A Survey on Question Paper Generation System

Aniruddha Joshi
BE Computer Engineering
Viva Institute of Technology

Prathamesh Kudnekar
BE Computer Engineering
Viva Institute of Technology

Mayur Joshi
BE Computer Engineering
Viva Institute of Technology

Siddhesh Doiphode
Asst-Prof(comp. Engg.)
Viva Institute of Technology

ABSTRACT
The effort needed for creating question paper is reduced because of the advanced system. Firstly people require to spend time on generating paper by referring syllabus and references book etc, which was time consuming. Because of the advanced system there is no need for human to think and spend time which can be used on some more important tasks instead of creating question paper. The system fully automates the process of question paper generation. The advanced system generates question paper based on database such that all types of questions such as (MCQS, Numerical type, Theory based etc) are stored in database. The system randomly selects questions from database and generates a question paper such that it covers entire syllabus.

General Terms
Artificial Intelligence, Security, Algorithms.

Keywords
CSV file, PDF file, Question paper format, pattern composer, question aggregator.

1. INTRODUCTION
Previously question papers were generated manually, which time was consuming. To overcome this problem new

System is generated that uses keywords from syllabus and

978-93-5254-975-7/ ©2016 NCRENB

EBook to generate question considering the rules and

Restriction, paper is generated very effectively and efficiently. The question paper generation system that currently exists is totally based on database. It just randomly selects the questions from database and generates question paper. In proposed system instead of adding each question into database for question paper generation, CSV files are used to generate question paper based on keywords that are extracted from that CSV file.

For generating CSV file that is imported into database for keyword extraction, system will use simple PDF file of syllabus given by university is used and further process of selecting number of question and type of question generated on each module takes place with help of CSV file.

So proposed new question paper generation system is very much efficient and time saving than the other ones that currently exist.

Comma-separated values (CSV) file stores the tabular data (i.e. numbers and text) in plain text. Each line of the file is like a data record and each record consists of one or more fields, separated by commas. The use of comma as a field separator or delimiter is the source of the name for this file format. An official standard for the CSV file format does not exist, but RFC 4180 provides a de facto standard for many aspects of it. In popular usage, the term CSV may denote some closely related delimiter-separated formats, which may use a variety of different field delimiters. These formats include tab-separated values and space-separated values, both of which are popular. Such files are often even given a .CSV extension, despite the use of a different field separator than the comma. This loose terminology creates problems for data exchange [8].

2. LITERATURE REVIEW

2.1 Generation of Paper from Question Bank[1]
The structure of question database, paper database and template database are discussed .First question type is designed including subjective and objective questions. The test database contains variety of question with difficulty levels. Using the system user can choose paper setting manual.

Functions
Through user interface user can access following functions:

1) **Test question database maintenance:** The question can added, deleted and modified from database.
2) **Manual paper combination:** In proposed system user can select the required question and add it into paper.
3) **Intelligent paper combination:** In proposed system question are selected randomly according to specific requirement and conditions.
4) **Test paper generation:** It is the combination of manual and intelligent paper combination.
5) **Other function:** Other function like setting, system exit, library open etc [1].

System Structure and Composition

1) **Structure of test question database:** Most important database containing different question like sums, theory question etc.
2) **Structure of paper database:** Used to Store Generated paper.
3) **Structure of template database:** Used to apply intelligent paper combination to generate a paper, it also stores how many questions and difficulty level etc.

4) **System implementation:** Using intelligent function system will generate a paper as per as specific requirement. After selecting the question, number of question and difficulty level the percentage of marks should be calculated by Score Distribution Algorithm.

2.2 Question Paper Manipulation[2]

The development of proposed system follows prototype model of software development life cycle, which help us to create a product in short time. The system has several test engines with randomization of question that can create different sets of question.

System Structure and Design

Overall aim is to generate question automatically, first select number of question set, secondly the number of questions in each set. Then user needs to select examination question from test bank, if user satisfy then he can continue with further process else select different question from test bank. Lastly user should select output file i.e. in (PDF or text file) [2].

2.3 Keyword Extraction [3]

Fang Yuan in year 2005 has proposed a method for extracting information from PDF files. The rule set is constructed for manually or from learned training data for information extraction and modified later. There are many Information Extraction (IE) approached can be used e.g. STALKER, CCWRAP. Main task of IE algorithm to find out symbol of left and right side of each attribute. But these methods cannot be applied directly to PDF files.

The process of Information extraction consists of following modules.

1) Construction of PDF files parser.

   PDF document consist of various information such as text, images and table information. PDF file parser parses this information.
2) Construction of tag injector.
   In PDF files information is organized in the form of co-ordinates, so for extracting information just inject user defined tags for example font size, font type, empty row etc...
3) The process of tag preprocessor.
   It process the user defined tags [3].

2.4 Question Paper Template Generation[4]

Vaibhav M. Kale in year 2013 has proposed the framework for creating question paper. The quality of question paper depends upon various constraints like question paper format, coverage of syllabus, coverage of difficulty levels, coverage of cognitive levels. The algorithm provided in the proposed system solves the problem by dividing the task into two subtasks namely question paper template generation and question paper generation.

Constraints

1. Unit wise distribution of marks
2. Difficulty level wise distribution of marks.
3. Distribution across cognitive level of Blooms Taxonomy.
4. Question paper format [4].

Algorithm Working

1) Round 1: Algorithm takes two constraints input question [], unit [] and stores the output in Question temp[] and Unit No[]
2) Round 2: Same algorithm takes two input Cognitive_Level [], Question [] and stores the output in Question temp [] and CoglevNo []
3) Round 3: algorithm takes two inputs Difficulty Level [] and Question []And final output will be stirred in Question_temp [] and DiffLevNo[] [4].

2.5 Intelligent Test Paper Generation [5]

Jing Mei Li in year 2009 has proposed two strategies for intelligent paper generating.

1. Random selection.
2. Backtrack .

Constraints

Test paper generation algorithm has following constraints:

1. Score in total test paper.
2. Required proportion of each module.
3. Difficulty in total test paper.
4. Required proportion of question type.

Algorithm

1) Judge whether the input control parameters is legal, if not, jump to 11), otherwise, system acquire question types parameter and chapter parameter and initialize the difficulty factor, then go to next step.
2) Sort order based on the percentage difference between required question types and current question types, select question type which can minish the percentage difference most, then go to next step.
3) Judge whether the percentage of current question type can meet the need, if yes, go to next step, otherwise, jump to 5).
4) Judge whether the percentage of all current question types can meet the need, if yes, jump to 11), otherwise, continual.
5) Sort order based on the percentage difference between required chapters and current chapters select chapter which can minish the percentage difference most, and then go to next step.
6) Judge whether the percentage of current chapter can meet the need, if not, go to next step, otherwise, jump to 5).
7) Look up difficulty partition table (M) according to the difficult grade in the whole test paper. Compare the percentage of each area between exam question that existed in current test paper and required. Select area that have most different and judge whether there is any exam question existed in selected area. If yes, jump to 9), otherwise, go to next step.
8) Judge whether all of areas has been searched. If not, jump to 5), otherwise, jump to 7).
9) Judge whether there is any exam question that can match the selected question type, chapter and area

constraint existed. If not, jump to 7), otherwise, go to next step.

10) Sort all of exam questions that existed in database based on selected frequency. Select exam question with low frequency and put it into test paper, then jump to 2).

11) Test paper generation process complete [5].

2.6 Extract Table Information from PDF[6]

Burcu Yildiz in year 2014 has proposed a way to decompose table and store extracted data in structured data format (XML) for easy reuse. Table often contains high density information. PDF file is taken as an input. Table Extraction is process of decomposing table information in document. In this process author made use of PDFtohtmltool. For each text chunk in PDF file it returns text in XML with following attributes.

Attributes

Attributes are the properties related to the text

1. Top=vertical distance from the top of page
2. Left=horizontal distance from the left border of page
3. Width=width of text part
4. Height=height of text part
5. Font= this attribute describe the size, family and color of text part.

Heuristics

Here author has studied and explored different types of table and their structure and given some heuristics. The heuristics used can be grouped in two categories

1. Heuristics intended to recognize table.
2. Heuristics intended to decompose table.

2.7 Framework of Question Paper Generation System [7]

Ashok Immanuel in 2015 proposed a system, which is trying to tell how to generate an efficient paper by using set of keywords of syllabus particular course subject. There are seven main components to generate the paper; they are Question Bank Engine – Framework, Bank Interface, Syllabus Engine, Pattern Composer, Question Aggregator, Bank Management, and Question Paper Generation.

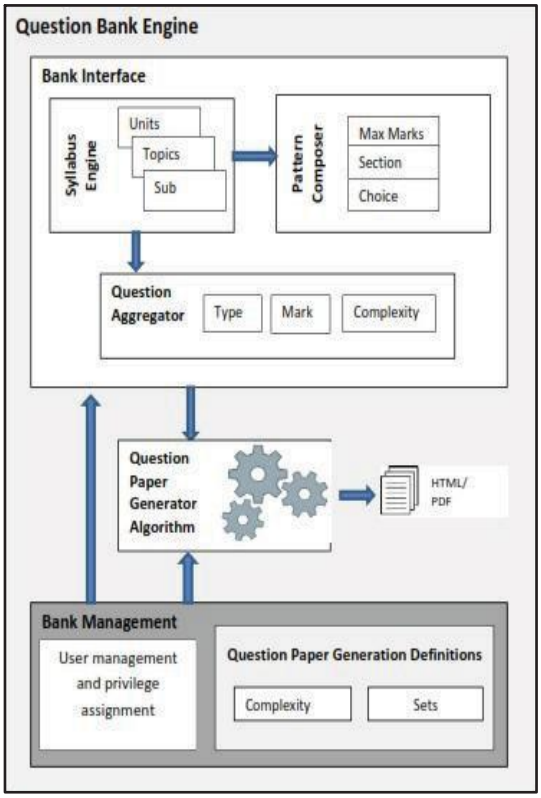

Fig 2.1: Framework Diagram[7]

Above figure 2.1 represents framework diagram syllabus engine consist of units, topics and subtopics which is input to pattern composer and question aggregator .pattern composer consist marks ,section, choice .question aggregator consist of type mark and complexity of question which will be input to the question paper generation algorithm and after that the PDF of question paper is generated[7].

3. CONCLUSION

This paper presents various systems to create effective question and automate the complete procedure of question paper generation. Some system uses searching algorithm and questions can be generated that are relevant to syllabus and generate a question paper according to weight age of each chapter. Thus system simplifies the whole process of question paper generation. And further security can be added in future to system.

In future proposed system can be extended using Natural Language Processing (NLP) for generating questions.NLP can be used in system to make sure that same question will not be generated again and again, and with the help of NLP we can add calculation type and cases type questions in question paper. Also EBook can be used to generated question from sub-topics

4. REFERENCES

[1] Peijiang Chen, ”Construction of Test Question Database for Electrical and Electronic Technology”, International Conference on Electronic & Mechanical Engineering and Information Technology, IEEE, August 2011, p 5.

[2] Syahaneim Marzukhi**,** Muhammad Firdaus Sulaiman. “Automatic Generation Question System to Enhance Preparation of Learning Assessment”, p 6.

[3] Fang Yuan, Bo Liu,. "A New Method of Information Extraction from PDF Files", The Fourth International Conference on Machine learning and Cybernetics, IEEE, August 2005, p 4.

[4] Vaibhav M. Kale, Arvind W Kiwelekar. "An algorithm for question paper template generation in question paper generation system". IEEE, 2013, p 8.

[5] Jing Mei Li, Jing Li, Jian Pei Zhang, Nan Ding. "Research on intelligent test paper generation based on multi-variable asymptotic optimization". IEEE ,2010, p 6.

[6] Burcu Yildiz, Katharina Kaiser, Silvia Miksch. PDF2table: "A method Extract table information from PDF files'. IEEE, 2014, p 6.

[7] Ashok M., Tulasi .B. "Framework For Automatic Paper Generation System", International Journal of Computer Applications, IEEE, 2014, p 9.

[8] S. S. Sheik,† Sumit K. Aggarwal," A FAST Pattern Matching Algorithm", Bioinformatics Centre and Supercomputer Education and Research Centre, Indian Institute of Science, *J. Chem. Inf. Comput. Sci*, 2004, *44*, *P.*1251-1256.

[9] Mr. Rahul B. Diwate, Prof. Satish J. Alaspurkar," Study of Different Algorithms for Pattern Matching", International Journal of Advanced Research in Computer Science and Software Engineering, IJARCSSE, Volume 3, Issue 3, March 2013.

[10] Vasile Rus, Brendan Wyse," Question Generation Shared Task and Evaluation Challenge, Proceedings of the 13th European Workshop on Natural Language

[11] Husam Ali ,Yllias Chali," Automatic Question Generation from Sentences, University of Lethbridge, TALN 2010, p. 19–23.

[12] S. Chen ,S. Diggavi," Efficient String MatchingAlgorithms for Combinatorial Universal Denoising". Generation (ENLG), Association for Computational Linguistics, September 2011, pages 318–320