Automated Question Paper Generator and Answer Checker Using Information Retrieval Approach

Mansi Palav#, Pranjal Singh#, Pratik Vishwakarma#, Pooja Pandit#, Prof. Neelam Phadnis#
B.E, Department of Computer Engineering, Mumbai University
Mumbai, Maharashtra, India

Abstract
This is a challenging span due to the growth in the field of computer science and demand we are facing today. Hence, examinations play a vital role in testing student’s performance. That is why it is important to have a smart development question model for growth of students as well as to test their learning skills thereby keeping a check on student performance. Generating an effective question paper is a task of great importance for any educational institute. The traditional method, where lecturers manually prepare question paper, it is very tedious and challenging. Our System allows faculty to generate question papers with random questions, which covers the chapters selected by the faculty. In this study, we propose an automated scoring approach for descriptive answers by using Jaro-Winkler

Keywords - Question Paper, Answer Checker, Randomize Algorithm, Jaro-Winkler

I. INTRODUCTION
Generating an effective question paper is a task of great importance for any educational institute. The traditional method, where lecturers manually prepare question paper, is very tedious and challenging. Automated Question Paper Generator and Answer Checker System can reduce time consumption by replacing the conventional method of question paper generation. The system fully automates the process of question paper generation and selective answer checker. The advanced system generates question paper based on database such that all types of questions and answers such as (MCQs, Theory based, and objectives.) are stored in database. The questions are randomly selected by the system from database and generates a question paper such that it covers all the chapters, which are selected, and allows student to attempt the examination and to score accordingly. Answer Checker checks the paper and generates marks/score for the student by finding the difference between model answer and student answer by using Jaro-Winkler

II. LITERATURE REVIEW
This section presents the significant approaches in the field of information retrieval, the techniques used for text similarity and to understand the need for automatic generation of question paper and answer checking. Different models used to evaluate the results of these techniques are also reviewed.

A. Background Study
1. Question Paper Generation
A literature survey was started to understand the need for automatic generation of question paper. As mentioned in, many existing LMS support tagging feature but users may not utilize this feature fully. The comparative study shows that Moodle is the best LMS for any educational institution and to support large number of users. 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. Hence it is very much necessary to tag the questions before adding them to repository. A system which offers generation of question paper using user given input parameters considers only fixed range of values. 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.[5]

2. Answer Checker System
Different models used to evaluate the results are reviewed.

a. Intelligent Essay Assessor (IEA)
It uses statistical model to compare descriptive answers and checks the semantic similarity between two answer sets. It is also used to analyse and to give score for the essay type of answers.

b. E-rater
E-rater is use to analyse the essay type of answers and specifies syntactical and lexical issues in the text. [3]
c. C-rater

C-rater is primarily use for assigning the marks as per the student’s correct answers. It also deploys the similar kind of words used in the answer, spelling errors, syntax variations, which are checked automatically. [2]

3. Text to Text Similarity Approaches

The primary methods of similarity are classified as knowledge-based similarity, corpus-based similarity, and string-based similarity measures. [4]

a. Knowledge-based similarity

It applies text-to-text similarity to determine the shortest path of similarity by detecting lexical chains between pairs in a text using the WordNet hierarchy.

b. Corpus-based Similarity

It is used to find similarity between words according to the sets called as corpus. It checks the occurrences of the word in the particular answer.

c. String-based Similarity

String-based similarity evaluates the measures of similarity or dissimilarity between two text strings. There are two different types of string-based algorithms for evaluating similarity between the student’s answer (SA) and the model answer (MA).

1. Character-based similarity

Character-based similarity is used to find out the distance between two strings and perform minimum operations. Operations such as insertion, deletion, substitution and transposition of a single character.

c.2 Term-based similarity

Term-based similarity is the distance between two item's sum of the distances of their corresponding items

D. Cosine Similarity

The SA and MA are represented as vectors, where student’s answer and model answer are a set of terms; each term has a weight, which reflects its importance on that MA or SA. There are several ways to calculate this weight, such as the Term Frequency-Inverse Document Frequency (TF-IDF), where the (TF) refers to the term frequency in the model answer, and the IDF represents the importance of a term with respect to the entire corpus. It is calculated by the number of answers in the corpus divided by the number of answers containing a term.

The cosine similarity measure is based on term weighting scheme, which is the TF-IDF. It is usually used as a weighting factor in information retrieval and text mining. The formulas of TF, IDF and TF-IDF are illustrated below as follows:

\[ TF = \frac{\text{number of occurrences of term in answer}}{\text{number of terms in answer}} \]

\[ IDF = \frac{N}{\log nj+1} \]

Where N is the total number of answers, NJ is the number of answers containing the term.

\[ TFIDF = TF \times IDF \]

The main idea behind this model is to calculate the weight of each term in each answer with respect to the entire corpus.

The TF-IDF is used to compare a student’s answer vector with a model answer vector using cosine similarity measure. Cosine similarity measures the cosine of the angle between two vectors. Two vectors of attributes, SA and MA, the cosine similarity and cosine (θ) are represented by using a dot product and magnitude as follows:

\[ \text{cosine similarity}(SA, MA) = \frac{\text{Dot Product} (SA, MA)}{|SA| \times |MA|} \]

Where Dot Product is:

\[ \text{Dot Product} (SA, MA) = SA [0] \times MA [0] + \ldots + SA [n] \times MA [n] \]

And Distances | | and |MA| is defined as:

\[ | | = [0] + [1] + \ldots + [ ] \]

And

\[ |MA| = MA [0] + MA [1] + \ldots + MA [n] \]

After the cosine similarity between the model answers and student answers are calculated, marks are assigned.

4. Automated Tagging

The following four tags were identified for the automatic generation of the question paper based on Blooms Taxonomy.

<table>
<thead>
<tr>
<th>Tags</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cognitive Level</td>
<td>Recall, Understand, Apply, Analyse, Evaluate, Create</td>
</tr>
<tr>
<td>Question Type</td>
<td>Fill in the blanks, Multiple choice, Match the following, True False, Answer in one word, Definition</td>
</tr>
<tr>
<td>Content</td>
<td>Topics and subtopic from syllabus</td>
</tr>
<tr>
<td>Difficulty Level</td>
<td>Low, Medium, High</td>
</tr>
</tbody>
</table>

a. Cognitive level Identification

It is the process to Understand, Percept, Evaluate, Analyse, Recall and Create the functioning of the system

b. Question-type Identification

Question-type is to identify the objective or subjective type of questions.
c. Content Identification
It is to identify the topics as well as subtopics.

d. Difficulty-Level Identification
The difficulty level of a question depends on concept involved, type of question, cognitive level and decide whether it is low, high or medium.

III. PROPOSED SYSTEM
We are presenting an Automated Question Paper Generator System and Answer Checker System that can reduce time consumption by replacing the traditional method of question paper generation.

B. Question paper generation system
The examiner will input the questions as per their need in their respective database record if they want to manipulate (Add, Delete or Change) the data through a GUI. Questions will get updated as per the function obtained. As this is a web-based application Faculty can set the difficulty level, structure of the answer, Chapters that they want to add according to their subject, Exam Score for how many marks they want to set their papers. The system will generate random questions of the chapters specified by extracting it from database by using algorithm the questions will be well organized

1. Randomized Algorithm
Randomized Algorithm checks the duplicate questions and use to display random questions. The algorithm is as followed, N = total no. of questions in the database. The randomized algorithm randomly generates these questions
Step 1: Create an array of N locations.
Step 2: Generate random number.
Step 3: if (lock==0)
Store generated number.
Else
Compare the generated number with previous number in array.
If matching value found, go to step 2;
Else
Store the no in next location.
Step 4: Repeat step 2 for N numbers.
Step 5: Select questions from DB matching with values from array location one by one [1].

B. Answer Checker System
The Student appearing for the test needs to login and appear for the test (Test will be of Subjective as well as objective pattern). The more accurate the answers the more he/she will earn the marks. The difference between model answer and student answer is done by using Jaro-Winkler and checks all the possible mistakes that the student has made and gives the final score of the test.
Answer Checking Process

1. Pre-processing
Pre-processing plays a very important role in answer checking. The pre-processing operations needed are segmentation, stop-words removal, normalization, finding synonyms and extracting roots.

   a. Tokenization
   Tokenization is needed to identify the end of each sentences. The sentences are ended with various punctuation marks that can be dot (.), comma (,), colons (:) etc.

   b. Stop-words Removal
   Stop-words are the most frequent words, which we use in our answer such as prepositions, articles, conjunctions and are not much useful for automatic scoring. Removal of these stop-words will improve the performance of the system.

   c. Normalization
   Normalization is needed to modify the text to make it definite as per needed by removing unnecessary characters, non-alphanumeric characters to improve the performance of the system.

   d. Root Extraction
   The sources of the keyword are extracted from both student answer as well as model answer.

   e. Jaro-Winkler distance
   Jaro-Winkler is used to compare strings by measuring the edit distance between two strings. The minimum the distance, the more the strings are similar. The score is normalize such that 0 is an exact match and 1 is for similarity. Jaro-Winkler similarity is 1 - Jaro-Winkler distance.

   \[ \text{simj} = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3} \left( \frac{m}{|s1|} + \frac{m}{|s2|} + \frac{m-t}{m} \right) & \text{otherwise} \end{cases} \]

   Where \(|s1| = \text{length of the strings}
   \quad M = \text{Number of matching characters}
   \quad T = \text{is half the number of transposition.}

   Two strings \(s1\) and \(s2\) are considered matching only if it is same and not greater than

   \[ \left[ \frac{\max(|s1|, |s2|)}{2} \right] - 1 \]

   Jaro-Similarity uses prefix scale (p) which gives ratings to the strings that are matched from very start. So for this Jaro-Winkler similarity will be

   \[ \text{Simw} = \text{simj} + \text{lp} \times (1 - \text{simj}) \]

   Where

   \( L \) is length of common prefix max up to 4 characters. \( p \) is constant scaling factor to adjust upwards to have common prefix. \( p \) should not be greater than 0.25 else the distance will become greater than 1. Standard value of \( p = 0.1 \)

   Jaro-Winkler distance is \( dw \).

   \[ dw = 1 - \text{simw} \]

1.6 Automatic Scoring
The similarity measure value is converted into a score using the following formula:

Mark = Similarity – Value * Mark

Similarity is calculated by Jaro-Winkler method.
IV. IMPLEMENTATION OF THE SYSTEM

1. Login Page

In our system login page is common for admin, staff as well as students. All the username and password are by default inserted in the database uniquely for all type of user (Admin, Faculty and Student).

2. Add Question and Answer

The faculty can insert additional questions in the database if required. They only need to enter subject name, Chapter Number of which the question belongs to. Then enter the question as well answer as the faculty need to give their model answer in the database. They have to select marks (1, 2, and 5) for
that particular question that they need to allocate. Select the difficulty level (Low, Medium, and High) and the type of question (MCQ, FIB, Define, and Brief).

3. **Generate Question Paper**

Faculty need to enter the Subject name that they need to generate. Pattern of the paper (20, 40, 60, 80). Chapter names are displayed faculty need to select at least two Chapters to generate the paper pattern.

Exam Date is selected when the exam is conducted. By clicking on generate question paper the question paper is generated with random questions as per the selected chapters.

3.1 **Sample Paper Generated Inputs**
Output

4. Student- Start Exam

Question Paper

- PLEASE SELECT -

- LIST IS EMPTY -

-
Student login and start its session when it needs to appear for the exam. Student need to select the question paper, which is hosted and then the exam session gets started by clicking on Let's Begin. With the help of marks of individual questions, total marks is then calculated automatically and result is displayed.

5. **Answer Sheet**

![Answer Sheet Image]

**Q.1**

a. __________ is a breadth-first search with all paths having some cost.

b. Which environment cannot provide end-less information, as the environment cannot be seen completely? a. Stochastic b. Discrete c. Continuous d. Partially Observable

c. __________

d. Forward state-space search is also called as __________

**Q.2**

a. What is Agent Function

b. Define Prognosis Planner

c. What is Expert System?

d. Define Planning

**Q.3**

a. List Types of Environment

b. List Uninformed Search Algorithms

c. What are IEAS Descriptors?
5.1 Sample Answer Generated

**Inputs**

![Image of question paper input]

**Output**

![Image of answer paper output]
V CONCLUSIONS

We have implemented automatic question paper generation using randomized algorithm. System can generate 20, 40, 60 and 80 marks of question papers with various types of questions like MCQ’s (Multiple Choice Questions), Fill in the Blanks, Answer in one sentence and Short answers also. The answer checking system generates the marks for all the individual questions based on the similarity measure between the student answer and the model answer. This system can be used further in various colleges and schools to reduce their work and effective time utilization.

REFERENCES


