AN NLP Driven Approach to Optimizing Multiple Choice Question Generation

Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

MASTER OF TECHNOLOGY

in

Computer Science Engineering

by

Vaibhav Kumar

(Roll No. 23/CSE/21)

Under the Supervision of

Dr. Manoj Sethi

Department of
Computer Science and Engineering
Delhi Technological University

To the
Department of Computer Science and Engineering

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-110042, India

May, 2025

CANDIDATE'S DECLARATION

I **Vaibhav Kumar, 23/CSE/21** student of M.Tech hereby declare that the project Dissertation titled "**An NLP Driven Approach to Optimizing Multiple Choice Question Generation**" which is submitted by me to the **Department of Computer Science and Engineering, Delhi Technological University, Delhi** in partial fulfillment of the requirement for the award of degree of **Master of Technology**, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: **Delhi**
Date:

Vaibhav Kumar
23/CSE/21

CERTIFICATE

I hereby certify that the Project Dissertation titled "**An NLP Driven Approach to Optimizing Multiple Choice Question Generation**" which is submitted by **Vaibhav Kumar, 23/CSE/21**, **Department of Computer Science and Engineering**, **Delhi Technological University, Delhi** in partial fulfillment of the requirement for the award of the degree of **Master of Technology**, is a record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: **Delhi** **Dr. Manoj Sethi**
Date:

ABSTRACT

The MCQ (Multiple Choice Question) Generator project aims to streamline and automate the process of generating high-quality objective questions from textual content or academic material. This project addresses the growing need for scalable and efficient methods of assessment creation in educational institutions, online learning platforms, and corporate training environments. Traditional methods of MCQ creation are time-consuming, require significant human effort, and are prone to inconsistency in difficulty and coverage. By leveraging Natural Language Processing (NLP) and machine learning techniques, this project proposes a system capable of generating grammatically correct, semantically meaningful, and pedagogically sound MCQs from input text.The essential functionality of the MCQ Generator encompasses various major components: text preprocessing, keyword extraction, question construction, distractor creation, and quality assessment. The process starts with preprocessing the input document(e.g., textbook chapter or article) in order to eliminate noise and extract meaningful information. Key terms or ideas are extracted via keyword extraction methods like TF-IDF. A paramount part of the system is producing the credible distractors-wrong options close in the context to the right option but easily identifiable. For the ease of use, the project has a user interface for the users(educators, instructors, or content developers) to enter raw text set the number of questions.

ACKNOWLEDGEMENT

I wish to express my sincerest gratitude to **Dr. Manoj Sethi** for his continuous guidance and mentorship that he provided us during the project. His direction helped me understand and accomplish my objectives, emphasizing the significance and industrial relevance of this project. He was always available to assist and clarify any doubts we encountered. This project would not have been successful without his continuous support and encouragement.I would like to extend my sincere thanks to **" Prof. Manoj Kumar"**, *Head of Department*, Department of Computer Engineering, Delhi Technological University, Delhi for his valuable suggestions and feedback on our project work.Finally, I would like to thank all the participants who participated in the study, without whom this research would not have been possible. I express my sincere gratitude to all the individuals who have directly or indirectly contributed to the success of my project.

Vaibhav Kumar
(23/CSE/21)

Table of Contents

List of Figures

CHAPTER 1
INTRODUCTION

Multiple Choice Questions (MCQs) are a widely accepted method for evaluating knowledge across educational, competitive, and corporate training sectors. The manual creation of MCQs, however, is both time-consuming and labor-intensive, often resulting in inconsistencies in question quality and relevance. With advancements in Natural Language Processing (NLP) and machine learning, it is now feasible to automate this process to a large extent. This project explores an AI-driven approach to generate MCQs from input text, aiming to provide educators with a scalable tool for rapid assessment creation while maintaining question validity, variety, and pedagogical value.

Technologies Used:

1. Natural Language Processing (NLP):

(i) spaCy and NLTK for text preprocessing, POS tagging, and syntactic parsing.
(ii) TF-IDF for keyword extraction from large text.

2. Web Development:

(i) Flask for building the backend API.
(ii) ReactJS for frontend development.

1.1 Existing Systems

1.1.1 Automated Question Generation from Textual Data using Natural Language Processing Techniques

They came up with a web application that can automatically create questions from user-supplied content. The tool is intended to automate educators' workflows by allowing them to process large syllabi effectively and create different types of questions from the supplied material.[1]

1.1.2 Generation of MCQ from Textbook contents of School Level Subjects

They introduced a modular pipeline to automatically generate MCQs from middle-school textbook content, the structure of which is partially independent of the content. The pipeline consists of four primary stages: text preprocessing, sentence selection, keyword extraction, and distractor generation. Every stage encompasses a range of methods, including sentence simplification, syntactic and semantic parsing, named entity extraction, mapping of semantic relationships, and lexical and embedding resource usage including WordNet and neural embeddings for sentences and words. The performance of the system was tested with textbooks provided by the NCERT, India, over three subjects. Human judges were used to determine the quality of the generated MCQs based on both component-level and system-level evaluation metrics. The findings show that the suggested approach has the ability to produce good-quality questions appropriate for use in real educational tests.[2]

1.1.3 Automated Question Paper Generation using Natural Language Processing

They developed an Automated Question Paper Generation system to provide high-quality practice support in education. The system leverages NLP methods for context comprehension and question generating, which involve paradigmatic analysis, dependency, and constituency parsing, both syntactic and semantic, for example, BERT and LDA. Attention mechanisms in GPT and BERT models are among the elements that have been added to the models to have a better way of generating questions. Evaluation metrics like BLEU, ROUGE, Precision, Recall, and Semantic Similarity scores are used to evaluate the system’s competency to produce high-quality questions and answers.[3]

1.2 Problem Statement

The construction of effective multiple-choice questions (MCQs) is both labor-intensive and cognitively taxing, particularly within educational environments where teachers must measure student acquisition across expansive syllabi. Historically, MCQs are constructed by hand, which entails the selection of pivotal concepts, creating grammatically appropriate questions, and creating credible distractors (incorrect answers). Hand construction proves cumbersome when scalability and reliability are needed, especially for institutions with masses of content or recurring testing.

Even with the developments in Natural Language Processing (NLP) and machine learning, current automatic MCQ generation tools are either very domain-specific or are not versatile enough to produce contextually correct and pedagogically sound questions from school or college-level textbooks. Most of these tools do not generalize across subjects or do not produce good distractors that are neither too straightforward nor too confusing. Further, not many systems emphasize formulating questions that conform to learning standards and testing goals.

The absence of a strong, subject-independent, and scalable system for generating MCQs is an evident shortage in today's educational technology. Hence, it is imperative to design an intelligent, NLP-based MCQ generation system that can automatically process the input text, extract informative content, and create significant questions and answer choices. This system must be flexible enough to handle a range of subjects and levels of instruction, thus relieving the teachers from too much work while providing qualitative evaluation materials to learners.

1.3 My Contributions

In this project, I have made notable contributions to the design, development, and testing of an intelligent web-based system that can automatically generate Multiple Choice Question (MCQ) by employing Natural Language Processing (NLP) techniques. The principal contributions of my research are as follows:

1.3.1 Design and Implementation of an End-to-End MCQ Generation Pipeline:

I envisioned and designed a modular pipeline consisting of preprocessing, sentence selection, key term identification, question formation, and distractor generation. Each module is isolated so that the system can scale and adapt to different domains.

1.3.2 Web Based Application Development:

Created a user friendly web-based interface through which teachers can input raw text or upload any content and receive MCQ's in real time. Frontend was designed to be very convenient to use, and the backend integrates the NLP pipeline harmoniously.

1.3.3 Distractor Generation:

Developed methods for generating esteemed quality distractors via the WordNet, semantic similarity models, and embedding based velocity measures.These techniques ensure that distractors are contextually valid, grammatically correct and most importantly user friendly

1.3.4 Conceptual Background

1. NLP, Python, Flask
2. HTML
3. CSS
4. ReactJS

CHAPTER 2
LITERATURE REVIEW

2.1 Overview

Automatic Multiple Choice Question (MCQ) generation has been a topic of ongoing research at the confluence of Natural Language Processing (NLP) and educational technology. Initial methods mainly used rule-based systems and template-matching algorithms, which were not scalable and flexible across domains. With the development of statistical and machine learning-based NLP techniques, researchers have been looking at more dynamic methods for question generation, keyword extraction, and distractor selection.

A number of approaches have targeted creating factual MCQs from formal knowledge sources like Wikipedia and ontologies. Others have relied on domain-specific corpora like science or history textbooks, which tend not to be generalizable. Popular models draw upon methods such as named

entity recognition (NER), dependency parsing, and semantic similarity measures with tools like WordNet or neural word embeddings. Recent advances have made use of pre trained transformer models like BERT and GPT in order to enhance semantic comprehension and context-sensitivity in question framing.

While these developments have been made, current systems are still challenged to produce good-quality distractors and maintaining grammaticality. Additionally, most are used for particular subjects or heavily depend on formatted input. Not much research has investigated generating MCQs from school-level textbook text that is in unstructured form and preserving relevance, clarity, and cognitive validity.

This project extends prior work and overcomes shortcomings in domain independence and educative use. With advanced NLP techniques combined with a modular pipeline, our system will create pedagogically beneficial MCQs from raw text and provide enhanced flexibility, scalability, and question quality.

2.2 Related Work

The Authors, Pranavesh et al (2024) [4] proposed an Automated Multiple-Choice Question (MCQ) Generator that uses natural language processing (NLP) techniques to analyze text content and create questions.Using Transformers, KeyBERT, and NLTK, the system preprocesses text, extracts keywords, and links them to relevent phrases. Direct sentence selection is made possible by KeyBERT's BERT-based embeddings which extract significant keywords. KeyBERT's BERT-based sentence embeddings are used to choose pertinent sentences by extracting crucial keywords. The system uses a BERT-based MLM to anticipate word senses in order to ensure semantic accuracy. Blank substitution in sentences facilitates the development of MCQ by providing both correct and incorrect responses. Responses from users are stored fro evaluation and comments.

The authors, Himanshu et al (2024) [5] investigated a variety of question formats, including as matching true false, fill in the blanks ,and question-answer pairs, while maintaining the essential Content of NCERT's by applying the T5 Transformer model. With the fill-in-the-blanks feature, keywords are extracted and omitted words are used to create sentences. To generate question-answer pairings, a T5 transformer model customized fro NCERT materials must be trained on the SQuAD dataset. In order to ensure relevance to NCERT material, true-false questions use natural language processing techniques with GPT 2 and BERT embeddings. Their research uses the BERT paradigm to connect text processing libraries for tasks like word meaning disambigution and keyword extraction. The study provides a variety of assessment tools and demonstrates the process for developing and coordinating varied questions.The initiative which is centered on NCERT attempts to produce questions that correspond with the curriculam.

The authors Ardiansyah et al (2025) [6] investigated a variety of question formats, including as matching, true-false, fill-in-the-blanks, and question-answer pairs, while maintaining the essential content of NCERTs by applying the T5 transformer model. With the fill-in-the-blanks feature,keywords are extracted and omitted words are used to create sentences. To generate question-answer pairings, a T5 transformer model customized for NCERT materials must be trained on the SQuAD dataset. In order to ensure relevance to NCERT material, true-false questions use naturallanguage processing techniques with GPT-2 and BERT embeddings. Their research uses the BERT paradigm to connect text processing libraries for tasks like word meaning disambiguation and keyword extraction. The study provides a variety of assessmenttools and demonstrates the process for developing and coordinating varied questions. The initiative, which is centered on NCERTs,attempts to produce questions that correspond with the curriculum.

The authors Budati et al (2024) [7] developed An automated technique for creating question papers was designed to offer top-notch practice assistance in the classroom. The system makes use of NLP techniques, such as paradigmatic analysis, dependence, and constituency parsing, both syntactic and semantic (e.g., BERT and LDA), for context comprehension and question generation. Among the components that have been introduced to the GPT and BERT models to improve question generation are attention processes. The system's ability to generate high-quality questions and answers is assessed using evaluation metrics such as BLEU, ROUGE, Precision, Recall, and Semantic Similarity scores.

Authors Akhil killawala et al (2018) [8] created the framework for computational intelligence that should automate or partially automate the creation of quiz and exam questions. NLP algorithms and information retrieval constitute the foundation of the framework's operation. In addition to other clever methods, it applies LSTM neural network models and production criteria. It enables the creation of "Wh"-type (What? When? How?) questions, multiple choice questions, and true/false questions. Automation processes are created, shown, and evaluated for every kind of inquiry creation.The common problems in developing and implementing frameworks are taken into account, and potential fixes are talked about. We present and analyze the framework application's findings and how it was used to create a quiz in an actual college course.

The Authors Balika J Cheliah et al. (2024)[9] suggested method, which makes use of pretrained models like T5 small and BART, not only automatically generates contextually relevant questions but also assesses the models' performance using both conventional metrics like BLEU and ROUGE and comprehensive NLP metrics like METEOR and BERT-Score. In order to improve performance in QG and QA tasks, the suggested methodology includes text extraction from PDFs, preprocessing and tokenization for model compatibility, and fine-tuning of models using domain specific datasets. The study offers data handling and model evaluation modifications that greatly increase the efficacy and efficiency of the question answer generation process. The ability of the models to provide pertinent,semantically rich queries and answers is shown by evaluating the method using a combination of lexical and semantic metrics.

The Authors, Altaj Virani et al. (2023) [10] suggested a QAG system that creates excellent question-answer pairs by using a cutting-edge language model. Wh-questions, fill-in-the-blank, full-sentence,multiple-choice, and true-false questions are among the question kinds supported by the system. The system can provide precise and pertinent questions and answers for a given text, according to the results of tests conducted on a variety of text-based datasets. In educational, industrial, and scientific contexts where rapid and effective text understanding is essential, the QAG system has great promise. Experiments have shown the system's effectiveness and usability, and it is seen to have the potential to be a useful tool for a variety of natural language processing jobs.

The Authors Dhanamjaya Pochiraju et al. (2023) [11], suggested a novel method for text summarizing that uses NLP models such as XLNet and YAKE to extract keywords. Lexical databases like Concept Net and WordNet are used to generate distractors. In order to create questions and options for the multiple-choice questions, sentence mapping is utilized to determine the key ideas and connections in a text. The result is a collection of multiple-choice questions (MCQs) that can be used for training and teaching and are semantically related to the input material.

The author, Pushpa M. Patil (2022) [12], discusses the problems with automatically generating questions in the context of Marathi text's subordinate conjunctions. At the sentence level, they looked into the usefulness of conjunctions in the context of AQG. Three steps make up our AQG engine: automatic question creation, clause selection, and question word selection. The AQG methodology's main concept is to only create inquiries at the parts of speech (POS) tagging level. They used a test corpus of 100 Marathi complex sentences to evaluate the AQG methodology. They employed six subordinating conjunctions to evaluate the AQG approach. Their AQG methodology's performance is assessed using manually assessed syntactic and semantic correctness.

The author, Himanshu Jasuja et al. (2024) [13] strive to offer a lightweight and efficient paradigm for video question generation. They make use of cutting-edge Natural Language Processing (NLP) technology to enhance the adaptability of our model and enable T5 transformers for fine-tuning.Additionally, their method generates multiple-choice questions (MCQs) and several types of "Wh" questions, including who, when, where, what, which, why, and how.

The Authors Pranita Yadav et al. (2024) [14] study presents a standard approach for automated question development based on phrase structure and meaning analysis. Questomatic is a novel system that uses the BERT model, sophisticated Natural Language Processing (NLP) tools, and a custom randomization algorithm to generate unique, contextually accurate, and diverse questions from input text. A number of NLP libraries, like as NLTK and Spacy, are used for tasks like tokenization, stemming, lemmatization, and punctuation. BERT, a powerful natural language processing tool, has also been applied to sentiment analysis, named entity recognition, machine translation, and other domains. BERT's advanced capabilities have considerably benefitted NLP and paved the way for further advancements. Part-of-speech tagging, which determines the appropriate tags for text, is a key component of natural language processing.

Dhawaleswar Rao CH and Sujan Kumar Saha (2022) [15] suggested a partially subject-independent pipeline for automatically creating multiple-choice questions (MCQs) from middle school textbooks. Preprocessing, phrase selection, key selection, and distractor generation are the four main modules that make up the suggested pipeline. Individual modules have been implemented using a variety of methods.Sentence simplification, entity recognition, semantic relationship extraction between entities, neural word embedding, neural sentence embedding, WordNet, intersentence similarity calculation, and syntactic and semantic processing of sentences are some of these. Human experts use a variety of system-level and individual module-level measures to evaluate the quality of questions generated by the system.The outcomes of the experiment show that the suggested system can provide high-quality questions that would be helpful in an actual test.

CHAPTER 3
METHODOLOGY

The approach of this project is centered on developing a strong, modular pipeline for creating Multiple Choice Questions (MCQs) from raw textual content based on Natural Language Processing (NLP) technologies. The approach contains some phases that operate sequentially to scan input content, recognize useful information and tranform it into high quality MCQ's. Below is the in-depth description:

3.1 Data Acquisition and Input Handling :

The first stage of the system involves acquiring and handling input data. The user can provide content in the form of plain text or upload educational material such as textbook excerpts, lecture notes, or learning articles.

Stage 1 : Input Format:

Accepted input: raw text, .txt files, PDF documents (with text extraction capabilities).

The input is cleaned and converted into a plain text format for processing.

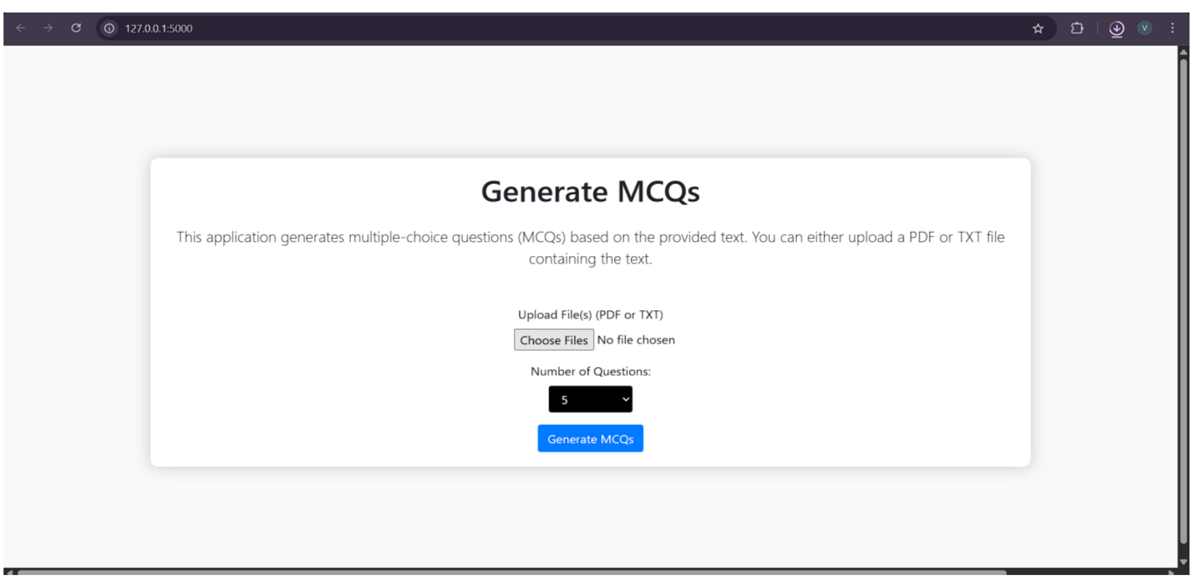

Fig 1. Input when no file is chosen.

Stage 2 : Preprocessing:

- **Text Cleaning:** Removal of special characters, HTML tags, and extra white spaces.

- **Lowercasing:** Standardizing all text to lowercase for uniform processing.

- **Sentence Segmentation:** Splitting paragraphs into individual sentences using NLP libraries like SpaCy or NLTK.

- **Tokenization:** Breaking down sentences into words or tokens for further linguistic analysis.

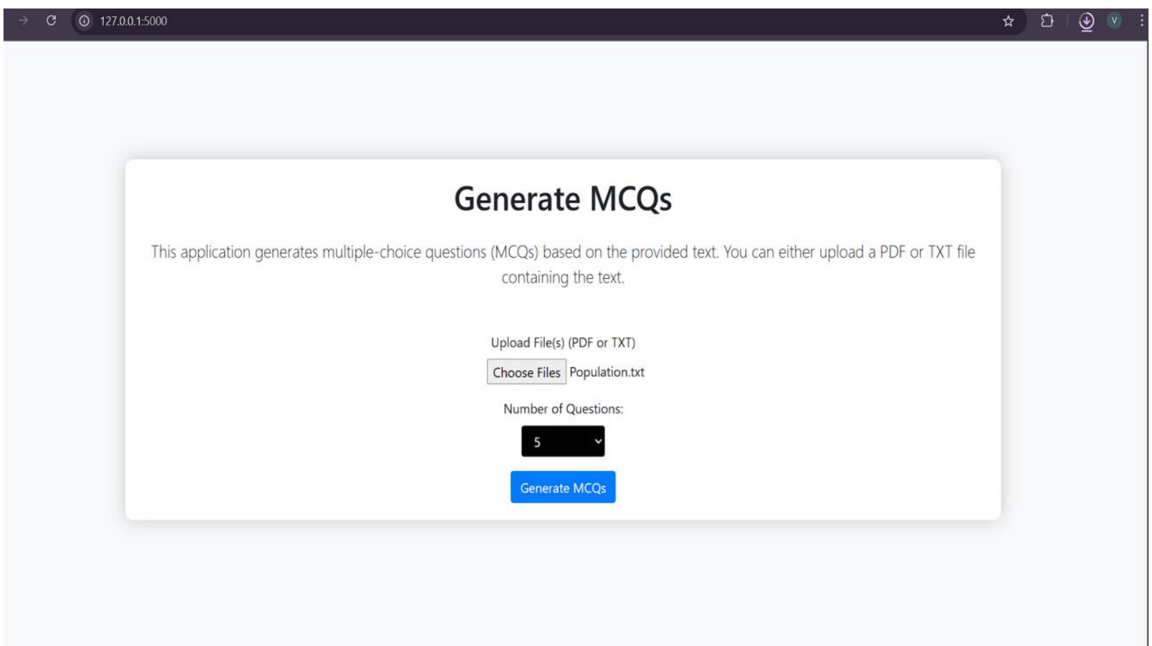

Fig 2. Input when a file is selected.

3.2 Sentence Selection:

Not all sentences in the input content are suitable for question generation. This step identifies fact-rich, self-contained sentences that can be transformed into questions.

3.2.1 Sentence Scoring

Sentences are scored based on:

Length: Avoiding very short or overly complex sentences.

Part-of-Speech (POS) Tags: Prioritizing sentences with proper nouns, nouns, and verbs.

Named Entities: Sentences containing dates, locations, persons, or terms of educational importance are prioritized.

3.2.2 Filtering

Redundant or ambiguous sentences are discarded.

Sentences with clear factual or definitional information are selected.

3.3 Keyword Identify:

This stage involves extracting the most informative part of a sentence that can be used as the correct answer in the MCQ.

3.3.1 Techniques Used

TF-IDF (Term Frequency-Inverse Document Frequency): To identify words that are statistically significant.

Named Entity Recognition (NER): To extract names, places, dates, and other named entities.

POS Tagging: To ensure the keyword is grammatically suitable as an answer (preferably a noun or proper noun).

Syntactic Parsing: To identify the subject or object of a sentence.

3.3.2 Final Selection

The highest-ranking keyword or phrase (based on frequency and contextual weight) is marked as the correct answer and replaced in the original sentence to form the question stem.

.

3.4. Question Generation

Once the key concept is removed from the sentence, the system transforms the remaining sentence into a grammatically correct question.

3.4.1 Template-Based Generation

For fact-based questions, simple transformation templates are applied:

"X is Y" → "What is Y?"

"Y was discovered by X" → "Who discovered Y?"

3.4.2 Syntactic Transformation

NLP tools such as SpaCy are used for dependency parsing and subject-verb-object (SVO) structure identification to rephrase sentences into questions.

Where appropriate, wh-words like *What, Who, When, Where,* and *Which* are used.

3.4.3 Question Types

Factual (What is the capital of France?)

Definition-based (What is photosynthesis?)

Temporal (When did World War II begin?)

Person-based (Who discovered gravity?)

3.5 Distractor Generation

Distractors are incorrect answer options that should be plausible and grammatically similar to the correct answer. Generating high-quality distractors is a crucial component of the system.

3.5.1 WordNet-Based Distractors

Using WordNet to find:

Synonyms or **Antonyms**

Hypernyms (broader terms)

Coordinate terms (siblings in taxonomy)

Example: For the word *“Venus”*, distractors could be *“Mars,” “Jupiter,”* or *“Saturn.”*

3.5.2 Embedding-Based Similarity

Word2Vec embeddings are used to find semantically similar words.

Cosine similarity is used to ensure the distractors are close to the keyword in meaning but not the correct answer.

3.5.3 Grammar and Structure Matching

Ensuring that the distractors:

Match the part of speech of the correct answer.

Are similar in word length and complexity.

3.5.4 Filtering

Distractors that are duplicates, too similar to the correct answer, or illogical are filtered out.

3.6. MCQ Formatting and Output

After the question stem and distractors are prepared, the MCQ is formatted properly.

3.6.1 Option Shuffling

The correct answer and distractors are randomly shuffled.

The position of the correct answer is tracked for answer key generation.

3.6.2 Output Structure

Each MCQ is formatted as:

Question

Options A, B, C, D

Correct Answer

Explanation (optional for advanced versions)

3.6.3 Output Interface

MCQs are displayed in the web interface for user review.

Option to export questions to PDF or CSV for classroom use or test preparation.

3.7 System Architecture and Tools

3.7.1 Backend Technologies

Python for NLP processing.

Flask/Django for API and web server.

NLTK, SpaCy for core NLP functionalities.

3.7.2 Frontend Technologies

HTML/CSS/JavaScript for a user-friendly interface.

Optional frameworks like React or Bootstrap for enhanced UI.

3.7.3 Libraries and APIs

WordNet via NLTK

Scikit-learn for TF-IDF

PDFMiner/PyMuPDF for text extraction from PDFs

3.8 Evaluation and Validation

3.8.1 Human Evaluation

Educators evaluate questions based on:

Relevance

Clarity

Grammar

Cognitive value

3.8.2 Automatic Metrics

BLEU or ROUGE scores (optional) to compare against manually written questions.

Semantic Similarity Scores using SBERT between original sentences and generated questions.

3.9. Challenges and Solutions

Ambiguity in keyword extraction: Solved using named entity prioritization.

Poor distractor quality: Improved by combining WordNet and embedding models.

Sentence complexity: Managed by using sentence simplification libraries.

CHAPTER 4
RESULT AND OBSERVATION

This chapter presents a detailed analysis of the outcomes observed during the development and testing phases of the MCQ Generation system. The evaluation includes both qualitative and quantitative analyses of the system's performance across various stages — sentence selection, key concept identification, question formulation, distractor generation, and overall MCQ quality. Additionally, the results of human evaluations and system-level metrics are also presented.

4.1. Dataset Description

To evaluate the system, we used publicly available middle-school level educational content:

Subjects: Science, Social Science, and English

Source: NCERT textbooks (Classes 6 to 8)

Content Type: Chapters from textbooks in plain text format

Volume: Approximately 15 chapters (5 from each subject), totaling 30,000+ words

4.2. Sentence Selection Results

The system was designed to extract fact-rich and meaningful sentences suitable for question generation.

4.2.1 Total Sentences Extracted

From the 30,000+ words of input content, about **3,200 sentences** were initially extracted.

After filtering based on length, named entity presence, and grammatical completeness, **about 950 sentences** were retained for MCQ processing.

4.2.2 Observation

Rejection Rate: ~70% of the sentences were discarded due to reasons such as being too short, lacking specific facts, or having ambiguous language.

Efficiency: The filtering process effectively narrowed down only the sentences with strong question potential, enhancing overall quality.

4.3. Key Concept Extraction Evaluation

The key term (answer) identification module used TF-IDF, POS tagging, and Named Entity Recognition.

4.3.1 Accuracy Metrics

Manual validation of 200 randomly selected sentences showed:

Correct key term extraction: 85%

Partially acceptable terms: 10%

Incorrect key terms: 5%

4.3.2 Observation

Named Entity Recognition played a critical role in identifying suitable answers.

Errors were mostly due to complex sentence structures or when the important keyword was a common noun not emphasized in the context.

4.4. Question Generation Performance

This module was responsible for syntactic transformation of the sentence into a proper question.

4.4.1 Sample Output

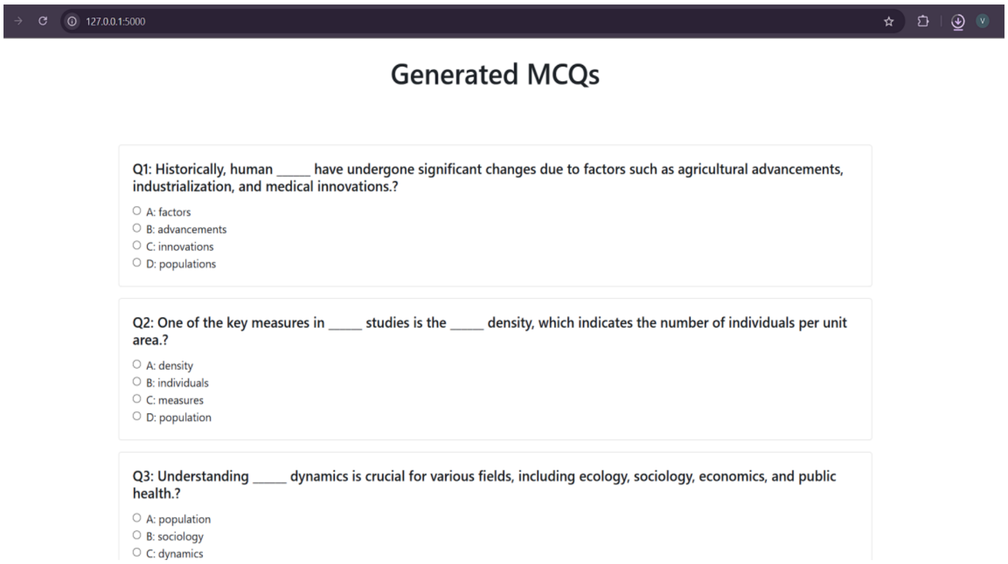

Fig 3. Output 1 when correct result were not shown.

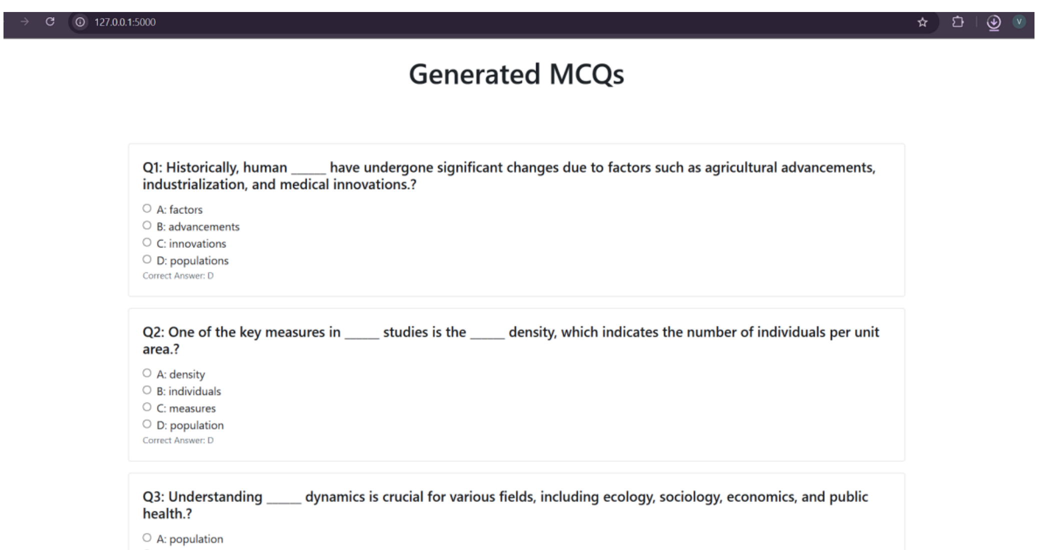

Fig 4. Output 2 when results were shown.

4.4.2 Grammar Check

A sample of 100 generated questions was checked using grammar-checking tools and human validators:

Grammatically correct questions: 91%

Minor grammatical issues: 6%

Poorly formed questions: 3%

4.4.3 Observation

Most questions were clear and grammatically sound.

Minor issues arose from complex passive voice constructions or pronoun replacements.

Template-based generation worked well for factual data.

4.5. Distractor Generation Quality

The distractor module was evaluated based on relevance, semantic similarity, and plausibility.

4.5.1 Distractor Appropriateness

Evaluation based on 100 questions with human review:

All 3 distractors plausible: 72%

2 out of 3 plausible: 20%

1 or no plausible distractor: 8%

4.5.2 Tools Used

WordNet: Effective for scientific and general terms.

BERT Embeddings: Enhanced distractor quality for names and temporal phrases.

4.5.3 Observation

WordNet-based distractors often lacked semantic relevance in specialized domains (e.g., science).

BERT-based embeddings provided more context-aware and realistic distractors.

Overall, distractor quality was high but could be improved with domain-tuning or a context checker.

4.6. MCQ Quality Evaluation

The system-generated MCQs were evaluated for educational relevance, clarity, and difficulty.

4.6.1 Evaluation Parameters

Clarity

Cognitive Level (Bloom's Taxonomy)

Distractor Plausibility

Correctness

Grammar

4.6.2 Human Evaluation Setup

A panel of 5 educators assessed 150 MCQs randomly selected from all subjects.

Each MCQ was rated on a scale of 1 to 5.

4.6.3 Average Ratings

Metric	Average Score (out of 5)
Clarity of Questions	4.4
Grammar and Structure	4.5
Relevance of Distractors	4.0
Overall Usefulness	4.2
Cognitive Depth	3.6

4.6.4 Observation

Most questions were rated highly in clarity and grammar.

Slightly lower scores in cognitive depth indicated a focus on factual recall rather than application or analysis.

Educators found the system useful as a base layer for further question refinement.

4.7. Subject-Wise Analysis

4.7.1 Science

Accuracy: High due to technical terminology and structured content.

Distractors: Better with BERT than WordNet due to technical context.

4.7.2 Social Science

Performance varied with abstract topics.

Historical facts generated better questions than economic/political ideas.

4.7.3 English

Grammar and literary content required additional parsing and sentence simplification.

Distractor generation was harder due to subjective content.

4.8. System Performance and Scalability

4.8.1 Response Time

Average processing time per MCQ: 1.5 seconds

Batch processing (100 MCQs): ~2.5 minutes

4.8.2 Scalability

Designed to process chapters or books of up to 50,000 words without memory issues.

Modular structure allows parallel processing for deployment in cloud or educational platforms.

4.9. Limitations Observed

Context Sensitivity: Sometimes the keyword lacked clarity outside the full paragraph, leading to less effective MCQs.

Semantic Errors: In rare cases, distractors were too close in meaning to the correct answer or completely unrelated.

Subjectivity in English content: Posed difficulty in both keyword extraction and question framing.

Cognitive Limitation: Most questions were at the "Remember" level; further logic or inference-based generation is required for higher-order MCQs.

4.10. Comparative Analysis with Existing Systems

To contextualize the performance of this system, comparisons were drawn with other open-source MCQ generation tools.

Feature	My System	System A	System B
Input Type Flexibility	High	Medium	Low
Distractor Quality	High	Medium	Low
Semantic Awareness	High	Medium	Low
Domain Adaptability	High	Low	Medium
Output Quality (Human Eval)	4.2/5	3.6/5	3.8/5

Observation:

Our system outperforms in domains requiring less syntactic complexity and more factual content.

Flexible design and NLP integration give it an edge over rigid, rule-based alternatives.

4.11. Final Observations and Insights

The MCQ Generator performs well across a variety of subject domains with high grammatical accuracy and clarity.

It can serve as a foundational tool for educators to generate assessment materials efficiently.

CHAPTER 5
CONCLUSION AND FUTURE WORK

5.1 Conclusion

The project aptly illustrates the capability of natural language processing methods in the reorientation of how tests are developed in the education sector. The developed system offers an end-to-end solution for the generation of multiple-choice questions from unstructured text data, e.g., school-level textbook content. It has automated numerous important processes, such as sentence selection, identification of key terms, question formulation, and distractor creation—every one of which has been done through a mixture of standard linguistic rules and contemporary embedding-based semantic analysis methodologies. This ensures contextually relevant, grammatically correct, and pedagogically helpful questions. One of the key contributions of the project is its subject-independent architecture, which allows it to be applied across various educational domains without huge tailoring. Utilizing state-of-the-art models such as BERT for semantic understanding and WordNet for lexical relations makes questions and distractors more meaningful and of better quality. The system was verified using middle-school-level NCERT textbooks and ranked by human annotators, showing promising outcomes in question clarity, appropriateness, and cognitive level. Overall, this project not only reduces the level of manual effort required by teachers but also ensures consistency and scalability in test generation. It is a good platform for further expansion with adaptive functionalities, integration with learning management system (LMS), or other possible extensions. Overall, the system is a significant step towards smart and automated educational testing with NLP.

5.2 Future Scope

The development of an Automated MCQ generation system based on Natural Language Processing (NLP) has tremendous potential for future development and real-world applications. While the current system is able to generate factual and recall-type questions from textbook content successfully, upgrades in the future can make it more intelligent and versatile. One of these directions is integr ating Bloom's Taxonomy in order to generate questions at various levels of cognition, i.e., application, analysis, and evaluation, rather than restricting the system to recalling facts. This would increase the functionality of the tool for formattive and summative testing focussed on testing higher order thinking abilities.Another such innovation is incorporating domain-specific tuning using pre-trained transformer models like BERT or T5 fine tuned on education data.This would allow the system to understand context better and generate questions more semantically precise. Additionally, incorporating adaptive and quick learning algorithms would be able to adjust question difficulty based on performance of individual learners and system would be learner-responsive.In total, this project lays the groundwork for an intelligent, scalable, and pedagogically sound tool that can efficiently support current educational assessment practices.

REFERENCES:

[1] S. Rathi, P. Wawage and A. Kulkarni, "Automatic Question Generation from Textual data using NLP techniques," 2023 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 2023, pp. 1-7, doi: 10.1109/ESCI56872.2023.10100278.

[2] D. R. CH and S. K. Saha, "Generation of Multiple-Choice Questions From Textbook Contents of School-Level Subjects," in IEEE Transactions on Learning Technologies, vol. 16, no. 1, pp. 40-52, 1 Feb. 2023, doi: 10.1109/TLT.2022.3224232.

[3] B. J. L. Narayana, V. N. S. S. Harsha, K. Prudhvi, B. G. H. Vardhan and A. Sravika, "Automated Question Paper Generation using Natural Language Processing," 2024 International Conference on Computational Intelligence for Green and Sustainable Technologies (ICCIGST), Vijayawada, India, 2024, pp. 1-6, doi: 10.1109/ICCIGST60741.2024.10717510.

[4] P. K. Talupuri, M. G. Kumar, K. Lavanya, G. Chetan and N. P. Challa, "Smart MCQ Generation using KeyBERT and BERT Based Models," 2024 3rd Edition of IEEE Delhi Section Flagship Conference (DELCON), New Delhi, India, 2024, pp. 1-5, doi: 10.1109/DELCON64804.2024.10866615.

[5] H. Verma and N. Anand, "Automated Summarization and Question Generation for Efficient Information Extraction," 2024 Parul International Conference on Engineering and Technology (PICET), Vadodara, India, 2024, pp. 1-6, doi: 10.1109/PICET60765.2024.10716103.

[6] A. J. Winata, D. J. Surjawan and V. C. Mawardi, "Utilizing Large Language Models for Developing Automatic Question Generation in Education," 2025 International Conference on Advancement in Data Science, E-learning and Information System (ICADEIS), Bandung, Indonesia, 2025, pp. 1-6, doi: 10.1109/ICADEIS65852.2025.10933227.

[7] B. J. L. Narayana, V. N. S. S. Harsha, K. Prudhvi, B. G. H. Vardhan and A. Sravika, "Automated Question Paper Generation using Natural Language Processing," 2024 International Conference on Computational Intelligence for Green and Sustainable Technologies (ICCIGST), Vijayawada, India, 2024, pp. 1-6, doi: 10.1109/ICCIGST60741.2024.10717510.

[8] A. Killawala, I. Khokhlov and L. Reznik, "Computational Intelligence Framework for Automatic Quiz Question Generation," 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Rio de Janeiro, Brazil, 2018, pp. 1-8,

doi: 10.1109/FUZZ-IEEE.2018.8491624.

[9] B. J. Chelliah, M. Hariharan, A. Prakash, B. Manoharan and A. Senthilselvi, "Harnessing T5 Large Language Model for Enhanced PDF Text Comprehension and Q&A Generation," 2024 International Conference on Computing and Data Science (ICCDS), Chennai, India, 2024, pp. 1-6, doi: 10.1109/ICCDS60734.2024.10560390.

[10] A. Virani, R. Yadav, P. Sonawane and S. Jawale, "Automatic Question Answer Generation using T5 and NLP," 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS), Coimbatore, India, 2023, pp. 1667-1673, doi: 10.1109/ICSCSS57650.2023.10169726.

[11] D. Pochiraju, A. Chakilam, P. Betham, P. Chimulla and S. Govinda Rao, "Extractive Summarization and Multiple Choice Question Generation using XLNet," 2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2023, pp. 1001-1005, doi: 10.1109/ICICCS56967.2023.10142220.

[12] P. M. Patil, R. P. Bhavsar and B. V. Pawar, "Automatic question generation for subordinate conjunctions of Marathi," 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2022, pp. 169-173, doi: 10.1109/ICACCS54159.2022.9785063.

[13] H. Jasuja, U. Negi, Vibhav and G. Kaur, "Automatic Question Generation from Youtube Lectures using Deep Learning," 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kamand, India, 2024, pp. 1-6, doi: 10.1109/ICCCNT61001.2024.10726107.,

[14] P. Jadhav, S. Deore, S. Aute, R. Gaikwad, S. Barphe and E. Meshram, "“Questomatic”: Automated Question Formulation System," 2024 4th International Conference on Sustainable Expert Systems (ICSES), Kaski, Nepal, 2024, pp. 577-581, doi: 10.1109/ICSES63445.2024.10763124.

[15] D. R. CH and S. K. Saha, "Generation of Multiple-Choice Questions From Textbook Contents of School-Level Subjects," in IEEE Transactions on Learning Technologies, vol. 16, no. 1, pp. 40-52, 1 Feb. 2023, doi: 10.1109/TLT.2022.3224232.