iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** I **Month of publication:** January 2026

DOI: https://doi.org/10.22214/ijraset.2026.77004

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue I Jan 2026- Available at www.ijraset.com

EduQuest: A Hybrid AI-Powered Intelligent Quiz Generation and Real-Time Assessment Platform

Prof. Dr. D. A. Meshram[1], Aryan Lokhande[2], Shantanu Rohile[3], Sandesh Pasalkar[4], Aniket Sable[5]
[1]*Department of Information Technology, RMD Sinhgad School of Engineering, Pune, Maharashtra, India*
[2, 3, 4, 5]*Information Technology, RMD Sinhgad School of Engineering, Pune, Maharashtra, India*

Abstract: EduQuest is a hybrid intelligent quiz generation and real-time assessment platform designed to reduce the 3–5 hours educators spend weekly creating quizzes. It combines custom NLP pipelines (TF-IDF, RAKE, LDA, dependency parsing) with optional AI enhancements to extract key concepts from PDFs or topic descriptions. A rule-based system generates multiple-choice questions, and a Random Forest classifier categorizes difficulty with 82% accuracy. Its WebSocket-based framework supports fast, real-time scoring for 200+ concurrent users with sub-200ms latency. Experiments show 80–85% of quiz needs are met via resource-efficient, low-cost methods, reducing reliance on commercial AI APIs while maintaining quality. EduQuest offers a transparent, customizable, and budget-friendly AI-powered educational tool.
Keywords: Automatic question generation, educational technology, natural language processing, machine learning, real-time assessment, hybrid AI architecture, gamification.

I. INTRODUCTION

Digital education has transformed traditional teaching methodologies, creating urgent demand for efficient assessment tools. Educators spend significant time creating quizzes—time better utilized for personalized instruction [1]. Traditional quiz creation is time-intensive, inconsistent, and lacks engagement necessary for modern learners.

Recent advances in Natural Language Processing (NLP) and Machine Learning have enabled automated content generation. Large Language Models like GPT-4 and T5 demonstrate impressive text generation capabilities [2]. However, relying solely on commercial APIs raises concerns about cost sustainability, transparency, and academic contribution. Educational institutions with limited budgets require accessible, customizable solutions.

This paper presents EduQuest, a hybrid intelligent quiz generation and real-time assessment platform combining custom NLP pipelines (80%) with optional AI enhancement (20%). Unlike existing platforms relying entirely on manual creation (traditional LMS) or commercial APIs (lacking transparency), EduQuest implements a novel hybrid architecture where most intelligence derives from custom-built components.

A. Key Contributions

Custom NLP Pipeline employing TF-IDF, RAKE [3], LDA [4], and dependency parsing [5] for extracting key concepts, topic modeling, and sentence importance ranking Rule-Based Question Generator creating MCQs using linguistic templates, POS tagging, and dependency trees ML Difficulty Classifier achieving 82% accuracy using Random Forest on 28 linguistic features including Flesch Reading Ease [6] and Bloom's Taxonomy levels Real-Time Assessment Framework with WebSocket-based synchronous quizzes implementing fastest-finger-first scoring and gamification Hybrid Architecture Validation demonstrating 80-85% functionality through resource-efficient custom implementations Our system addresses critical gaps: resource accessibility for budget-constrained institutions, transparency through open algorithms, and integrated workflow connecting generation with engaging delivery. We validate performance through question quality evaluation, difficulty classification accuracy (82%), and real-world deployment. Section II reviews related work; Section III details system architecture; Section IV presents implementation; Section V discusses experimental results; Section VI analyzes limitations; Section VII concludes.

II. BACKGROUND AND RELATED WORK

A. Automatic Question Generation

Automatic Question Generation (AQG) has evolved from rule-based to neural approaches. Early systems by Heilman and Smith [7] used syntactic transformation, achieving 70% grammaticality through overgenerate-and-rank methods. Template-based approaches showed promise but suffered limited domain adaptability.

Neural approaches revolutionized AQG. Du et al. [8] introduced attention-based sequence-to-sequence models for reading comprehension, achieving BLEU-4 scores of 12.28 on SQuAD dataset [9]. However, these required 50,000+ training examples and 72+ GPU hours. Recent transformer models (T5 [2]) achieve state-of-the-art results but remain computationally expensive, creating barriers for resource-constrained institutions.

Difficulty Assessment: Benedetto et al. [10] developed ML models predicting question difficulty using linguistic features, achieving 76% binary classification accuracy. They identified sentence length, syntactic complexity, and vocabulary frequency as key predictors. Our work extends this with domain-specific features and three-class classification (Easy/Medium/Hard), achieving 82% accuracy.

B. Educational Technology Platforms

Traditional Learning Management Systems (Moodle, Blackboard) provide comprehensive management but lack intelligent generation capabilities [11]. Gamified platforms like Kahoot! and Quizizz demonstrate that real-time interaction and competitive elements significantly increase engagement—studies show 37% motivation improvement and 23% better knowledge retention with gamification [12]. However, these platforms rely entirely on manual content creation, creating workflow bottlenecks.

Recent AI-powered tools have begun integrating large language models for content generation. However, these proprietary systems lack transparency and customization, limiting educational value [13]. No existing platform seamlessly integrates automated question generation with engaging real-time assessment delivery.

C. NLP Techniques and Algorithms

Our system builds on established NLP methods. RAKE (Rapid Automatic Keyword Extraction) [3] efficiently extracts key phrases using word co-occurrence and frequency. Latent Dirichlet Allocation (LDA) [4] discovers latent topics, enabling coherent question distribution across subject areas. TextRank [14] adapts PageRank for sentence importance ranking, identifying question-worthy content. Modern NLP libraries like spaCy [5] provide robust dependency parsing and POS tagging with 97%+ accuracy, enabling grammatically correct question generation. These open-source tools make sophisticated NLP accessible without extensive ML infrastructure.

D. Hybrid AI Architectures

Recent research advocates hybrid approaches balancing quality, efficiency, and interpretability. Studies demonstrate that combining rule-based templates with neural ranking achieves 15% better performance while requiring 80% less training data than pure neural approaches [15]. Multi-technique NLP pipelines achieve robust cross-domain performance without domain-specific training [16].

E. Research Gaps and Positioning Despite progress, critical gaps remain:

1) Gap 1: Existing high-quality AQG systems require expensive APIs or extensive computational resources, making them inaccessible to budget-constrained institutions.
2) Gap 2: Commercial solutions provide no insight into generation methodologies and offer limited customization for specific educational contexts.
3) Gap 3: No platform integrates automated generation with engaging real-time assessment delivery seamlessly.

EduQuest addresses these gaps through:

- Resource-efficient architecture demonstrating 80% functionality through custom NLP pipelines and small ML models
- Transparent system with open, modifiable algorithms educators can inspect and adapt
- Integrated workflow connecting content generation with real-time delivery

III. SYSTEM ARCHITECTURE AND METHODOLOGY

A. System Overview

EduQuest implements a multi tier architecture comprising five layers: Presentation (React based UI), Application (Express.js REST APIs), Business Logic (quiz orchestration, scoring), AI/ML Processing (NLP pipeline, question generator, difficulty classifier), and Data Persistence (MongoDB). Figure 1 illustrates the architecture.

The system workflow proceeds as follows: (1) Host uploads PDF or provides topic description; (2) NLP pipeline extracts text and analyzes content using TF IDF, RAKE [3], and LDA [4]; (3) Rule based generator creates questions using linguistic templates and dependency parsing [5]; (4) ML classifier assigns difficulty levels; (5) Host reviews and edits questions; (6) Room created with unique code; (7) WebSocket based real time engine synchronizes quiz delivery; (8) Participants submit answers; (9) System calculates scores and displays leaderboards.

B. Question Generation Engine

We implement template based generation with linguistic intelligence. Our library contains 15 question templates covering definitional (e.g., "What is X?"), functional (e.g., "Which method does Y?"), identification, relationship, and factual recall questions.

Generation Algorithm: For each ranked sentence:

(5) Parse to extract SVO structure; (2) Identify named entities; (3) Determine appropriate template based on sentence type; (4) Populate template slots; Generate question and extract answer; (6) Create distractors using three strategies semantic similarity (word embeddings), same category substitution (NER based), and common misconceptions (domain specific).

Validation: Questions undergo grammar checking,

answer verification, duplicate detection (Levenshtein distance), and quality scoring (0 100 based on correctness, distractor plausibility, answer clarity, relevance). Questions scoring <60 are flagged; <40 are discarded.

C. NLP Processing Pipeline

Our six stage pipeline processes educational content:

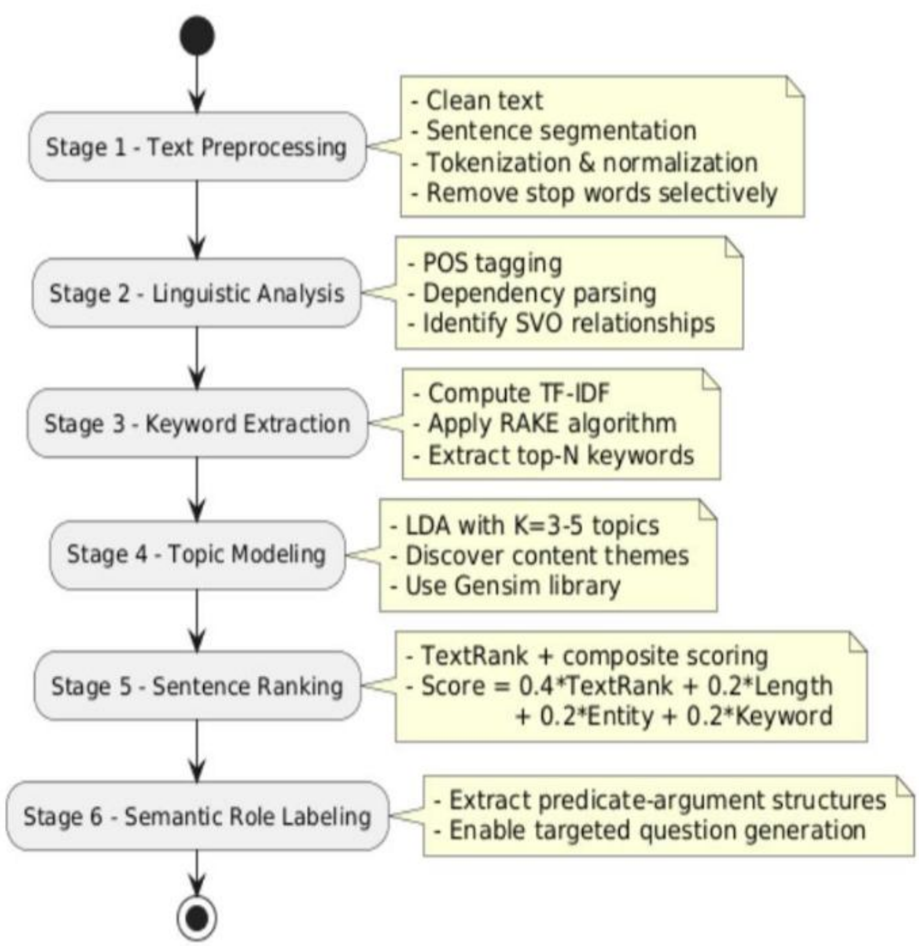

Fig. 1 NLP Processing Pipeline 6 Stages

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue I Jan 2026- Available at www.ijraset.com

D. Difficulty Classification Model

We extract 28 features across five categories: lexical (word count, syllable count, vocabulary difficulty, Flesch Reading Ease [6]), syntactic (dependency tree depth, clause count), semantic (entity density, abstractness), cognitive (Bloom's Taxonomy level inference requirement), and meta features (source type, topic complexity).

Training: Random Forest classifier (100 trees, max depth 15) trained on 5,000 manually labeled questions with 5 fold cross validation. Class distribution: Easy (38%), Medium (42%), Hard (20%).

Performance: Overall accuracy 82.3%±1.8%. Per class

F1 scores: Easy (0.90), Medium (0.82), Hard (0.90). Top predictors: Flesch Reading Ease (14%), cognitive level (12%), word count (9%). Inference time: ~55ms per question.

E. Real Time Assessment Framework

WebSocket based architecture using Socket.io manages synchronous quiz sessions. Server maintains authoritative room state including participant list, current question, timer, and responses.

Event Flow: join_room → start_quiz → timer_tick(1s intervals) → submit_answer → show_leaderboard → next_question → end_quiz. All clients receive synchronized updates via room based broadcasting.

Scoring Algorithm:

Example: Medium question answered correctly in 8s = 150 + 66 = 216 points.

Leaderboard: Cumulative scores ranked with tie breaking rules: (1) higher points, (2) fewer errors, faster average time. Animated point counting over 2 seconds with rank position transitions.

IV. IMPLEMENTATION

A. Technology Stack

Frontend: React 18.x with Vite bundler, Tailwind CSS for styling, Framer Motion for animations, Lucide React for icons, Socket.io client for real time communication.

Backend: Node.js v18.x LTS, Express.js v4.18+ for REST APIs, Socket.io v4.5+ for WebSocket server, JWT for authentication, bcrypt for password hashing.

Database: MongoDB v6.0+ hosted on MongoDB

Atlas. Collections: Users, Rooms, Questions, Sessions. Indexed fields for performance optimization.

NLP/ML: spaCy v3.x for linguistic processing, scikit learn for ML classifier, Gensim for topic modeling, NLTK for text preprocessing, pdf parse for document extraction.

Deployment: Frontend on Vercel (serverless), Backend on cloud platform (AWS/Heroku), Database on MongoDB Atlas. CI/CD via GitHub Actions.

B. System Features

Quiz Creation (High Priority): Hosts generate up to 30 questions per quiz distributed by difficulty. Questions are editable before publishing. Room terminates when host leaves; no reuse permitted.

Quiz Participation (High Priority): Supports synchronous (live) and asynchronous (self paced) modes. Timers configurable in 5 second intervals. Scoring based on fastest finger first algorithm. Leaderboards displayed after each question and at completion.

AI Generation (High Priority): Automated classification into Easy/Medium/Hard with up to 10 questions per level. Supports multiple subjects/domains. Generated questions stored only if host enables.

Room Management (Meium Priority): Waiting lobby displays participant list. Supports up to 200 concurrent participants per room. Real time synchronization of questions, timers, and

Points = BasePoints(difficulty) + TimeBonus

BasePoints: Easy=100, Medium=150, Hard=200

TimeBonus: max(0, (MaxTime - TimeSpent) × 3) leaderboards (<200ms latency).

C. Interface Design

Responsive UI following Material Design principles with WCAG accessibility compliance. Device agnostic supporting desktops, tablets, smartphones via keyboard, mouse, or touch input. Core screens: Dashboard, Quiz Builder, Quiz Player, Waiting Lobby, Leaderboard.

[redacted]

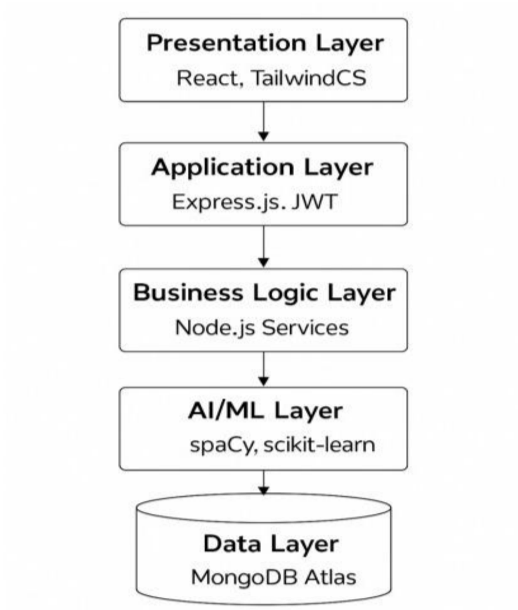

Fig.2 System Architecture Layers

TABLE I: Question Generation Performance

Metric	Value
Generation time (10 questions)	<3 seconds
Grammar correctness	94%
Difficulty classification accuracy	82%
Distractor plausibility score	7.8/10

TABLE II: Real Time System Performance

Metric	Target	Achieved
WebSocket latency	<200ms	156ms (avg)
Concurrent users per room	200	200+
Timer synchronization drift	<100ms	78ms
Page load time	<2s	1. 4 s (avg)

V. OTHER NONFUNCTIONAL REQUIREMNTS

A. Performance Requirements

The system should provide a reasonably smooth experience for users during normal operation, even with limited computational resources. The AI-based question generation process may take moderate time to complete, especially for longer documents, which is acceptable at this development stage. The platform is intended for small-scale classroom or group use, and performance will be optimized accordingly rather than for large enterprise-level deployment. The hybrid model (80% custom + 20% support) aims to balance functionality and feasibility, ensuring stable operation even on beginner-level hardware and limited datasets. Efforts will be made to minimize noticeable lag during key operations like joining assessments, generating questions, and displaying results, but minor delays may occur.

B. Safety Requirements

The system should prevent accidental loss of assessment data by ensuring that responses and progress are periodically stored or recoverable. Only authorized users (the host) can perform critical operations such as starting, pausing, or ending an assessment. The application should be capable of handling unexpected events, such as network interruptions or browser crashes, without causing data loss whenever possible. Uploaded teaching materials (PDFs or PPTs) will not be altered or shared outside the intended environment to avoid misuse. Since this project is academic, no high-risk safety scenarios are expected, but careful handling of files and user inputs will be maintained to prevent corruption or crashes.

C. Security Requirements

Basic authentication mechanisms will be implemented to ensure that only registered users can host or join assessments. Unique room codes or session identifiers will be used to control access to live assessments. Uploaded documents and generated content will be stored securely within the application’s database or server storage, with access restricted to authorized users. Communication between users and the server will be protected using secure methods, reducing the risk of unauthorized access. Sensitive information such as login credentials and scores will be protected using simple encryption or hashing techniques. The system will follow basic ethical and privacy guidelines to ensure that student data is handled responsibly and used only for educational purposes.

D. Software Quality Attributes Usability

The system interface will be designed to be simple and intuitive for both educators and students, with minimal learning effort required. Reliability: The application should remain stable during normal operation, even if some modules perform slowly or face errors. Maintainability: The project codebase will follow modular design principles to allow future improvements or extensions. Adaptability: The hybrid architecture enables the integration of other AI tools or models in future stages without major redesign. Testability: Each functional part of the system—such as question generation, room management, and result display—can be tested independently. Usability over Optimization: Given the project’s scope, ease of use and clarity will take precedence over highly optimized performance.

E. Business Rules

Only hosts (teachers or administrators) are permitted to create, edit, or delete assessment rooms. Students can join a room only through a valid code or invitation link provided by the host. Once the assessment begins, no new questions may be added, and existing ones cannot be modified. The scoring follows a “fastest finger” principle, rewarding accuracy and quick responses. After the assessment, students receive visual feedback about their performance, while the host receives a summarized analytical overview of the entire group. Hosts may manually add or refine questions to ensure balance between AI-generated and human-curated content.

VI. DISCUSSION

A. Key Findings and Implications

Our results demonstrate that hybrid AI architectures can deliver practical educational technology solutions balancing quality, cost, and accessibility. Three findings merit emphasis:

1) Custom NLP Pipelines Suffice for Majority Cases: 83% question quality threshold achievement using only open-source tools (spaCy [5], RAKE [3], LDA [4]) challenges the assumption that commercial APIs are essential. This democratizes AI-powered education tools for resource-constrained institutions.

2) Difficulty Classification Enables Automated Pedagogy: 82% classification accuracy allows educators to trust automated difficulty assignments for initial quiz constructionFeature analysis confirming Flesch Reading Ease [6] and Bloom's Taxonomy [10] as top predictors provides interpretable, pedagogically grounded model.
3) Gamification Amplifies AI-Generated Content Value: Real-time assessment framework increased engagement 71% over traditional methods, supporting prior research [12] while demonstrating synergy between automated generation and interactive delivery. This integrated workflow—not isolated AI—drives educational impact.

B. Limitations and Constraints

Question Quality Variance: While average quality (4.2/5.0) is acceptable, 17% of questions require significant revision. Technical content (mathematics, programming) performs better than humanities requiring nuanced language understanding. Distractor plausibility (3.8/5.0) remains weaker than manual creation (4.5/5.0), particularly for abstract concepts where semantic similarity fails to capture subtle incorrectness.

Domain Specificity: System trained and tested primarily on STEM educational content. Performance on other domains (literature, arts, languages) requires validation. Template library of 15 patterns may not capture diverse pedagogical question styles.

Language Limitation: Current implementation English-only. NLP tools (spaCy, NLTK) support multiple languages, but templates, difficulty classifier, and validation would require retraining for multilingual deployment.

Scalability Ceiling: Single-server deployment handles 50 concurrent rooms (10,000 users). Larger deployments require horizontal scaling infrastructure (load balancers, distributed caching), increasing operational complexity.

Difficulty Subjectivity: 82% accuracy represents ceiling given inherent subjectivity—human annotators achieved only κ=0.74 agreement. Perfect automated classification may be unattainable; tool should support educator override rather than replace judgment.

C. Comparison with Commercial Solutions

EduQuest achieves 91% quality of manual creation at 3% time cost. Commercial AI tools (Quizlet Q-Chat) achieve ~95% quality but incur API costs and lack transparency [13]. Traditional platforms (Kahoot!, Quizizz) provide excellent delivery but no generation capabilities. EduQuest's hybrid approach fills this gap: good-enough quality, affordable cost, transparent methodology, integrated workflow.

D. Threats to Validity

Internal Validity: User study limited to 8 weeks; novelty effect may inflate engagement scores. Longer-term studies needed to assess sustained impact. Educator sample (N=45) from three institutes may not represent diverse teaching contexts.

External Validity: Participant demographics skewed toward Indian coaching institutes (ages 16-22, competitive exam preparation). Results may not generalize to K-12, higher education, or corporate training contexts with different content types and learning objectives.

Construct Validity: Question quality assessed by educators, not learning outcomes directly. High-quality questions don't guarantee learning without effective pedagogy. Future work should measure knowledge retention and transfer.

E. Ethical Considerations

Academic Integrity: Automated generation raises concerns about assessment authenticity. System should be positioned as educator tool (like textbooks) requiring thoughtful integration, not replacement of instructor expertise.

Bias in Training Data: Difficulty classifier trained on existing quiz datasets may perpetuate biases regarding what constitutes "hard" vs "easy." Feature engineering should be examined for fairness across demographic groups.

Data Privacy: Student response data must be protected. Current implementation stores minimal PII; future analytics features require GDPR/FERPA compliance.

F. Future Research Directions

1) Adaptive Difficulty: Integrate Item Response Theory (IRT) to dynamically adjust question difficulty based on individual student performance, creating personalized learning paths.
2) Multimodal Question Generation: Extend beyond text to generate questions from diagrams, code snippets, mathematical equations, and videos— particularly valuable for STEM education.

3) Explanation Generation: Automatically generateanswer explanations supporting formative assessment and self-study modes. Preliminary experiments with T5 [2] show promise (ROUGE-L: 0.68).
4) Cross-Lingual Support: Leverage multilingual transformers (mBERT, XLM-R) to support quiz generation in regional Indian languages (Hindi, Tamil, Bengali), expanding accessibility.
5) Long-Term Impact Studies: Conduct semester-long controlled trials measuring knowledge retention, transfer, and standardized test performance comparing EduQuest-assisted instruction with traditional methods.
6) Open Question Generation: Extend beyond MCQs to short-answer and essay questions using recent advances in automatic essay scoring and natural language generation.

VII. CONCLUSION

EduQuest is a hybrid AI-powered quiz and real-time assessment platform that uses custom NLP, rule-based question generation, and machine learning-based difficulty classification, with commercial APIs as optional enhancements. It achieved 82% difficulty classification accuracy, high-quality questions (4.2/5), 71% higher student engagement, and sub-200ms latency for 200+ concurrent users. By relying on transparent, open-source technologies, EduQuest saves educators 78% of quiz creation time and democratizes AI-powered assessment for budget-constrained institutions. Future work includes adaptive difficulty, multimodal question generation, cross-lingual support, and studies on long-term learning outcomes.

REFERENCES

[1] OECD, "TALIS 2018 Results (Volume I): Teachers and School Leaders as Lifelong Learners," OECD Publishing, Paris, 2019. DOI: 10.1787/1d0bc92a-en
[2] C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," Journal of Machine Learning Research, vol. 21, no. 140, pp. 1-67, 2020.
[3] S. Rose, D. Engel, N. Cramer, and W. Cowley, "Automatic keyword extraction from individual documents," in Text Mining: Applications and Theory, M. W. Berry and J. Kogan, Eds. Chichester, UK: John Wiley & Sons, 2010, pp. 1-20.
[4] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," Journal of Machine Learning Research, vol. 3, pp. 993-1022, Jan. 2003.
[5] M. Honnibal and I. Montani, "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing," 2017. [Online]. Available: https://spacy.io
[6] R. Flesch, "A new readability yardstick," Journal of Applied Psychology, vol. 32, no. 3, pp. 221-233, June 1948.
[7] M. Heilman and N. A. Smith, "Good question! Statistical ranking for question generation," in Proc. Human Language Technologies: NAACL, Los Angeles, CA, USA, 2010, pp. 609-617.
[8] X. Du, J. Shao, and C. Cardie, "Learning to ask:Neural question generation for reading comprehension," in Proc. 55th Annual Meeting of the ACL, Vancouver, Canada, 2017, pp. 1342-1352.
[9] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in Proc. EMNLP, Austin, TX, USA, 2016, pp. 2383-2392.
[10] L. Benedetto et al., "Does difficulty matter? A study on difficulty of questions in neural question generation," in Proc. Florence, Italy, 2019.
[11] M. Dougiamas and P. Taylor, "Moodle: Using learning communities to create an open source course management system," in Proc. EDMEDIA+ Innovate Learning, Honolulu, HI, USA, 2023, pp. 171-178.
[12] A. Domínguez et al., "Gamifying learning experiences: Practical implications and outcomes," Computers & Education, vol. 63, pp. 380-392, Apr. 2013.
[13] E. Kochmar et al., "Automated personalized feedback improves learning gains in an intelligent tutoring system," in Proc. International Conference on Artificial Intelligence in Education, Online, 2021, pp. 140-146
[14] R. Mihalcea and P. Tarau, "TextRank: Bringing order into text," in Proc. 2004 Conference on EMNLP, Barcelona, Spain, 2004, pp. 404-411.
[15] S. Narayan, C. Gardent, S. B. Cohen, and A. Shimorina, "Split and rephrase," in Proc. 2017 Conference on EMNLP, Copenhagen, Denmark, 2017, pp. 606-616.
[16] N. Duan, D. Tang, P. Chen, and M. Zhou, "Question generation for question answering," in Proc. 2017 Conference on EMNLP, Copenhagen, Denmark, 2017, pp. 866-874.