A Smart Problem Solving Environment

Nguyen-Thinh Le and Niels Pinkwart

Department of Informatics
Clausthal University of Technology
Germany
{nguyen-thinh.le, niels.pinkwart}@tu-clausthal.de

Abstract. Researchers of constructivist learning suggest that students should rather learn to solve real-world problems than artificial problems. This paper proposes a smart constructivist learning environment which provides real-world problems collected from crowd-sourcing problem-solution exchange platforms. In addition, this learning environment helps students solve real-world problems by retrieving relevant information on the Internet and by generating appropriate questions automatically. This learning environment is smart from three points of view. First, the problems to be solved by students are real-world problems. Second, the learning environment extracts relevant information available on the Internet to support problem solving. Third, the environment generates questions which help students to think about the problem to be solved.

Keywords: constructivist learning, information extraction, question generation

1 Introduction

A smart learning environment may provide adaptive support in many forms, including curriculum sequencing or navigation [1], student-centered e-learning settings [2], or intelligent support for problem solving [3]. For the latter class, a smart learning environment should be able to provide students with appropriate problems and intervene in the process of problem solving when necessary. Researchers of constructivist learning suggest that students should learn with real-world problems, because real-world problems are motivating and require the student to exercise their cognitive and metacognitive strategies [4]. In the opposite, traditional learning and teaching approaches typically rely on artificial (teacher-made up) problems. Often though, students can then solve a problem which is provided in a class, but would not be able to apply learned concepts to solve real-world problems.

Hence, Jonassen [5] suggested that students should rather learn to acquire skills to solve real-world problems than to memorize concepts while applying them to artificial problems and proposed a model of constructivist learning environments. We adopt this model and propose a learning environment which supports students as they solve real-world problems collected from various crowd-sourcing problem-solution

adfa, p. 1, 2011.

exchange platforms. For example, the platform *Stack overflow*[1] is a forum for programmers for posting programming problems and solutions; The platform *Wer-Weiss-Was*[2] provides a place for posting any possible problem and users who have appropriate competence are asked to solve a problem or to answer a question. These crowdsourcing platforms can provide the learning environment to be developed with real-world problems. In order to coach and to scaffold the process of student's problem solving, the learning environment is intended to provide two cognitive tools: 1) an information extraction tool, and 2) a question generation tool.

The information extraction tool is required to provide students selectable information when necessary to support meaningful activity (e.g., students might need information to understand the problem or to formulate hypotheses about the problem space). The process of seeking information may distract learners from problem solving, especially if the information seeking process takes too long and if found information is not relevant for the problem being investigated. Therefore, the information extraction tool is designed to help the student to select relevant information. It can crawl relevant websites on the Internet and represent required information in a structured form.

Land [6] analyzed the cognitive requirements for learning with resource-rich environments and pointed out that the ability of identifying and refining questions, topics or information needs is necessary, because the process of formulating questions, identifying information needs, and locating relevant information resources forms the foundation for critical thinking skills necessary for learning with resource-rich environments. However, research has reported that students usually failed to ask questions that are focusing on the problem being investigated. For example, Lyons and colleagues reported that middle school children using the WWW for science inquiry failed to generate questions that were focused enough to be helpful [7]. For this reason, a question generation tool can potentially be helpful for students during the process of gathering relevant information. If a student is not able to come up with any question to investigate the problem to be solved, the learning environment should generate relevant questions for the student.

The constructivist learning environment to be developed is smart and a novel contribution due to three features. First, it deploys real-world problems for students to acquire problem solving skills. Second, even though information extraction is an established technology, it has rarely been deployed for enhancing the adaptive support for problem solving in smart learning environments. Third, while automatic question generation has also been researched widely, strategies of deploying question generation into educational systems are rarely found in literature [8]. This learning environment can be regarded as an open-ended learning environment which supports students acquire problem solving skills using information technology [6].

---

[1] http://stackoverflow.com/

[2] http://www.wer-weiss-was.de/

2 A Smart Constructivist Learning Environment

Currently, we are initiating a project which promotes the idea of learning by solving real-world problems. For this purpose, we develop a learning environment which collects real-world problems from crowd-sourcing platforms. Real-world problems occur almost every day, e.g., ”in my area, it is snowing heavily. How can I bind a snow chain for my car?”, “my bank offers me a credit of 100 000 Euro for a period of 10 years with an interest rate on 5%. Should I choose a fixed rate mortgage or a variable rate mortgage? Which one is better for me?”, “I have a blood pressure of 170/86. Could you diagnose whether I have to use medicine?”

Two actors will play roles in this learning environment: instructors and students. The roles of instructors who are the expert of a specific learning domain include choosing the category of problems for their class and selecting real-world problems which are relevant for the learning topic being taught and at the right complexity level for their students. The challenge might here be how the platform should support instructors to choose appropriate problems, because if it takes too much time to search for relevant problems, instructors might give up and think of artificial problems. Through human instructors, real-world problems which are tailored to the level of their students can be selected. It is unlikely that the learning system might be able to select the right problem automatically for a given student model (this would require that a problem has a very detailed formal description, including complexity level).

Students can solve problems assigned by their instructors by themselves or collaboratively. They can use two cognitive tools during problem solving: information extraction for retrieving relevant information available on the Internet, and automatic question generation for helping students ask questions related to the problem to be solved. After attempting to solve these problems, students can submit their solution to the system. There, they can get in discussion with other students who are also interested in solving these problems. Let’s name our learning environment SMART-SOLVER. In the following we illustrate how these tools can be deployed.

A university professor of a course *Banking and Investment* has collected the following problem from the SMART-SOLVER platform for his students:

“I want to buy a house and a bank for a loan of 100.000 Euro. The bank makes two offers for a yearly interest rate of 5%: 1) Fixed rate mortgage, 2) Variable rate mortgage. Which offer is better for me?”

John is a student of this course. He is asked to solve this problem using SMART-SOLVER. His problem solving scenario might be illustrated in Figure 1.

Peter is also a student of this course. However, he does not have an as good performance as John and is stuck. He does not know what kind of information or questions can be input into the information extraction tool. Therefore, he uses the question generation tool which proposes him several questions. The question generation uses the problem text as input and might generate the following questions which help Peter to understand basic concepts: “What is fixed rate mortgage?”, “What is variable rate mortgage?” After receiving these questions, Peter might have a look into his course book or input these questions into the information extraction tool in order to look for definitions of these investment concepts.

John tries to solve the problem above using SMART-SOLVER. First, he uses the information extraction component to look for formulas for calculating the two mortgage options. He might have learned the concepts "fixed rate mortgage" and "variable rate mortgage" in his course. However, he might still have not understood these concepts; therefore John inputs the following questions into the information extraction tool:

- *"What is fixed rate mortgage?"*
- *"What is variable rate mortgage?"*

SMART-SOLVER searches on the Internet and shows several definitions of these concepts (not the whole websites). After studying the definitions, John may have understood the concepts and may want to calculate the two loan options. Again, he uses the information extraction tool in order to search for mathematical formulas. John might input the following questions into the tool:

- *"How is fixed rate mortgage calculated?"*
- *"How is variable rate mortgage calculated?"*

Using the formulas extracted from the Internet, John calculates the total interest amount for each loan option. He analyzes the advantage and disadvantage of each option by comparing the total amount of interest rate.

Fig. 1. A learning scenario using the information extraction tool

3 Architectural Approach

The architecture of the learning environment being proposed consists of five components: a user interface for students, a user interface for instructors, a database of real-world problems, an information extraction component, and a question generation component (Figure 2).

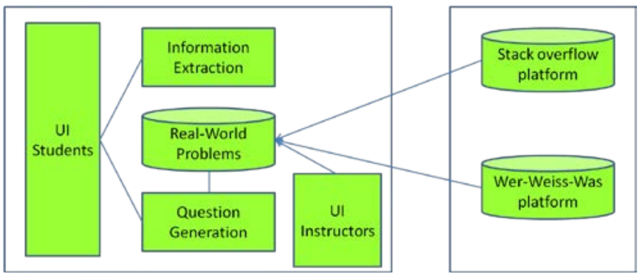

Fig. 2. The architecture of the smart learning environment

The database is connected with one or more crowd-sourcing platforms (e.g., Wer-Weiss-Was or Stack overflow) in order to retrieve real-world problems. The user interface for instructors is provided to support instructors in choosing appropriate problems for their students according to the level of their class. The user interface for students depends on the domain of studies. For each specific domain, a specific form for developing solutions should be supported, e.g., for the domain of law, the learning environment could provide tools for users to model an argumentation process as a graph. While attempting to solve problems, students can retrieve relevant information from the Internet by requesting the information extraction component, or they can ask the system to suggest a question via the question generation component. In the following, we will explain how these two components (information extraction and question generation) can be developed in order to make the learning environment in line with the constructivist learning approach.

3.1 Information Extraction

In order to extract relevant information on the Internet, we usually have to input some keywords into a search engine (e.g., Google or Bing). However, such search engines would find a huge amount of web pages, which contains these keywords, but do not necessarily provide relevant information for a task at hand.

Information extraction techniques can be used to automatically extract knowledge from text by converting unstructured text into relational structures. To achieve this aim, traditional information extraction systems have to rely on a significant amount of human involvement [9]. That is, a target relation which represents a knowledge structure is provided to the system as input along with hand-crafted extraction patterns or examples. If the user needs new knowledge (i.e., other relational structures) it is required to create new patterns or examples. This manual labor increases with the number of target relations. Moreover, the user is required to explicitly pre-specify each relation of interest. That is, classical information extraction systems are not scalable and portable across domains.

Recently, Etzioni and colleagues [10] proposed a so-called Open Information Extraction (OIE) paradigm that facilitates domain independent discovery of relations extracted from text and readily scales to the diversity and size of the Web corpus. The sole input to an OIE system is a corpus, and its output is a set of extracted relations. A system implementing this approach is thus able to extract relational tuples from text. The Open Information Extraction paradigm is promising for extracting relevant information on the Internet: TextRunner was run on a collection of 120 million web pages and extracted over 500 million tuples and achieved a precision of 75% on average [10]. Etzioni and colleagues suggested that Open Information Extraction can be deployed in three types of applications. The first application type includes question answering: the task is providing an answer to a user's factual question, e.g. "What kills bacteria?" Using the Open Information Extraction, answers to this question are collected across a huge amount of web pages on the Internet. The second application type is opinion mining which asks for opinion information about particular objects (e.g., products, political candidates) which is available in blog posts, reviews, or other texts. The third class of applications is fact checking which requires identifying claims that are in conflict with knowledge extracted from the Internet. The first type of applications using Open Information Extraction meets our requirement for developing an information extraction tool which helps students to submit questions for extracting relevant information.

3.2 Question Generation

Before students use the information gathering tool to retrieve relevant information for their problem, first they have to know what kind of information they need. Some of them may be stuck here. In this case, they can use the question generation tool which generates appropriate questions in the context of the problem being solved.

Graesser and Person [11] proposed 16 question categories for tutoring (verification, disjunctive, concept completion, example, feature specification, quantification, definition, comparison, interpretation, causal antecedent, causal consequence, goal orientation, instrumental/procedural, enablement, expectation, and judgmental) where the first 4 categories were classified as simple/shallow, 5-8 as intermediate and 9-16 as complex/deep questions. In order to help students who are stuck with a given problem statement, it may be useful to pose some simple or intermediate questions first. For example: "**What** is fixed rate mortgage?" (definition question), "**Does** a constant monthly rate include repayment and interest?" (verification question). According to Becker et al. [12], the process of question generation involves the following issues:

- Target concept identification: Which topics in the input sentence are important so that questions about these make sense?
- Question type determination: Which question types are relevant to the identified target concepts?
- Question formation: How can grammatically correct questions be constructed?

The first and the second issue are usually solved by most question generation systems by using different techniques in the field of natural language processing (NLP): parsing, simplifying sentence, anaphor resolution, semantic role labeling, and named entity recognizing. For the third issue, namely constructing questions in grammatically correct natural language expression, many question generation systems applied transformation-based approaches to generate well-formulated questions [13]. In principle, transformation-based question generation systems work through several steps: 1) delete the identified target concept, 2) a determined question key word is placed on the first position of the question, 3) convert the verb into a grammatically correct form considering auxiliary and model verbs. For example, the question generation system of Varga and Le [13] uses a set of transformation rules for question formation. For subject-verb-object clauses whose subject has been identified as a target concept, a "Which Verb Object" template is selected and matched against the clause. By matching the question word "Which" replaces the target concept in the selected clause. For key concepts that are in the object position of a subject-verb-object, the verb phrase is adjusted (i.e., auxiliary verb is used).

The second approach, which is also employed widely in several question generation systems, is template-based [14]. The template-based approach relies on the idea that a question template can capture a class of questions, which are context specific. For example, Chen et al. [14] developed the following templates: "*What would hap-*

pen if <X>?” for conditional text, “*When would <X>?*” and “*what happens <temporal-expression>?*” for temporal context, and “*Why <auxiliary-verb> <X>?*” for linguistic modality, where the place-holder <X> is mapped to semantic roles annotated by a semantic role labeler. These question templates can only be used for these specific entity relationships. For other kinds of entity relationships, new templates must be defined. Hence, the template-based question generation approach is mostly suitable for applications with a special purpose. However, to develop high-quality templates, a lot of human involvement is expected.

From a technical point of view, automatic question generation can be achieved using a variety of natural language processing techniques which have gained wide acceptance. Currently, high quality shallow questions can be generated from sentences. Deep questions, which capture causal structures, can also be modeled using current natural language processing techniques, if causal relations within the input text can be annotated adequately. However, successful deployment of question generation in educational systems is rarely found in literature. Currently, researchers are focusing more on the techniques of automatic question generation than on the strategies of deploying question generation into educational systems [8].

4 Discussion and Conclusion

We have proposed a vision and an architectural framework for a learning environment based on the constructivist learning approach. This learning environment is smart due to three characteristics: 1) this environment provides authentic and real-world problems, 2) the problem solving process performed by students are supported by exploration using the information gathering tool, and 3) the reflection and thinking process is supported by the question generation tool.

We are aware that some real-world problems might be overly complex especially for novice students. However, real-world problems can range from simple to highly complex – some of them might even be appropriate for students of elementary schools. In addition, since the learning environment being proposed provides cognitive tools (information extraction and question generation) which scaffold the process of problem solving, we think that using this learning environment, by solving real-world problems, students may improve their problem solving skills which can be used later in their daily life.

Numerous research questions can be identified in the course of developing this proposed learning environment. For instance, how should real-world problems be classified so that instructors can select appropriate problems easily? With respect to research on question generation with a focus on educational systems, several research questions need to be investigated, e.g., if the intent of a question is to facilitate learning, which question taxonomy (deep or shallow) should be deployed? Given a student model, which question type is appropriate to pose the next question to the student? Another area of deploying question generation in educational systems may be using model questions to help students improve the skill of creating questions, e.g., in the legal context. With respect to research on information extraction, several questions will arise, e.g., how should the problem solving process be designed so that students request appropriate information for solving an assigned problem? How much information should be retrieved for problem solving? We are sure, this list of research questions is not complete.

The contribution of this paper is twofold. First, it proposes a smart constructivist learning environment which enables students to solve real-world problems. Using this learning environment, students request the system for relevant information from the Internet and the system can generate questions for reflection. Second, the paper identifies challenges which are relevant for deploying information extraction and question generation technologies for building the learning environment.

5 References

1. Yudelson, M., Brusilovsky, P.: NavEx: Providing Navigation Support for Adaptive Browsing of Annotated Code Examples. In: the 12th International Conference on AI in Education, pp. 710–717. IOS Press, (2005)
2. Motschnig-Pitrik, R., Holzinger, A.: Student-Centered Teaching Meets New Media: Concept and Case Study. Journal of Edu. Technology & Society, 5(4), 160-172, IEEE, (2002)
3. Le, N. T., Menzel, W.: Using Weighted Constraints to Diagnose Errors in Logic Programming-The Case of an Ill-defined Domain. Journal of AI in Edu., 19, 381-400 (2009)
4. Bransford, J. D., Sherwood, R. D., Hasselbring, T. S., Kinzer, C. K., Williams, S. M.: Anchored Instruction: Why We Need It and How Technology Can Help. In: Cognition, Education, Multimedia - Exploring Ideas in High Technology. Lawrence Erlbaum, NJ, (1990)
5. Jonassen, D. H.: Designing Constructivist Learning Environments. In: Reigeluth, C. M. (Ed.) Instructional Design Theories and Models: A New Paradigm of Instructional Theory, Vol. 2, pp. 215-239, Lawrence Erlbaum, (1999)
6. Land, S. M.: Cognitive Requirements for Learning with Open-ended Learning Environments. In: Educational Technology Research and Development, 48(3), 61-78, (2000)
7. Lyons, D., Hoffman, J., Krajcik, J., Soloway, E.: An Investigation of the Use of the World Wide Web for On-line Inquiry in a Science Classroom. Presented at the meeting of the National Association for Research in Science Teaching, (1997)
8. Mostow, J., Chen, W.: Generating Instruction Automatically for the Reading Strategy of Self-questioning. In: Proceeding of the Conference on AI in Education, pp.465-472, (2009)
9. Soderland, S.: Learning Information Extraction Rules for Semi-structured and Free-text. Machine Learning, 34(1-3), pp. 233-272, (1999)
10. Etzioni, O., Banko, M., Soderland, S., Weld, D. S.: Open Information Extraction From the Web. Communication ACM, 51(12), 68-74, (2008)
11. Graesser, A. C., Person, N. K.: Question Asking during Tutoring. American Educational Research Journal, 31(1),104–137, (1994)
12. Becker, L, Nielsen, R. D., Okoye, I., Sumner, T., Ward, W. H.: What's Next? Target Concept Identification and Sequencing. In: Proceedings of the 3rd Workshop on Question Generation, held at the Conference on Intelligent Tutoring Ssystems, pp. 35-44, (2010)
13. Varga, A., Le, A. H.: A Question Generation System for the QGSTEC 2010 Task B. In: Proc. of the 3rd WS. on Question Generation, held at the ITS Conf., pp. 80-83, (2010)
14. Chen, W., Aist, G., Mostow, J.: Generating Questions Automatically From Informational Text. In: Proceedings of the 2nd Workshop on Question Generation, held at the Conference on AI in Education, pp. 17-24, (2009)