category: **THEORY PAPER**

Taking the Initiative for System Goals in Cooperative Dialogue[1]

Kevin Matthews

CUCS-150-85

Kevin Matthews

Department of Computer Science
Columbia University
New York, New York 10027

Abstract

We discuss the need for natural language systems to have a more flexible, conversational ability to interact with the user particularly for problem-solving dialogue. In particular, such systems need to have their own goals in addition to responding to the user's requests and the linguistic ability to manage such goals in the dialogue. We discuss some motivations for taking the initiative, the information necessary to do it, and present a partial solution to a subproblem. The approach is exemplified by output from our experimental dialogue ADVISOR which functions as an academic advisor to students.

[1]This work was partially supported by ONR grant N00014-82-k-0256

Introduction

In order for natural language interfaces to computer systems such as database, cai, help, and expert systems, to be more effective they need to develop more conversational abilities. Human/human problem-solving dialogues are characterized by a more flexible, give-and-take interplay between the participants than is true of the typical examples of these systems. In particular, it is seldom the case in natural dialogue that one conversant asks all the questions while the other provides all the answers. In this paper we consider what would be necessary to enable an automated advice-giving system to shift from respondent to the more active, questioning position in a dialogue, some reasons why it might be desirable for a system to do so, and how it might accomplish such a maneuver, a problem which has not previously been addressed.

We propose a solution for initiative-taking as part of a system that carries on a dialogue as a cooperative problem-solver. Such a system will have goals of its own to fulfill in addition to its intention to respond to the user. For instance, a medical advisor such as MYCIN, in addition to its intention to answer the user's questions, might have a goal of determining that all of the tests relevant to the patient in question have been administered. We assume that it is desirable to permit the user to guide the discussion as much as possible. When it happens however, that a goal of the user comes into conflict with one of the system's goals the system should have the ability to take the active role to advise the user of the apparent conflict. In addition, the system ought to be able to decide whether the signs of a conflict in goals are strong enough to indicate that the system should completely take control of the dialogue and address the conflict, continue in its present role of replying to the user's request while alerting the user to the potential problem, or of ignoring the conflict and simply continuing to respond to the user's request. A system which was unable to scale its response in this manner would run the risk of frustrating the user by taking control too frequently on the one hand or of failing to fulfill its Gricean responsibility of completeness on the other. We present a method for detecting one type of such goal conflict and choosing among the alternative responses to it. We also include details of the implementation in an experimental dialogue system.

We chose student advising as a domain for problem-solving because it is rich in the kind of flexible, conversational behavior we wish to simulate. Our approach was first to collect and examine transcripts of student/advisor dialogues. We then implemented the model of discourse behavior in an experimental system system, ADVISOR, which functions as a faculty advisor of undergraduates. ADVISOR is designed basically as a question-answering system, but with an additional component, the discourse manager, interposed between the parser and the surface generator whose function is to manage shifts in dialogue roles.

Related Work

Previous work in natural language relates to ours in several ways. There are several experimental front-ends to databases, that have some ability to take the initiative and ask questions of the user. These include Codd's RENDEZVOUS [78] and Pazzani's KNOBS system [83], both frame-based systems. While these systems can generate requests to the user for the additional information necessary to form a successful query against the database, they cannot vary their response depending upon circumstances and their interventions are restricted to domain-level inquiries alone and not communication-level goals.

Our work presupposes an ability to monitor and effect changes in the topic along the lines suggested by the work of Grosz [77] and by the concept of *context spaces* in the work of Reichman [78]. Grosz and Reichman did not consider changes in the roles of initiator and reactor.

The Roles of Initiator and Reactor

The speaker's choice during his turn is between two linguistic roles which we distinguish as follows:

1. **initiator**- controls the conversation during a segment of dialogue, by asking questions, requesting information, or by informing as stage-setting for either of the previous goals. The dominant expectation is that the other conversant in the dialogue will respond to the direction supplied by the initiator. In example 1 the student is the initiator from the beginning until the professor's second turn when the latter takes the initiator role.

2. **reactor**- responds to the questions or requests for information from the

initiator or makes back-channel responses that indicate his continuation of the reactor role without contributing content (e.g. "mmhmm").

It is important to note that these roles may change independently of turn and topic changes. In example 2 below, for example, the advisor changes the topic and his role within his last turn from reactor to initiator. Other examples from our data include cases in which a conversant preserves his initiator or reactor role over several turns, introduces a new topic without changing his role, or preserves the same topic over a role change.

Motivations for Taking the Initiative

There are a number of reasons for a cooperative, problem-solving system such as an automated advisor to take the initiative. These include

- clarification: to ask a question to resolve a pronoun reference, for example. Occurs in turn 4 of example 1 below.
- request for additional information: as, in reply to an underspecified query from the user.
- correct a misconception: system should correct misconception implied in user's question.
- redirect discussion: to introduce a new topic or problem that the user should know about. Occurs in turn 4 of example 2 below.
- identify conflict between user's goal and system's: for instance, in advising domain, student may wish to pursue some action the system knows would hurt the student's schedule.

STUDENT: ***um yes. I didn't take Fortran language.***

PROF: ***umhm.***

STUDENT: ***I only took Pascal. If I take that would I get a credit for it? It's a beginning course.***

PROF: ***To take another 1000?***

STUDENT: ***uh huh***

PROF: ***no***

STUDENT: ***no?***

PROF: *no*

Example 1, from transcripts
Taking Initiative for Clarification

STUDENT: *ok um, how about computer arts-video games which one do you think will be better?*

PROF: *Well it depends on what you are interested in. Computer arts-video games actually it's an introduction to computer graphics.*

STUDENT: *umhm*

PROF: *And if you are interested in computer graphics this is your chance to learn about it. Uh, if you are interested in computer networks you should take the other one.*
[pause]
Let me first check up. Do you have all your prerequisites? Let's go through the list:
You have calc I,...

Example 2, from the transcripts
Taking Initiative to Redirect Discussion

The last two motivations may both be considered taking the initiative because of a conflict in goals. In redirecting discussion the intervention by the system is likely to be more extended than when it detects a conflict or requires clarification. In this paper we will present a partial solution to the problem of taking the initiative because of a conflict in goals about domain-level actions.

Information Needed to Manage Roles

In order to manage shifts in the initiator and reactor roles the system needs to maintain the following information which together we call the "discourse position" of a conversant:

1. **role**: which may be one of either the initiator or the responder. This information is necessary because if one of the system's plans calls for asking a question or introducing a new topic, for instance, the system must choose the appropriate linguistic means depending on which role it is currently filling.

2. **topic**: by this we mean a local subtopic whose change may be marked by the linguistic means suggested by Reichman [78], for instance. The advising system needs to know the topic because if its new goal involves a shift in topic it might need to signal the shift linguistically to preserve coherence.

3. **current speech act**: this is the goal of the utterance the conversant is currently forming and may be to answer a question, to ask a question, or other speech act(s). The current speech act may come from the default associated with the role (e.g. "answer" for the reactor) or be provided by some knowledge based-component in the system.

4. **domain-goal**: if the conversant is the initiator he may also have some extended domain-level goal which he is pursuing of which the current speech act forms a single step. For an advisor such an extended domain-goal might be to ask a series of questions to determine which courses the student has already taken, for example.

In addition to the discourse position the system has a set of goals or plans of its own, the activation of which provide the stimulus for taking the initiative. There are two types of these. An ***opportunistic plan*** which is triggered by some aspect of the last utterance of the user that calls for action by the advising system. A pronoun reference that cannot be resolved would activate a plan to ask a clarification question, for instance. The implemented example in the next section is another case. The system may also have one or more ***session-level plans*** which are goals which it is committed to achieve during the session regardless of which issues the user raises. In the student advising domain a system might have a goal of verifying that the student has a workable schedule for the coming semester which it will address even if the student does not.

Responding to a Conflict in Goals- An Example

We now consider an example from the operation of ADVISOR in which the system detects a conflict between the user's goal and its own by means of an opportunistic system plan. The system, initially in the role of reactor responding to the user's questions, must decide among its options of continuing to answer the questions, answering while pointing out the conflict, or taking the initiative and addressing the conflict directly. On the basis of an evaluation of the strength of its belief in the violation it will choose the appropriate one among a set of responses to becomes its new current speech act. Here we have assumed a segment of discourse about a single topic and so have not implemented a method of monitoring and shifting topic. In this example ADVISOR does not have a domain-goal.

The relevant system resources utilized in this mini-dialogue include an opportunistic plan called *check-prerequisites* which attempts to verify that a student has satisfied the prerequisites for a course that he appears to be interested in taking. In order to be able to make inferences about such things as when a student intends to take a course without saying so explicitly the system has a database of likely student plans to which it applies an inferencing method after Perrault and Allen [80]. This inferencing method has been extended to produce inferences which are sometimes definite as well as the plausible ones produced by Perrault and Allen's rules. (For a fuller description of this aspect of the system see McKeown et al. [85].) For checking prerequisites it has a list of courses the student is known to have taken although it does not assume the list to be complete. Depending on the contents of the list ADVISOR can decide that the belief that a student has not taken a particular prerequisite course, i.e. the "violation," is definite or plausible.

Since ADVISOR'S belief that the student is actually pursuing the inferred plan and the belief that he has violated a precondition of the plan can both be either "definite' or "plausible" the various possibilities can be ordered in this way from weakest to strongest:

Case I: plan = plausible, violation = plausible < **Case II: plan = definite, violation= plausible**

< **Case III: plan = plausible, violation = definite** < **Case IV: plan = definite, violation = definite**

Figure 1: Ordering of Intervention Cases According to Strength of Belief

The available responses can also be ordered by strength as follows:

Answer only < Answer + Question < Question only

< Answer + Warning < Warning only

Figure 2: Ordering of Possible Responses

Our solution is then to map the responses to the cases in the following manner:

Case I: Answer only
Case II: Answer + question *or* Question only
Case III: Answer + warning
Case IV: Warning only

Figure 3: Responses Tailored to Situation

Examples for each of the cases have been implemented. Figure 4 shows ADVISOR'S output for an example of CASE II. Initially ADVISOR'S discourse position shows that it is in the role of reactor and its (default) current speech act is to answer. It has no current domain-goal and the topic is (and will continue to be) prerequisites. The student's question about who is teaching nlp would be sufficient for the system to infer that the student plausibly has the goal of selecting nlp. Since he announces this goal explicitly it becomes definite. When the student's plan of selecting a course is inferred ADVISOR'S opportunistic plan check-prerequisites fires. The check-prerequisites plan tries to determine whether, in this case, the student has taken ai, the prerequisite to nlp. It can not determine whether he has or not and so marks the violation, the failure to take ai, as plausible. It can now determine that the situation corresponds to CASE II and chooses the answer + question option for ADVISOR'S next current speech act. Since its new current speech act involves asking a question the system must now take the initiative and so marks the role slot in its new discourse position as initiator and updates the current speech act slot. The other slots do not change. The current speech act would then be passed to the surface generator for transformation into English. (Since the interface between the discourse manager and surface generator has not yet been established ADVISOR'S only output in this case is a deep structure representation of the response for which we have supplied a gloss in Figure 4.)

ADVISOR's prior knowledge: list of courses taken by student does not include artificial intelligence which is the prerequisite for nlp.

ADVISOR's current discourse position:
(role reactor)
(topic prereqs)
(current-speech-act answer)
(domain-goal nil)

student: I want to take nlp. Who is teaching nlp this semester?

inferred goal: (select c:nlp)
strength of belief in inferred goal: definite
strength of belief in violation of prerequisite: plausible

ADVISOR'S new discourse position:
(role initiator)
(topic prereqs)
(current-speech-act ((answer) (conj but)
(askif (taken (agent user)
(object c:ai)))))
(domain-goal nil)

ADVISOR: ((answer) (conj but) (askif (taken (agent user)
(object c:ai))))

;; Mckeown, but have you taken ai?

Figure 4: ADVISOR system output
Showing Taking of Initiative: Case II

By contrast if the student had asked simply, "Who is teaching nlp this semester?" The same inferences about plan and violation would have occurred, but with a belief-level in each case of plausible and a response including a question about prerequisites would be inappropriate. ADVISOR would therefore only answer the question.

Future Work

Having shown an example of initiative-taking for an opportunistic plan, we would like to add one or more session-level plans. Such a plan will determine when and how the system should take over the dialogue in a more extended way.

In addition to taking the initiative the model of role handling should be developed toward complete manipulation of the initiator and reactor roles including seizure, maintenance, surrender, and abandonment of each role.

Conclusion

We have discussed the desirability of nl systems that can be flexible enough to engage in dialogue with the user both by responding to the user and by addressing its own goals. For this purpose such a system needs the ability to change its role from responder to initiator. An example of a conflict in goals was presented from the output of an automated advisor and a student user in which the

system demonstrates its ability to recognize the conflict and grade its response according to the apparent severity of situation. Thus, we have presented the information needed for the change in linguistic roles, some motivations for doing so, and a means that the system can use to accomplish it.

Acknowledgments

The author wishes to express his gratitude to Prof. Kathleen McKeown, Columbia University, for the benefit of her continuing guidance throughout this and related work, and to Dr. Myron Wish, A.T. & T. Bell Laboratories, Murry Hill, NJ, for helpful discussions on an earlier version of this paper.

References

Allen, J.F., "Argot: A System Overview", Comp. & Maths. with Appl. Vol. 9, No. 1, pp. 97-109, 1983

Carberry, S., "Tracking User Goals in An Information Seeking Environment", *Proc. AAAI*, 1983

Carbonell, J.G., "Subjective Understanding: Computer Models of Belief Systems", UMI Research Press, Ann Arbor, Michigan 1981

Codd, E. F., et al., "Rendezvous Version 1: An experimental English-Language Query Formulation System for Casual Users of Relational Databases, IBM Research Laboratory, San Jose, Ca., Tech. Report RJ2144(29407), 1978

Grosz, B. J., The Representation and use of focus in dialogue understanding. Technical note 151, Stanford Research Institute, Menlo Park, Ca. (1977)

Kaplan, S. J., "Cooperative Responses from a Portable Natural Language Darabase Query System.", Ph.D. Dissertation, U. of Penn., Phila, Pa., 1979

Lehnert, W.G., *The Process of Question Answering*, Lawrence Erlbaum Associates , Hillsdale, NJ, 1977

Mckeown, K., Wish, M. & Matthews, K., "Tailoring Explanations for the User", submitted to IJCAI '85

Pazzani, M.J., "Interactive Script Instantiation", *Proc. AAAI*,

1983

Perrault, C. R., Allen, J.F. "A Plan-Based Analysis of Indeirect Speech Acts", Am. Journal of Comp. Ling., Vol. 6, No. 3-4, July-Dec. 1980

Reichman, R., "Plain Speaking: A Theory and Grammar of Spontaneous Discourse", Report No. 4681, Bolt Beranek and Newman Inc. June 1981

Reichman, R., "Conversational Coherency", Tech. Report #95., Harvard University and Bolt Beranek and Newmand Inc., 1978

Schank, R.C. and Wendy Lehnert, "The Conceptual Content of Conversation", Research Report #160, Yale University, 1979

Webber, B.L. and Tim Finin, "In Response: Next Steps in Natural Language Interaction", to appear in W. Reitman (ed), *Artificial Intelligence Applications for Business*, Norwood NJ: Ablex Publ. Co., 1984