the previous utterance can be used as the basis for interpreting the ellipsis. Thus, "the length" above can be plugged into the semantic tree and recognized as "Give me the length of the Intrepid." Similarly, if case frames are used, then when there is an anaphoric reference, the slot conditions of the case frame representing the current input can be very useful for seeking out appropriately matching predecessors in case frames representing previously interpreted utterances. In the anaphora example above, the "It" in the sentence must be something animate if the case frame is to be successfully filled; thus the search through previous discourse can look for something animate, rather than any old "it" at all.

The pragmatics level is very complex, and the techniques developed to handle issues at this level are by no means complete. A few of the ideas that have been explored will be presented

here to give a flavour of the approaches being taken. Many of the special difficulties confronting tutorial dialogue emerge from pragmatics issues. Thus, work on modelling tutorial dialogues will have to focus on pragmatics issues especially.

One of the key ideas at the pragmatics level is that language is purposeful behaviour. Like any action, linguistic actions carry out higher level goals of a speaker. These linguistic actions are called speech acts (after Austin, 1962), and one of the important things about understanding language is to discern what speech act underlies an utterance. Thus, "Could you give me the class of the Intrepid?" is a request, "Stand at ease!" is a command, and "Taiwan is an island in the western Pacific ocean." asserts a fact. Speech acts may not obviously map onto a surface linguistic form. Thus, "The library is a place to be quiet." on the surface may seem to be an assertion, but if uttered by a librarian to a noisy patron may in fact be a command (to shut up!) A linguist, Searle (1969), formulated a theory of speech acts that has formed the basis for most computational linguistics approaches that use them.

Speech acts have three important aspects: locution, what is said (or written); illocution, what is meant; and perlocution, the intended effect. It is important to understand all three aspects in order to fully understand an utterance. In the library example, the locution and illocution together still do not capture the full effect: perlocution is the crucial aspects. Determining perlocutionary effects complicates the interpretation process because it demands an understanding of other contextual issues, in this case knowing who is making the utterance and in what situation that person finds himself/herself.

The librarian example is not unique: in many domains knowing who is making an utterance is often a critical factor in fully understanding what is said. In fact this need is so critical that an entire sub-discipline of artificial intelligence has slowly emerged over the last 15 years in order to study issues of user modelling. To give a flavour for the kind of work that has gone on, look at the "Mangos" example above, adapted from Cohen's (1978) work on planning speech acts. The clerk is able to use the single word phrase, "Mangos?", to mean "Where are the mangos?" when talking to the stock-boy because both the clerk and the stock-boy know that the stock-boy's expertise is the location of goods in the store. The same phrase can be used by the clerk to mean "How much do the mangos cost?" when talking to another clerk because both parties know that a clerk's expertise is prices.

Perrault, Allen, and Cohen (Allen & Perrault, 1980, Cohen, 1978) were the first to seriously try to model how speech acts can be planned out in the context of understanding the user to whom the speech acts are directed. The assumption underlying their work is that speech acts are planned in order to achieve a linguistic goal, just as motor actions are planned in order to achieve a goal in the physical world. The speech acts, however, are planned using information about the user stored in "belief spaces", representing the system's beliefs about the beliefs (and goals) of the user. These can be arbitrarily nested to represent the system's beliefs about the beliefs of the user about the system, or the system's beliefs about the beliefs of the user about the system's beliefs about the user, and so on. Using these nested belief spaces, Cohen showed how to plan an appropriate sequence of speech acts to achieve communication with a particular user, and Allen showed how to recognize the plan underlying a particular user's utterance in order to fully understand the perlocutionary effect of the utterance. This research established a "planning speech acts" paradigm that has been under exploration ever since.

Another important pragmatics phenomenon is our ability to "read between the lines" of what is said, especially when we are understanding language in stereotypical situations. Thus, in a story about somebody eating at a restaurant, if it is explicitly mentioned that the customer ordered fish, then it need not be explicitly mentioned that she ate the fish for the reader to later infer that she probably did so. To handle such situations, Schank & Abelson (1977) devised the concept of

scripts (a linguistic version of the similar idea of frames (Minsky, 1975) to represent stereotypical packages of information. Sort of extensions of case frames, scripts encode the standard sequence of events underlying a particular stereotypical situation in a series of slots. As the events unfold in the story, the natural language understanding system can fill these slots appropriately. If an event is not explicitly mentioned, then it can be inferred to have happened. So, in the fish example, the failure to mention that the customer ate fish is no problem since the eating step is a normal follow-up to the ordering step in the script, so the missing information can be inferred. Moreover, that she ate fish will also be inferrable since the *food-ordered* slot (or whatever it is called) will be bound to have the same value as the *food-eaten* slot. In the ticket-selling example above, the "Yes?" is interpretable by the ticket buyer because the buyer will be following the ticket-selling script. Scripts can be very useful, but have fairly narrow applicability to domains where stereotypical behaviour can be expected. Unfortunately, such stereotypical situations occur less frequently in tutoring domains than in many of the domains studied in computational linguistics.

When moving out into more general domains, scripts lose some of their power. In order to handle unconstrained natural language it is important to understand the social contract underlying linguistic interaction. Grice (1975) postulated that underlying all human conversation were a set of social rules. We use these rules to generate language, and can assume that others do so as well. In particular Grice proposed the following "maxims of conversation":

Maxims of quantity

- make your contribution as informative as required
- but not more so

Maxims of quality

- don't say what you believe to be false
- don't say that for which you lack adequate evidence

Maxim of relation

- be relevant

Maxims of manner

- avoid obscurity of expression
- avoid ambiguity
- be brief

The underlying assumption of most computational linguistics research is that the user and the system are both being as lucid and linguistically well-behaved as possible, that each is trying to help rather than hurt the other. This "helping" relationship thus allows Grice's maxims to be used directly. If somebody uses a partial sentence, then all of the relevant information must be there, by the first maxim of quality. This allows the natural language understanding system to reject the assumption that the input is incomplete and mandates it to make inferences through its knowledge base and user model in order to fill in the missing parts. If somebody says something that apparently does not follow from something else, thus violating the maxim of relation, again the natural language understanding system can try to infer the missing steps. As will be seen, the "helping" relationship may not pertain in tutorial situations, thus at the very least complicating the use of Grice's maxims directly in tutorial dialogue systems.

Much other work in artificial intelligence has applicability to the pragmatics level of discourse analysis. Thus, knowledge representation is a critical need for any intelligent understanding system, not only to encode the user model, but also so the system can reason about the domain being discussed. Plan recognition is needed, not only to figure out sequences of speech acts, but also to understand the non-linguistic goals a user may be attempting to achieve.

Planning is needed when scripts fail to capture the sequence of events, and the sequence must be generated from first principles. The situated cognition perspective has been usefully applied to the problem of analyzing the relationships between language and social behaviour (e.g. see Barwise & Perry, 1983). And so on. This need for techniques from all over artificial intelligence is typical of any sub-area of AI: in some sense each of the parts of AI is as great as the whole. This "holographic" property is especially true in intelligent tutoring systems as will be seen.

Before tackling tutorial situations, however, it might be useful to summarize the state of the art in "standard" natural language understanding:

- syntax is relatively well understood;
- semantics still tends to be domain-specific and somewhat ad hoc;
- pragmatics is wide open for investigation; only relatively sporadic forays have been made to explore most of the pragmatics issues;
- the focus has been on limited domains, often domains where the user is carrying out one, fairly straightforward task;
- the focus has been on "helping" systems;
- the focus has been on interpretation not generation;
- the focus has been largely on single sentences/questions not connected discourse.

In the next section differences between standard natural language approaches and the needs of tutorial dialogue will be presented.

Contrasting Standard Natural Language Approaches and Tutorial Dialogue

As discussed in the introduction, considerable challenges face a natural language understanding system that is meant to handle tutorial dialogue. It's not that a fully general natural language understanding system won't face the same challenges. It's that most applications for natural language understanding that have been explored are limited in some way, and this makes the understanding problem somewhat easier. Here are some of the differences between standard applications and tutorial dialogue that are the source of the challenges.

In tutorial dialogue, the student model is the analogue to the user model. The student model differs from the standard user model, however, in that it has to deal with two fundamental truths: a student's knowledge is incomplete and a student's knowledge is changing. These properties are in fact definitional to a dialogue being tutorial: students are by definition learners whose knowledge of the domain is incomplete from the start and whose knowledge evolves throughout the tutorial interaction. Many basic user modelling techniques may crumble under the necessity of dealing with incomplete and changing knowledge.

Another difference is that the role of a learner differs from that of a "normal" user in a task-oriented domain in that the learner's goals are multi-faceted, ill-specified, hard to infer, and interrelated. The learner isn't just carrying out one task in a limited domain, but is in fact trying to learn concepts in an intellectually challenging domain at many levels (analytic, synthetic, fact). The learner's own goals and plans often conflict with activities that he or she should be engaged in. In trying to understand the domain, the learner can create his or her own knowledge, perhaps conceptually incorrect, perhaps conceptually different from the tutor's own perceptions.

Similarly, the role of the tutor differs from that of a standard natural language help system, where the help system is essentially a slave to the user, doing his or her bidding, or at least attempting to. In tutorial dialogue, there can be tutor control, student control, mixed

initiative. For human tutors, it is sometimes the case that tutor goals are ill-specified in that the tutor can sometimes essentially try a variety of pedagogical interventions in the hope that one of them work.

In traditional natural language applications, both the user and the natural language system have a tacit agreement over goals. Thus, a natural language front end to a database system assumes the user will be asking questions about the database. In the tutorial dialogue situation, however, both parties come to the interaction with their own goals. There is, of course, a tacit agreement at the highest level that the learner is there to learn the domain and the tutor is there to help the learner to learn, but at the level of subgoals, a mismatch in tutor and student goals can occur which may require negotiation over what subsequent discourse will be about. This one issue of negotiation is so complex that, all by itself, it forms a nexus of interesting research in intelligent tutoring systems (Moyse & Elsom-Cook, 1992).

Not every issue is harder in tutorial dialogue. While in many ways interactions are more flexible in tutorial dialogue than in standard domains, the existence of pedagogical goals helps to constrain the full range of natural language interaction. Particular pedagogical goals of the tutor or learning goals of the learner can help to focus understanding. Unlike a normal conversation, the tutor can legitimately take control of a conversation, and justify this as pedagogically necessary (even if it isn't). This is useful if the dialogue starts to go beyond the abilities of the tutorial dialogue system to handle it.

In many standard domains, dialogue is a means to an end, e.g. to order a ticket, to get a question answered. In learning situations, however, it is often the linguistic interaction itself that is central to learning. As John Seely Brown has said: "We have been misled into thinking that natural language, *per se*, is so powerful. Instead, I think it is the dialogue process that is so powerful." (John Seely Brown, quoted in Petrie-Brown, 1989). In short, the pragmatics processes at the heart of understanding have the side effect of causing domain learning to occur. The implication for modelling tutorial dialogue is that it is essential to get the pragmatics right. Even though this is hard, at least it means that tutorial dialogue is a good domain for investigating the under explored areas of pragmatic analysis, since these issues cannot be downplayed as they can be in other domains.

Even those standard natural language approaches that deal with pragmatics issues would be sorely tested in the tutoring domain. The speech acts that must be handled by a tutorial dialogue system are more varied than in standard domains. Tutorial dialogue systems do not just have to deal with *request* and *inform*, but also many other speech acts like *promise*, *criticize*, *censure*, *argue*, *approve*, etc. Worse yet, Grice fails in the tutoring domain. Learners do not subscribe to the "helping" philosophies central to direct application of the Gricean maxims. In fact, there may not be *any* mutually agreed upon ground rules. Certainly there is reason to question Grice's particular maxims. Tutors use false information all the time to achieve pedagogical goals, students often say things for which they have little or no evidence. Because of the failure of the Gricean assumptions, it is problematical whether the body of work on "helping" systems can readily apply to the tutoring domain.

Although it hasn't been mentioned much in this paper so far, the standard paradigm for natural language processing does have a generation step in which the system produces natural language output for the user. Generation is often quite uninspired, requiring the production of a pre-stored natural language "blurb" or a straightforward transliteration of an internal formalism (e.g. a database tuple or a predicate calculus expression). In tutoring situations, however, generation is non-trivial. What is said to the learner, and how it is said, is crucial for motivation and other things. All sorts of issues like focus, phraseology, etc., come up. The natural language system can't just produce database tuples and hope that the learner understands them.

Selected Tutorial Dialogue Systems

There is relatively little work on tutorial dialogue in the intelligent tutoring systems literature. Most intelligent tutoring systems, like most AI systems, rely on end runs around the dialogue problem, the more so in ITS because of the great difficulty in finding any reasonable compromise between needing full dialogue capabilities and being able to get by with no dialogue capabilities. Of the relatively few systems that use dialogue in any serious way, I have selected fewer still. The main point of the section is to illustrate how the basic natural language issues manifest themselves in the tutoring context, rather than to do an exhaustive survey of all extant tutorial dialogue research. I have selected some works because of their historic contributions, others because they illustrate an interesting issue, still others because they are work in which I have been involved.

Historically, two tutorial dialogue systems stand out above all others: Scholar and SOPHIE. Scholar (Carbonell, 1970) was the first in a series of tutorial dialogue systems developed at BBN in Cambridge, Massachusetts by Carbonell, Collins, and others. Scholar is perhaps the purest of the tutorial dialogue systems in that, just as John Seely Brown would recommend, dialogue is the central activity. In fact, Scholar has no directive pedagogical style at all and the conversation between the system and the student is free flowing; the student learns or not from the conversation, as he or she sees fit. The system has a semantic network knowledge base encoding knowledge of South American geography and is able to generate natural language descriptions of this knowledge in answer to questions. It is also able to ask questions of its own. This not only makes Scholar the first mixed initiative dialogue system, but it is historically the first computer assisted instruction system to actually itself understand the domain, at least to some extent. Interpretation of the student's input is carried out by a case grammar in some versions of Scholar, and by a semantic grammar in other versions. It is interesting that case and semantic grammars are used in a context where free flowing dialogue is the norm rather than static question/answer couplets. This is as one might expect. Generation is from linguistic templates whose slots are filled in with knowledge gleaned from the semantic network. For its day, Scholar was a major advance, and in some ways it still stands out today.

SOPHIE is the collective name of a sequence of tutoring systems developed in the mid-seventies at BBN by Brown, Burton, & de Kleer (1982) and others. Learners using SOPHIE are given a buggy electronic circuit and are asked to debug it. SOPHIE can answer questions about the circuit, or pose them if appropriate. If asked how the circuit will behave, SOPHIE can actually simulate the circuit's operation and thereby answer very specific questions about voltages, resistances, etc., at particular points in the circuit. Like Scholar, then, SOPHIE understands its domain (the circuit) and can engage in a mixed initiative dialogue. SOPHIE's knowledge base is procedural as opposed to Scholar's more static network representations, as befits the contrasting nature of the domains. This is another lesson illustrated by the two systems: the choice of appropriate knowledge representation will vary from domain to domain; universal principles usually don't apply. Semantic grammars are used by SOPHIE to interpret the input. In fact SOPHIE introduced the notion of semantic grammar to artificial intelligence. Using semantic grammars is a particularly appropriate interpretation strategy for SOPHIE since the limited and technical nature of the domain means the coverage problems of semantic grammars will not be a big handicap. The use of semantic grammars also allows SOPHIE to interpret ellipsis and pronouns appropriately in context. Generation is from prestored natural language templates, with slots filled using information gathered from the simulation of the circuit.

MENO-Tutor (Woolf & McDonald, 1984) is also a series of tutors designed to be the dialogue component for a socratic tutor. It works in two domains: helping students learn the principles underlying the rainy climate of the American Pacific northwest coast and helping them program in Pascal. The key idea in MENO-Tutor is to use a subtle pedagogic strategy to direct the socratic dialogue. This is done by traversing a "discourse management network" which encodes various pedagogical states in a three level hierarchy. At the top of this hierarchy are various high level pedagogic states, such as *tutor* that indicate the general pedagogic style. These expand into lower level strategic states, such as *explore competency*, which in turn unfold into tactical states such as *exploratory question*. At any given point the tutor is carrying out one line of this discourse management network in its attempt to influence the student: its actual output is directed by the tactical state, but this is carrying out higher level strategic and pedagogical goals. MENO-Tutor is only a prototype for one component of a tutoring system, so cannot stand alone. In fact, it does not deal with raw natural language input, having assumed that some other component will produce the propositions with which it works. Its output is from prestored templates, although in a later version of the tutor McDonald's MUMBLE natural language generation system was strapped on to give realistic surface language.

MENO-Tutor's main contribution has been to start to try to delineate the special pedagogical rules that underlie tutorial dialogue. Once understood, these rules should make it easier to subtly constrain tutorial dialogues, thus making them tractable. Work by Wasson (1990) explored these notions further, dividing the pedagogical level into two parts: content and delivery. In her approach pedagogical decisions are made by two parallel planning systems, a content planner to decide what to say, and a delivery planner to decide how to say it. Pedagogical planning gives tutorial dialogue systems an advantage over free form dialogue systems in that in a tutoring context the planning rules can constrain the interaction quite naturally. The planning rules in some sense also replace the need for the Gricean maxims since they now form very precise rules of engagement for the tutorial interaction.

The next system I would like to discuss is the KANT system (Baker, 1989). KANT engages its learner in critical argument in two domains: musical structures and Prolog. KANT is a purely pragmatics-level system. Its input is from menus and its output is canned. At the pragmatics level, KANT negotiates mutual goals with the learner, using planning and plan recognition ideas similar to the Perrault-Allen-Cohen group. Dialogue focus is maintained through closeness metrics applied to a semantic network knowledge representation (echoes of Scholar). KANT is the first system to recognize the validity of the learner's goals in the dialogue, and to begin to tackle the problem of how to understand these goals and to deal with them. A number of researchers have followed up on the idea of negotiated learning (see the edited collection of Moyse & Elsom-Cook, 1991).

Gutwin's PMM system (Gutwin & McCalla, 1992a, 1992b) grappled with the issue of how tutors are able to use pedagogically motivated misrepresentations to influence the learner's understanding of the domain. As in the KANT system, Gutwin's PMM deals only at the pragmatics level under the assumption that other components can interpret and generate the surface language that real dialogue requires. The PMM system tries to impose a consistent, but misrepresented reality on the student, in order to aid learning. In PMM's application domain, LISP programming, it is sometimes useful to simplify or even to lie about concepts until the learner is ready to handle the complex reality. Hence, telling the learner that there is only one kind of base case in a LISP recursion will offload the necessity for him or her to worry about all the others, and the learner can focus on the central issue of how to recursively reduce a problem rather than on how to stop the recursion.

Gutwin created an extension of the idea of frames in order to represent his PMMs. All natural language input and output would flow through the PMM frame in order to be massaged into a form compatible with the imposed misrepresentation. This way, the misrepresentation could be consistently applied and the learner would not be distracted by various versions of reality. The various alternative realities are carefully imposed and carefully removed through special sub-dialogues that ensure the student is not confused and the system can still maintain its credibility even after the misrepresentation is corrected. The main contribution of Gutwin's approach is that he has challenged Grice directly and he gets away with it. The PMM idea explicitly recognizes that people do say things they know to be false, but that this is not always bad. Moreover, as Gutwin shows, it is possible to come up with a computational approach to handle such a non-Gricean world.

The final system I would like to discuss is the G.E.N.I.U.S. (McCalla & Murtagh, 1991) PL/C help/advising system. G.E.N.I.U.S. engages the learner in a dialogue about problems in their PL/C program, and, like Scholar, the learner is supposed to correct these problems through the dialogue process itself. Unlike Scholar and the other tutorial dialogue systems, however, G.E.N.I.U.S. does not attempt to incorporate deep knowledge in order to interact with the user. Instead, it makes an end run around the need for deep knowledge. At the heart of G.E.N.I.U.S. is a discrimination network. Keywords recognized in the student's input map onto nodes in the network. Attached to any network node is a natural language blurb that can be generated as the system's response to the student's input when the keyword associated with that node is recognized in the input. This prestored blurb can be phrased in such a way as to direct the student toward producing certain responses which the system can follow up in context. For example, if the student has asked about the meaning of a particular error message, then the node associated with that error message may produce a blurb that asks the student about what he or she is doing at that stage. Depending on what the student answers, G.E.N.I.U.S. can branch to any of a set of adjacent nodes to provide specific follow-up information about what the error message means in the context of what the student is trying to accomplish.

G.E.N.I.U.S.'s "ignorance-based" approach is very robust. Student input can be very badly mangled, and the system can still respond. With a cleverly designed discrimination network, the tutor can give the student a good deal of control. Moreover, it can appear to be understanding the student, even when that understanding is shallow and can appear to follow up on questions, even when it has directed the conversation to the follow-up response. Yet, G.E.N.I.U.S. is remarkably easy to program and build.

When G.E.N.I.U.S. was tested in a first year programming class, it was marginally successful. It worked robustly and the students in the class were in fact occasionally helped to solve their problems. However, the illusion of expertise that is essential to maintaining G.E.N.I.U.S.'s credibility crumbled when students talked to one another and discovered the system's true state of ignorance. At least G.E.N.I.U.S. was tested: too few tutorial dialogue modelling systems have actually been stress tested by the harsh users in the real world. This is because most of the experimental work in tutorial dialogue ignores surface language in order to make headway on the deeper pragmatics issues. G.E.N.I.U.S., with its end-run around the need for deep knowledge, was able to achieve surface performance that was of enough quality to allow real world testing.

Fundamentally, G.E.N.I.U.S. takes advantage of idiosyncracies in the human-tutor interaction to offload onto the student the responsibility for understanding the conversation and learning the domain. In this sense, it is a situated agent (see Clancey's chapters in this volume) since it depends on there being an appropriate understanding of the social context for the system work smoothly. The lesson of the G.E.N.I.U.S. approach is that sometimes it may be possible to

engage in useful tutorial dialogues with no deep understanding of the domain, or of the student. All that may be needed is an understanding of the social context.

In concluding this section, it is worthy of note that all of the systems discussed are focussed at the pragmatics level. That's where the action is in tutorial dialogue research. Unlike many other areas of natural language understanding, the pragmatics level is central and cannot be avoided. That's why modelling tutorial dialogue is hard, but that's also why it is interesting because it is in pragmatics that many of the main unsolved linguistics questions still lie.

Conclusion

What are the general lessons to take out of the research into tutorial dialogue? First, tutorial dialogue will become a more central topic in intelligent tutoring systems research if styles of tutoring such as collaborative learning and negotiated tutoring become widespread. These styles take John Seely Brown's admonition about the centrality of dialogue very seriously. Dialogue between the learner and the system in order to negotiate learning goals, and dialogue with collaborative learners are definitional features of these styles of tutoring.

Another important lesson is that tutorial dialogue can make use of standard AI natural language techniques. Research at all three linguistic levels can be adapted for use in modelling tutorial dialogue. Parsing techniques to handle the syntax, semantic grammars for meaning analysis, and speech act planning, to name just a few ideas, can all be applied. However, tutorial dialogue confronts standard natural language techniques with difficult new problems, and requires the solution to numerous pragmatics problems that can be downplayed in standard natural language approaches.

In the near term, most tutorial dialogue systems will be quite limited and will continue to rely on templates and massaged input in order to make progress on the deeper pragmatics issues. Occasional end runs, such as G.E.N.I.U.S., will allow practical tutorial dialogue systems to actually be constructed and used in the real world. In the middle term, interesting integration should take place between the natural language component and graphics interface so that some combination of pointing and speaking can enhance learner-tutor interaction. A full solution of the problems confronting tutorial dialogue will not happen in the foreseeable future. If this solution does happen, it will mean that the full natural language understanding problem will also be solved, for any domain, not just the tutoring domain.

References

- Allen, J.F. & Perrault, C.R. (1980). Analyzing intention in dialogues. In *Artificial Intelligence*, 15 (3), 143-178.
- Austin, J.L. (1962). *How to Do Things with Words*. Oxford: Oxford University Press.
- Baker, M. (1989). A model for tutorial dialogues based on critical argument. In *Proc. of 4th Int. Conference on AI and Education*, Amsterdam, Netherlands, 2-8.
- Barwise, J. & Perry, J. (1983). *Situations and Attitudes*. Cambridge: MIT Press.
- Bruce, B. (1975). *Belief Systems and Language Understanding*. BBN Report #2973, Bolt, Beranek, and Newman, Inc., Cambridge, Mass.
- Brown, J.S., Burton, R.R., & deKleer, J. (1982). Pedagogical, natural language, and knowledge engineering techniques in SOPHIE I, II, and III. In D. Sleeman and J.S. Brown (eds.), *Intelligent Tutoring Systems*. London: Academic Press, 227-282.
- Carbonell, J. (1970). AI in CAI: an artificial intelligence approach to computer-aided instruction. *IEEE Trans. on Man-Machine Systems*, 11 (4), 190-202.

- Chafe, W. (1970). *Meaning and the Structure of Language*. Chicago: University of Chicago Press.
- Cohen, P.R. (1978). *On Knowing What to Say: Planning Speech Acts*. TR 118, Department of Computer Science, U. of Toronto.
- Fillmore, C. (1968). The case for case. In E. Bach and R. Harms (eds.), *Universals in Linguistic Theory*. New York: Holt, Rinehart, and Winston, 1-88.
- Grice, H.P. (1975). Logic and conversation. In P. Cole & J.L. Morgan (eds.), *Studies in Syntax, Vol. III*. New York: Seminar Press, 41-58.
- Gutwin, C. & McCalla, G.I. (1992a). The use of pedagogic misrepresentation in tutorial dialogue. In *Proceedings of the Second International Conference on Intelligent Tutoring Systems*. Montreal, Quebec, 507-514.
- Gutwin, C. & McCalla, G.I. (1992b). Would I lie to you? Modelling misrepresentation and context in dialogue. In *Proceedings of the 30th Conference of the Association for Computational Linguistics*, U. of Delaware, Newark, Delaware, 152-158.
- Hendrix, G.G., Sacerdoti, E.D., Sagalowicz, D., & Slocum, J. (1978). Developing a natural language interface to complex data. *ACM Transactions on Database Systems*, 3 (2), 105-147.
- Kurup, M., Greer, J., & McCalla, G.I. (1992). The Fawly article tutor. In *Proc. of the Second International Conference on Intelligent Tutoring Systems*, Montreal, Quebec, 84-91.
- McCalla, G. I. & Murtagh, K. (1991). G.E.N.I.U.S.: An experiment in ignorance-based automated program advising. *AISB Newsletter 75*. United Kingdom, 13-20.
- Minsky, M. (1975). A framework for representing knowledge. In P. Winston (ed.), *The Psychology of Computer Vision*. New York: McGraw-Hill.
- Moyse, R. & Elsom-Cook, M. (eds.) (1992). *Knowledge Negotiation*. London: Academic Press.
- Petrie-Brown, A. (1989). Intelligent tutoring dialogue: the structure of an interaction. In *Proc. of 4th International Conference on AI and Education*, Amsterdam, 195-201.
- Schank, R.C. (1972). Conceptual dependency: a theory of natural language understanding. *J. of Cognitive Psychology*, 3 (4), 552-631.
- Schank, R.C. & Abelson, R.P. (1977). *Scripts, Plans, Goals, and Understanding*. Hillsdale, New Jersey: Erlbaum.
- Schwarz, M. & Yazdani, M. (eds.) (1992). *Intelligent Tutoring Systems for Foreign Language Learning*. Berlin: Springer-Verlag.
- Searle, J.R. (1969). *Speech Acts: An Essay in the Philosophy of Language*. Cambridge: Cambridge University Press.
- Wasson (Brecht), B. (1990). *Determining the Focus for Instruction: Content Planning for Intelligent Tutoring Systems*. Ph.D. Thesis, Department of Computational Science, University of Saskatchewan, Saskatoon, Canada.
- Winograd, T. (1972). *Understanding Natural Language*. New York: Academic Press.
- Woods, W.A. (1970). Transition network grammars for natural language analysis. *C. of the ACM*, 13, 591-606.
- Woolf, B. & McDonald, D. (1984). Context-dependent transitions in tutoring discourse. In *Proc. 1984 Conference of the American Association for Artificial Intelligence*, Austin, Texas, 355-361.

Acknowledgements: I would like to thank the organizers of the 1993 International Conference on Computers in Education, in particular Tak Wai Chan and John Self, for asking me to participate as an invited speaker and tutorial presenter. I also want to acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada for my research.

