GyanPatra (ज्ञानपत्र): AI-Powered Marksheet Extractor

30 Aug, 2025

Approach Note: AI-Powered Marksheet Extraction API

- **Author:** Gautam Kumar
- **Email:** gautamk8760@gmail.com
- **GitHub:** github.com/GitGautamHub

---

1. Project Overview & Objective

The primary objective of this project was to design and build a robust, scalable, and intelligent API that can extract structured information from a wide variety of academic marksheets. The system is engineered to handle diverse formats, including both image-based (JPG, PNG) and document-based (PDF) files, and to be resilient to variations in layout, language, and grading systems. The final output is a clean, structured JSON object containing the extracted data, with a derived confidence score for each field, providing a measure of the extraction's reliability.

---

2. System Architecture & Pipeline

The extraction process is implemented as a multi-stage, asynchronous pipeline, designed for efficiency and accuracy. The entire workflow is wrapped in a FastAPI backend, which exposes a batch processing endpoint.

Stage 1: File Ingestion & Validation

The process begins when a user uploads one or more files to the API.

- **Endpoint:** A FastAPI endpoint (`/extract/`) accepts a list of files.
- **Input Validation:** The frontend enforces a 10 MB file size limit. The backend uses the `python-magic` library to verify the true MIME type of each file by inspecting its binary content, making the system robust against incorrectly labeled files.

Stage 2: Intelligent OCR with DocTR

Raw text is extracted using **DocTR**, a state-of-the-art optical character recognition (OCR) engine.

- **Tool Selection:** DocTR was chosen over traditional tools like Tesseract due to its superior performance in layout analysis, which is critical for correctly parsing the complex tabular structures found in marksheets.
- **PDF Handling:** To ensure consistency, PDF files are first converted into a series of high-resolution images (300 DPI) using the Poppler library. This allows the OCR engine to process both native and scanned PDFs with high accuracy.

Stage 3: LLM-based Data Structuring

The unstructured text from the OCR stage is then passed to a Large Language Model (Google Gemini) for structuring.

- **Flexible Prompt Engineering:** The core of the system's intelligence lies in a highly flexible prompt. The prompt explicitly instructs the Gemini model to adapt its extraction logic based on the type of marksheet it encounters.
- **Adaptive Schema:** The prompt contains a detailed JSON schema that includes fields for both **marks-based** systems (`max_marks`, `obtained_marks`) and **grade-based** systems (`credits`, `grade_obtained`, `grade_point`, `sgpa`). The model is instructed to populate only the relevant fields, using `null` for those that don't apply, ensuring a consistent yet adaptive output structure.

---

3. Confidence Scoring Methodology

A key requirement was to provide a meaningful confidence score (0-1) for each extracted field. These scores are not generated by the LLM directly but are derived through a post-processing step using a **Hybrid Heuristic Model**.

This model combines a baseline trust in the LLM with a series of objective validation checks:

1. **Baseline Score:** Each successfully extracted non-null value starts with a high baseline confidence of **0.85**.
2. **Rule-Based Adjustments:** This score is then increased or decreased based on a set of validation rules:
    - **Format & Type Validation:** Fields like `date_of_birth` are checked if they can be parsed into a valid date (`+0.15` on success, `-0.40` on failure). Numeric fields are checked to ensure they are valid numbers. A new check for `exam_year` validates that the year is within a reasonable range (e.g., 1900-2100), increasing confidence.

    - **Logical Validation:** A critical check ensures a subject's `obtained_marks` are not greater than `max_marks` or negative. `max_marks` is also validated to be a positive number. A failure in these logical checks results in a significant penalty (`-0.70`), flagging the data as highly unreliable.
    - **Content Heuristics:** Text-based fields like `name` are checked for anomalies, such as the presence of numbers, which would lower the confidence score.

This hybrid approach makes the confidence scores transparent, explainable, and directly tied to the logical consistency of the extracted data.

---

4. Key Design & Technology Choices

- **FastAPI (Backend):** Chosen for its high performance, native asynchronous support, and automatic OpenAPI documentation. Its async capabilities were essential for building the efficient, concurrent batch processing endpoint using `asyncio.gather`.
- **Docker (Deployment):** Adopted to create a consistent, portable, and reliable production environment. This approach solved initial deployment challenges related to platform-specific system dependencies (like Poppler) and ensures the application builds and runs predictably anywhere.
- **DocTR (OCR):** Selected for its advanced deep learning models that excel at document layout analysis, providing cleaner and more structured text output for the LLM compared to alternatives.
- **Streamlit (Frontend):** Used for rapid prototyping of a user-friendly demo page. Its tight integration with Python allowed for the quick development of a functional UI for file uploads and result visualization.

---

5. Challenges & Solutions

- **Challenge:** High variability in marksheet formats, languages, and grading systems.
    - **Solution:** This was addressed by designing a flexible, adaptive prompt for the LLM and a comprehensive JSON schema that could accommodate different data types.
- **Challenge:** Inconsistent build environments during deployment, leading to permission errors and missing dependencies.
    - **Solution:** The project was migrated to a Docker-based deployment. By defining the entire environment in a `Dockerfile`, we ensured a consistent and reproducible build process, resolving all platform-specific issues.

---

6. Conclusion

The final system is a robust and intelligent API capable of handling the complexities of real-world academic documents. By combining a powerful OCR engine, flexible LLM prompting, and a logical confidence scoring model, the application delivers accurate, structured, and reliable data extraction, fulfilling all the core requirements of the project.