Automatic Generation of Questions for On-line Evaluation

M. Lemos, R. Muralidharan, V. V. Kamat
Department of Computer Science & Technology
Goa University, Goa –403206, India
E-mail: vvkamat@goatelecom.com

Abstract

Evaluation is an integral part of learning. A good objective test at the end of each learning objective can reveal a lot about the level of understanding of the learner. Question bank based evaluation often looses its freshness over a period, unless the questions are updated from time to time. In this paper we propose an alternate method wherein questions are generated from a template. Whenever a request is made to the template, a new instance of the question is generated. The concept is implemented for learning/testing tree data structures.

1. Introduction

As the saying goes, the proof of the pudding is in eating; real proof of concept being understood by the learner is in evaluation. A good test can reveal a lot about the level of understanding of the learner. Earlier the learner discovers the conceptual bugs in understanding, better the learning. Hence, an objective test at the end of each learning objective can prove very useful. Very often these objective questions are pulled out of a common repository or so called question bank. One of the limitations of the question bank based testing is the effort involved in keeping the question bank updated. Unless new questions are added from time to time, the question bank soon becomes stale and looses its significance. Setting good objective questions is a tedious and time-consuming process. One way to over come this problem is to have a template of questions stored in the question bank rather than questions themselves. Each question template can be designed to test one learning objective. Whenever a learner decides to take a test, a template would be initialized with a question from a broad range of plausible values. This gives an impression to the learner that there is a vast store of questions stored in the database. From the teacher's perspective, it increases the confidence in testing. In this paper we discuss design and implementation of template based questions for learning and testing fundamental concepts on tree data structures. The next section describes the motivation behind automatic generation of questions. This is followed by section on review of literature and a section that describes our approach to the problem. In the end we conclude by discussing the limitations of our method and possible directions for improvements and future work.

2. Motivation

Many teachers find correcting answer papers very boring and mundane activity. The experience shows that setting an essay type paper is easy but correcting such a question paper is very time consuming. Often in an essay type of question paper, it is difficult to judge the level of understanding of the learner. On the other hand, setting a good objective type questions is very challenging. Some of the advantages of objective type testing include automatic grading by the machine and better scalability. Considering the effort involved in setting an objective questions, it is very natural to ask if a template could be created per question from which many questions could be generated automatically. This will not only save significant time of the teacher but it also shows the possibility of conducting the test in an asynchronous manner. Since the chances of two students getting the same question will be very insignificant, it can also effectively handle copying and related menace. This method can also become very useful in CD based distribution of learning material where there is no scope for update once the CD-ROM is burnt and distributed.

3. Related work

With increasing number of courseware being delivered online, a wide variety of knowledge testing and evaluation systems are currently available [1]. Many of these tools [2] are for administrating the test on-line and hence address the issues such as formulating different types of objective questions, keeping student profiles, grading individual performance of the student and collecting question related statistics. Although

WebCT like tools provide excellent support for managing learner related data and the learning material, it lacks support for automatic question generation and testing.

Although the basic concept behind using question templates for generation of the questions is quite old, the recent interest in the subject is more from the point of view of on-line testing rather than development of expert system. Some work on automatic generation of questions is reported in the recent paper by Ravi Prakash et. al in development of Vyasa system [3]. Vyasa attempts to provide a general framework for multiple-choice question using constraint satisfaction approach. On the contrary we have taken a simple algorithmic approach to generate correct answer as well as wrong answer from the space of plausible answers. The wrong answers are typically generated based on buggy or misunderstood concept.

4. The basic idea

Solving a multiple-choice question can be viewed upon as a journey through network of nodes. There are start nodes, several intermediate nodes and 3-4 terminal nodes. Each terminal node represents a choice for an answer. Currently we will assume that there is only one correct answer and therefore a learner who has understood the concept correctly should be able to reach the correct terminal node starting from the initial node without much difficulty. The total number of nodes in the network model complexity of the question. If the learner has not understood the concept correctly, it is very likely that she takes a wrong path that leads to incorrect answer. The quality of the question will not only depend upon how well the correct answer separates from wrong answers but also how well the wrong answers distracts learner from the potential answers if the concept is understood wrongly.

5. Implementation

The domain that we have chosen to illustrate the concept is from tree data structures. The three important concepts in tree data structures are binary tree, binary search tree and AVL tree. We have chosen these concepts because they are fundamental to computer programming and are very often misunderstood by the beginners. Further, each of these concepts is based on sound mathematical foundation and forms a natural inheritance hierarchy. For instance AVL tree is a binary search tree and binary search tree is a binary tree. This helps us in developing the solution incrementally, reusing already implemented concepts. We will now discuss the learning phase and testing phase separately. Although from the implementation point of view the underlying algorithm is same for both the phases.

5.1. Learning

The main page listing all the three tree concepts is shown in Figure 1. When learner clicks on selected concept to learn, an HTML page corresponding to that concept is displayed on the screen. Here the learner has a choice to either read and understand the concept or visualize the concept through a simulation. When visualize button is clicked on the page, an applet as shown in Figure 2 gets started which allows the learner to build the binary tree interactively.

Figure 1: Main page on tree concepts

There are two ways to build the tree. In the first method the tree can be built step-by-step. In the second method the machine can generate the tree automatically. In step-by-step method, learner can decide to insert a node either to the left or to the right of selected node. The initial node is created by clicking on create tree button as shown in Figure 2. In the second method, an algorithm randomly inserts a node of some random value either to the left or to the right.

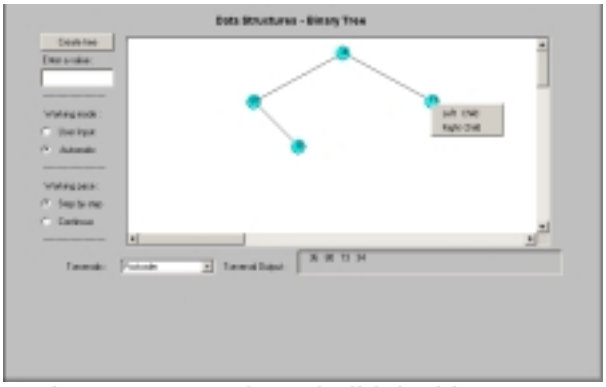

Figure 2: An applet to build the binary tree

The main purpose of the simulation is to illustrate the important concept of tree traversal. The learner can interactively select appropriate tree traversal mechanism and see the result in output box. An animation also shows the order of selection of node during traversal.

5.2. Testing

At the end of the learning phase, the learner can take an objective test by clicking on take a test button as shown in Figure 1. This in turn calls a binary tree generation algorithm that randomly inserts a node to generate a binary tree. The four choices corresponding to answers are also generated algorithmically with only one choice giving the correct answer. A sample output of automatic test generation is shown in Figure 3. After writing the correct choice in the answer box, learner can click on submit button, which will evaluate the answer. An appropriate message is displayed on the screen depending upon whether the answer is correct or wrong. Also in case the answer is wrong the correct choice is indicated. At this point the learner has a choice of either continuing with evaluation by selecting next question or quitting the testing phase.

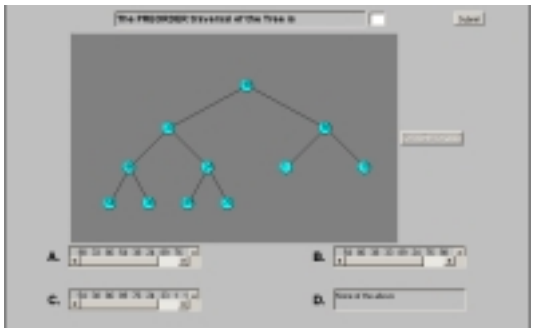

Figure 3: A sample multiple-choice question generated

The basic idea of generation of binary tree also applies to remaining two concepts mainly binary search tree and AVL tree. However there are additional constraints that need to be satisfied while inserting or deleting a node in the process of constructing binary search tree and AVL tree. A sample question generated automatically for testing binary search tree concept is shown in Figure 4.

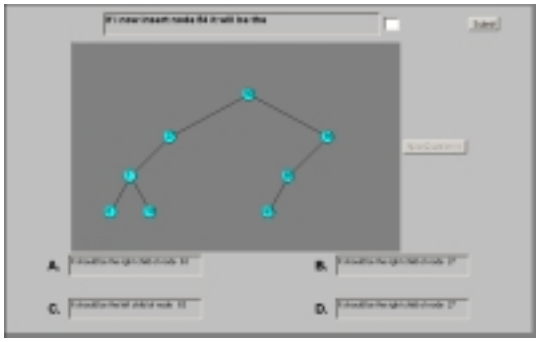

Figure 4: A sample question on binary search tree

6. Conclusion

Automatic generation of questions has a definite role to play in on-line testing systems. Some of the direct benefits of the system vis-à-vis traditional question bank based evaluation include, better scalability, reduced chances of exposure to question over repeated usage and suitability to asynchronous mode of evaluation. The indirect benefit includes elimination of copying.

Thus, automatic question generation system is well suited for on-line examination.

The method proposed is simple and easy to implement. One of the limitations of the current implementation is its usefulness over limited domain. The method is too specific and lacks necessary framework for other domains. The proposed future work is in the direction of extending the system to a general framework.

7. Acknowledgement

This work is supported through the financial grant received from AICTE as part of a project entitled "Web based evaluation tool".

8. References

[1] E. J. Gibson et. al. "A Comparative Analysis of Web-based Testing and Evaluation Systems" (http://renoir.csc.ncsu.edu/)

[2] ISIS "A Web-based quizzing system developed by ISIS computing group of Virginia Tech." (http://whizquiz.isis.vt.edu/)

[3] Ravi Prakash et. al."Vyasa: A System for Multiple-Choice Generative Questions" KBCS2000 Conference Proceedings (http://www.ncst.ernet.in/~kbcs/vyasa.html)