International Journal of Creative and Open Research in Engineering and Management
ISSN: 3108-1754 (Online)
Volume 02 Issue 10 October-2026 | Impact Factor: 3.5



EduArchive AI 2.0: An AI-Powered Academic Intelligence and Previous Year Question Paper Analytics Platform

Ms. Tanvi Jagtap[1], Ms. Vaidehi Kale[2], Ms. Shamal Jadhav[3], Ms. Akshada Diwate[4], Dr. Maithili S. Deshmukh[5]

[1234]Student, Information Technology, Prof. Ram Meghe Institute of Technology and Research, Badnera, Maharashtra, India

[5] Associate Professor, Information Technology, Prof. Ram Meghe Institute of Technology and Research, Badnera, Maharashtra, India

Student Email: vaidehikale192@gmail.com

Abstract—

Previous Year Question Papers (PYQs) are valuable academic resources that help students understand examination patterns, identify frequently asked questions, recognize important topics, and plan their preparation effectively. However, conventional question-paper repositories mainly provide document storage and downloading facilities, requiring students to manually browse and compare papers from different years. This process is time-consuming and makes it difficult to identify repeated questions and important topics systematically. To address these limitations, EduArchive AI 2.0 is proposed as an artificial intelligence-powered academic intelligence and question-paper analytics platform. The system integrates Optical Character Recognition (OCR), Natural Language Processing (NLP), sentence embeddings, semantic search, Large Language Models (LLMs), and Retrieval-Augmented Generation (RAG) to transform historical examination papers into a searchable academic knowledge base. Its multi-agent architecture consists of five specialized agents: Ingestion, Analysis, Search, Recommendation, and Admin Intelligence. The platform supports processing digital and scanned PDF documents, detecting semantically similar questions, identifying frequently occurring topics, classifying question difficulty, and retrieving relevant academic information through natural-language queries. The RAG-based "Ask Your Papers" assistant is designed to generate responses using relevant content retrieved from uploaded question papers. The proposed implementation uses React.js, Python Flask, Sentence Transformers, ChromaDB, LangChain, Redis, and Docker. EduArchive AI 2.0 aims to reduce manual document analysis, improve access to academic information, and support more organized examination preparation. Its effectiveness should be evaluated using representative question-paper datasets and measurable performance metrics.

HOW TO CITE THIS ARTICLE:

Jagtap, T., Kale, V., Jadhav, S. & Diwate, A. (2026). EduArchive AI 2.0: An AI-Powered Academic Intelligence and Previous Year Question Paper Analytics Platform. International Journal of Creative and Open Research in Engineering and Management, 02(10), 1-9. https://doi.org/10.55041/ijcope.v2i10.140

License:

This article is published under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

© The Author(s). Published by International Journal of Creative and Open Research in Engineering and Management.





https://doi.org/10.55041/ijcope.v2i10.140

Keywords— Artificial Intelligence; Previous Year Question Papers; Natural Language Processing; Optical Character Recognition; Semantic Search; Retrieval-Augmented Generation.

I. INTRODUCTION

Previous Year Question Papers (PYQs) are important academic resources that help students understand examination patterns, identify frequently asked questions, recognize important topics, and plan their examination preparation. However, these papers are commonly available as separate PDF files, scanned documents, photocopies, or files stored in different digital repositories. As a result, students often need to manually open and compare multiple papers to find relevant questions, making the preparation process time-consuming and inefficient.

EduArchive AI 2.0 is proposed as an Artificial Intelligence (AI)-powered academic intelligence platform that transforms conventional PYQ repositories



into a searchable and analytical knowledge base. The system integrates Optical Character Recognition (OCR), Natural Language Processing (NLP), sentence embeddings, semantic search, Large Language Models (LLMs), and Retrieval-Augmented Generation (RAG) to process and analyze historical examination papers. OCR enables text extraction from scanned documents, while semantic embeddings support the identification of questions with similar meanings despite differences in wording. RAG enables the system to retrieve relevant information from uploaded question papers before generating responses to student queries [1], [2].

The proposed platform supports natural-language search, repeated-question detection, topic analysis, question-difficulty classification, unit-wise importance identification, and personalized recommendations. It also introduces an "Ask Your Papers" assistant that aims to provide responses grounded in retrieved academic content. The system uses React.js, Python Flask, LangChain, Sentence Transformers, ChromaDB, Redis, and Docker to support its proposed web-based and AI-driven architecture.

EduArchive AI 2.0 organizes these capabilities through five specialized agents: Ingestion Agent, Analysis Agent, Search Agent, Recommendation Agent, and Admin Intelligence Agent. Each agent is assigned a specific responsibility within the overall workflow, from document processing to academic analysis and administrative reporting. The proposed system aims to reduce the manual effort involved in searching and comparing question papers while making historical examination information easier to access and analyze.

A. Motivation

The primary motivation behind EduArchive AI 2.0 is to reduce the manual effort required to search, compare, and analyze historical examination papers. Traditional repositories mainly provide storage and downloading facilities without offering intelligent analysis of their contents. Students therefore spend considerable time identifying repeated questions and important topics.

The proposed system aims to make academic information more accessible by converting static question papers into searchable and analyzable resources. It also aims to support students in understanding examination trends and organizing their preparation more effectively through AI-based assistance.

B. Objectives

The main objectives of EduArchive AI 2.0 are:

1. To develop a document-ingestion system for processing digital and scanned Previous Year Question Papers.
2. To extract text from scanned question papers using Optical Character Recognition (OCR).
3. To generate semantic embeddings for questions and document segments.
4. To provide natural-language semantic search across multiple question papers.
5. To identify repeated questions and frequently occurring topics.
6. To analyze topic frequency, unit-wise importance, and question difficulty.
7. To implement a RAG-based academic assistant that answers queries using uploaded question papers.
8. To provide personalized recommendations for relevant questions and academic resources.
9. To offer administrators usage statistics and academic trend analytics.
10. To develop a modular and scalable architecture for intelligent academic resource management.

II. LITERATURE REVIEW

The development of intelligent academic-support systems has progressed with advances in Natural Language Processing, semantic search, Large Language Models, and Retrieval-Augmented Generation. Existing research has explored these technologies for document retrieval, examination-question analysis, educational assistance, and personalized learning. EduArchive AI 2.0 draws on these research directions to propose an integrated platform for Previous Year Question Paper analysis.

Lewis et al. [1] introduced Retrieval-Augmented Generation (RAG), which retrieves relevant information from external knowledge sources before generating a response. This approach provides the foundation for the "Ask Your Papers" feature of EduArchive AI 2.0, where responses are intended to be grounded in uploaded academic documents.

Reimers and Gurevych [2] proposed Sentence-BERT (SBERT), which generates meaningful sentence



embeddings for semantic similarity comparison. This technique is relevant to identifying questions with similar meanings despite differences in wording and supports the proposed platform's semantic-search functionality.

Thawkar et al. [3] presented Learnova, an ML-powered learning system that uses OCR and NLP techniques to process examination papers and identify recurring and important topics. EduArchive AI 2.0 aims to extend this direction by integrating semantic retrieval, RAG-based assistance, recommendations, and multi-agent analysis into a unified academic platform.

Zotos et al. [4] investigated the use of NLP methods for estimating examination-question difficulty. Their research is relevant to the proposed difficulty-classification component, which aims to categorize questions according to their estimated difficulty.

Dimitrov [5] explored duplicate-question detection using word embeddings and convolutional neural networks. This research supports the use of semantic representations to identify conceptually similar questions even when their wording differs.

Dan et al. [6] introduced EduChat, a large-language-model-based chatbot system for intelligent education. Their work demonstrates the potential of conversational AI for educational support. EduArchive AI 2.0 applies a related approach to historical question papers by using retrieved academic content to support responses to student queries.

Song and Shmatikov [7] examined vulnerabilities in OCR systems and highlighted challenges associated with document quality and image distortions. This research is relevant to the proposed system because scanned question papers may contain low-resolution text, distorted images, or complex layouts that affect extraction quality.

Khanal et al. [8] reviewed machine-learning-based recommendation systems for e-learning and discussed the use of learner-related information in educational recommendations. This provides background for the Recommendation Agent of EduArchive AI 2.0, which aims to suggest relevant academic resources based on available student interactions and subject information.

III. RESEARCH GAP

The reviewed studies provide foundations for individual capabilities such as RAG-based question answering, semantic embeddings, OCR, question-difficulty estimation, duplicate detection, educational chatbots, and personalized recommendations. However, these capabilities are generally addressed as separate tasks rather than as components of a unified Previous Year Question Paper analytics platform.

EduArchive AI 2.0 addresses this integration gap by proposing a platform that combines PDF and OCR processing, Sentence Transformer embeddings, ChromaDB-based semantic retrieval, repeated-question identification, topic analysis, difficulty classification, RAG-based question answering, personalized recommendations, and administrative analytics.

The principal contribution of the proposed system is the integration of these capabilities through five specialized agents. This architecture aims to transform conventional question-paper repositories into an interactive academic knowledge-management system that supports searching, analysing, and retrieving historical examination information within a common platform.

on procedures, and analysis techniques in detail. This section should provide sufficient information for other researchers to replicate the study. Mention tools, software, or frameworks used, and ensure ethical considerations are addressed where applicable. The data collection process involved a mixed-methods approach, combining quantitative surveys with qualitative interviews. A sample of 500 participants was recruited using stratified random sampling to ensure representation across demographic groups. Statistical analysis was conducted using SPSS software, while qualitative data was coded and analyzed thematically using NVivo.

IV. METHODOLOGY

The methodology of EduArchive AI 2.0 focuses on converting Previous Year Question Papers (PYQs) into searchable and analyzable academic information. The system integrates document ingestion, text extraction, preprocessing, semantic embedding generation, vector storage, and AI-based analysis. It supports both digital and scanned PDFs to improve access to historical examination content.



A. Document Processing-The system accepts question papers uploaded by the administrator. Text is extracted directly from digital PDFs, while Tesseract OCR is used for scanned documents. The extracted text is cleaned to remove unnecessary content and organized into meaningful question segments.

B. Semantic Embedding and Storage-Sentence Transformer models convert the processed text into semantic embeddings. These embeddings and their associated metadata are stored in ChromaDB, enabling efficient retrieval of relevant questions based on their meaning.

C. Semantic Search and AI Analysis-When a student enters a query, the system retrieves relevant content using semantic similarity. The Analysis Agent identifies repeated questions, important topics, unit-wise distribution, and question difficulty.

D. RAG-Based Question Answering-The "Ask Your Papers" feature uses Retrieval-Augmented Generation (RAG) to retrieve relevant question-paper content and provide contextual responses through a Large Language Model. The Recommendation Agent suggests relevant academic resources, while the Admin Intelligence Agent provides document and usage analytics.

Thus, the methodology combines document processing, semantic retrieval, and AI-based analysis to support efficient examination preparation.

V. SYSTEM ARCHITECTURE

The system architecture of EduArchive AI 2.0 consists of multiple interconnected layers that work together to process question papers and provide intelligent academic services. Each layer performs specific functions, from user interaction to AI-based analysis and deployment.

1. User Layer: This layer provides access to students and administrators through desktop, laptop, and mobile devices. Students can search question papers, ask questions, view results, and receive recommendations, while administrators manage academic documents.

2. Frontend Layer: The frontend is developed using React.js with Vite. It provides dashboards, semantic search, AI assistance, paper viewing, and analysis features. Tailwind CSS or Material UI supports interface design, Redux Toolkit manages application state, and Axios handles API communication.

3. Backend Layer: Python Flask manages application logic and communication between frontend and AI services. Flask REST APIs handle requests, Flask JWT Authentication supports secure authentication, and Flask Blueprints provide a modular backend structure.

4. AI Services Layer: This layer processes uploaded documents using PDF text extraction and OCR. Sentence Transformers generate semantic embeddings, LangChain coordinates AI workflows, and Large Language Models accessed through OpenRouter support question classification, topic analysis, and response generation.

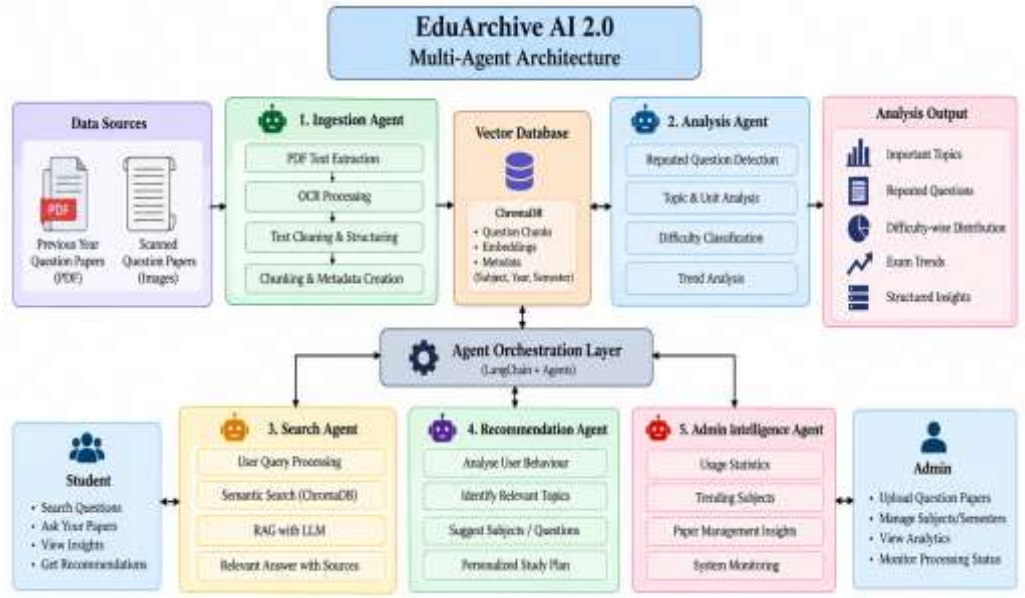
5. Data Layer: ChromaDB stores semantic embeddings for efficient retrieval. PostgreSQL or MySQL manages core application data, while Redis supports caching and session handling.

6. AI Agents Layer: The architecture includes five specialized agents: Ingestion Agent for document processing, Analysis Agent for identifying important and repeated questions, Search Agent for semantic retrieval and RAG-based answers, Recommendation Agent for personalized academic suggestions, and Admin Intelligence Agent for usage statistics and academic analytics.

7. Infrastructure and Deployment Layer: Docker supports application containerization, Nginx handles web requests, and cloud platforms such as AWS, Render, or DigitalOcean can be used for deployment.

These layers work together to transform static question-paper documents into a searchable academic knowledge base and provide intelligent examination-preparation support

Fig. 1: System Architecture



VI. IMPLEMENTATION

EduArchive AI 2.0 integrates frontend development, backend services, document processing, AI models, and database technologies to provide academic intelligence features. The implementation is organized into the following components.

1) Frontend Development: React.js with Vite provides the user interface for dashboards, question-paper search, and AI-assisted interaction. Tailwind CSS or Material UI supports interface styling, while Redux Toolkit and Axios manage application state and API communication.

2) Backend Development: Python Flask provides REST APIs for handling user requests, document uploads, and communication with AI services. Flask Blueprints organize the backend into modules, while JWT authentication supports secure access.

3) Document Processing: Uploaded PDFs are processed using PyMuPDF or pdfplumber for direct text extraction. Tesseract OCR recognizes text from scanned question papers, allowing both digital and scanned documents to be processed.

4) Text Preprocessing and Embedding: The extracted text is cleaned, structured, and divided into meaningful chunks. Sentence Transformers convert these chunks into semantic embeddings that represent their textual meaning.

5) Vector Storage and Retrieval: ChromaDB stores embeddings and associated document information. The system uses semantic retrieval to identify question-paper segments relevant to student queries.

6) AI and RAG Integration: LangChain coordinates the retrieval workflow, while OpenRouter provides access to Large Language Models. The RAG pipeline

7) retrieves relevant document content before generating answers through the “Ask Your Papers” assistant.

8) Multi-Agent Processing: The Ingestion, Analysis, Search, Recommendation, and Admin Intelligence Agents perform their respective tasks, including document processing, repeated-question detection, topic analysis, academic recommendations, and administrative reporting.

9) Database and Deployment: PostgreSQL or MySQL stores core application data, while Redis supports caching and session handling. Docker packages application components for deployment, with Nginx and cloud-hosting



platforms supporting infrastructure requirements.

The integration of these components provides the foundation for semantic search, academic question analysis, personalized recommendations, and AI-assisted examination preparation.

VII. RESULTS AND DISCUSSION

EduArchive AI 2.0 is designed to transform conventional Previous Year Question Paper repositories into an intelligent academic platform by integrating Artificial Intelligence, Natural Language Processing, Optical Character Recognition, semantic search, and Retrieval-Augmented Generation (RAG). The proposed system supports the processing of digital and scanned question papers and aims to improve access to historical examination information. The major expected outcomes of the system include:

1. Document Processing: Extracts text from digital PDFs and scanned question papers.
2. Semantic Search: Retrieves relevant questions based on the meaning of student queries.
3. Repeated-Question Detection: Identifies potentially similar questions across different examination years.
4. Topic Analysis: Identifies frequently occurring topics and unit-wise question distribution.
5. Difficulty Classification: Supports the classification of questions into different difficulty levels.
6. AI-Based Question Answering: Generates responses using relevant information retrieved from uploaded papers.
7. Personalized Recommendations: Suggests relevant questions, subjects, and academic resources.
8. Administrative Analytics: Provides upload statistics, document-processing status, and information about frequently accessed papers.

A. Performance Evaluation

The effectiveness of EduArchive AI 2.0 can be assessed by evaluating each module using suitable performance metrics. Table I presents the proposed evaluation parameters.

Table I. Evaluation Parameters of EduArchive AI 2.0

Sr. No.	Module	Evaluation Parameter
1	PDF Text Extraction	Text extraction accuracy
2	OCR Processing	Character or word recognition error
3	Semantic Search	Precision@k and Recall@k
4	Repeated-Question Detection	Precision, recall, and F1-score
5	Topic Classification	Accuracy and F1-score
Sr. No.	Module	Evaluation Parameter
6	Difficulty Classification	Classification accuracy
7	RAG-Based Assistant	Answer correctness and relevance
8	Recommendation System	Recommendation relevance

B. Discussion

The proposed system combines document processing, semantic retrieval, and AI-based analysis within a unified platform. OCR supports the processing of scanned question papers, while Sentence Transformer embeddings enable semantic search based on the meaning of queries rather than exact keyword matches. ChromaDB supports vector-based retrieval, and the RAG pipeline uses relevant document content to generate contextual responses.

The five-agent architecture separates document ingestion, academic analysis, search, recommendations, and administrative analytics, providing a modular structure for the platform. However, the system's effectiveness depends on document quality, OCR performance, retrieval relevance, and the reliability of language-model responses. Errors in scanned text may affect subsequent analysis, and semantically similar



questions may not always be exact duplicates. Therefore, generated insights and answers require validation against the original question papers.

The proposed evaluation should use representative question-paper datasets and manually verified results to measure the performance of individual modules. Since controlled experimental measurements are not established in the available project information, numerical accuracy values and claims of reduced examination-preparation time cannot yet be reported as achieved results. Future testing should evaluate retrieval quality, question-similarity detection, answer correctness, and user experience to determine the practical effectiveness of EduArchive AI 2.0.

VIII. CONCLUSION

EduArchive AI 2.0 presents an AI-powered approach to transforming conventional Previous Year Question Paper repositories into an intelligent academic knowledge base. The system integrates OCR, NLP, semantic embeddings, ChromaDB, Large Language Models, and Retrieval-Augmented Generation to support document processing, semantic search, repeated-question detection, topic analysis, and AI-based question answering.

The proposed five-agent architecture organizes document ingestion, academic analysis, search, personalized recommendations, and administrative analytics into separate components. These features aim to help students identify relevant questions, understand examination patterns, and organize their preparation more efficiently. The system also provides a centralized platform for managing and accessing historical question papers.

Future work will focus on evaluating the system using real question-paper datasets, improving OCR and semantic retrieval quality, validating AI-generated responses, and measuring the effectiveness of personalized recommendations. Overall, EduArchive AI 2.0 aims to make academic resources more accessible and support intelligent, data-driven examination preparation.

REFERENCES

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 9459–9474.

[2] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, Hong Kong, 2019, pp. 3982–3992, doi: 10.18653/v1/D19-1410.

[3] P. Thawkar, M. Vaishnavi, M. Aneesha, S. Dabhade, and S. Salve, "Learnova – ML Powered Smart Learning System," in *Proceedings of the International Conference on Recent Advances in Intelligent and Sustainable Technologies (RAIST 2026)*, Atlantis Highlights in Intelligent Systems, vol. 17, 2026, pp. 100–109, doi: 10.2991/978-94-6239-707-1_9.

[4] C. Song and V. Shmatikov, "Fooling OCR Systems with Adversarial Text Images," arXiv:1802.05385, 2018.

[5] Y. Dan, Z. Lei, Y. Gu, Y. Li, J. Yin, J. Lin, L. Ye, Z. Tie, Y. Zhou, Y. Wang, A. Zhou, Z. Zhou, Q. Chen, J. Zhou, L. He, and X. Qiu, "EduChat: A Large-Scale Language Model-based Chatbot System for Intelligent Education," arXiv:2308.02773, 2023.

[6] Y. Dimitrov, "Combining Word Embeddings and Convolutional Neural Networks to Detect Duplicated Questions," arXiv:2006.04513, 2020.

[7] L. Zotos, I. P. de Jong, M. Valdenegro-Toro, A. I. Sburlea, M. Nissim, and H. van Rijn, "NLP Methods May Actually Be Better Than Professors at Estimating Question Difficulty," arXiv:2508.03294, 2025.

[8] S. S. Khanal, P. W. C. Prasad, A. Alsadoon, and A. Maag, "A Systematic Review: Machine Learning Based Recommendation Systems for E-Learning," *Education and Information Technologies*, vol. 25, pp. 2635–2664, 2020.