Adaptive Conversational Agents for Task-Oriented Dialogues

Cynthia A. Thompson
Center for the Study of Language and Information
Stanford University
Stanford, CA 94305-4115
cthomp@csli.stanford.edu

Abstract

This paper describes ongoing research on an adaptive conversational agent supporting task-oriented dialogues. The system will use spoken input, and spoken and screen-based output, with the addition of other modalities over time. The issues addressed by this research include adjusting dialogue management to suite an individual user's needs, viewing dialogue as the interactive construction of database queries, and effectively managing dialogues for such interfaces. Two demonstration tasks are being developed, one for destination recommendation, specifically restaurants, and the other for finding information and locations in a library setting.

As growing numbers of users interact with the World Wide Web, more of its inadequacies as a method to efficiently access information and solve problems become apparent. The specific interface improvements addressed by this paper are the addition of more natural human-computer interaction through human language, and system adaptation, over time, to the preferences of an individual user. The agent described here integrates natural language understanding technology with an adaptive ability to reason about and carry out dialogues. The goal is not just to allow users to engage in conversation, but to allow them to achieve some goal with their conversation. The adaptive component makes this process more efficient over time.

For example, a user may want to retrieve a recommendation for a restaurant. Current Web interfaces force a keyword or other highly constrained search. An adaptive conversational interface, on the other hand, would allow the user to quickly narrow down his options in a more natural way, and would become more efficient at doing so as it learns a particular user's preferences. As this example illustrates, the types of tasks we are examining are those for which a large amount of information is available, but for which the user does not know the organization or extent of the information, but instead just some characteristics of the information, such as type of food, that she wants to retrieve. Figure 1 shows an example of a conversation that might take place in this domain.

There has been some pioneering work on conversational interfaces for simple tasks in limited domains (Seneff et al., 1996; Allen et al., 1995; Dowding et al., 1993). These systems have reached the point where they are fairly robust for conversations that fit

1. `U:` Where should I eat today?
2. `S:` What type of cuisine would you like?
3. `U:` What are my choices?
4. `S:` You can pick from types of food like Chinese, Indian, or Mexican.
5. `U:` Chinese.
6. `S:` What quality rating would you like?
7. `U:` I don't care, as long as it's cheap.
8. `S:` How do you want to pay?
9. `U:` Let me have Indian instead.
10. `S:` Ok, switching from Chinese to Indian.
11. What city do you prefer?
12. `U:` How about Berkeley.
13. `S:` There are 4 cheap Indian restaurants in Berkeley, would you like the name of one?

...

Figure 1: Interaction Example

their design, and we plan to build on this work. Our work expands on this work by proposing an interactive model for task-oriented dialogues, by combining reactive and deliberative planning to manage those dialogues, and by allowing unobtrusive adaptivity based on past interactions with a user. Two demonstration tasks are being developed, one for destination recommendation, specifically restaurants, and the other for finding information and locations in a library setting.

1 System Overview

As with the example of the Web, our system is the intermediary between the user and large amounts of information. Instead of the typical interface which forces users to either step through a broad and deep taxonomy or wade through many tens of answers to a keyword search, our interface will carry out a conversation until it finds a manageable set of results that are acceptable to the user. Specifically, the system will ask questions about the type of information that the user is interested in, and the user's replies then help the system to narrow in on the satisfying results. Our approach is to view the dialogue process as an interactive, cooperative database search, in which the participants narrow in on information of interest, as in Figure 1.

The system components are illustrated in Figure 2. The speech recognition component is an off-the-shelf recognizer capable of speaker-independent, continuous recognition. After a user utterance has been recognized, the modules described below determine the best response, and the system replies with a question, displayed or spoken result, or the answer to questions that the user might ask. No robust natural language generation is currently planned, but we are instead using canned text and predesigned screens.

1.1 Parser

The output of the speech recognizer is passed to the parser. Most of the time, parsing will be easy, since utterances will be answers to system queries. However, since we

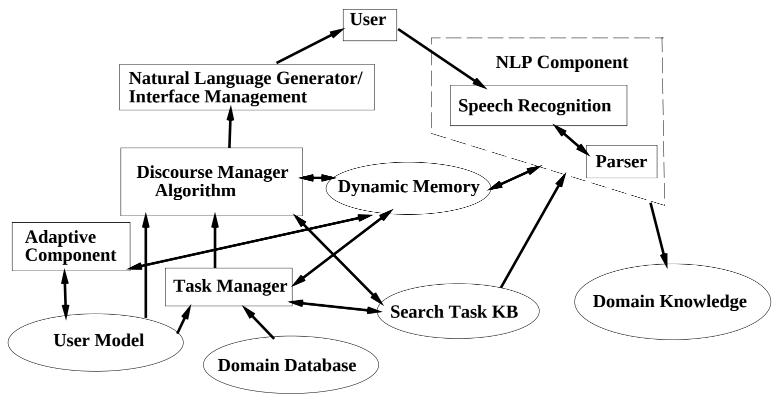

Figure 2: System Organization

Act	Definition	Line #
Accept	User accepts the feature or value suggested by the system.	5, 12
Reject	User rejects a system suggestion.	7, 9
Query	User requests information about the content of the database or the state of the task.	1, 3
Suggest	User fills in information without regard for system suggestions.	7
Request	User asks for the results of the search.	None
Modify	User changes a previously chosen value, or completely eliminates a value for a feature.	9

Table 1: User Speech Acts

also want to allow the user to control the conversation, more robust parsing will be needed. To support this, we will expand on parsing methods developed at CSLI, based on Head-Driven Phrase Structure Grammar (Pollard & Sag, 1994). The associated Linguistic Grammars Online (LinGO) project is building a computational platform to support this theory, and earlier versions have already been developed in support of the Verbmobil translation project (J. Alexandersson, 1998). We will adapt this platform with a specialized grammar and vocabulary to produce output that is compatible with the rest of our system.

Since LinGO is not built to handle the unconstrained language of spoken utterances or errors due to speech recognition, backup modules will be used. We are expanding on previous work on learning grammars for database queries (Zelle & Mooney, 1996) to make the learned grammars more robust and sensitive to context, so that this more robust parser can be used to create partial or ranked parses when LinGO breaks down.

Part of the parser's task is to determine the user's "speech act," which indicates her aim, or intention, for an utterance. That is, for each sentence the system must determine whether the user is asking a question, stating an answer, or giving a command. As shown in Figure 2, the parser will use the current context, held in dynamic memory, and also domain specific knowledge, to help determine the user's speech act. Table 1 shows some of the speech acts that the user may make, together with a pointer to the line in Figure 1 in which that type of utterance is made.

1.2 Discourse Manager Algorithm

The discourse manager algorithm is the heart of the agent, as it is responsible for determining and coordinating each response to the user's input. The progression of a task-oriented dialogue can be usefully viewed as transitions in two distinct search spaces, the task space and the discourse space. The task space describes the operators for progressing on the task towards a solution that satisfies the user. The discourse space describes the coherent system-user interactions needed to decide on which task operators to pursue. This distinction is general enough to manage many types of task-oriented dialogues.

While typically we expect the user to cooperatively answer system queries, at times she will want more control. For example, she might ignore a question, ask for other options, or ask for examples of the current search results. To handle this, the dialogue manager uses a combination of deliberative and reactive planning to respond appropriately to the user. We will use ICARUS (Langley, 1997a), an architecture for intelligent agents, to manage the dialogue. ICARUS combines reactive and deliberative ability, making it ideal for modeling both abstract, conceptual plans, and for handling the faster calculations needed when the user abruptly changes his focus during a conversation.

ICARUS will use several sources of information to determine what transitions to make in the task or discourse space, and then makes a call to the appropriate module to make that transition. One source of this information is the task and discourse context information, and another is the search task knowledge base, which holds *intentions* and *protocols*. Intentions, or goals, specify movements through the search space that the user or system might make at any given time. These intentions might activate one or more protocols that attempt to carry out the actions and subdialogues mapped out in an intention. ICARUS determines when an intention is relevant and it then becomes active. For example, to constrain the search, the system may request that the user provide a value for a specified feature. If the user refuses to provide the value, a new intention may need to become activated. In sum, our dialogue management design incorporates elements from multi-agent communication techniques, speech act theory, and a mix of deliberative and reactive planning.

1.3 Task Manager

The task manager is the intermediary between the domain knowledge base, which is the source of information that the user is searching, and the discourse manager. It is the job of the task manager to take an appropriate database query action and to suggest the next feature or value to query the user about, if any. This suggestion is based both on the user model and on the feature most likely to reduce the search space quickly. The latter is determined by information-theoretic measures including the size of the search result and the number of data items satisfying each feature. The task manager will also handle user queries about the domain being searched, the system's capabilities, and other knowledge-level information.

1.4 Adaptive Component

Because people differ in their standard operating procedures, we intend to include an adaptive component in our conversational agent. Automatically learning user models is a growing sub-field within both machine learning and user modeling. However, training data for learning systems is often difficult to acquire in any useful form. Our approach avoids the latter issue by unobtrusively monitoring interactions with the user. Instead of asking generic questions about the user's preferences, it tracks a user's interaction with the system and uses that to infer their preferences.

Most past adaptive systems for recommendation and information filtering model user preferences at the item level, and thus require user feedback about complete items such as movies or books. Our approach instead models the user in terms of the questions she finds useful and her likely responses. The aim of adaptation is not to better rank hundreds of items, but rather to make conversations more efficient and effective. Our adaptive interface will quickly learn that some questions are irrelevant and others unnecessary, and thus focus on ones that provide constraints most useful for most effectively managing the conversation with the user.

Our approach to user modeling takes advantage of trace data collected during the course of each conversation. This knowledge will then be used in future interactions with that user to predict his questions and responses to system queries. Briefly, the system will store each question the user accepts from the system, along with the questions and answers from earlier in that dialogue. Our first prototype uses a case-based classifier, though we are not tied to this method and will employ other methods as required. At each step in a new dialogue, when it must decide on a question to ask the user, the system invokes a simple case-based mechanism to find the question with the most similar historical context. Similarly, the system stores the user's answer to each question, along with the questions and answers that preceded it, then retrieves the best match when selecting a recommended answer in a new dialogue. In this way, we integrate user modeling with dialogue management: the learned information will influence the progression of the dialogue for an individual user. For example, after several interactions such as that in Figure 1, where the user rejects questions about quality, the system will avoid asking that question.

1.5 Evaluation Plan

We plan to evaluate our system on its ability to support two rather distinct tasks. The first, a destination advice task, will be rather limited in the range of dialogues allowed, but will demonstrate the basic ideas of cooperative database search, simple dialogue management, and adaptive ability. An initial adaptive prototype has already been implemented for this domain. The second task is information seeking in a library domain, with the goal of reducing the load of human reference librarians. An initial interface with speech interaction has been implemented for this domain. Both prototypes avoid the issues of natural language understanding and dialogue management. Further, an evaluation of the user modeling component has not yet been performed.

To evaluate the discourse management component, we will measure the ability of the basic interface to understand sentences and answer questions correctly (e.g., (Zelle & Mooney, 1996)). A second measure, more qualitative, will be on the coherence of the conversational interactions. To evaluate adaptivity, we will first connect the adaptive component to a menu-based GUI, initially keeping its evaluation distinct from the conversational aspects. The average number of interactions required to complete a task will be measured with and without the adaptive component.

After the separate evaluations, the two components will be united and evaluated as one unit. The addition of the adaptive module to the conversational interface should result in even more effective dialogues. We will compare this increase in effectiveness with the increase achieved using the adaptive GUI alone.

2 Related Work

The TRAINS project at Rochester (Allen et al., 1995) is one of the most ambitious and well known related efforts in implemented conversational interfaces. It focuses on the development of an intelligent planning assistant that converses with users in spoken natural language. While also capable of mixed initiative dialogues, the system is not adaptive to individual users. The system is targeted towards assisting a person in planning activities, as opposed to the search-oriented tasks addressed here.

The effort discussed by (Smith & Hipp, 1994) is also closely related. They discuss a system for spoken interaction, focusing on task-oriented dialogues. They also use independent reasoning modules for task assistance and dialogue processing, but instead connect them with what they call the "Missing Axiom Theory" for language use. For example, the detection of a missing axiom in a proof attempt for a sub-dialog initiates a dialogue interaction. However, their system is not truly mixed initiative, and is demonstrated in only one domain. Other systems for spoken language applications include (Seneff et al., 1996) and (Dowding et al., 1993). However, these systems primarily focus on the language understanding portion of conversational interfaces, and do not demonstrate them in the context of an adaptive interface.

Early work on user modeling for interactive computer systems was based on predefined, hand-coded knowledge about users or groups of users. Learning these preferences has recently emerged as a more workable and flexible approach. Results have been demonstrated for recommendation systems and information filtering, among other applications (Pazzani, Muramatsu, & Billsus, 1996). (Smith & Hipp, 1994) do some limited user modeling, but only infer user knowledge over the course of a single interaction, not over the long term as we plan to do. (Langley, 1997b) gives a more thorough overview of adaptive user interfaces. The idea of using interaction with the user in an unobtrusive way is not new. Applied machine learning methods have been using this method for quite some time, but have not situated the work in the framework of a conversational interface.

3 Conclusion

In conclusion, we have described the design of an agent that can support the management of task-oriented dialogues in several settings, and that will adapt to user preferences. Prototypes have been developed for several pieces of the system, and are currently being integrated into a complete system. The core processing modules are flexible enough to plug in new discourse and sentence grammars for new domains. Planned evaluations will shed light on the usability, efficiency, and correctness of the resulting system.

4 Acknowledgments

This work was done in collaboration with Pat Langley, and expands on earlier work on dialogue management for a destination advisor by Renee Elio and Afsaneh Haddadi.

References

Allen, J., Schubert, L., Ferguson, G., Heeman, P., Hwang, C., Kato, T., Light, M., Martin, N., Miller, B., Poesio, M., & Traum, D. (1995). The TRAINS project: a case study in building a conversational planning agent. *Journal of Experimental and Theoretical Artificial Intelligence*, *7*, 7–48.

Dowding, J., Gawron, J., Appelt, D., Bear, J., Cherny, L., Moore, R., & Moran, D. (1993). Gemini: A natural language system for spoken-language understanding. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*.

J. Alexandersson, e. (1998). Dialogue acts in VERBMOBIL-2. Tech. rep. 226, DFKI Saarbrücken.

Langley, P. (1997a). Learning to sense selectively in physical domains. In *Proceedings of the First International conference on Autonomous agents* Marina del Rey, CA.

Langley, P. (1997b). Machine learning for adaptive user interfaces. In *Proceedings of the 21st German Annual Conference on Artificial Intelligence*, pp. 53–62 Freiburg, Germany.

Pazzani, M., Muramatsu, J., & Billsus, D. (1996). Syskill & Webert: Identifying interesting web sites. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pp. 54–61 Portland, OR.

Pollard, C., & Sag, I. (1994). *Head-driven Phrase Structure Grammar*. University of Chicago Press.

Seneff, S., Hurley, E., Lau, R., Pao, C., Schmid, P., & Zue, V. (1996). Galaxy-II: A reference architecture for conversational system development. In *Proceedings ICSLP 1996* Sydney, Australia.

Smith, R., & Hipp, D. (1994). *Spoken natural language dialog systems: a practical approach*. Oxford University Press, New York, NY.

Zelle, J. M., & Mooney, R. J. (1996). Learning to parse database queries using inductive logic programming. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence* Portland, OR.