A tutorial dialogue system with unrestricted spoken input

A tutorial dialogue system with unrestricted spoken input

Peter Bell, Myroslava Dzikovska, Amy Isard

School of Informatics, University of Edinburgh, UK

{peter.bell,m.dzikovska,amy.isard}@ed.ac.uk

Abstract

We present our work in building a spoken language interface for a tutorial dialogue system. Our goal is to allow natural, unrestricted student interaction with the computer tutor, which has been shown to improve the student's learning gain, but presents challenges for speech recognition and spoken language understanding. Here we describe the system design, focusing on the components used for speech recognition.

Index Terms: spoken dialogue system, speech recognition, computer tutoring

1. Introduction

Most research in spoken dialogue systems has focussed on systems which are task-oriented, designed to help the user achieve some fixed goal in a minimum number of dialogue turns, often using a slot-filling paradigm. We believe that spoken dialogue systems could be deployed more widely in the domain of computer tutoring, where, in contrast, the primary aim is to maximise the student's learning gain from using the system.

A substantial body of research eg. [1] has shown that an effective tutoring technique is to encourage students to produce their own explanations and generally to talk more about the domain during problem-solving. This motivated the development of dialogue-based intelligent tutoring systems (ITS) which ask students open-response questions (rather than multiple-choice questions), and in particular explanation questions. However, to date such systems have largely been limited to using typed interactions; existing speech-enabled tutorial dialogue systems such as [2] have been constrained to small-vocabulary scenarios which restrict the student to a limited range of answers, and therefore restrict opportunities for self-explanation.

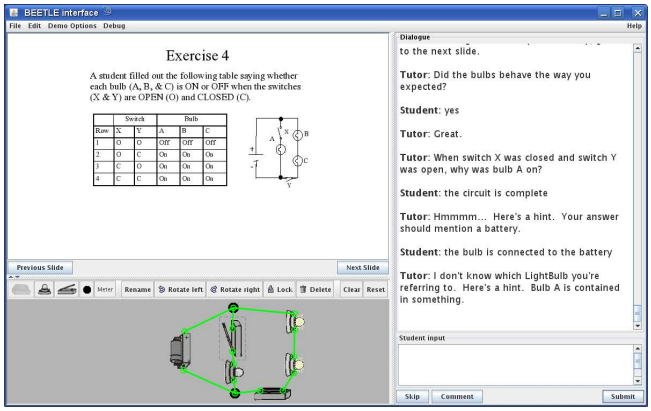

Figure 1: *Screenshot of the* BEETLE II *tutor showing text-based interaction*

Recently, typed systems attempted to encourage long, open-ended student answers through asking explanation questions and giving targeted feedback. While this approach was effective in some cases [3], there is evidence that in human-human tutoring spoken dialogue is more effective than typing [4].

This motivates our work in adding the capability for natural, unrestricted spoken interaction to BEETLE II, our existing typed tutorial dialogue system [5]. The BEETLE II system teaches students basic electricity and electronics by introducing them to the fundamental concepts, then challenging their understanding by asking them to predict the outcome of exercises in a circuit simulator and to explain their reasoning. This approach encourages deep thinking, requiring the students to articulate the causes and effects of the events they are observing. The example dialogue shown in Figure 2 illustrates the natural language capabilities of the system; a screenshot of the Graphical User Interface (GUI) is shown in Figure 1.

Tutor: Why was bulb A on when switch Y was open and switch Z was closed?
Student: because it had a closed path
Tutor: Right. There is a closed path. Keep going.
Student: it has a closed path with the battery
Tutor: You're on the right track. Here's the answer. Bulb A is still contained in a closed path with the battery.

Figure 2: *Example interaction with the system from the* BEETLE II *corpus*

To our knowledge, the addition of speech modality to complement the NLP capabilities of BEETLE II will make it the first ITS capable of processing long spoken explanation answers. Moving from typed to spoken interactions in this type of system presents a number of challenges, which we discuss in the following sections.

2. Architecture

The system is highly modular in design, illustrated in figure 3. On the input side, the system employs a deep parser, TRIPS [6] which provides a domain-independent semantic representation, followed by higher-level domain reasoning and diagnostics components which determine the correctness of student explanations. Based on this input, the tutorial planner module selects which tutorial strategy to use, which is implemented via a deep generation module which constructs tutorial feedback using a domain-specific content planner together with relevant content from the student's own answer.

The new ASR module uses ATK[1] to perform one-line speech parametrisation, voice activity detection and speech recognition in real-time using a multi-threaded design (though

[1] http://htk.eng.cam.ac.uk

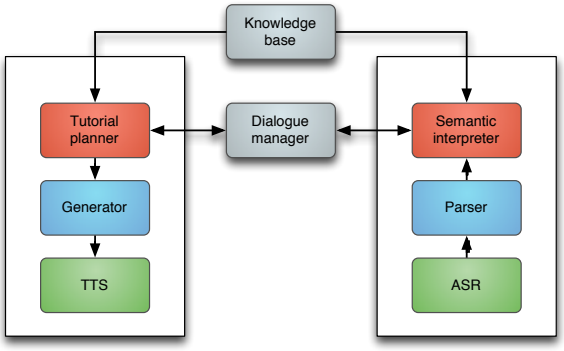

Figure 3: *The modules of the* BEETLE II *system*

dialogue-management functions are delegated to the existing BEETLE II dialogue manager). We replaced ATK's native Viterbi decoder with our own online version of HTK's HDecode, to allow efficient large-vocabulary recognition. The language models and acoustic models are described in the following sections. Spoken output is provided using the Festival text-to-speech engine. In addition to the natural language components, the GUI includes an area to display reading material and an interactive circuit simulator.

3. Language modelling

In many spoken dialogue systems, ASR is performed using hand-crafted finite-state networks selected according to the dialogue state. This is not appropriate for our system, where it is important to allow unrestricted speech, at least in principle, because students often struggle with unfamiliar terminology: effective tutoring requires knowing the words that the student said, even if they are out of domain. Therefore recognition is performed using an n-gram language model (LM).

We have a corpus available of domain-specific data comprising 90,000 words of typed interactions with the earlier BEETLE II system, collected during 2009. However, we would expect the lexical content of the spoken input to differ considerably from to the typed inputs: the switch to the spoken modality is likely to result in more verbose responses, and furthermore, the speech may contain disfluencies characteristic of spontaneous speech. As an illustration of this, Figure 4 shows an example of two different spoken student responses from our development data, illustrating the contrast with typed answers.

Student one: Row one. If bulb A is out bulb B and C will remain on. So number one is correct. Row two. Bulb B is out therefore bulb C will be out so that is incorrect and vice versa for row number three. If C is out B will also be out.

Student two: X is it open? Row two is incorrect. Um. Row three is incorrect. Rows two and three are incorrect.

Figure 4: *Two example responses to the question "Which rows do you think are incorrect?" from our development collection of spoken interaction. Punctuation has been added for readability.*

To solve this problem, we created an interpolated LM using two further corpora: the Fisher corpus of transcribed telephone conversations, and a small development corpus of spoken interactions with the system. We restricted the recogniser's vocabulary to the complete set of words from the corpus of typed interactions, plus filled pauses and common contractions such as "it's", "you've" etc.

4. Acoustic modelling

Due to the limited quantities of development audio data available, we did not attempt to train acoustic models on in-domain data, but instead used models available to us from the AMIDA corpus [7], which were trained on approximately 130 hours of speech from multiparty meetings. They are a reasonable match for our domain in terms of the recording conditions, speaking style and speaker demographic. The models were standard HMM-GMMs, trained on PLP features using MPE training. A global HLDA transform was used, and online CMN was performed using ATK's standard method. We implemented online speaker adaption using a smoothed version of CMLLR [8]

5. Future work

Considering that the output from ASR will always contain errors, a number of other problems must be solved to create an effective spoken language system. Clearly a major challenge is ensuring robust spoken language understanding when the WER is relatively high, given that the student utterances often have a complex semantic representation. The TRIPS parser is designed to provide robust parses over lattices; however, since the higher-level modules are deterministic in nature, we are not yet able to use the deep domain knowledge available to them to re-score ASR lattices. Furthermore, the parser is tuned to maximise the chance of finding a complete spanning parse, rather than to discriminate between alternative hypotheses. We plan to address this in future work.

Additionally, the system does not yet use statistical dialogue management. We propose to employ reinforcement learning in a future version of the system. Major unsolved issues to consider will be determining a suitable low-dimensional state-space for the dialogue, and selecting which measures of system or student performance should be optimised.

6. References

[1] M. T. H. Chi, N. de Leeuw, M.-H. Chiu, and C. LaVancher, "Eliciting self-explanations improves understanding." *Cognitive Science*, vol. 18, no. 3, pp. 439–477, 1994.

[2] D. J. Litman and S. Silliman, "ITSPOKE: an intelligent tutoring spoken dialogue system," in *Demonstration Papers at HLT-NAACL 2004*, 2004, pp. 5–8.

[3] N. Person, A. C. Graesser, L. Bautista, E. C. Mathews, and TRG, "Evaluating student learning gains in two versions of AutoTutor," in *Proceedings of AIED-2001*, 2001.

[4] D. Litman, C. P. Rosé, K. Forbes-Riley, K. VanLehn, D. Bhembe, and S. Silliman, "Spoken versus typed human and computer dialogue tutoring," *International Journal of Artificial Intelligence in Education*, vol. 16, pp. 145–170, 2006.

[5] M. Dzikovska, D. Bental, J. D. Moore, N. B. Steinhauser, G. E. Campbell, E. Farrow, and C. B. Callaway, "Intelligent tutoring with natural language support in the Beetle II system," in *Proceedings of ECTEL-2010*. Springer, October 2010, pp. 620–625.

[6] J. Allen, M. Dzikovska, M. Manshadi, and M. Swift, "Deep linguistic processing for spoken dialogue systems," in *Proceedings of the ACL-07 Workshop on Deep Linguistic Processing*, 2007.

[7] T. Hain, L. Burget, J. Dines, P. Garner, A. el Hannani, M. Huijbregts, M. Karafiat, M. Lincoln, and V. Wan, "The AMIDA 2009 meeting transcription system," in *Proc. Interspeech*, 2010.

[8] C. Breslin, K. Chin, M. Gales, K. Knill, and H. Xu, "Prior information for rapid speaker adaptation," in *Proc. Interspeech*, 2010.