Automatic Generation of Question Paper from User Entered Specifications using a Semantically Tagged Question Repository

Gauri Nalawade
Department of Computer Technology
Shah and Anchor Kutchhi Engineering College
Chembur, Mumbai, India
e-mail- suvarna.benke@gmail.com

Rekha Ramesh
Department of educational Technology,
Indian Institute of Technology Bombay,
Mumbai, India
e-mail- rekha.ramesh@iitb.ac.in

Abstract— In any educational course curriculum, the courses are defined with learning objectives. Teachers conduct assessments to know if students have achieved certain learning objectives or not. Teachers generate variety of question papers as per the universities' assessment requirements. It is very challenging for the teachers to make question papers with varied questions and which meet learning objectives of the course. There are no standardized methods to ensure quality of question paper. Hence there arises a need to have a system which will automatically generate the question paper from teacher entered specification within few seconds. Researchers recommend different sets of tags such as cognitive level, difficulty level, type of question, content /topic for defining a question etc. The existing tools are rigid and support very basic or limited tags. The proposed system will automatically generate a question paper from semantically tagged question repository. This system offers flexibility by supporting all four tags and allows entry of every property in the form of ranges i.e. lower bound and upper bound. The question paper is generated in xml format and as Microsoft Word document.

Keywords— Learning objectives, assessments, question paper, cognitive level, tagging, question repository

I. INTRODUCTION

In the present competitive world, an examination plays a crucial role in checking the intellectual growth of students [1]. In fact, the quality of the students is judged by the nature of the exam questions of particular institutions [2]. Thus examination is the only measure of competence in the current education system of our country and has a decisive role in career building of students. For various examinations conducted in a year in any academic course, teachers need to generate variety of question papers as per the universities assessment requirements. The university guidelines concentrate only on the format of the question paper rather than on the quality of the question paper. It is very challenging for the teachers to cover all aspects of the course objectives and avoid duplication of questions in the subsequent exams. There are no standardized methods and hence the quality of the question paper depends completely on an individual teacher's experience and expertise [3]. The fact that there is shortage of experienced teachers makes situation even worse. At times, all these factors may deteriorate quality of the question paper.

As per researchers, a good question paper is a proper blend of items (questions) guided by various parameters such as cognitive level, difficulty level, type of item, distribution of marks across the question paper, etc. [4] [5]. Making a good question paper containing varied questions aligned with learning objective of the course in terms of contents and cognitive level is very difficult [6]. A good question paper is always fit for ordinary students but it also encompasses challenging items for intelligent students.

The existing Learning Management Systems (LMS) support very basic level or limited tags such as question types [7]. Even the most preferred LMS, Moodle allows creating only subjective/objective type of questions. Thus automatically generating question paper from a teacher's entered specification using a semantically tagged QR is the need of the hour today. The system to semi automatically tag the questions of a repository is in place [8]. Here we are proposing a system which automatically generates the question paper from this semantically tagged question repository. Since the existing systems are rigid and lack the flexibility of supporting all types of tags, the generated question paper may not be totally aligned with its given objectives. Our system supports all four tags and also flexible enough to provide an interface that allows user to enter specifications for each tag/property in the form of lower and upper bounds. Each property is specified with a range indicating that value should not be lower than minimum value and not exceed the maximum value of the range. Also, it is rule base system which takes all the combinations of the tags and generates output based on the rule applicable. The output is generated in xml format and in word document.

Section 2 discusses the related literature survey. The design approach is explained in section 3. The implementation aspects and the user interface are described in section 4. System Evaluation is available in section 5. Section 6 consists of discussions and future scope of our work followed by the references used for entire work.

II. LITERATURE SURVEY

A literature survey was started to understand the need for automatic generation of QP. As mentioned in [7], many

978-1-5090-6115-0/16 $31.00 © 2016 IEEE
DOI 10.1109/T4E.2016.37

existing LMS support tagging feature but users may not utilize this feature fully. The comparative study shows that Moodle is best LMs to support large number of users and also for any educational institution [7] [9]. But it allows user to define only question type. Hence the questions in the repository may have only basic tags or no tags at all. So it becomes overhead for teachers to tag these questions before using them. Properly tagged questions can be efficiently retrieved from repository [10]. Hence it is very much necessary to tag the questions before adding them to repository. Most teachers and researchers recommend four types of tags such as topic, question type, cognitive level and difficulty level.

A system which offers generation of question paper using user given input parameters considers only fixed range of values [1]. Our system not only supports upper and lowers bounds for inputs but also supports more granular level of topics than chapters and more question types as compared to only three types offered by this system. We are using automatically tagged question repository as input instead of untagged questions. Automatic Question Paper Generator System described in [2] has great interface, but supports only question type tag. Hence the question paper generated would have only one difficulty level.

III. DESIGN

We have implemented a system that facilitates automatic generation of question paper from semantically tagged question repository. The system would be useful for institutes, publishers and test paper setters who have a huge repository of tagged questions and need to frequently generate question paper with ease. In this paper, the working of the system is demonstrated for written examination of Data Structure domain of an engineering curriculum.

A. System Architecture

The system block diagram is shown in figure. 1

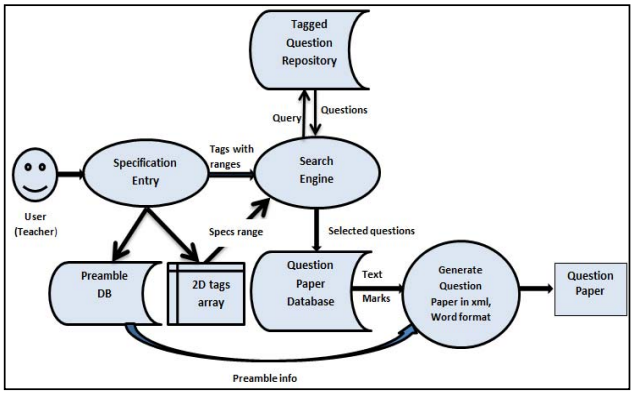

Fig. 1. System Block Diagram

We are using the existing exhaustively tagged question repository as an input to our new system. User will be asked to enter the values for each tag in the form of lower and upper bounds. The search engine extracts questions from question repository based on the specifications entered by the user. The output is produced in the xml format and further as word document if required.

B. Input Database

A well tagged question repository contains questions with four tags: topic (content), question type, cognitive level, and difficulty level [8]. We are using Bloom's taxonomy for cognitive level [11]. The two tables i.e. Concept table and Question table of the existing repository will be used to extract information from database as per requirement. The Concept table stores the q_id, topic names and chapter title of the question from a particular domain. The Question table stores the metadata about the semantically tagged questions. The attributes include id, cognitive level, question type, topic, difficulty level, question text and max_marks allotted to the question. The various attributes (tags) and their value ranges are given in table 1.

Table 1: Various tags with value range [8]

Tags	Values
Cognitive level	Six levels of Blooms taxonomy: Recall, Understand, Apply, Analyze, Evaluate, Create
Question Type	Fill-in-the-blanks, Multiple-choice, Match the following, True-false, Answer-in-one-word, Arrange in order, Definition, Short-note, Differentiate/Comparison, Program-implementation, Short-answer, Long-answer
Content	Topics and subtopics from the syllabus
Difficulty level	Low, Medium, High

C. Specification Entry

User will be asked to enter two types of specifications (i) the header or preamble; (ii) question paper specification.

a) Header/Preamble:

At the first level user will be asked to enter the question paper preamble specifications such as university, course, course_year, semester, subject, total marks, date of examination and notes if any. All the fields are validated for non-blank values. Only numeric values are accepted in marks field whereas rest of the fields accept alphanumeric values. The valid fields go to qp_header table to print them further on question paper.

b) Question Paper Specification:

The next web page interface will ask the user to select the Question Paper Specifications (QPS) and enter the values for each property in the form of lower and upper bounds. The validations on this page are done as follows: (i) only numeric values will be accepted in all the minimum and maximum range. (ii) The lower bound total for a tag must always be equal to 100 and upper bound total for tag must always exceed 100. (iii) Any upper bound value of a tag attribute must always be greater than or equal to lower bound value of the respective tag attribute. These validations also remain valid for question type, cognitive level and difficulty level min-max values.

User may also prefer to give partial specification to generate question paper on specific topics only irrespective of question type, cognitive level or difficulty level. For the given specification, the search engine will apply the rule base and extract the questions from the database which fits in all criteria specified.

D. Search Engine

After the valid specifications are received, the search engine starts searching for the question in a tagged repository which fit within the minimum and maximum marks range specified for a selected tag. The questions are picked up randomly every time from the repository. So, the user will get new set of questions for the same specifications next time.

ALGORITHM SEARCH ENGINE:

1. Get the input specification..
2. Marks_Left=Total_marks.
3. For all the values present in Topic, Question type, cognitive level and difficulty level, search the question in Min range column
4. If Marks_Left >0, then search in Max range columns.
5. For each question id from concept table, search the question in Question table using following query:
   select id, text, max_marks from question where id = concept_qid and max_marks <= Topic[j][i] and (cognitive = coglevel and (max_marks <= Coglevel[x][z])) and (q_type = Qtype and (max_marks <= QType[l][m])) and (difficulty = Difflevel and (max_marks <= DiffLevel[v][y]))
6. If question found, add it to Question_paper database
7. Decrement the values of the respective arrays and Marks_Left by marks of the question.
8. While Marks_Left>0, go to step 4.
9. If no question added to Question_paper database, issue a warning, else display the blueprint.

E. Question Paper Generation

The preamble information available in qp_header table and question contents from question_paper table is combined to produce question paper in xml format.

IV. IMPLEMENTATION

The system is implemented as a web application using NetBeans IDE. The default servers i.e. Glassfish web server and derby database were used. However the application is tested on Apache Tomcat server and it runs absolutely fine without any issues.

A. Specification Entry

The first web interfaces allows user to enter question paper header specification while the second one allows question paper specification entry.

Header/Preamble Form:

The beginning screen of the Question Paper Generation System is as shown in figure 2. The teacher must enter the university name, course name, academic year, semester, subject, total marks and date of examination along with any instruction notes required. All the valid fields are stored in qp_header table.

Question Paper Specification Entry Form:

In the next interface as shown in figure 3, user will be provided tables with input boxes to enter the specifications in the range between 1 to 100 for topics, question types, cognitive levels and difficulty levels. The search engine validates the inputs, stores them and prints the blueprint as shown in figure 4.

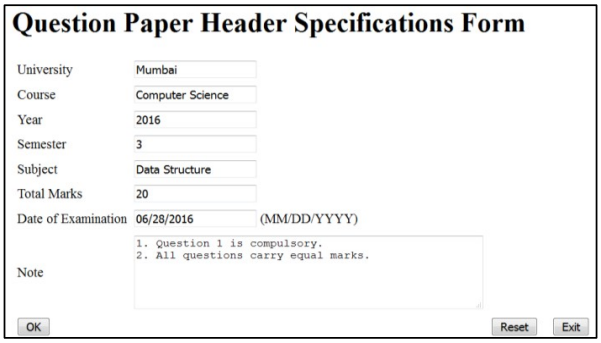

Fig. 2. Question Paper Preamble Information

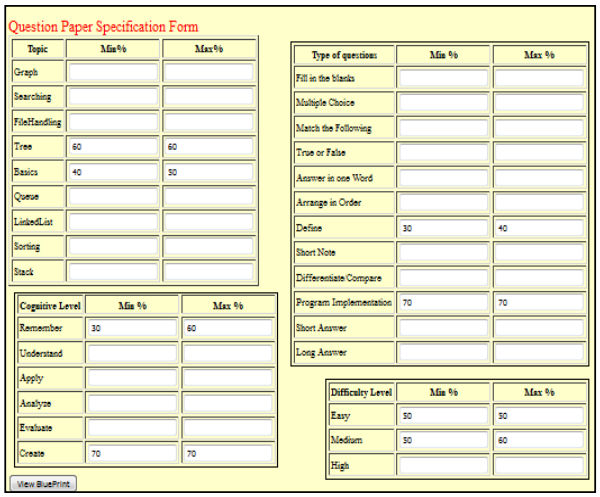

Fig. 3. Question Paper Specification form

When user chooses to generate question paper, the search engine begins its processing as per search engine algorithm.

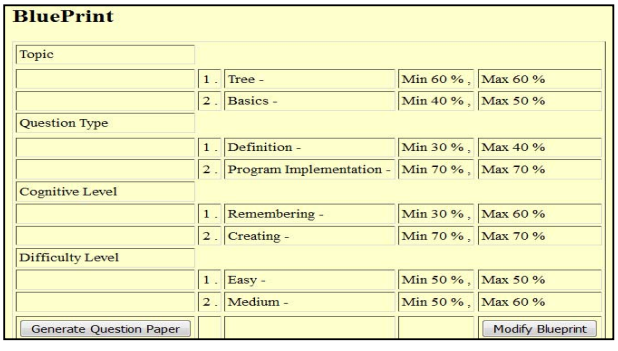

Fig. 4. Blueprint of the selected specifications

B. *Generate Question Paper*

The next process would take the preamble information available in qp_header table and combine it with question_paper table contents to produce question paper in xml format. We preferred xml format as it is supported by all browsers. The xml output would look like as shown in figure 5. User will get an interface with an option to convert it into word document which looks as shown in figure 6.

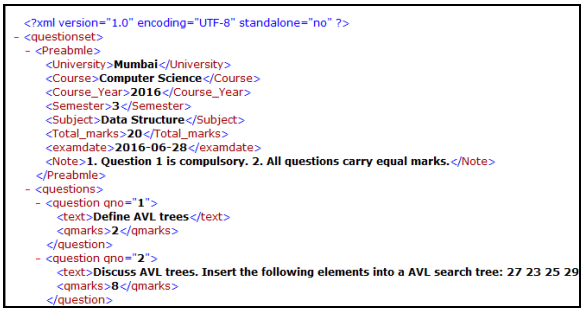

Fig. 5. Question Paper in xml format

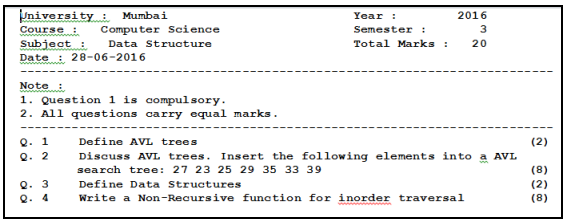

Fig. 6. Question paper in word doc format

V. System Evaluation

We performed an accuracy-test with the help of two Data Structure expert teachers. It was found that the accuracy with respect to questions selection is 95%. The usability and user friendliness was checked by 5 Computer Science instructors with an experience of at least 5 years. In the first phase of user testing, they were given simple set of instructions, viz.(i) "explore each components of the system"; (ii) "Validate each entry on every screen with positive and negative inputs "; (iii) "Check the correctness of generated question paper in xml format and word format". In the second phase, users were given ten questions to test the System Usability (SUS) score of the Question Generator [12]. For each of the SUS question, we asked the users to write an open-ended response using likert scale score which were qualitatively analyzed and coded to test the usability of the Question Generator. Four out of five users found the system to be useful.

Most prominent benefits of the system as reported by the users are: (i) the easy GUI (ii) nil or least requirement of technical knowledge to use the system (iii) generation of question paper in xml and ready to print word format.

VI. Discussions and Future Scope

To frequently generate good question paper which meets learning objectives of the course, we need expert teachers. But there is shortage of experienced teachers in institutions or university. Existing systems lack the flexibility to support all types of tags identified by researchers. Our system is a great aid for teachers in generating question papers automatically from tagged question repository.

While the system designed by us stands out in all available systems, there is scope for more enhancements to make it more useful. Depending on the type of assessment required, the system can be made to select particular question types. For example, if user wants assessment for online quiz, it could smartly include all MCQs. Or if user is choosing term test assessment, more objective type and short answer questions must be preferred. Also, users would be delighted to have a feature to present statistics for gaps in user given specifications and system generated specifications.

References

[1] Vijay Krishan Purohit', Abhijeet Kumar', Asma Jabeen', Saurabh Srivastava', R H Goudar' ,Shivanagowda, "Design of Adaptive Question Bank Development and Management System", 2nd IEEE International Conference on Parallel,Distributed and Grid Computing, 2012

[2] Kapil Naik, Shreyas Sule, Shruti Jadhav, Surya Pandey, " Automatic Question Paper Generation System Using Randomization Algorithm", (IJETR) ISSN: 2321-0869,Volume-2, Issue-12, December 2014

[3] Kumar, Ajit, and N. Shukla. "Criteria for evaluating the quality of a question paper." Journal of Technical Education and Training 3.1 (2011): 1-6.

[4] J. Millman, and J. A. Arter, "Issues in item banking," Journal of Educational Measurement," 21(4), pp.315-330, 1984.

[5] W. Harlen and M. James, "Assessment and learning: differences and relationships between formative and summative assessment," Assessment in Education, 4(3), pp.365-379, 1997.

[6] J. Biggs, "Aligning teaching and assessing to course objectives," Teaching and Learning in Higher Education: New Trends and Innovations, 2, pp.13-17, 2003.

[7] Caminero, A.C. ; Dept. de Sist. de Comun. y Control, Univ. Nac. de Educ. a Distancia, Madrid, Spain ; Hernandez, R. ; Ros, S. ; Robles-Gomez, A. , "Choosing the right LMS: A performance evaluation of three open-source LMS", Global Engineering Education Conference (EDUCON), 2013 IEEE, March 2013

[8] Ramesh, R. ; Mishra, S. ; Sasikumar, M. ; Iyer, S., "Semi- Automatic Generation of Metadata for Items in a Question Repository", Technology for Education (T4E), 2014 IEEE Sixth International Conference, Dec 2014

[9] Vyshnavi M R , "A rubric to Evaluate Learning Management Systems", Teaching, Assessment and Learning for Engineering (TALE), 2013 IEEE International Conference, 2013

[10] Saha, A.K. ; Univ. of Saskatchewan, Saskatoon, SK, Canada; Saha, R.K. ; Schneider, K.A., " A discriminative model approach for suggesting tags automatically for Stack Overflow questions", Mining Software Repositories (MSR), 2013 10th IEEE Working Conference , May 2013

[11] D. R. Krathwohl, " A revision of Bloom's taxonomy: An overview. Theory into practice," Vol. 4 41, pp. 212-218, 2002.

[12] J. Brooke, "SUS-A quick and dirty usability scale. Usability evaluation in industry, " 189, 194.