RESEARCH ARTICLE OPEN ACCESS

BloomQGen: An Intelligent Question Paper Generation System Using Machine Learning and Bloom's Taxonomy

Pritesh Premnath Waghmare*, A. H Auti**
*(Department of Computer Engineering, SAE Kondhwa,
Email: Priteshwaghmare8855@gmail.com)
** (Department of Computer Engineering, SAE Kondhwa,)

--************************----------------------------------

Abstract:

Preparing examination question papers manually is a time-consuming and challenging task for educators. Faculty members must ensure balanced syllabus coverage, appropriate marks distribution, and proper assessment of students across different cognitive levels. Manual preparation often results in inconsistencies, repeated questions, and increased workload. This paper presents BloomQGen, a web-based intelligent question paper generation system that automates the process using Machine Learning techniques and Bloom's Taxonomy. The proposed system preprocesses question datasets, converts textual questions into numerical features using Count Vectorizer, predicts appropriate marks through a Linear Regression model, and categorizes questions according to Bloom's cognitive levels using predefined action verbs. Based on user-defined constraints such as subject, total marks, and Bloom's Taxonomy distribution, the system automatically generates balanced and syllabus-oriented question papers. The application is implemented using the Django framework with Python as the backend and SQLite as the database. The proposed solution reduces manual effort, improves consistency in assessment design, and supports Outcome-Based Education (OBE) by enabling structured and balanced examination paper generation.

Keywords — **Machine Learning, Bloom's Taxonomy, Question Paper Generation, Educational Assessment, Linear Regression, Count Vectorizer, Django Framework, Natural Language Processing.**

--************************----------------------------------

I. INTRODUCTION

The rapid advancement of digital technologies has transformed various aspects of education, including teaching, learning, and assessment. Despite these developments, the preparation of examination question papers in many educational institutions continues to be a manual process. Faculty members are required to select suitable questions, assign appropriate marks, ensure complete syllabus coverage, and maintain a balanced distribution of cognitive difficulty levels. This process is not only time-consuming but is also prone to inconsistencies, repetitive questions, and subjective decision-making.

Bloom's Taxonomy provides a widely accepted educational framework for classifying learning objectives into six cognitive levels: Remember, Understand, Apply, Analyze, Evaluate, and Create. These levels help educators design assessments that measure different degrees of student understanding rather than focusing only on memorization. However, manually categorizing questions according to Bloom's Taxonomy requires considerable effort and expertise, making it difficult to maintain consistency across different examinations.

Machine Learning and Natural Language Processing (NLP) offer effective solutions for automating text-based educational tasks. Text preprocessing techniques and feature extraction methods enable computers to analyze question statements and assist in decision-making. By combining these techniques with educational frameworks such as Bloom's Taxonomy, it is possible to develop intelligent systems capable of supporting educators in generating balanced and standardized examination papers.

This paper presents **BloomQGen**, a web-based intelligent question paper generation system designed to simplify the examination paper preparation process. The proposed system accepts question datasets, preprocesses textual data, predicts suitable marks using a Machine Learning regression model, categorizes questions according to Bloom's Taxonomy using predefined cognitive action verbs, and automatically generates balanced question papers based on user-defined constraints. The application is developed using the Django framework, Python, Bootstrap, and SQLite, providing a user-friendly interface for managing question banks and generating examination papers efficiently.

The major objectives of the proposed system are:

- To automate the question paper generation process.
- To reduce the manual workload involved in preparing examination papers.
- To maintain balanced syllabus coverage and marks distribution.
- To classify questions according to Bloom's Taxonomy.
- To support Outcome-Based Education (OBE) through standardized assessment.

The proposed solution provides an efficient and practical approach for educational institutions by reducing manual effort while improving the consistency, reliability, and quality of examination paper preparation.

II. LITERATURE REVIEW

1. The automation of examination systems has become an active area of research due to the increasing demand for efficient and standardized assessment methods. Several researchers have proposed systems that utilize Natural Language Processing (NLP), Machine Learning (ML), and educational frameworks to improve question generation, classification, and evaluation. However, most existing approaches focus on only one aspect of the examination process, such as automatic question generation or cognitive-level classification, while very few integrate all these functionalities into a single system.
2. Benjamin S. Bloom introduced Bloom's Taxonomy in 1956 as a hierarchical framework for classifying educational objectives into six cognitive levels: Remember, Understand, Apply, Analyze, Evaluate, and Create. The taxonomy has since become a standard guideline for designing examinations that assess different levels of student learning rather than focusing solely on factual recall.
3. Rus et al. proposed automatic question generation techniques using Natural Language Processing for educational applications. Their work demonstrated that NLP could effectively generate questions from textual documents. However, the generated questions required additional manual validation to ensure educational quality and appropriate cognitive-level distribution.
4. Heilman and Smith developed an automatic question generation framework that combines syntactic and semantic analysis to produce educational questions from textual resources. Although their approach improved the quality of generated questions, it primarily focused on question extraction and did not address balanced marks allocation or examination paper preparation.
5. Pedregosa et al. introduced Scikit-learn, an open-source Machine Learning library that provides efficient algorithms for classification, regression, clustering, and feature extraction. The library has become one of the most widely adopted Machine Learning frameworks in educational data mining and text analytics due to its

simplicity, flexibility, and computational efficiency.

6. Jurafsky and Martin extensively discussed Natural Language Processing techniques such as text preprocessing, tokenization, feature extraction, and language modeling. Their work established a strong foundation for applying NLP methods to educational applications, document classification, and intelligent information retrieval.
7. Despite these advancements, existing systems often lack an integrated solution capable of classifying examination questions according to Bloom's Taxonomy, predicting suitable marks, and automatically generating balanced question papers. The proposed BloomQGen system addresses these limitations by combining Machine Learning-based marks prediction with Bloom's Taxonomy-based cognitive classification within a unified web application, thereby reducing manual effort while improving assessment quality and standardization.

TABLE I

COMPARISON OF EXISTING STUDIES

Authors	Technique Used	Limitation
Bloom (1956)	Bloom's Taxonomy	No automation for question generation
Rus et al.	NLP-based Question Generation	Limited cognitive-level assessment
Heilman & Smith	Automatic Question Generation	No marks prediction mechanism
Pedregosa et al.	Scikit-learn Machine Learning Library	General-purpose ML library; not specific to education
Proposed BloomQGen	Machine Learning + Bloom's Taxonomy	Generates balanced question papers with reduced manual effort

III. PROPOSED METHODOLOGY

The proposed BloomQGen system follows a structured methodology to automate the generation of examination question papers using Machine Learning and Bloom's Taxonomy. The system processes a question dataset through multiple stages, including data preprocessing, feature extraction, marks prediction, Bloom's level identification, and intelligent question paper generation. The overall workflow of the proposed methodology is illustrated in Fig. 1.

The primary objective of the methodology is to reduce manual intervention while ensuring balanced marks distribution, syllabus coverage, and cognitive-level representation in the generated examination paper.

1. DATASET COLLECTION

The system uses a structured dataset containing subject-wise examination questions collected from previous university examinations and institutional question banks. Each record includes the question statement and the corresponding marks information required for model training and question selection. The dataset is maintained in CSV format, allowing easy modification and future expansion.

2. DATA PREPROCESSING

The collected question dataset undergoes preprocessing to improve data quality before analysis. During this stage, unwanted characters, punctuation symbols, extra spaces, and formatting inconsistencies are removed. The text is standardized to ensure uniform representation of examination questions. This preprocessing step improves the quality of feature extraction and increases the reliability of subsequent predictions.

3. FEATURE EXTRACTION

After preprocessing, the textual questions are transformed into numerical representations using the Count Vectorizer technique. Count Vectorizer creates a vocabulary from the complete dataset and represents each question as a vector based on the frequency of words present in the question text. This numerical representation enables Machine Learning algorithms to process textual information effectively.

4. MARKS PREDICTION

The extracted feature vectors are provided to a Linear Regression model to estimate suitable marks for each examination question. The regression model learns the relationship between textual features and marks from the available training dataset and predicts appropriate marks for newly added questions. This assists in maintaining consistent marks allocation throughout the generated question paper.

5. BLOOM'S TAXONOMY CLASSIFICATION

The cognitive level of each question is identified using Bloom's Taxonomy action verbs. The system examines the keywords present in each question and categorizes it into one of the six Bloom's Taxonomy levels: Remember, Understand, Apply, Analyze, Evaluate, or Create. This classification helps maintain a balanced cognitive distribution in the final examination paper.

6. QUESTION PAPER GENERATION

Finally, the generated marks and Bloom's Taxonomy levels are combined with user-defined constraints such as subject selection and total marks. The system intelligently selects appropriate questions while avoiding duplication and maintaining balanced cognitive-level distribution. The generated question paper is displayed through the web interface and stored in the database for future reference.

Algorithm 1: Proposed BloomQGen Workflow
Input:
Question Dataset (CSV)
Step 1:
Load the question dataset into the system.
Step 2:
Preprocess the textual questions by removing unwanted characters and formatting inconsistencies.
Step 3:
Convert textual data into numerical vectors using Count Vectorizer.
Step 4:
Predict appropriate marks using the Linear Regression model.
Step 5:
Determine Bloom's Taxonomy level using predefined cognitive action verbs.
Step 6:
Select questions according to subject, marks, and Bloom's Taxonomy distribution.
Step 7:
Generate and display the final examination question paper.
Output:
Balanced Question Paper
Mathematical Model
Let

$$Q = \{q_1, q_2, q_3, \ldots, q_n\}$$

represent the collection of examination questions.
The proposed system can be represented as

$$Q \rightarrow P \rightarrow F \rightarrow M \rightarrow B \rightarrow G$$

where
Q = Input Question Dataset
P = Data Preprocessing
F = Feature Extraction (Count Vectorizer)
M = Marks Prediction
B = Bloom's Taxonomy Classification
G = Generated Question Paper
The output is

$$O = G$$

where O represents the automatically generated balanced examination question paper.

Fig. 1

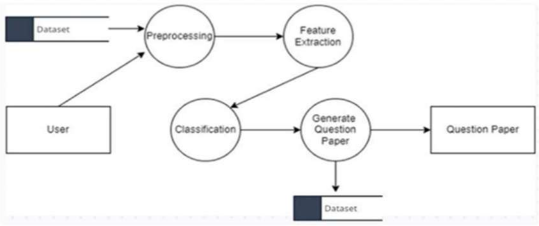

Fig. 1: DFD Level 1

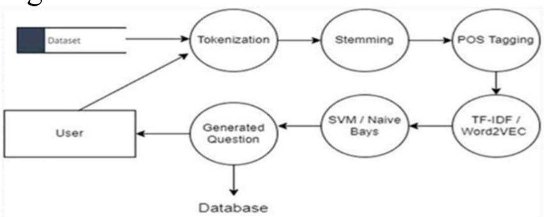

Fig. 1.1: DFD Level 2

IV. SYSTEM ARCHITECTURE

The proposed BloomQGen system is designed using a modular architecture that integrates user interaction, data preprocessing, Machine Learning, Bloom's Taxonomy analysis, and automated question paper generation. The architecture follows a client-server model in which the user interacts with the web application through a browser, while the backend processes requests and generates the required examination paper.

The frontend of the application is developed using HTML, CSS, Bootstrap, and JavaScript, providing an intuitive interface for user authentication, question bank management, and question paper generation. The backend is implemented using the Django framework, which manages business logic, user requests, Machine Learning integration, and database operations. SQLite is used as the relational database to store user information, generated papers, and application data.

The complete system architecture consists of five major modules:

User Management Module

This module manages user authentication, registration, and login. Authorized users can securely access the system, upload question datasets, generate question papers, and view previously generated papers. The authentication mechanism ensures that only registered users can access the application.

Dataset Management Module

The dataset management module allows users to upload subject-wise question banks in CSV format. The uploaded datasets are validated before processing and are used as the primary source for question selection. This module also maintains the consistency and organization of the available question repository.

1. MACHINE LEARNING MODULE

   The Machine Learning module performs preprocessing, feature extraction, and marks prediction. Initially, the uploaded question text is cleaned and standardized. Count Vectorizer converts the textual questions into numerical feature vectors. These vectors are supplied to a Linear Regression model, which predicts suitable marks for each examination question.

2. BLOOM'S TAXONOMY ANALYSIS MODULE

   This module analyzes each question by identifying predefined cognitive action verbs associated with Bloom's Taxonomy. Based on these action verbs, the system categorizes every question into one of the six cognitive levels: Remember, Understand, Apply, Analyze, Evaluate, or Create. This classification helps maintain balanced cognitive-level distribution during question paper generation.

3. QUESTION PAPER GENERATION MODULE

   The final module combines the predicted marks, Bloom's Taxonomy levels, and user-defined constraints such as subject selection and total marks. Based on these parameters, the system selects appropriate questions while minimizing duplication and ensuring balanced assessment. The generated question paper is displayed through the web interface and stored in the database for future access.

ADVANTAGES OF THE PROPOSED ARCHITECTURE

The proposed architecture provides the following benefits:

Modular and scalable system design.
Easy integration of additional subjects and datasets.
Reduced manual intervention in examination paper preparation.
Improved consistency in marks allocation and cognitive-level distribution.
Efficient storage and retrieval of generated question papers.

V. IMPLEMENTATION AND RESULTS

The BloomQGen system has been developed as a web-based application using the Django framework and Python programming language. The implementation integrates Machine Learning techniques with Bloom's Taxonomy to automate the process of question paper generation. The application follows a modular architecture in which each module performs a specific task while communicating seamlessly with the others.

The frontend of the system is designed using HTML, CSS, Bootstrap, and JavaScript, providing a simple and user-friendly interface for faculty members. The backend is implemented using Django, which handles user authentication, dataset processing, database operations, and question paper generation. SQLite is used as the backend database to store user information, uploaded datasets, and generated examination papers.

Initially, the administrator uploads the subject-wise question dataset in CSV format. The uploaded dataset is preprocessed to remove unwanted characters and formatting inconsistencies. The cleaned textual data is converted into numerical feature vectors using Count Vectorizer. These feature vectors are then supplied to the Linear Regression model, which predicts suitable marks for each question.

After marks prediction, the system identifies the Bloom's Taxonomy level of each question by examining predefined cognitive action verbs. Questions are categorized into the appropriate cognitive level, allowing the system to maintain balanced cognitive distribution during question paper generation.

Based on the subject selected by the user and the required marks distribution, the system automatically selects appropriate questions while avoiding unnecessary repetition. The generated examination paper is displayed through the web interface and can also be stored for future reference.

The implementation successfully automates several activities that are traditionally performed manually by faculty members. The developed application demonstrates that integrating Machine Learning with educational assessment frameworks can significantly simplify question paper preparation while maintaining consistency and standardization.

A. Experimental Results

The developed application was tested using subject-wise examination question datasets. Various test cases were executed to verify the correctness of user authentication, dataset management, marks prediction, Bloom's Taxonomy classification, and question paper generation.

The experimental observations indicate that the proposed system successfully generates structured question papers based on the user-selected subject and predefined examination constraints. The generated papers maintain balanced marks distribution and include questions from different Bloom's Taxonomy levels, thereby supporting comprehensive student assessment.

Table II

Functional Testing Summary

Module	Status
User Authentication	Successfully Implemented

Module	Status
Dataset Upload	Successfully Implemented
Data Preprocessing	Successfully Implemented
Marks Prediction	Successfully Implemented
Bloom's Taxonomy Classification	Successfully Implemented
Question Paper Generation	Successfully Implemented
Database Storage	Successfully Implemented

Table III

Technology Stack

Component	Technology Used
Programming Language	Python
Web Framework	Django
Frontend	HTML, CSS, Bootstrap, JavaScript
Database	SQLite
Feature Extraction	Count Vectorizer
Marks Prediction	Linear Regression
Bloom's Classification	Bloom's Action Verb Mapping

Discussion

The proposed BloomQGen system significantly reduces the manual effort involved in examination paper preparation. By automating marks prediction and Bloom's Taxonomy classification, the system assists faculty members in generating standardized question papers within a shorter time compared to traditional manual methods. The modular implementation also allows future enhancements such as additional Machine Learning models, support for multiple universities, and cloud-based deployment.

VI. CONCLUSION AND FUTURE SCOPE

The proposed BloomQGen system successfully automates the process of examination question paper generation by integrating Machine Learning techniques with Bloom's Taxonomy. The developed web application reduces the manual effort involved in preparing examination papers while ensuring balanced marks distribution and cognitive-level representation. By utilizing Count Vectorizer for feature extraction, Linear Regression for marks prediction, and Bloom's action verbs for cognitive-level classification, the system generates structured and standardized question papers based on user-defined requirements.

The modular architecture of the proposed system enables easy management of question datasets, efficient processing of examination questions, and rapid generation of balanced question papers through a user-friendly web interface. The implementation demonstrates that educational assessment can be significantly improved through intelligent automation while supporting the principles of Outcome-Based Education (OBE).

Although the current implementation provides an effective solution for automated question paper generation, several enhancements can be incorporated in future versions. Advanced Natural Language Processing techniques and transformer-based language models may be integrated to improve semantic understanding of examination questions. Additional Machine Learning models can be explored to enhance marks prediction and cognitive-level classification accuracy. Future work may also include support for multiple universities, additional academic disciplines, multilingual question banks, cloud-based deployment, and automatic PDF generation with customizable examination templates.

Overall, the proposed BloomQGen system provides an efficient, scalable, and practical solution for educational institutions seeking to modernize the examination paper preparation process while reducing manual workload and improving assessment quality.

ACKNOWLEDGMENT

The authors express their sincere gratitude to the Department of Computer Engineering, Sinhgad Academy of Engineering, Pune, for providing the facilities and academic support required for the completion of this work. The authors also thank the project guide and faculty members for their valuable suggestions and continuous encouragement throughout the development of the proposed system.

REFERENCES

[1] B. S. Bloom, Taxonomy of Educational Objectives: The Classification of Educational Goals, Handbook I: Cognitive Domain. New York, NY, USA: Longmans, Green, 1956.

[2] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 4th ed. Pearson, 2021.

[3] C. D. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval. Cambridge, U.K.: Cambridge University Press, 2008.

[4] D. Jurafsky and J. H. Martin, Speech and Language Processing, 3rd ed. Pearson, 2023.

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," arXiv preprint arXiv:1301.3781, 2013.

[7] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning, 2nd ed. New York, NY, USA: Springer, 2009.

[8] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow, 3rd ed. Sebastopol, CA, USA: O'Reilly Media, 2022.

[9] Django Software Foundation, "Django Documentation." [Online]. Available: https://docs.djangoproject.com/

[10] Python Software Foundation, "Python Documentation." [Online]. Available: https://docs.python.org/3/

[11] NumPy Developers, "NumPy Documentation." [Online]. Available: https://numpy.org/

[12] Pandas Development Team, "Pandas Documentation." [Online]. Available: https://pandas.pydata.org/

[13] Scikit-learn Developers, "CountVectorizer." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

[14] World Wide Web Consortium (W3C), "HTML Living Standard." [Online]. Available: https://html.spec.whatwg.org/

[15] Bootstrap Team, "Bootstrap Documentation." [Online]. Available: https://getbootstrap.com/docs/