

Project Documentation: Automated CBSE/ICSE Textbook Scraper & Metadata Extractor

1. Executive Summary

This project delivers an end-to-end automated pipeline designed to extract structured metadata (Chapter Titles, Numbers, Page Ranges, Subject, Class, and Board) from unstructured PDF textbooks (specifically NCERT/CBSE).

The system employs a Hybrid Heuristic Architecture that dynamically switches extraction strategies based on the subject domain (STEM vs. Humanities) and cross-verifies data between the Table of Contents (ToC) and physical document layout to ensure high-fidelity output.

2. Design Philosophy & Approach

2.1. The Challenge of Unstructured PDFs

Educational PDFs are highly heterogeneous.

STEM Textbooks (Physics, Math): Contain complex layouts, equations, and diagrams that often confuse standard OCR. Titles often contain spaced typography (e.g., "U N I T S").

Humanities Textbooks (English, History): Feature literary excerpts, heavy use of running headers, and introductory paragraphs that mimic titles (e.g., "A short story is...").

2.2. Solution Architecture: Domain-Adaptive Extraction

Instead of a "one-size-fits-all" model, we implemented a routing logic:

1. Metadata Intelligence Module:

Decodes standardized filename nomenclatures (e.g., lekl101.pdf) to instantly determine Class, Subject, and Board with 100% accuracy, bypassing error-prone OCR for these fields.

2. Dual-Strategy Extraction Engine:

Strategy A (Vector-Layout Analysis): Used for Science/Math. Prioritizes font size hierarchy and spatial clustering to handle sparse text and equations. It includes a custom "Kerning Normalization Layer" to reconstruct words split by stylistic formatting.

Strategy B (Semantic-Layout Analysis): Used for Arts/English. Implements a "False Positive Rejection System" that filters out running headers, genre definitions (e.g., "Introduction to Poetry"), and non-title entities using frequency analysis and stop-word heuristics.
t.

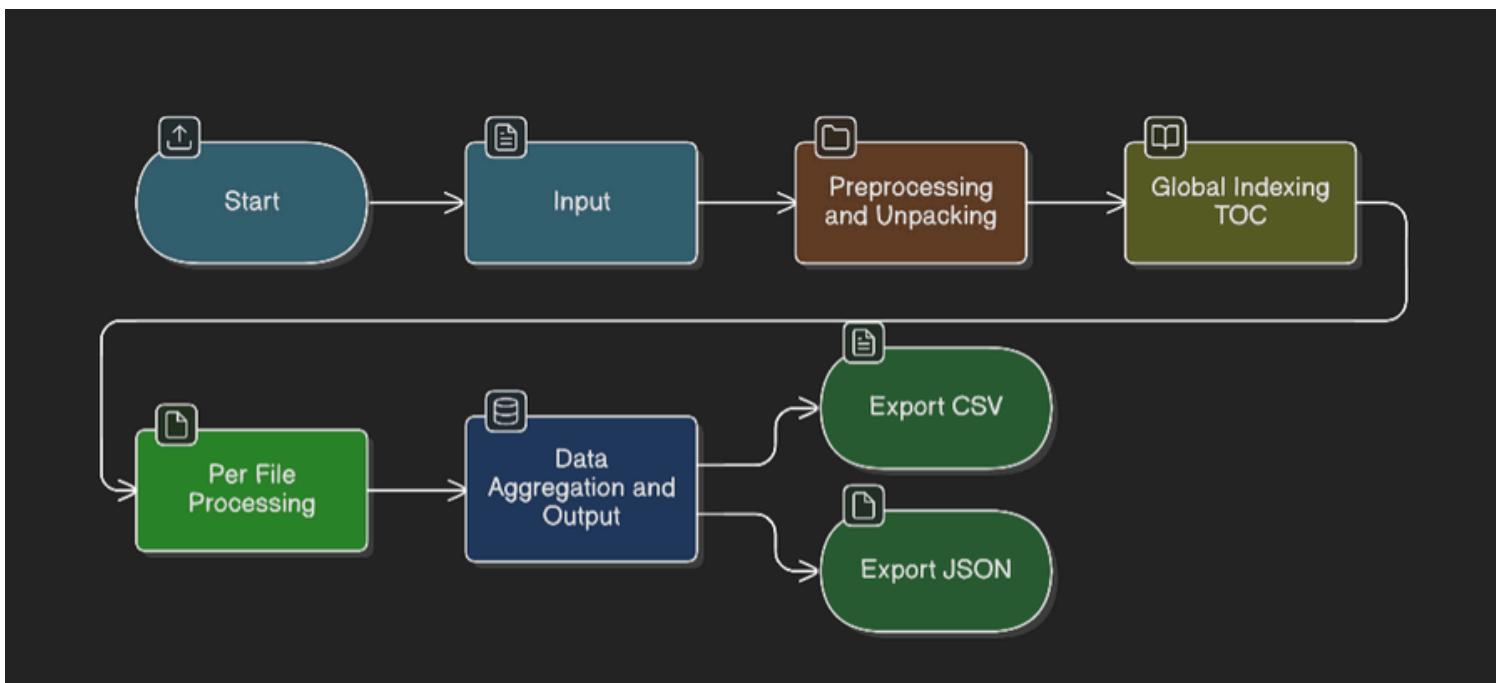
3. Multi-Source Validation (The "Golden Logic"):

The system parses the Table of Contents (ToC) separately to build a "Ground Truth" map. If the physical page number on the PDF matches a page entry in the ToC, the system prioritizes the ToC title, overriding any visual extraction errors.

Signal Processing & Normalization:

A post-processing layer applies Text Normalization to resolve OCR artifacts, character encoding errors, and typographic inconsistencies before finalizing the output.

3. System Architecture & Pipeline Flow



4. Pseudo-Code Implementation

CLASS Pipeline:

```
FUNCTION Run(input_file):
    files = Unzip(input_file)
    toc_map = Parse_Table_Of_Contents(files) # Ground Truth

    final_data = []

    FOR file IN files:
        # Step 1: Context
        metadata = Decode_Filename(file.name)

        # Step 2: Extraction Strategy
        IF metadata.Subject IS Science_Stream:
            raw_title = Extract_Largest_Vector_Text(file)
            # Handles spatial merging of letters
            cleaned_title = Apply_Kerning_Normalization(raw_title)
        ELSE:
            raw_title = Extract_Heuristic_Text(file)
            # Removes headers and definitions
            cleaned_title = Filter_False_Positives(raw_title)

        # Step 3: Verification
        physical_page_num = OCR_Bottom_Corner(file)
        IF physical_page_num IN toc_map:
            final_title = toc_map[physical_page_num] # Override with Truth
        ELSE:
            final_title = cleaned_title

        # Step 4: Final Polish
        # Advanced cleaning for artifacts and repetition
        final_title = Normalize_Text_Quality(final_title)

        final_data.APPEND({
            metadata,
            final_title,
            Calculate_Page_Range(physical_page_num, file.length)
        })

    RETURN Export_CSV_JSON(final_data)
```