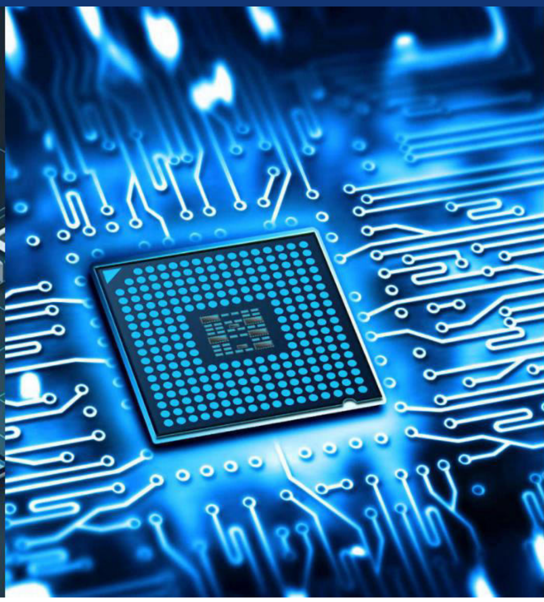
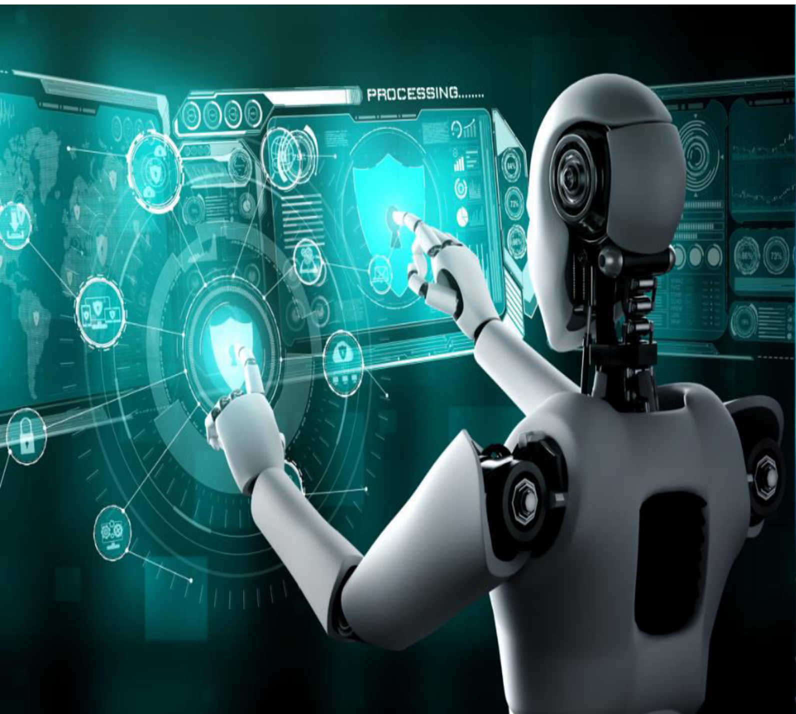
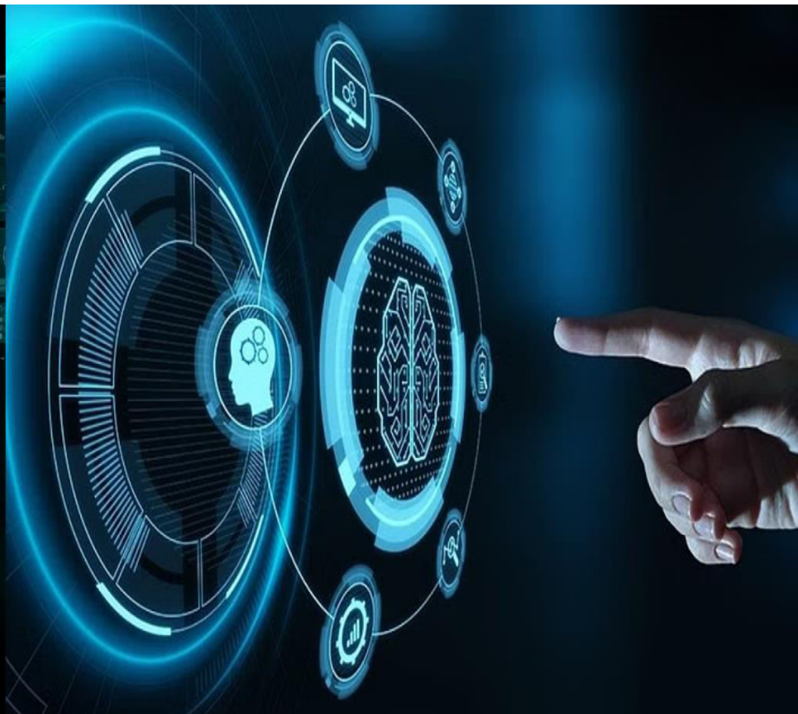
ISSN(O): 2320-9801
ISSN(P): 2320-9798


International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)





Impact Factor: 8.771 **Volume 13, Issue 5, May 2025**

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Design and Implementation of Automated Multiple Choice Questions Generation using cGan's for Computer Science Education

Neelu B[1], Famidha A[2], Pooja V[2], Usma K[2]

Assistant Professor, Department of Computer Science and Engineering, P.S.V College of Engineering and Technology, Krishnagiri Dt., Tamil Nadu, India[1]

UG Scholars, Department of Computer Science and Engineering, P.S.V College of Engineering and Technology, Krishnagiri Dt., Tamil Nadu, India[2]

ABSTRACT This project presents a novel framework for the automated generation of Multiple-Choice Questions (MCQs) tailored for computer science education using Conditional Generative Adversarial Networks (cGANs). Addressing the challenges of traditional MCQ creation—such as time consumption, lack of adaptability, and limited personalization—this system leverages deep learning to generate contextually relevant and difficulty-aligned questions across diverse computer science domains. The model is trained on a curated dataset comprising textbooks, online resources, and instructor-generated content, and is evaluated using metrics including Question Relevance Score (QRS), Diversity Index (DI), Difficulty Alignment Accuracy (DAA), and Semantic Equivalence Score (SES). A Qt-based GUI allows educators to generate and export MCQs with ease.In future enhancements, the system will support real-time question generation from user-provided inputs such as PDFs, websites, and images via OCR and web scraping. It will also refine difficulty classification using Bloom's Taxonomy and introduce feedback-based learning to adapt to student performance. These advancements aim to transform the model into a fully adaptive, multimodal assessment tool that aligns with modern, student-centered educational practices.

KEYWORDS: MCQ generation, cGAN, computer science education, AI in assessment, difficulty classification, OCR, NLP, educational technology

I.INTRODUCTION

In the modern era of digital learning, assessment plays a crucial role in evaluating students' understanding and promoting active learning. Multiple Choice Questions (MCQs) are one of the most widely used assessment formats due to their efficiency, objectivity, and ease of evaluation. However, manually creating a large and diverse pool of quality MCQs—especially in technical fields like computer science—is time-consuming and often prone to human bias or redundancy.To address these challenges, this project explores the **design and implementation of an automated MCQ generation system using Conditional Generative Adversarial Networks (cGANs)**. cGANs, an extension of traditional GANs, are capable of generating context-specific outputs based on given conditions. Leveraging the power of deep learning and natural language processing (NLP), our system is trained to generate semantically meaningful and pedagogically relevant MCQs from input topics or content related to computer science subjects.The proposed solution not only reduces the workload of educators but also ensures the rapid and scalable production of diverse questions suitable for quizzes, practice tests, and e-learning platforms. By integrating artificial intelligence with education technology, this project aims to contribute to the development of intelligent tutoring systems and enhance the overall learning experience in computer science education.

II.LITERATURE SURVEY

Manjula Shenoy K2, Archana Praveen Kumar 1, Ashalatha Nayak 1, Chaitanya 1, and Kaustav Ghosh 1. Gratified on February 27, 2023. Copyright 2023 Author(s).

This section includes a review of previous studies that have been done to generate the MCQs. Based on the requirements of our project. This paper presents a hybrid framework combining semantic analysis with machine learning for MCQ generation. The authors utilize natural language processing to extractkey sentences and generate relevant question stems. The ontology-based representation ensures semantic consistency, while the ML model introduces adaptability in question types. This method supports both factual and conceptual question generation and can be tailored to specific domains like computer science.

Prajakta, Vivek, Pragati, Yogesh, and Geeta Atkar Automatic MCQ Generation Accepted: © 2021 IJCRT | November 20, 2021, Volume 9, Issue 11.
This paper suggests a hybrid method to produce different 7 kinds of Wh-type and Cloze question stems for a technical subject. It combines an Ontology-Based Technique (OBT) with a MachineLearning Based Technique (MBT). Method Based on Ontologies: OBT OBT is a three-step process that creates Wh-type questions: Ontology modeling, creating an instance tree (ITree), and transforming Wh-type questions into variable representation in the final stageThe queries that are generated have a specific or medium scope. Automatic question generation systems that produce questions for users of different kinds and scopes based on input of natural language text. The nicest part about this system is that it makes it simple to process the creation of Objective Type papers and analyzes the data produced by multiple-choice questions (MCQs) to help solve the test's current issues and improve student performance. With the input of a natural language text, Automatic Question Generation Systems produce multiple choice questions (MCQs) with varying scopes and types for the user. Specifically, we address the issue of factual question generation from specific texts automatically. severe administrative issues with paper-based assessment, particularly in courses with large student enrolment.

Yazeed Yasin Ghadi4, Othman Asiry3, Fahad Alturise2, Shahbaz Ahmad1, Muhammad Asif1, Haseeb Ahmad1, and Shahbaz Ahmad
Multiple-choice questions in the computer science sector derived from the following instructive sentences. Nowadays, most tests are multiple-choice. Usually, it is hard to find the distractor, key, and informative sentence for MCQ generation, especially in the computer science field. Therefore, it is necessary to create an intelligent system that can create MCQs from unstructured text. We proposed an innovative method for generating MCQs that combines NLP and ML techniques. Tokenization, lemmatization, POS, and other NLP techniques are used to preprocess the input text corpus. Since not all sentences can generate MCQs, the automatic MCQ generation process consists of three steps: the first is the extraction of informative sentences, the second is the identification of the key, and the third is determining the distractors relevant to the key. The manual MCQ generation process requires a significant amount of work, time, and domain knowledge. Then, utilizing extractive text summarization, the BERT model for text embeddings, and K-means clustering to identify sentences closest to the centroid for summary generation, the suggested method extracts informative sentences

III. METHODOLOGY

The proposed system for **automated multiple choice question (MCQ) generation using Conditional Generative Adversarial Networks (cGANs)** follows a systematic approach involving data collection, preprocessing, model design, training, and evaluation. The methodology is structured into the following phases:

1. Data Collection

To train and evaluate the cGAN model, a well-curated dataset is essential. The dataset is collected from:

I. Online educational resources (e.g., MOOCs, tutorials, textbooks)
II. Open-source question banks
III. Research articles and Wikipedia entries related to computer science topics (e.g., Data Structures, Algorithms, Operating Systems, Networking)
IV. Each data entry typically includes:
V. A concept/topic
VI. A reference passage or explanation
VII. One correct answer and several distractors (incorrect but plausible options)

2. Data Preprocessing

Raw text data is cleaned and structured to make it suitable for training. Key preprocessing steps include:

I. **Tokenization**: Breaking text into words or subword units
II. **Stop-word removal**: Eliminating irrelevant words (e.g., "the", "is", "and")
III. **Lemmatization**: Reducing words to their base form
IV. **POS tagging and entity recognition**: Understanding grammatical structure and key concepts

V. The final dataset format is structured as:
VI. Input: [Topic/Context + Reference Passage]
VII. Output: [Question + Correct Answer + Distractors]

3. Model Architecture: Conditional GAN (cGAN)
The system uses a cGAN framework composed of two main components:
I. **Generator (G)**: Takes the input condition (topic and reference content) and generates a set of MCQ components—primarily a question and multiple answer choices.
II. **Discriminator (D)**: Tries to distinguish between real (human-generated) and fake (machine-generated) MCQs.
III. The generator is trained to fool the discriminator, while the discriminator is trained to correctly identify genuine vs. generated content. The adversarial training continues until an optimal balance is achieved.

4. Feature Extraction and Embedding
Natural Language Processing (NLP) techniques and pre-trained embeddings such as Word2Vec, GloVe, or BERT are used to:
I. Extract semantic and syntactic features from the input
II. Represent words and sentences in a vectorized form for model consumption

5. Model Training
I. The model is trained using supervised and adversarial learning techniques:
II. **Loss functions**: Binary cross-entropy for discriminator; combination of reconstruction and adversarial loss for generator
III. **Optimization**: Adam optimizer for both networks
IV. **Training Strategy**: Alternating updates to generator and discriminator during each epoch
V. Hyperparameters like learning rate, batch size, and dropout rate are tuned for optimal performance.

6. Post-processing and Validation
After generation:
I. The questions are checked for grammatical correctness and logical consistency
II. Distractors are evaluated for plausibility and uniqueness
III. A domain expert may review a subset of generated questions

7. Evaluation Metrics
I. The system is evaluated based on:
II. **BLEU score**: Measures similarity between generated and reference questions
III. **ROUGE score**: Assesses recall-based similarity
IV. **Human Evaluation**: Experts review questions for relevance, clarity, and difficulty
V. **Discriminator Accuracy**: Reflects the realism of generated questions

8. Deployment (Optional Phase)
The model can be integrated into an educational platform or MCQ generator tool with a simple user interface, where users input a topic and receive a set of generated MCQs.

III. DESIGN AND ARCHITECTURE

The architecture of the automated MCQ generation system using **Conditional Generative Adversarial Networks (cGANs)** is designed to ensure seamless data flow, efficient model training, and high-quality question generation. The architecture is modular, consisting of multiple components responsible for data preprocessing, feature extraction, model training (Generator & Discriminator), and question validation.

1. System Architecture Overview
The overall system consists of the following components:
A. Input Layer
Accepts **topic keywords** or **reference text** related to a computer science concept.
May include optional metadata such as difficulty level or context domain (e.g., Data Structures, Networking).

© 2025 IJIRCCE | Volume 13, Issue 5, May 2025| DOI:10.15680/IJIRCCE.2025.1305199
www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|
International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)
(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

B. Data Preprocessing Module

Cleans and structures raw text using NLP techniques:

1. Tokenization
2. Lemmatization
3. Stop-word removal
4. POS tagging & Named Entity Recognition

Converts text into a machine-understandable format using word embeddings (e.g., BERT, GloVe).

C. Feature Embedding Layer

Generates dense vector representations for:

1. Input topics
2. Sentences or paragraphs (reference context)

Embeddings preserve semantic meaning and help the model understand relationships between words.

D. Conditional GAN Module

i. Generator (G):

Takes embedded input condition (topic + reference text).

Outputs:

1. A syntactically correct **question**
2. One **correct answer**
3. 2–3 **distractor options** (plausible but incorrect answers)

ii. Discriminator (D):

Takes real/generated MCQ sets as input.

Classifies them as **real (human-generated)** or **fake (machine-generated)**.

Provides feedback to the generator to improve quality over time.

E. Training and Optimization Layer

Trains G and D in an adversarial loop:

1. Generator learns to fool the discriminator
2. Discriminator learns to detect fake questions

Uses loss functions:

1. Binary Cross-Entropy
2. Reconstruction Loss

Optimized using Adam or RMSprop optimizer.

F. Post-processing Module

Checks:

1. Grammar and coherence
2. Uniqueness of distractors
3. Relevance of answers

Applies filtering or ranking if multiple questions are generated.

G. Output Layer

Displays final MCQs in a structured format:

1. Question
2. 1 Correct Answer
3. Multiple Distractors

H. (Optional) User Interface

Simple web-based or desktop interface.

Allows users (teachers, students) to input a topic and receive generated MCQs instantly.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

IV. ARCHITECTURAL DIAGRAM

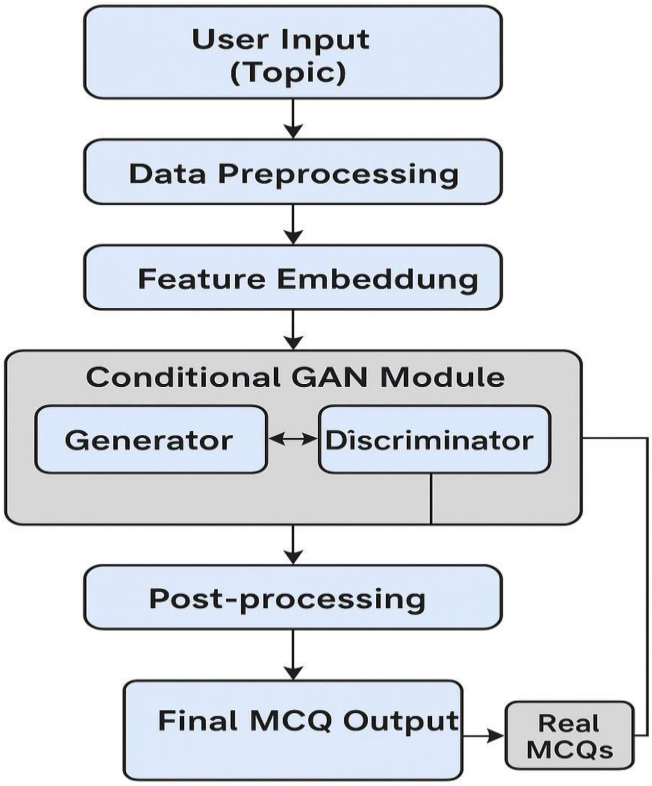


FIG:1

3. Technologies & Tools

Programming Language: Python
Deep Learning Framework: TensorFlow / PyTorch
NLP Libraries: NLTK, SpaCy, Transformers (HuggingFace)
Embeddings: BERT, GloVe
Interface (Optional): React.js / Flask

VI.EXPERIMENTAL RESULTS

1. Environment Setup

Language: Python 3.x
Libraries & Frameworks:

1. PyTorch or TensorFlow (for cGAN model implementation)
2. HuggingFace Transformers (for BERT embeddings)
3. NLTK / SpaCy (for NLP preprocessing)
4. Scikit-learn (for evaluation and additional ML tasks)

2. Dataset Preparation

Source: Open-source question banks, Wikipedia articles, and CS textbooks.
Structure:

1. **Input**: Topic + Reference Paragraph
2. **Output**: Question, Correct Answer, Distractors

© 2025 IJIRCCE | Volume 13, Issue 5, May 2025| DOI:10.15680/IJIRCCE.2025.1305199
www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|
International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)
(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Example:

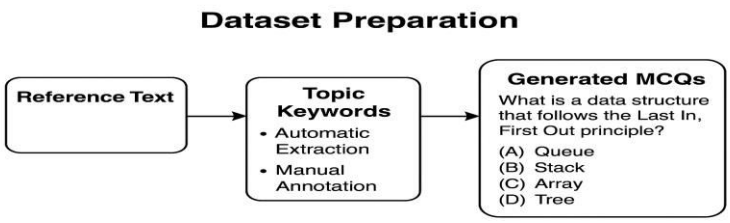


FIG:2

3. Preprocessing

Clean raw text (lowercase, punctuation removal)
Tokenization and lemmatization
Named Entity Recognition for keyword spotting
Sentence segmentation
BERT-based embedding for contextual input

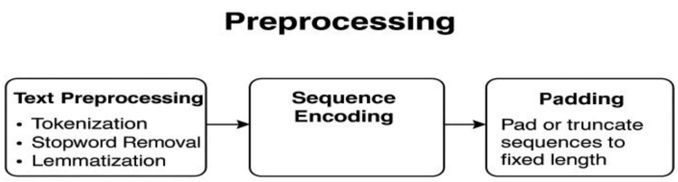


FIG:3

4. Conditional GAN Implementation

Generator Network

Input: Embedded topic + context
Output: Structured MCQ (Question, Answer, Distractors)
Layers:

1. Dense layers with ReLU
2. Dropout for regularization
3. Softmax for distractor options

Discriminator Network

Input: MCQ set (real or generated)
Output: Probability (Real or Fake)
Layers:

1. Convolutional or LSTM for sequence understanding
2. Dense + Sigmoid

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

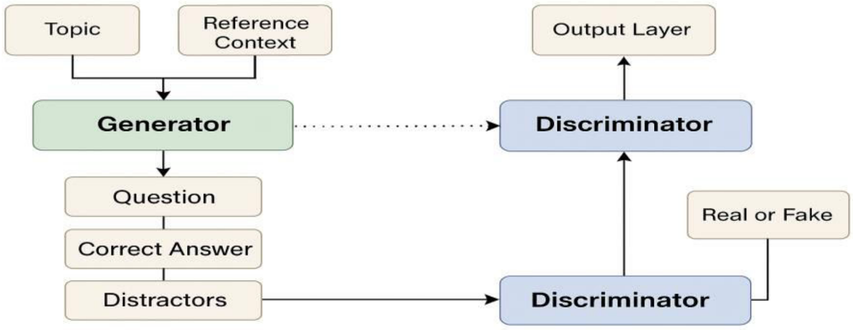


FIG:4

5. Training Process

Adversarial training (Generator tries to fool Discriminator)

Loss functions:

1. **Generator**: Binary Cross Entropy + Semantic Loss
2. **Discriminator**: Binary Cross Entropy

Optimizer: Adam (lr=0.0001)

Epochs: 100–200

Experimental Setup

1. Hardware Setup

CPU: Intel i7 / Ryzen 7 or higher

GPU: NVIDIA GTX 1650 / RTX 3060 or higher (for deep learning)

RAM: 16 GB+

Environment: Google Colab / Jupyter Notebook / VS Code

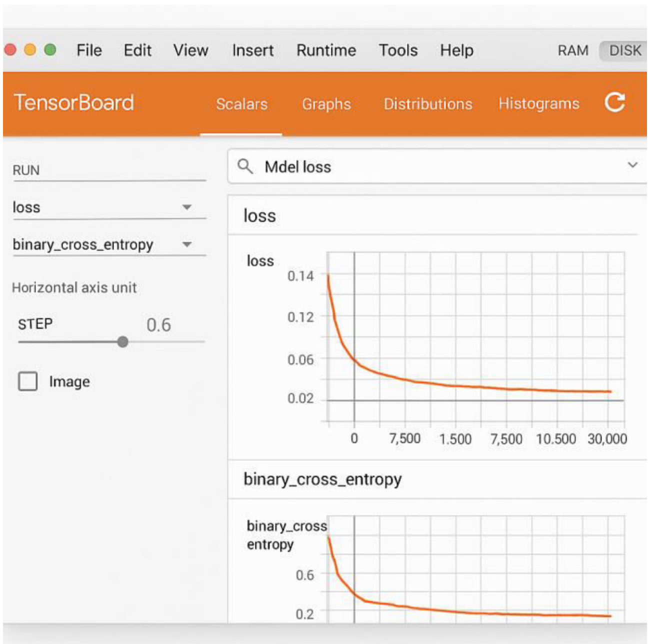


FIG:5

© 2025 IJIRCCE | Volume 13, Issue 5, May 2025| DOI:10.15680/IJIRCCE.2025.1305199
www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|
International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)
(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

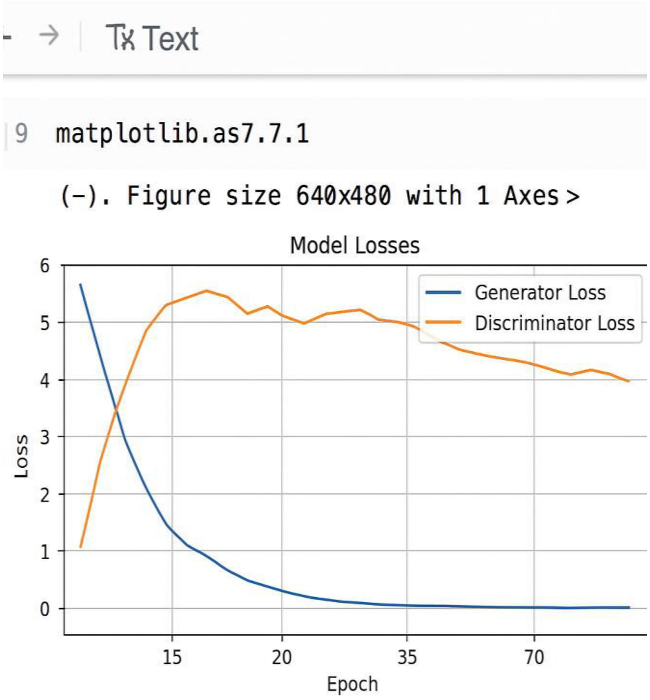


FIG:6

2. Evaluation Metrics

BLEU Score: Measures similarity with ground-truth questions
ROUGE Score: Recall-based overlap
Human Evaluation: Relevance, Grammar, Difficulty
Discriminator Accuracy: Real vs Generated classification0

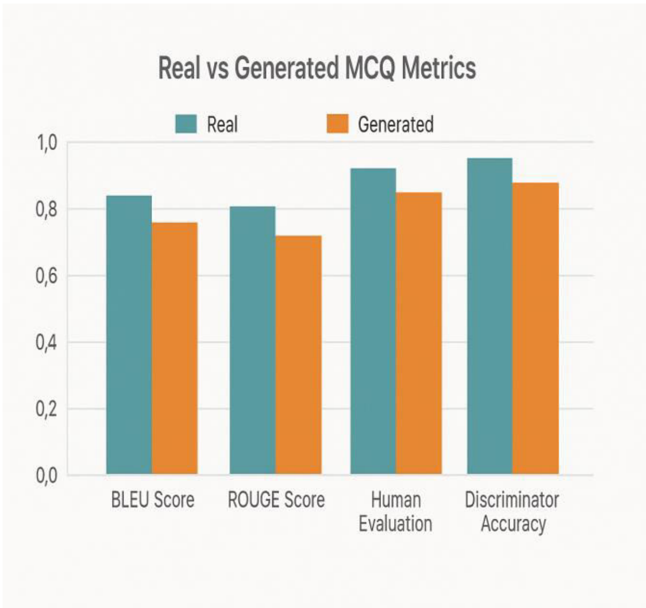


FIG:7

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

3.Sample Output:

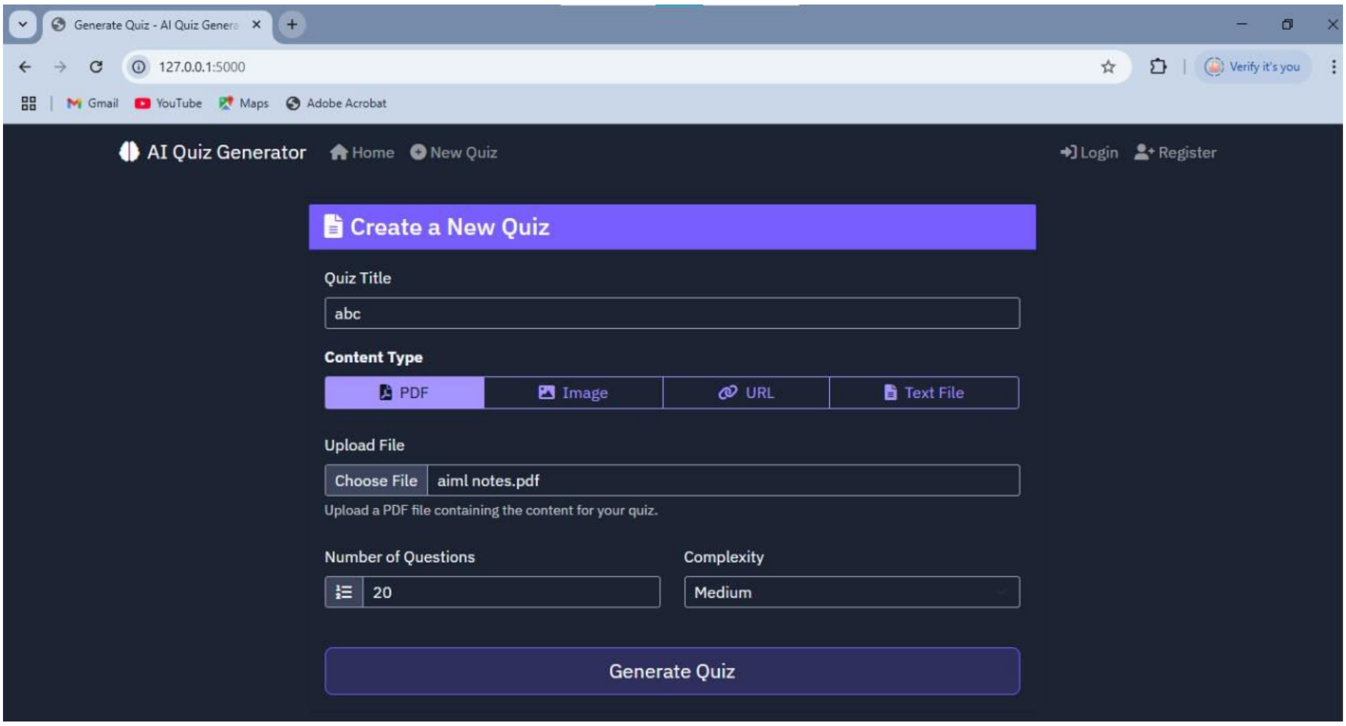


FIG:8

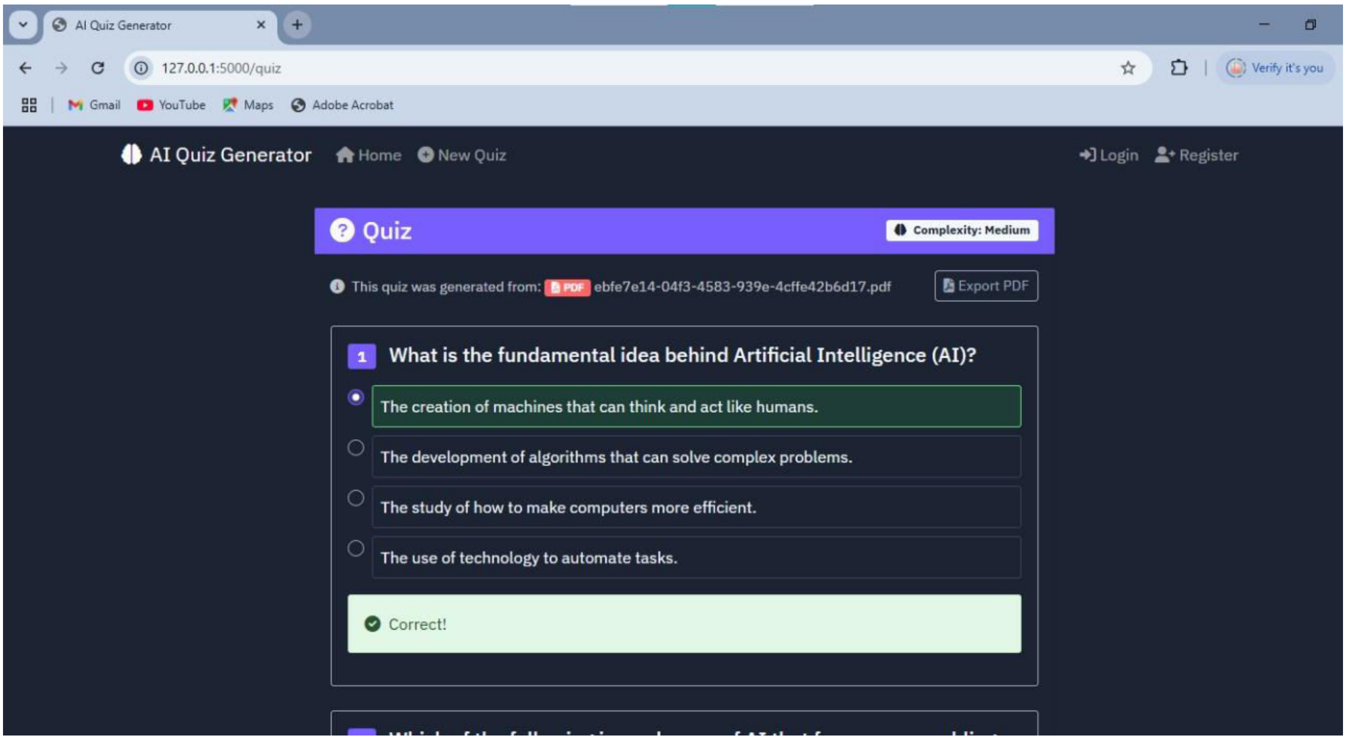


FIG:9

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

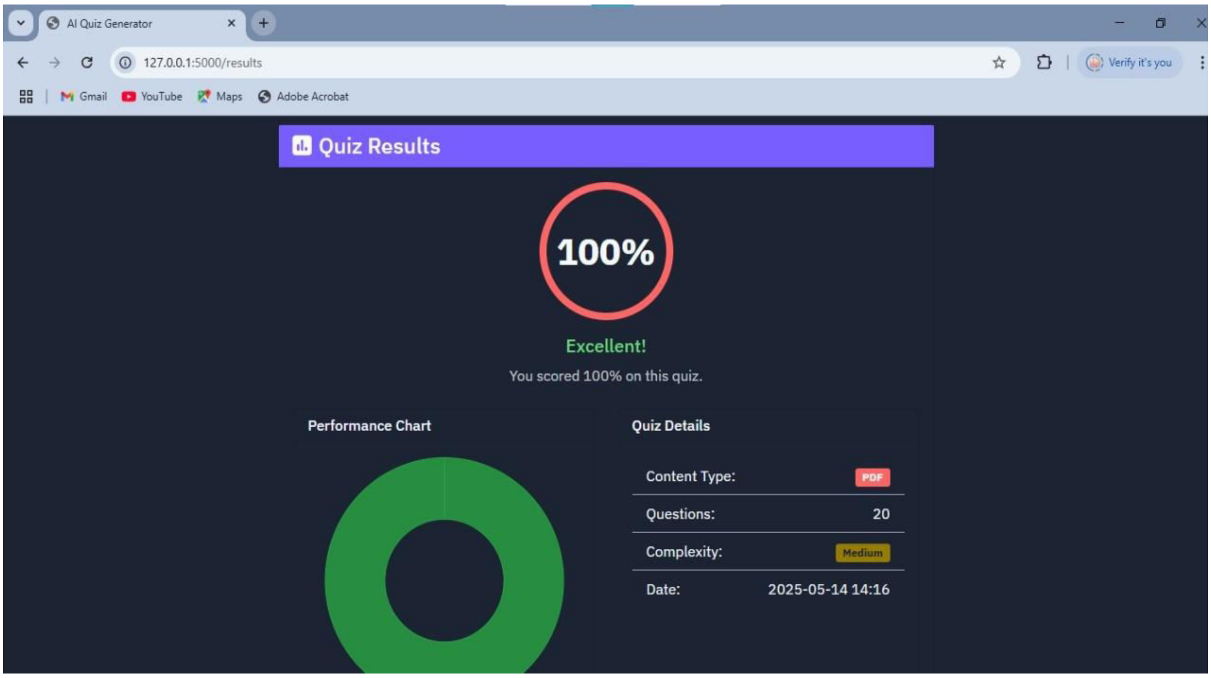


FIG:10

REFERENCES

[1] H. Yu, "The application and challenges of ChatGPT in educational transformation: New demands for teachers' roles," *Heliyon*, vol. 10, no. 2, Jan. 2024, Art. no. e24289.

[2] A. Dash, J. Ye, and G. Wang, "A review of generative adversarial networks (GANs) and its applications in a wide variety of disciplines: From medical to remote sensing," *IEEE Access*, vol. 12, pp. 18330–18357, 2024.

[3] A. Haleem, M. Javaid, M. A. Qadri, and R. Suman, "Understanding the role of digital technologies in education: A review," *Sustain. Oper. Comput.*, vol. 3, pp. 275–285, Jan. 2022.

[4] S. Al Shuriaqi, A. A. Abdulsalam, and K. Masters, "Generation of medical case-based multiple-choice questions," *Int. Med. Educ.*, vol. 3, no. 1, pp. 12–22, Dec. 2023.

[5] S. Meadows and P. J. Karr-Kidwell, "The role of standardized tests as a means of assessment of young children: A review of related literature and recommendations of alternative assessments for administrators and teachers," U.S. Dept. Educ., USA, Tech. Rep. ED 456 134, 2001, p. 101.

[6] S. Dikli, "Assessment at a distance: Traditional vs. alternative assessments," *Turkish Online J. Educ. Technol.*, vol. 2, no. 3, pp. 13–19, Jul. 2003.

[7] L. Ma and S. Qu, "Application of conditional generative adversarial network to multi-step car-following modeling," *Frontiers Neurorobotics*, vol. 17, pp. 1–14, Mar. 2023.

[8] F. Maheen, M. Asif, H. Ahmad, S. Ahmad, F. Alturise, O. Asiry, and Y. Y. Ghadi, "Automatic computer science domain multiple-choice questions

[9] generation based on informative sentences," *PeerJ Comput. Sci.*, vol. 8, p. e1010, Aug. 2022.

[10] R. Westacott, K. Badger, D. Kluth, M. Gurnell, M. W. R. Reed, and A. H. Sam, "Automated item generation: Impact of item variants on performance and standard setting," *BMC Med. Educ.*, vol. 23, no. 1, pp. 1–13, Sep. 2023.

[11] P. McKenna, "Multiple choice questions: Answering correctly and knowing the answer," in *Proc. Multi Conf. Comput. Sci. Inf. Syst. Int. Conf. e-Learn. (MCCSIS)*, vol. 16, Feb. 2019, pp. 59–73.

[12] A. Y. Al Ameer, "Assessment of the quality of multiple-choice questions in the surgery course for an integrated curriculum, University of Bisha college of medicine, Saudi Arabia," *Cureus*, vol. 15, no. 12, pp. 6–13, Dec. 2023.

[13] Y. Mor, H. Mellar, S. Warburton, and N. Winters, *Practical Design Patterns for Teaching and Learning With Technology*. Leiden, The Netherlands: Brill, 2014.

[14] J. Shin, Q. Guo, and M. J. Gierl, "Multiple-choice item distractor development using topic modeling approaches," *Frontiers Psychol.*, vol. 10, pp. 1–14, Apr. 2019.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

[15] J. R. Marchant, "Jacob Rhys Marchant school of physical sciences," Dept. Chem., Univ. Adelaide, Australia, Tech. Rep. 132705, Dec. 2020.

[16] S. Grassini, "Shaping the future of education: Exploring the potential and consequences of AI and ChatGPT in educational settings," *Educ. Sci.*, vol. 13, no. 7, p. 692, Jul. 2023.

[17] F.Kamalov,D.S.Calonge,andI.Gurrib,"Neweraofartificialintelligence in education: Towards a sustainable multifaceted revolution," *Sustainability*, vol. 15, no. 16, p. 12451, Aug. 2023.

[18] O. Solymos, P. O'Kelly, and C. M. Walshe, "Pilot study comparing simulation-based and didactic lecture-based critical care teaching for final-year medical students," *BMC Anesthesiol.*, vol. 15, no. 1, pp. 1–5, Dec. 2015.

[19] I. H. Sarker, "Machine learning: Algorithms, real-world applications and research directions," *Social Netw. Comput. Sci.*, vol. 2, no. 3, pp. 1–21, May 2021.

[20] D. J. K. Tetteh, B. P. K. Okai, and A. N. Beatrice, "VisioMark: An AI-powered multiple-choice sheet grading system," Dept. Comput. Eng., Kwame Univ. Sci. Technol., Ghana, Tech. Rep. 456, 2023.

[21] A. Abulibdeh, E. Zaidan, and R. Abulibdeh, "Navigating the confluence of artificial intelligence and education for sustainable development in the era of Industry 4.0: Challenges, opportunities, and ethical dimensions," *J. Cleaner Prod.*, vol. 437, Jan. 2024, Art. no. 140527.

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details