Beetle II: an adaptable tutorial dialogue system

Citation for published version:

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Publisher's PDF, also known as Version of record

Published In:
Proceedings of the SIGDIAL 2011 Conference

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.
Abstract

We present BEETLE II, a tutorial dialogue system which accepts unrestricted language input and supports experimentation with different dialogue strategies. Our first system evaluation compared two dialogue policies. The resulting corpus was used to study the impact of different tutoring and error recovery strategies on user satisfaction and student interaction style. It can also be used in the future to study a wide range of research issues in dialogue systems.

1 Introduction

There has recently been much interest in developing tutorial dialogue systems that understand student explanations (Graesser et al., 1999; Aleven et al., 2001; Nielsen et al., 2008; VanLehn et al., 2007), because it has been shown that high percentages of self-explanation and student contentful talk are correlated with better learning in human-human tutoring (Chi et al., 1994; Litman et al., 2009). However, most existing systems use pre-authored tutor responses for addressing student errors. The advantage of this approach is that tutors can devise remediation dialogues that are highly tailored to specific misconceptions, providing step-by-step scaffolding and potentially suggesting additional exercises. The disadvantage is a lack of adaptivity and generality: students often get the same remediation for the same error regardless of their past performance or dialogue context. It also becomes more difficult to experiment with different dialogue policies (including error recovery and tutorial policies determining the most appropriate feedback), due to the complexities in applying tutoring strategies consistently in a large number of hand-authored remediations.

The BEETLE II system architecture is designed to overcome these limitations (Callaway et al., 2007). It uses a deep parser and generator, together with a domain reasoner and a diagnoser, to produce detailed analyses of student utterances and to generate feedback automatically. This allows the system to consistently apply the same tutorial policy across a range of questions. The system’s modular setup and extensibility also make it a suitable testbed for both computational linguistics algorithms and more general questions about theories of learning.

The system is based on an introductory electricity and electronics course developed by experienced instructional designers, originally created for use in a human-human tutoring study. The exercises were then transferred into a computer system with only minor adjustments (e.g., breaking down compound questions into individual questions). This resulted in a realistic tutoring setup, which presents interesting challenges to language processing components, involving a wide variety of language phenomena.

We demonstrate a version of the system that underwent a user evaluation in 2009, which found significant learning gains for students interacting with the system. The experimental data collection compared two different dialogue policies implemented in the system, and resulted in a corpus supporting research into a variety of questions about human-computer dialogue interaction (Dzikovska et al., 2010a).
2 Example Interaction

The BEETLE II system delivers basic electricity and electronics tutoring to students with no prior knowledge of the subject. A screenshot is shown in Figure 1. The student interface includes an area to display reading material, a circuit simulator, and a dialogue history window. Currently all interactions with the system are typed. Students read pre-authored curriculum slides and carry out exercises which involve experimenting with the circuit simulator and explaining the observed behaviour. The system also asks some high-level questions, such as “What is voltage?”.

An example dialogue with the system, taken from the evaluation corpus, is shown in Figure 2. It shows three key system properties: after the student’s first turn, which was correct but incomplete, the system rephrases the correct part of the student answer and prompts for the missing information. In the second turn, the system cannot interpret the student utterance, so it responds with a targeted help message and a hint about the object that needs to be mentioned. Finally, in the last turn the system combines the information from the tutor’s hint and the student’s answers and restates the complete answer since the current answer was completed over multiple turns.

3 Data Analysis and Future Work

The data collected with the BEETLE II system has been used to investigate several research questions regarding discourse and dialogue: the effectiveness of different error recovery strategies (Dzikovska et al., 2010b); the underlying dimensions of user satisfaction and their relationship with learning gain (Dzikovska et al., 2011); the relationship between (student) alignment in dialogue and learning gain (Steinhauser et al., 2011); and the differences between students’ social and metacognitive statements depending on the interaction style (Dzikovska et al., 2010a). We are currently annotating the data with additional interaction parameters, including correctness of student answers and appropriateness of system hints. This will allow us to apply PARADISE
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: There was a closed path not effected by the switch.

Tutor: I’m sorry, I’m having a problem understanding. I don’t understand the word ‘effect’ here. Here’s a hint. Your answer should mention a battery.

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 our corpus.

methodology to get additional insight into which parameters of tutorial dialogue affect learning gain and user satisfaction.

We are also adding a speech interface to the system, which will open new and interesting research questions. Students often fail to use domain terminology correctly, and in the tutoring domain it is important to teach students to use proper terminology, so incorrect usage must be detected and corrected. This means that grammar-based language models are not appropriate for the language modeling, and opens new questions about robust ASR and language interpretation in such domains.

Acknowledgements

This work has been supported in part by US Office of Naval Research grants N000141010085 and N0001410WX20278. We thank Katherine Harrison, Leanne Taylor, Charles Scott, Simon Caine, Charles Callaway and Elaine Farrow for their contributions to this effort.

References


