Interactive Spoken Simulation Control and Conversational Tutoring

Karl Schultz, Brady Clark, Elizabeth Owen Bratt, Stanley Peters,
Heather Pon-Barry, Pucktada Treeratpituk, Zack Thomsen-Gray

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

David C Willdns, David M. Fried, Eugene Grois

Beckman Institute - University of Illinois
Urbana, Illinois 61801
{dcw,fried,e-grois}@uiuc.edu

Introduction

We describe how spoken dialogue interfaces make a simulation-based trainer and an intelligent tutoring system into a powerful package for Naval damage control education. An advanced simulator gives a student real-time experience that would otherwise be extremely costly to provide. The dialogue interface to the simulator uses a simple finite-state script to allow the user to issue orders and requests in a realistic way. Subsequently, the reflective tutor communicates to the student about their experience with the simulator entirely through a 'conversationally intelligent* dialogue system, with capabilities like topic management and coordination of multi-modal input and output.

Dialogue Systems

Dialogue systems can range from very simple finite-state scripts defining the bounds of interaction, to complicated plan-oriented agents that can have very complex conversational intelligence (Allen 2001). The two systems discussed below are among the extremes, one being very simple and the other containing more advanced forms of conversational intelligence, but both serving their purposes with a good degree of success.

DC-Train 4.1 Dialogue Manager

The shipboard Damage Control Trainer, or DC-Train, is a simulator of damage control aboard large Navy ships. There is a physical simulator of fire, flood, and smoke, as well as simulated shipboard damage control personnel. A student is placed in front of the simulator and presented with one or more damage control crises to solve. This involves giving orders to the various simulated agents, which should occur verbally not only for training purposes, but also to recreate the stressful nature of the job as accurately as possible.

To handle such interaction, it was possible to simulate the command-receiving agents using the simplest kind of dialogue system, a finite-state scripted system. Furthermore, only 3 states are needed: the agent is ready to receive a new command, the agent has an incomplete message and is awaiting a missing value(s), or the agent is actually issuing the command to the personnel that will carry it out.

The emphasis for the spoken interface to the simulator is on providing a realistic experience. Other benefits include not having to learn locations and organizations of simulator graphical menus when the spoken commands are already familiar, and gaining the speed of verbal interaction. We use clarification sub-dialogues to provide a way to address speech recognition problems, and to allow for building up complex commands incrementally.

SCOT Dialogue Manager

The Spoken Conversational Tutor (SCOT) is on the opposite end of dialogue-supporting capabilities from the simple finite-state script used in DC-Train. There is a more robust conversational intelligence (CI) at work in SCOT'S dialogue manager which is necessary to facilitate a prolonged conversation about damage control doctrine. Since the utterances here are not simply mapped into commands and parameters the dialogue structure becomes much more important. Topic management and coordination of multi-modal input and output (gestures) are among the improvements over a simple finite-state script to reach a more human-like level of conversation.

An annotated record of the student's performance with the DC-Train simulator is fed into SCOT, from which an initial plan is created. This plan is made by the tutoring component of SCOT, which is completely separate from the dialogue manager side. This extra 'planner' is one of the necessities with CI of topics and goals. The dialogue manager will only have limited abilities to manipulate the plan and attach a user utterance to the relevant place. It understands that there are topics and sub-topics, but the dialogue manager obviously has no knowledge of what the tutorial goals are or how they relate to each other. Dealing with these concerns is the job of an outside agent The separation of the CI from the tutoring knowledge allows the dialogue manager to be generic and usable for any domain, provided an outside agent contains the semantic understanding of the goals and topics.

The ability to reshape the current and future dialogue threads is a major advantage over a finite-state model. It creates an interactive style which comes much closer to humans, where topics can explored in more or less detail, or put aside for later, new topics can spring up at whim and old topics can never be brought back if it becomes unnecessary. The tutor agent monitors the conversation and makes these changes as it sees fit to suit the overall tutoring strategies for teaching and specific tactics applied to get points across.

This dialogue manager can also handle creating gestural output to the user and timing that with the speech, as well as interpreting gestural input from the user (in the form of mouse clicks). Gestures are a large part of typical human-to-human interaction (Clark 1996), and the more the student is able to interact with a common workspace which SCOT shares, the more natural-feeling and effective the interaction will be. Currently the student can only point at designated times, but ideally all mouse movement would be analyzed for pointing, just as all hand movements in conversation could be looked at for gestural meaning.

Voice Interaction

Of the major technical difficulties to overcome in creating a robust dialogue system the input and output is obviously one of the more pertinent. The concerns of good speech recognition are one of the limiting factors to how much variance in input is allowed. The audio side of speech output is somewhat less restrictive, but that varies based on how realistic the voice needs to be. However, both the input and output are governed by an underlying grammar which gives the system the ability to intelligently parse user input and create complex output using semantic constructs, or logical forms (LFs), rather than using canned phrases or sentences.

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. This enables us to give precise and reliable meaning representations which allow us to identify the discourse move types (e.g., a question) given a linguistic input or output; e.g., the question "What happened next?" has the LF: (ask(wh([past,happen]))).

Nuance

The Nuance speech recognition server takes a user utterance and a recognition model, which we compile directly from a Gemini grammar, and attempts to turn the utterance into text. The grammar plays a key role in defining the bias of the recognizer towards words and phrases within the current domain, as well as assuring that every recognized utterance has a corresponding LF. Limiting what a user can say has the bonus of significantly better recognition of expected utterances, but has the undesired effect of sometimes turning out-of-grammar phrases into in-grammar phrases, which would obviously cause problems. This is where a well engineered grammar plays an important role.

Festival and Festvox

The Festival text-to-speech system turns any text into audio output. How usable a speech-enabled system will be depends highly on how understandable the output is. There are a variety of voices to choose from when using Festival, but they are 'computer-sounding' and lack the clarity and subtle inflections of a real human voice. One solution to this is to use the Festvox add-in to Festival, which allows one to create a voice which sounds exactly like the person who creates it. This process involves recording a large number of phrases covering the range of words in the desired domain. These recordings are then analyzed along with the corresponding text, and a voice is compiled which uses the discovered human-sounding words to generate the audio output.

Acknowledgments

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

Allen, J.; Byron, D.; Dzikovska, M.; Ferguson, G.; Galescu, L.; and Stent, A. 2001. "Towards Conversational Human-Computer Interaction,'[1] *AI Magazine.*

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

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 ARPA Workshop on Human Language Technology.*