Generating Tailored Worked-out Problem Solutions to Help Students Learn from Examples

Cristina Conati and Giuseppe Carenini

Department of Computer Science

University of British Columbia

Vancouver, BC, Canada, V6T 1Z4

{conati, carenini}@cs.ubc.ca

1 Abstract

In this paper we describe a framework that helps students learn from examples. When presenting a new example, the framework uses natural language generation techniques and a probabilistic student model to tailor the example to the student's domain knowledge. Tailoring consists of selectively introducing gaps in the example solution, so that the student can practice applying rules learned from previous examples in problem solving episodes of difficulty adequate to her knowledge. Filling in solution gaps is part of the meta-cognitive skill known as self-explanation (generate explanations to oneself to clarify an example solution), which is crucial to effectively learn from examples. In this paper, we describe how examples with tailored solution gaps are generated and how they are used to support students in learning through gap-filling self-explanation.

1.1 Keywords

Interactive Information Presentation, Computer-aided Education, User Modeling, Natural Language Generation

2 Introduction

We describe a framework that helps students learn from examples by generating example problem solutions whose level of detail is tailored to the students' domain knowledge. The framework uses *natural language generation techniques* and a *probabilistic student model* to selectively introduce gaps in the example solution, so that the student can practice applying rules learned from previous examples in problem solving episodes of difficulty adequate to her knowledge.

The rationale behind varying the level of detail of an example solution lies on cognitive science studies showing that those students who self-explain examples (i.e., generate explanations to themselves to clarify an example solution) learn better than those students who read the examples without elaborating them [1]. One kind of self-explanation that these studies showed to be correlated with learning involves filling in the gaps commonly found in textbook example solutions (*gap filling* self-explanation). However, the same studies also showed that most students tend not to self-explain spontaneously. In the case of gap filling, this phenomenon could be due to the fact that gap filling virtually requires performing problem solving steps while studying an example. And, because problem solving can be highly cognitively and motivationally demanding [10], if the gaps in an example solution are too many or too difficult for a given student, they may hinder self-explanations aimed at filling them.

We argue that, by monitoring how a student's knowledge changes when studying a sequence of examples, it is possible to introduce in the examples solution gaps that are not too cognitively demanding, thus facilitating gap filling self-explanation and providing a smooth transition from example study to problem solving. We are testing our hypothesis by extending the SE-Coach [4], an Intelligent Tutoring System (ITS) designed to support self-explanation of physics examples like the one shown in Figure 1.

This problem is novel in ITS research, as it requires sophisticated natural language generation (NLG) techniques. While the NLG field has extensively studied the process of producing text tailored to a model of the user's inferential capabilities [e.g., 6, 7, 11], the application of NLG techniques in ITS are few and mainly focused on managing and structuring the tutorial dialogue [e.g., 8, 5], rather than on tailoring the presentation of instructional material to a detailed student model.

Several NLG computational models proposed in the literature generate concise text by taking into account the inferential capabilities of the user. [11] generates effective plan descriptions tailored to the hearer's *plan reasoning* capabilities. [6] is an example of models that take into account the hearer's *logical inference* capabilities. And [7] proposes a system that relies on a model of user's *probabilistic inferences* to generate sufficiently persuasive arguments.

In contrast, our generation system tailors the content and organization of an example to a *probabilistic model of the user logical inferences*, which allows us to explicitly represent the inherent uncertainty involved in assessing a learner's knowledge and reasoning processes. Furthermore, our system maintains information on what example parts are not initially presented (i.e., solution gaps), which is critical to support gap-filling self-explanations for those students who tend not to self-explain autonomously.

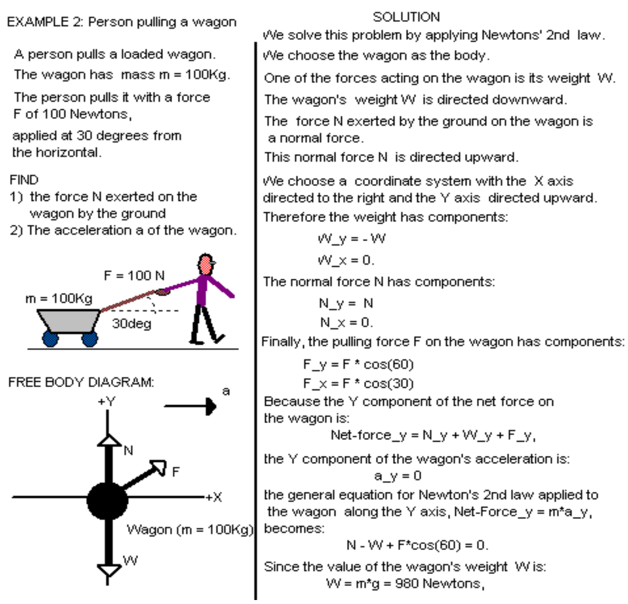


Figure 1: Sample Newtonian physics example

In the rest of the paper, we first briefly illustrate our general framework for example generation (see [2] for more details), and then we describe current research directions we are pursuing to improve its effectiveness.

3 The Framework for Example Generation

Assessing learning from instructional material requires to deal with uncertain information. We can only assess users' learning from indirect evidence; for instance by monitoring the users' reading behavior from their focus of attention or by examining users' explanations of portions of the material. The SE-Coach framework uses Bayesian networks to express and reason about the uncertainty involved in assessing learning from example solutions.

Figure 2 shows the architecture of the SE-Coach framework for generating tailored example presentations. The part of the framework labelled "before run-time" is responsible for generating the internal representation of an example solution from (i) a knowledge base (KB) of domain and planning rules (for Newtonian physics in this particular application); (ii) a formal description of the example initial situation, given quantities and sought quantities [4]. A problem solver uses these two knowledge sources to generate the example solution represented as a dependency network, known as the *solution graph*. The solution graph encodes how each intermediate result in the example solution is derived from a domain or planning rule and from previous results matching that rule's preconditions.

In the SE-Coach, every time a student is shown an example, the corresponding solution graph provides the structure for a Bayesian network that uses information on how the student reads and self-explains that example to generate a probabilistic assessment of how well the student understands the example and the related rules [3]. The prior probabilities to initialise the rule nodes in the Bayesian network come from the long-term student model (see Figure 2), which contains a probabilistic assessment of a student's current knowledge of each rule in the KB. This assessment is updated every time the student finishes studying an example, with the new rule probabilities computed by the corresponding Bayesian network.

The SE-Coach Example Generator (EG), see right part of Figure 2, uses the probabilistic student model to automatically tailor the detail level of an example description to the student's knowledge, in order to stimulate and support gap-filling self-explanation.

EG is designed as a standard pipelined NLG system [9]. A text planner [12] selects and organizes the example content, then a microplanner and a sentence generator realize this content into language. In generating an example, EG relies on two key communicative knowledge sources (right part of Figure 2): (i) a set of explanation strategies that allow the text planner to determine the example's content, organization and rhetorical structure; (ii) a set of templates that specifies how the selected content can be phrased in English.

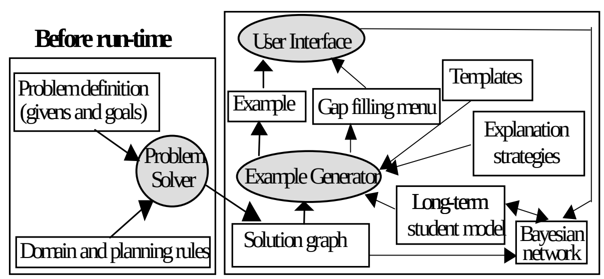
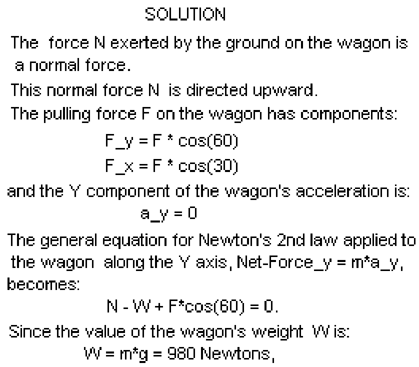


Figure 2: Framework for example generation

The selection and organisation of the example content relies on the solution graph and on the probabilistic long term student model. It consists of two phases, text planning and revision, to reduce the complexity of the plan operators and increase the efficiency of the planning process. Text planning selects from the solution graph a knowledge pool of all the propositions (i.e., goals and facts) necessary to solve a given example, and it organizes them according to ordering constraints also extracted from the solution graph. The output of this phase, if realized, would generate a fully detailed example solution. After text planning, a revision process uses the assessment in the student's long-term model to decide whether further content selection can be performed to insert appropriate solution gaps.

More specifically, the revision process examines each proposition specified by a primitive communicative action in the text plan and, if according to the student model there is high probability that the student knows the rule necessary to infer that proposition, the action is de-activated. De-activated actions are kept in the text plan but are not realized in the text, thus creating solution gaps. However, de-activated actions may be realized in follow-up interactions.

As an illustration of the effects of the revision process on content selection, compare the example solutions shown in Figure 1 and Figure 3


SOLUTION

The force N exerted by the ground on the wagon is a normal force.

This normal force N is directed upward.

The pulling force F on the wagon has components:

F_y = F * cos(60)

F_x = F * cos(30)

and the Y component of the wagon's acceleration is:

a_y = 0

The general equation for Newton's 2nd law applied to the wagon along the Y axis, Net-Force_y = m*a_y, becomes:

N - W + F*cos(60) = 0.

Since the value of the wagon's weight W is:

W = m*g = 980 Newtons,

Figure 3: Portion of Example 1 with solution gaps

Figure 1 displays the worked out solution for Example1 with no solution gaps. In contrast, the same portion of Example1 solution shown in Figure 3 is much shorter, including several solution gaps. As previously described, EG determines what information to leave out by consulting the long-term probabilistic student model. In particular, the concise solution in Figure 3 is generated by EG if the student model predicts that, among other things, the student has a good understanding of when Newton's second law should be applied, of how a body to apply the law should be selected, and of the weight force. Even if a student has sufficient knowledge to fill in the solution gaps inserted by the revision process, she may not actually perform the required inferences when studying the example. As a matter of fact, cognitive science studies show that most students tend not to self-explain spontaneously [1]. Thus, once the text plan is revised and realized, the system presents the concise example with tools designed to stimulate gap filling self-explanation. These tools help a student detect gaps in an example solution and fill them in. In particular, when the student has detected a gap and decides to fill it in, the SE-Coach's interface activates a dialogue box containing a blank for each missing step for the student to fill in. Since the interface currently does not process natural language input, the student fills each blank by selecting an item in the associated pull-down menu. EG generates the entries in this menu by applying the realisation component to unrealised communicative actions in the text plan, and uses them to provide immediate feedback on the student's selection.

4 Current Work

We are currently expanding the framework presented above along three directions. First, we are improving and testing the interface tools designed to help students in the gap filling process. We are currently experimenting with different interface designs and we plan to conduct user studies that can help us select the most effective one.

Second, we intend to exploit the *prediction capabilities* of the probabilistic student model to refine the strategies that EG uses to decide when to leave out a solution step. Currently the model is used to *diagnose* the student's understanding of the relevant physics rules given the student interaction with the system. The results of this diagnosis, i.e., the updated rule probabilities, are then used directly to decide if the student can use each rule to derive a corresponding solution step, and if so to deactivate that step in the presented solution. This approach is reasonable if we assume that we always insert only gaps that include only one missing solution step. However, if we want to insert gaps that consist of chains of two or more steps, the fact that the student has enough knowledge to infer each of these steps in isolation does not imply that the user can actually make the chain of inferences, because of the increased cognitive load that this process entails. Thus, we are working on using the student model to *predict*, given one or more chains of deactivated steps selected as described in the previous section, if the student can actually derive them given the current rule probabilities in the model. The third direction of research is in NLG. First, we plan to investigate how the example text plan can be used to maintain the coherence of the other example portions, when the student fills a solution gap. Second, we will start working on how to integrate the generation of the textual example solution and of the graphical elements of the example (e.g. the free body diagram). Third, we aim to verify whether a template-based approach to realization is sufficiently powerful to cover our target domain of physics examples.

REFERENCES

.

[1] Chi, M. T. H. Self-Explaining Expository Texts: The Dual Processes of Generating Inferences and Repairing Mental Models. *Advances in Instructional Psychology.* R. Glaser. Mahwah, NJ, Lawrence Erlbaum Associates**:** 161-238, 2000.

[2] Conati, C. and Carenini G. "Generating Tailored Example to Support Learning via Self-Explanation". *Proc. of the 17th International Joint Conference on AI*, Seattle, USA, 2001 .

[3] Conati, C. and K. VanLehn. "Providing adaptive support to the understanding of instructional material". *Proc. of IUI 2001, Int Conf on Intelligent User Interfaces,* Santa Fe, NM, January 2001.

[4] Conati, C. and K. VanLehn. "Toward Computer-Based Support of Meta-Cognitive Skills: a Computational Framework to Coach Self-Explanation." *Int. J. of AI in Education* **11**, 2000.

[5] Freedman, R. Plan-based dialogue management in a physics tutor. *Sixth Applied Natural Language Processing Conference,* Seattle, WA, 2000.

[6] Horacek, H. "A Model for Adapting Explanations to the User's Likely Inferences." *UMUAI* **7**(1): 1-55, 1997.

[7] Korb, K. B., R. McConachy, et al. A Cognitive Model of Argumentation. *Proc. 19th Cognitive Science Conf*, Stanford, CA, August 1997.

[8] Moore, J. "Discourse generation for instructional applications: Making computer-based tutors more like humans." *Journal of AI in Education* **7**(2), 1996.

[9] Reiter, E. and R. Dale. *Building NLG Systems*, Cambridge University Press, 2000.

[10] Sweller, J. "Cognitive load during problem solving: effects on learning." *Cognitive Science* **12**: 257-285, 1998.

[11] Young, M. "Using Grice's maxim of Quantity to select the content of plan descriptions." *Artificial Intelligence Journal* **115**(2): 215-256, 1999.

[12] Young, M. and J. Moore. DPOCL: A Principled Approach to Discourse Planning. *Proc. 7th Int. Workshop on NLG*, 13-20, 1994.