THE UNIVERSITY *of* EDINBURGH

Edinburgh Research Explorer

Intelligent Tutoring with Natural Language Support in the BEETLE II System

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Author final version (often known as postprint)

Published In:
SUSTAINING TEL: FROM INNOVATION TO LEARNING AND PRACTICE

OPEN ACCESS

Intelligent tutoring with natural language support in the BEETLE II system

Myroslava O. Dzikovska[1], Diana Bental[1], Johanna D. Moore[1], Natalie B. Steinhauser[2], Gwendolyn E. Campbell[2], Elaine Farrow[3], and Charles B. Callaway[4]⋆

[1] School of Informatics, University of Edinburgh, Edinburgh, United Kingdom
{m.dzikovska,d.bental,j.moore}@ed.ac.uk
[2] Naval Air Warfare Center Training Systems Division, Orlando, FL, USA
{gwendolyn.campbell,natalie.steinhauser}@navy.mil
[3] Heriot-Watt University, Edinburgh, United Kingdom
e.farrow@hw.ac.uk
[4] University of Haifa, Mount Carmel, Haifa, Israel
ccallawa@gmail.com

Abstract. We present BEETLE II, a tutorial dialogue system designed to accept unrestricted language input and support experimentation with different tutorial planning and dialogue strategies. Our first system evaluation used two different tutoring policies and demonstrated that BEETLE II can be successfully used as a platform to study the impact of different approaches to tutoring. In the future, the system can also be used to experiment with a variety of parameters that may affect learning in intelligent tutoring systems.

1 Introduction

Intelligent tutoring systems (ITS) have been shown to help learning as part of a technology-enhanced learning curriculum [1], and there has been a lot of interest in developing tutoring systems that understand student explanations (e.g. [2]). However, existing ITS with natural language support typically support only short answer questions; at most, they can analyze student essays but then revert to pre-authored short answer dialogues if remediation is necessary. Accepting longer utterances may be beneficial to learning, because high percentages of self-explanation and contentful talk by students are known to be correlated with better learning in human-human tutorial dialogue [3]. But students also need detailed feedback on their explanations [4], which is very labor-intensive to provide with pre-authored system responses.

We present BEETLE II, a tutorial dialogue system designed to accept unrestricted language input and to support experimentation with different approaches to tutoring. The BEETLE II architecture supports the understanding

Fig. 1. Screenshot of the BEETLE II system

of student explanations by using techniques from computational linguistics to analyze complex student utterances and generate feedback automatically [5].

Our research goal is to investigate whether contentful talk and self-explanation can be understood and encouraged in a computer system, given state-of-the-art in language technology. The BEETLE II system is designed to ask students to explain their answers, to give detailed feedback and to support students in arriving at the correct explanations without the system resorting to short-answer questions, and without having to hand-author every tutorial response.

We demonstrate a version of the system that has undergone successful user evaluation in 2009. The rest of this paper discusses the BEETLE II design (Section 2) the system architecture (Section 3), the user evaluation (Section 4), and further research questions that can be investigated using BEETLE II (Section 5).

2   Instructional Design

The BEETLE II system delivers basic electricity and electronics tutoring to students with no prior knowledge of the subject. A distinguishing feature of the system is that it is based on a course developed by experienced instructional designers for use in a human-human tutoring study, without taking into account the possible limitations of a computer-based tutor. Two research team members, an experienced electrician and a psychologist, jointly developed a curriculum covering topics including the components of a circuit, how to build a circuit, and how to find faults. The exercises were then transferred into a computer system with

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.

Fig. 2. Example interaction with the system from our corpus

only minor adjustments (e.g., breaking down compound questions into individual ones).

The instructional design interleaves short presentations of information with interactive exercises, activities and discussion. The activities use a circuit simulator which is realistic enough so that bulbs light up when connected correctly, bulbs may be bright or dim, and measurements may be taken by connecting a simulated meter. Many of the activities follow the "predict-verify-evaluate" (PVE) cycle, in which students are asked to predict the outcome of an activity before conducting an experiment using the simulator, and then discuss the actual outcome and its implications for the underlying principles [6].

A screenshot of the system is shown in Figure 1. The student interface includes an area to display reading material, a circuit simulator, and a dialogue history window. At present, students interact with the system via a typed chat interface to avoid the problems associated with automated speech recognition.

An example dialogue shown in Figure 2 shows how an answer is built jointly by the student and tutor over more than one turn. In response to the first student input, the system rephrases its understanding of the correct part of the answer, and prompts the student to supply the missing information. In the next turn the system combines the information from the tutor's hint and the student's answers and restates the completed answer. We will use this dialogue as a running example to explain system capabilities.

3   System Architecture

The system architecture is modular and combines domain-independent components for parsing and generation with domain-specific reasoners for decision making.

We use a natural language dialogue *parser* [7] to parse the student input. The parser extracts relevant semantic content from each utterance, recognizing paraphrases that can mean the same thing. For our example problem, the parser would recognize that "Bulb A and the battery are in the same closed path", "Bulb A is in a closed path with the battery" and "there is a closed path containing both the bulb and the battery" mean the same thing and constitute the correct answer to the tutor's question. The parser can handle fragmentary input, for example it can determine that in response to the question in Figure 1 "Which

bulbs will be on and which bulbs will be off?", the answer "off" can be taken to mean "all bulbs in the diagram will be off".

The output from the parser is passed on to the *domain reasoning* and *diagnosis* components [8, 9] to check the validity of the student's explanation. The diagnoser outputs lists of correct, contradictory and non-mentioned objects and relations from the student's explanation. For the first student utterance in our example, the diagnoser will determine that the student correctly mentioned a closed path as part of their explanation, but they forgot to mention that both the bulb and the battery must be in the same closed path.

The *tutorial planner* implements a set of generic tutoring strategies and a policy to choose an appropriate strategy at each point of the interaction. The currently implemented strategies are: acknowledge any correct parts of the answer; suggest a slide to read with background material; prompt for missing parts of the answer; provide a hint (at different levels of specificity); re-state an acceptable answer using better terminology; and give away the answer. The tutorial policy makes a high-level decision as to which strategies to use. In our example the first decision is to restate the correct part of the answer (student correctly mentioned a closed path), and give a contentless prompt for missing explanation parts. The tutorial planner also incorporates an error recovery policy to manage situations when the system cannot interpret the student's input [10].

The tutorial planner's decisions are realized by automatic *text generation* components, using a combination of domain-specific content planning and and a domain-independent text generation system [11] to produce the appropriate text. In our example, the text generation decides that the the chosen tutorial strategy (restate and contentless prompt) should be realized as "Right. There is a closed path. Keep going"

BEETLE II provides extensive logging facilities. All of the students' interactions with the system are logged, including both their text utterances and the experimental circuits they build with the simulator. The students' interactions with the system and the system's responses can be replayed. The system's utterances and its internal decisions are also logged. This provides a detailed corpus for later analysis.

4   Experimental work using BEETLE II

The first experimental evaluation involving 81 participants was completed in 2009. Participants were undergraduates recruited from a South-East US University, with little or no prior knowledge of the domain. Each participant took a pre-test, worked through a lesson with the system, took a post-test, and completed a user satisfaction survey. Each session lasted approximately 4 hours.

We implemented two different tutoring policies in the system for this evaluation, which was made possible by our flexible system architecture. In the baseline policy the students were given the correct answer straight away. The system made no attempt at remediation, and never indicated whether the stu-

dent was understood. In comparison, the full adaptive policy selected a strategy based on student answer analysis and dialogue context, as described above.

Out of 81 students, 76 successfully completed the evaluation (data from 5 participants had to be discarded due to system crashes). All students completed pre- and post- test questionnaires to assess their knowledge. The mean pre-test score was $34.56(SD = 12.38)$, and the mean post-test score was $73.91(SD = 15.64)$ for both conditions combined. The difference was statistically significant with $p < 0.0001$, indicating that the students successfully learned the material.

More detailed data analysis, comparison between conditions and comparison to human tutoring are given in [10, 12, 13].

5 Conclusions and Future Work

Our ultimate goal is to develop a system flexible enough to conduct a systematic investigation into the nature of effective tutoring in technology-enhanced learning environments. The experimental evaluation described in section 4 demonstrates that the BEETLE II system can be successfully deployed in experiments with naive users to test the effect of different tutorial policies. We now have a corpus of interactions between students and the computer tutor that can be analyzed in more detail and used to devise future experiments.

Our initial analyses indicate that additional improvements to remediation strategies, and especially strategies dealing with interpretation problems, are necessary to make the interaction less frustrating to the users [10]. We are planning to do this as part of future work. However, the success of our large-scale evaluation shows that the system can already be used to formulate hypotheses and conduct experimental studies.

Three factors whose effects we intend to investigate in the future are: linguistic alignment between system and user; the choice of modalities for input and output; and the impact of different tutorial strategies.

Techniques from computational linguistics allow us to vary and control the choice of terminology and phrasing that the tutor uses. Current research indicates that better alignment between students and tutors (computer or human) with respect to the terminology they use is correlated with higher learning gain and user satisfaction [14] and that student satisfaction is negatively affected if the system is using different terminology than the student [10]. However, it may be important that students learn specific terminology rather than the system adapt to the students. This can be investigated further through controlled experiments possible with BEETLE II since automatically generated feedback can be adjusted as necessary.

The timing and amount of feedback has been a topic of interest in the e-learning community (e.g.,[15]). The modular nature of BEETLE II will allow us to investigate the relevant issues. For example, it is easy to change the tutorial policy to give feedback immediately, and compare the outcome with exercises which make the student follow a PVE cycle which results in delayed feedback.

Other tutoring strategies, such as giving hints at different levels of specificity, can be investigated as well.

References

1. Anderson, J.R., Corbett, A.T., Koedinger, K.R., Pelletier, R.: Cognitive tutors: Lessons learned. The Journal of the Learning Sciences **4**(2) (1995) 167–207
2. Jordan, P., Makatchev, M., Pappuswamy, U., VanLehn, K., Albacete, P.: A natural language tutorial dialogue system for physics. In: Proceedings of the 19th International FLAIRS conference. (2006)
3. Chi, M.T.H., de Leeuw, N., Chiu, M.H., LaVancher, C.: Eliciting self-explanations improves understanding. Cognitive Science **18**(3) (1994) 439–477
4. Nielsen, R.D., Ward, W., Martin, J.H.: Learning to assess low-level conceptual understanding. In: Proceedings 21st International FLAIRS Conference. (2008)
5. Callaway, C.B., Dzikovska, M., Farrow, E., Marques-Pita, M., Matheson, C., Moore, J.D.: The Beetle and BeeDiff tutoring systems. In: Proceedings of SLaTE'07 (Speech and Language Technology in Education). (2007)
6. Guzzetti, B.J., Snyder, K.E., Glass, G.V., Gamas, W.S.: Promoting conceptual change in science: A comparative meta-analysis of instructional interventions from reading education and science education. Reading Research Quarterly **28**(2) (1993) 117–155
7. Allen, J., Dzikovska, M., Manshadi, M., Swift, M.: Deep linguistic processing for spoken dialogue systems. In: Proceedings of the ACL-07 Workshop on Deep Linguistic Processing. (2007)
8. Dzikovska, M.O., Callaway, C.B., Farrow, E.: Interpretation and generation in a knowledge-based tutorial system. In: Proceedings of EACL-06 workshop on knowledge and reasoning for language processing, Trento, Italy (April 2006)
9. Dzikovska, M.O., Campbell, G.E., Callaway, C.B., Steinhauser, N.B., Farrow, E., Moore, J.D., Butler, L.A., Matheson, C.: Diagnosing natural language answers to support adaptive tutoring. In: Proceedings 21st International FLAIRS Conference, Coconut Grove, Florida (May 2008)
10. Dzikovska, M.O., Moore, J.D., Steinhauser, N., Campbell, G.: The impact of interpretation problems on tutorial dialogue. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics(ACL-2010). (2010)
11. Elhadad, M., Robin, J.: Controlling content realization with functional unification grammars. In: Proceedings of the Sixth International Workshop on Natural Language Generation. (1992)
12. Steinhauser, N.B., Campbell, G.E., Harrison, K.M., Taylor, L.S., Dzikovska, M.O., Moore, J.D.: Comparing human-human and human-computer tutorial dialogue. In: Proceedings of the the 32nd Annual Conference of the Cognitive Science Society poster session. (2010)
13. Dzikovska, M.O., Steinhauser, N.B., Moore, J.D., Campbell, G.E., Harrison, K.M., Taylor, L.S.: Content, social and metacognitive statements: an empirical study comparing human-human and human-computer tutorial dialogue. In: Proceedings of ECTEL-2010. (2010)
14. Ward, A., Litman, D.: Cohesion and learning in a tutorial spoken dialog system. In: Proceedings of 19th International FLAIRS Conference. (2006)
15. Smits, M.H.S.B., Boon, J., Sluijsmans, D.M.A., van Gog, T.: Content and timing of feedback in a web-based learning environment: effects on learning as a function of prior knowledge. Interactive Learning Environments **16**(2) (August 2008) 183–193