International Journal of Science, Strategic Management and Technology
Volume **02** Issue **04** April-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

A Multilingual Document Summarization and Deadline Detection for Educational Institutions using Openai

Ms. Chitra Alavani and Ms. Pallavi Joshi
Assistant Professor, Kaveri College of Arts, Science and Commerce, Pune, India

https://doi.org/10.55041/ijsmt.v2i4.122

Cite this Article: Alavani, C. (2026). A Multilingual Document Summarization and Deadline Detection for Educational Institutions using Openai. International Journal of Science, Strategic Management and Technology, *02*(04). https://doi.org/10.55041/ijsmt.v2i4.122

Abstract

Educational institutes receives many documents from various authorities, most of the times these are either as text-based PDFs or scanned images and are commonly written in English or regional languages. Understanding such documents can be challenging due to their length, unstructured format, and official language complexity. Many a times, users often miss important deadlines embedded within these documents as they are in pdf formats are never revisited again. The proposed system automatically generates concise summaries and extracts critical dates and deadlines using the OpenAI GPT-4o-mini large language model (LLM). The system employs a hybrid processing strategy that dynamically selects between text-based and vision-based inputs to optimize accuracy and token cost, thus increasing the accessibility to key information in the official documents.

Keywords: document intelligence, large language models, OCR, deadline extraction, document summarization, multimodal processing

1. Introduction

In the Indian academic and administrative ecosystem, critical information is frequently disseminated through circulars, notices, and official communications in PDF or scanned image formats. These documents often contain important deadlines related to examinations, admissions, scholarships, and compliance requirements. Due to their length and unstructured nature, stakeholders such as students, faculty members, and administrative staff may overlook key information, leading to missed deadlines and operational inefficiencies.

Traditionally, these documents are manually read, summarized, and important dates are noted separately. This manual process is time-consuming and error-prone, often resulting in incomplete summaries or missed deadlines. Furthermore, once dates are noted, there is no automated reminder mechanism, increasing the likelihood of missing important deadlines.

This paper presents a AI-based system that automatically summarizes uploaded documents and extracts deadline information using the OpenAI GPT-4o-mini model. The proposed approach utilizes the OpenAI API to process either extracted text or document images and dynamically selects the appropriate model based on input type to improve accuracy and cost efficiency.

2. Review of the Literature

2.1 Using OCR to Get Information from Documents

People have used Optical Character Recognition (OCR) a lot to turn scanned documents into text that computers can read. Smith (2007) talked about how the Tesseract OCR engine works and what it can do to change scanned images of text into

editable format. OCR systems mostly focus on digitizing text, and they often have trouble with layouts that are noisy or complicated.

Huang et al. (2019) assessed OCR effectiveness on complicated scanned documents, including receipts, and underscored difficulties associated with layout variability, noise, and recognition precision. These limitations show that OCR by itself is not enough for more complex tasks that require semantic understanding, like extracting deadlines.

2.2 Document Understanding Based on AI

Recent improvements to large language models have made it much easier to understand language in context. The GPT-4 technical report (OpenAI, 2023) showed that multimodal large language models can do advanced reasoning on both text and images.

Zhou et al. (2022) demonstrated that structured prompting strategies can enhance reasoning performance in large language models while preserving token efficiency. The GPT-4o and GPT-4o-mini system card (OpenAI, 2024) also talks about how lightweight and larger multimodal models can be more efficient and how they can be less expensive.

Even with these improvements, not much work has been done on cost-aware hybrid pipelines for automatically extracting deadlines from Indian institutional documents.

3. Proposed System

The proposed system is designed to process uploaded documents and generate structured outputs consisting of concise summaries and extracted deadlines. It integrates both text-based and vision-based processing to handle diverse document formats commonly encountered in institutional environments.

The system operates through a token-based processing pipeline. Large Language Models process inputs in the form of tokens rather than full words. Each document undergoes tokenization before transformer-based processing generates the output tokens.

Processing Pipeline

Document Upload
↓
Text Extraction (if required using OCR)
↓
Prompt + Document Content
↓
GPT-4o-mini / GPT-4o Processing
↓
Structured Output (Summary + Deadlines)

The model selection is performed dynamically depending on whether the input PDF contains machine text or scanned images.

3.1 OCR-Based Text Pipeline

At first, the system used OCR to turn documents, especially scanned PDFs, into text that machines could read. After that, the GPT-4o-mini model was used to process the text that had been taken out to make summaries and find deadlines. This method had benefits like using fewer tokens and processing time faster, which made it more efficient. But it also had big problems because OCR wasn't always accurate, especially when working with low-quality scans or complicated layouts. These mistakes often led to partial data loss, which made it harder to get accurate summaries and deadlines. So, even though the OCR-only pipeline worked well, it wasn't good enough for understanding documents reliably.

3.2 Hybrid Multimodal Pipeline

A hybrid multimodal pipeline was made to get around the problems with the OCR-based method. In this method, the system chooses the processing method based on the type of input document. The GPT-4o-mini model processes text-based PDFs directly as text, which is quick and uses few tokens. On the other hand, scanned or image-based documents are processed directly as images by GPT-4o or GPT-4o-mini without going through OCR first. This helps the model understand the visual content better by keeping contextual and structural information that OCR might lose. This method uses more tokens, but it makes summarization and deadline extraction much more accurate. The hybrid strategy strikes a balance between performance and cost.

Text-based PDFs → processed using GPT-4o-mini

Scanned/image-based documents → processed directly using GPT-4o / GPT-4o-mini

4. Working of the OpenAI Model

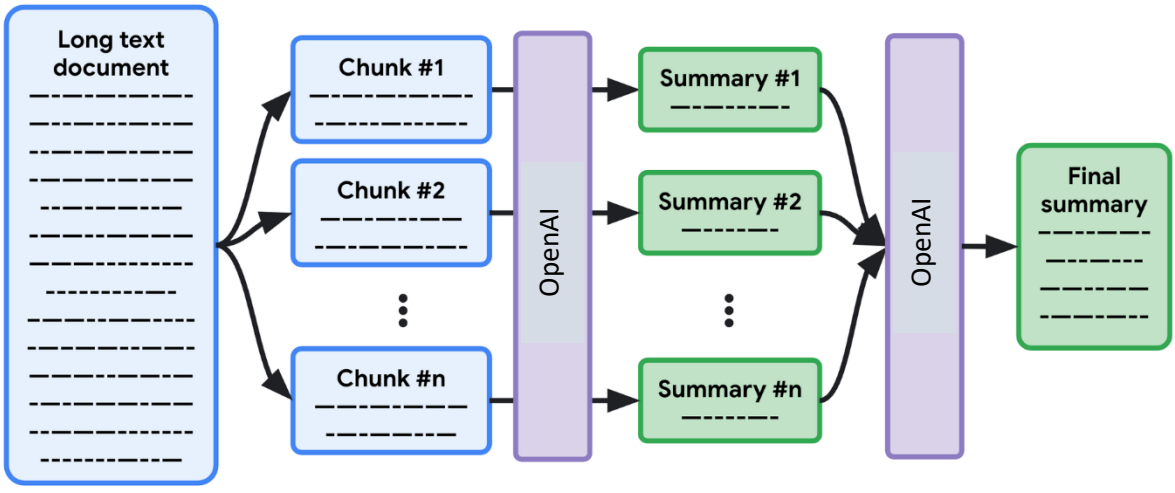
4.1 Large Language Model (LLM)

A Large Language Model (LLM) is a neural network that uses transformers and has been trained on a lot of text data to do things like understand language, summarize it, and pull out information. It uses tokenization and self-attention mechanisms to look at how words in a text relate to each other and make useful outputs. The proposed system relies heavily on the LLM to understand what documents mean and pull out useful information

4.2 Processing Based on Tokens

The model takes in all of its inputs as tokens instead of full words. Tokenization is the first step in processing the input document and the prompt. These tokens are then sent through several transformer layers, where self-attention mechanisms help them learn about contextual relationships. Lastly, the model makes output tokens that show the summary and the deadlines that were taken out. The total number of tokens processed includes both input tokens, which are the prompt and the content of the document, and output tokens, which are the response that was made.

4.3 Document Chunking and Summarization Process



The system uses a chunking-based method to summarize big documents. The LLM processes each smaller part of the document separately to make partial summaries. Then, these intermediate summaries are put together to make a final global summary. This method makes sure that big documents that go over token limits can still be processed correctly while keeping the meaning of the text.

4.4 Model Invocation (Implementation)

The system interacts with the OpenAI API to perform summarization and deadline extraction. A structured prompt is provided along with the document content to guide the model's response.

```
from openai import OpenAI

client = OpenAI()
response = client.chat.completions.create(
    model="gpt-4o-mini",
    messages=[
        {
            "role": "user",
            "content": f"""
            Summarize the document and extract all deadlines with dates.
            Document:
            {document_text}
            """
        }
    ]
)

summary = response.choices[0].message.content

input_tokens = response.usage.prompt_tokens
output_tokens = response.usage.completion_tokens
total_tokens = response.usage.total_tokens
```

4.5 Image-Based Processing

For scanned or image-based documents, the system directly sends the image to the model, enabling it to interpret the document visually without relying on OCR.

```
response = client.chat.completions.create(
    model="gpt-4o",
    messages=[
        {
            "role": "user",
            "content": [
                {"type": "text", "text": "Summarize and extract deadlines"},
                {"type": "image_url", "image_url": {"url": image_path}}
            ]
        }
    ]
)
```

5. Token and Cost Optimization

5.1 Token Calculation

The computational cost of the system is determined by the total number of tokens processed, which includes both input and output tokens. Input tokens consist of the prompt and document content, while output tokens correspond to the generated summary and extracted deadlines.

$$Total\ Tokens = Input\ Tokens + Output\ Tokens$$

5.1.1 Input Token Composition

Input tokens represent all tokens sent to the model during inference.

Input Tokens = Prompt Tokens + Document Tokens

Prompt Tokens:
These include the system instructions and task-specific prompts that guide the model to perform summarization and deadline extraction. Well-designed prompts improve accuracy but increase token usage. Therefore, prompt engineering was performed to keep instructions concise while preserving task clarity.

Document Tokens:
These correspond to the tokenized representation of the uploaded document content. Their volume depends on:

1. document length
2. language complexity
3. formatting noise
4. whether OCR text or raw image is used

Machine-generated PDFs typically produce cleaner and more compact token streams compared to OCR-generated text from scanned documents.

5.1.2 Output Token Composition

Output tokens correspond to the model-generated response and are computed as:

Output Tokens = Summary Tokens + Deadline Tokens

- Summary Tokens:
Tokens used to generate the abstractive summary of the document. The system controls this through prompt constraints (e.g., “summarize in 100 words”) to prevent unnecessary verbosity.

- Deadline Tokens:
Tokens used to produce the structured list of extracted dates and associated events. Since deadline outputs are relatively short, they contribute minimally to total cost.

5.2 Token Cost Example

Consider a scenario where the prompt consists of 120 tokens, the document contains 880 tokens, and the generated output contains 160 tokens. In this case, the total number of tokens processed is 1160. Based on typical pricing estimates, the cost per document is approximately $0.00074. In contrast, when image-based processing is used, the number of tokens increases (approximately 1500 or more), leading to higher cost but improved accuracy. This highlights the trade-off between efficiency and performance.

5.3 Cost Optimization Strategies

To ensure scalability for institutional deployment, the proposed system incorporates multiple optimization techniques:

1. Hybrid Processing Strategy
a. Text pipeline for machine-readable PDFs (lower token cost)
b. Vision pipeline for scanned documents (higher accuracy when OCR is unreliable)
2. Prompt Compression
Prompts were carefully engineered to be instruction-efficient while avoiding redundant wording.
3. Controlled Output Length
The summary length is explicitly bounded to prevent excessive output tokens.
4. Model Selection Policy
GPT-4o-mini is used as the default model due to its favorable cost–performance trade-off, while GPT-4o is invoked only when higher visual reasoning capability is required.

6. Experimental Observations

The system was evaluated using both English and Marathi (Devanagari) documents to assess its multilingual capabilities.

6.1 English Documents

For English documents, including both text-based PDFs and image-based PDFs, the system performed effectively. It successfully generated accurate summaries and correctly extracted deadlines. When image-based documents were processed directly without OCR, the accuracy improved further, although at the expense of increased token usage.

6.2 Marathi (Devanagari) Documents

For Marathi documents, the system demonstrated strong summarization capabilities. However, challenges were observed in deadline extraction, particularly in interpreting numerical values. In some cases, the model misread Devanagari numerals or misinterpreted date formats, leading to incorrect deadline extraction. This issue persisted regardless of whether the document was processed as text or as an image.

6.3 Key Insight

The experimental results indicate that while LLMs are highly effective in multilingual text understanding, they face challenges in accurately extracting numerical information from regional scripts, particularly in the context of deadline detection.

7. Results and Discussion

The OCR-only approach was found to be cost-efficient but less accurate due to information loss. The vision-based approach improved accuracy but increased token consumption. The hybrid pipeline provided the best balance, combining efficiency with improved performance, making it suitable for real-world deployment.

8. Sample output screenshot

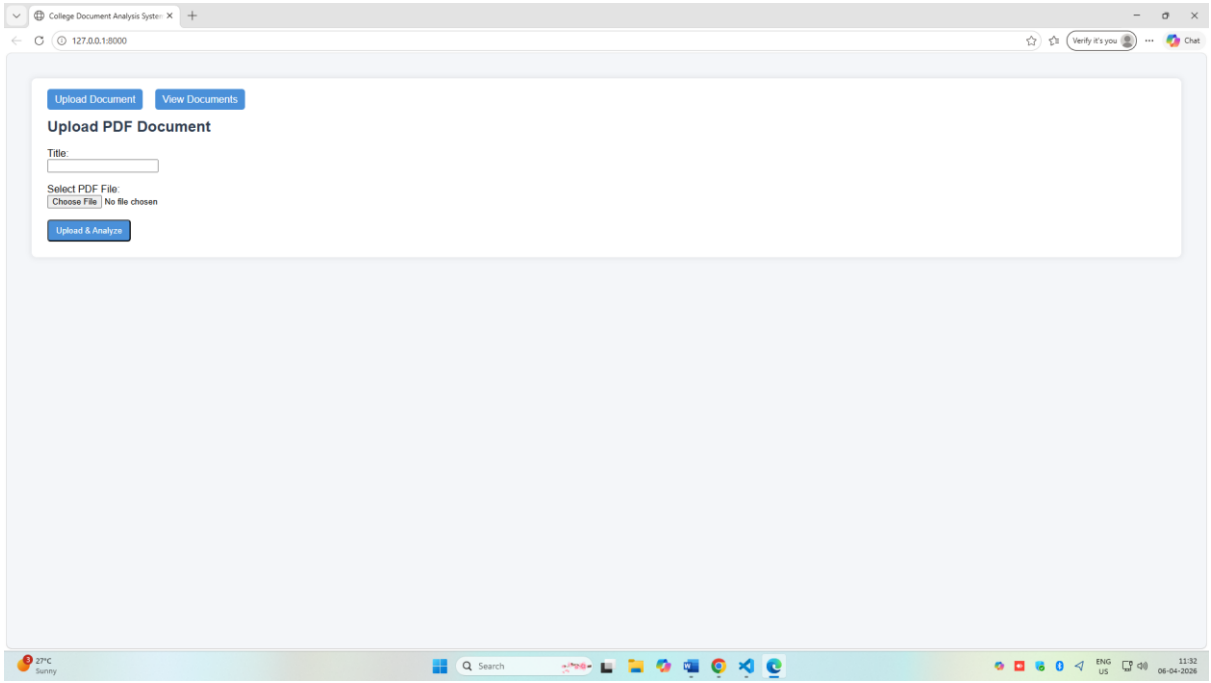


Figure 1: Screen to accept the document in PDF format

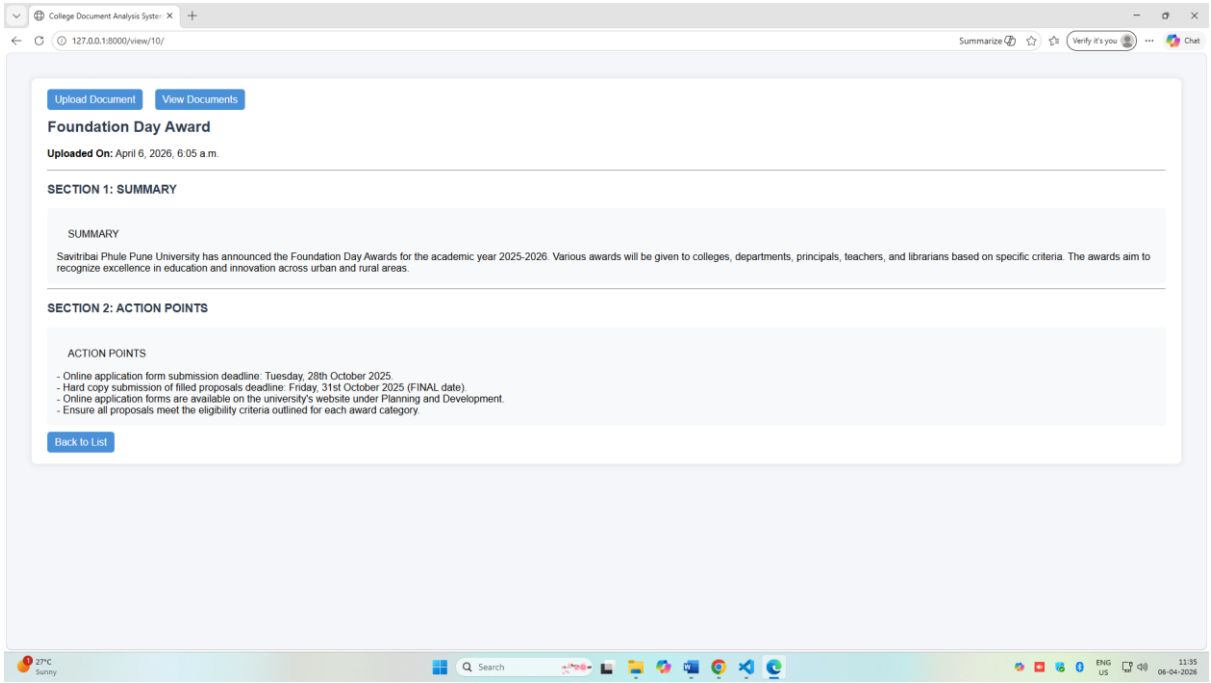


Figure 2: Summary and deadlines extracted from the uploaded document

9. Conclusion

The proposed system effectively automates document summarization and deadline extraction using a hybrid AI approach. By integrating OCR and multimodal LLM processing, it enhances accessibility to critical information and reduces the likelihood of missing important deadlines in institutional documents.

References

Huang, Z., et al. (2019). ICDAR 2019 robust reading challenge on scanned receipts OCR. *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*.

OpenAI. (2023). *GPT-4 technical report*. OpenAI.

OpenAI. (2024). *GPT-4o and GPT-4o mini system card*. OpenAI.

Smith, R. (2007). An overview of the Tesseract OCR engine. *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*.

Zhou, Y., et al. (2022). Least-to-most prompting enables complex reasoning in large language models. *Advances in Neural Information Processing Systems (NeurIPS)*.