SCoT: a Spoken Conversational Tutor

Karl Schultz, Brady Clark, Heather Pon-Barry, Elizabeth Owen Bratt, Stanley Peters

Center for the Study of Language and Information – Stanford University
Stanford, California 94305
{schultzk, bzack, ponbarry, ebratt, peters}@csli.stanford.edu

Abstract

We describe SCoT, a Spoken Conversational Tutor, which has been implemented in order to investigate the advantages of natural language in tutoring, especially spoken language. SCoT uses a generic architecture for conversational intelligence which has capabilities such as turn management and coordination of multi-modal input and output. SCoT also includes a set of domain independent tutorial recipes, a domain specific production-rule knowledge base, and many natural language components including a bi-directional grammar, a speech recognizer, and a text-to-speech synthesizer. SCoT leads a reflective tutorial discussion based on the details of a problem solving session with a real-time Navy shipboard damage control simulator. The tutor attempts to identify and remediate gaps in the student's understanding of damage control doctrine by decomposing its tutorial goals into dialogue acts, which are then acted on by the dialogue manager to facilitate the conversation.

Dialogue Management

Our overall framework attempts to separate the purely linguistic parts of dialogue interaction from the knowledge of tutoring. Thus, capabilities such as turn management, construction of a structured history of the dialogue, and appropriate use of discourse markers are part of the dialogue manager's general conversational intelligence. Our dialogue manager coordinates input from the user, interprets this input as a dialogue move(s), updates the dialogue context, and delivers speech and graphical output to the user according to a plan specified by the tutor.

The Architecture for Conversational Intelligence makes use of several recent ideas in dialogue modeling, described in detail in (Lemon 2001). It creates and updates an information state corresponding to a notion of dialogue context. Dialogue moves (e.g., an assertion) update information state and can be initiated by either the user or the system. A dialogue move might send a user response to the system, elicit an assertion by the system, or prompt a follow-up question.

The dialogue manager itself includes the following dynamically updated components: a *Dialogue Move Tree* containing a structured history of dialogue moves and 'threads', plus a list of 'active nodes' determining possible interpretations of user input; an *Activity Tree* representing a hierarchical structure of activities initiated by the system or the user; a *System Agenda* of issues to be raised; *Salience Groups* for objects referenced recently; and a *Modality Buffer* for storing gestures to be resolved.

Conversational Tutoring

The main purpose of SCoT is to investigate the value of natural language in tutoring, especially speech and multi-modal gesture. SCoT is a reflective tutor, and takes as input the details of a problem solving session from DC-Train (Wilkins 2000), a real-time Navy shipboard damage control simulator. This simulator is also speech-driven, taking advantage of many of the natural language components described below, though it has a dialogue management system that is much less complex. The initial plan for a dialogue is created by taking a DC-Train session and feeding an analysis of it from the knowledge representation into the tutorial recipes.

Tutoring Recipes

The tutorial recipes outline how to decompose a tutorial goal into other recipes or dialogue actions. They are written according to the scripting language set forth in Gruenstein (2002) and primarily consist of two groups: those corresponding to high level tutorial plans and those for responding to a specific student utterance. The recipes are responsible for creating an initial dialogue plan by examining a student's performance and identifying knowledge areas that need attention. They describe how to use information from the student assessment and the dialogue history to appropriately steer the conversation.

Knowledge Representation

The domain knowledge is represented as production rules acting on a working memory. The tutor can query this for procedural explanations about how to solve a problem. Justifications for each step in a solution are currently only derived from rule preconditions, and not a justification for each precondition. We are still working on how to use a more robust declarative structure for more descriptive justifications, especially for rule preconditions. In other words *why* a precondition exists, not merely *that* it exists.

Multi-modal Gesture

Gestures are a large part of typical human-to-human interaction and the more we allow a student and a tutor to interact within a common workspace the more effective the dialogue interaction becomes (Clark 1996). This capability also lifts a considerable burden from the tutor since the tutor doesn't always need to formulate a full textual description for everything it is trying to convey. Making use of a 3-D model of a Navy ship from DC-Train the tutor is able to point out locations on the ship and easily create a context for discussing a problem (e.g., a burning compartment). The student is also able to point and highlight within this display, but only at specified times.

Work in Progress

We are also working on a student model and a robust language generation component. The student model assigns a probability to each rule in the knowledge representation based on evidence seen during a problem solving session with DC-Train as well as evidence during dialogue. The language generation component takes a set of feature-value pairs which map onto particular logical forms and are then sent through Gemini (see Grammar section) to be turned into an utterance.

Speech Interaction

In creating any system which interacts with a user via natural language, especially spoken language, special attention is needed for the extra language components. The goal of accurate speech recognition dictates many of the developmental choices, particularly what phrases will be understood. To understand a user takes the combination of a powerful speech recognizer in addition to a robust parser, both of which depend on a hand-built grammar. There are fewer language restrictions for the text-to-speech, however creating a high quality audio voice takes a fair amount of effort. We discuss each of these below.

Grammar: Gemini

The Gemini NLP system (Dowding et al. 1993) uses a single unification grammar both for parsing strings of words into logical forms (LFs) and generating sentences from LF inputs, e.g., the question "What happened next?" has the LF: (ask(wh([past,happen]))). Therefore SCoT understands a student through deep semantic processing, which enables us to get a very precise and detailed understanding of student utterances. This is key to understanding the dialogue intentions of a student (e.g., a question; an answer).

Speech Recognition: Nuance

Another major aspect of understanding a user is the speech recognizer. Nuance takes the Gemini grammar and creates a recognition model of expected utterances using its built-in understanding of phonetics. The grammar, therefore, will bias the recognizer towards correct understanding by limiting the input to that which is pertinent to tutoring and the specific domain being tutored. However, this can have the unintended side effect of turning an out-of-grammar utterance into something in-grammar, which typically leads to problems. This is why a well engineered grammar is key to good language understanding.

Text-To-Speech: Festival and Festvox

We have seen from testing SCoT that likeability is highly correlated with the quality of the voice output, regardless of content. The Festival text-to-speech system can turn any text into audio, however it has too much of a 'computer-sounding' voice and at present lacks the clarity and subtle inflections of a real human voice. One solution we have to this is to use the Festvox add-in to Festival, which allows the developer to create a more natural sounding voice. It is a non-trivial process, which involves recording a large number of utterances within the desired domain. These are analyzed by a speech recognizer with their text representation, and a voice is created which can generate audio within the language coverage of utterances recorded.

Acknowledgements

This work is supported by the Department of the Navy under research grant N000140010660, a multidisciplinary university research initiative on natural language interaction with intelligent tutoring systems.

References

Lemon, O.; Gruenstein, A.; and Peters, S. 2002. Collaborative Activities and Multi-tasking in Dialogue Systems. *Traitement Automatique des Langues (TAL)*, 42(2):131-154, special issue on dialogue.

Gruenstein, A. 2002. Conversational Interfaces: A Domain-Independent Architecture for Task-Oriented Dialogues, M.S. Thesis, Symbolic Systems, Stanford Univ.

Dowding, J.; Gawron, J.; Appelt, D.; Bear, J.; Cherny, L.; Moore, R.C.; and Moran, D. 1993. *Gemini*: A Natural Language System for Spoken Language Understanding. *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*. Columbus, OH.

Wilkins, D. C.; and Sniezek, J. A. 2000. The DC-SCS Supervisory Control Systems for Ship Damage Control: Volume 1 - Design Overview, Knowledge Based Systems Report, UIUC-BI-KBS-2000-03, Beckman Institute, University of Illinois.

Clark, Herbert H. 1996. *Using Language*. Cambridge: Cambridge University Press.