Adobe India Hackathon 2025
Solution 1a: Title & Heading Extraction

Anuj Mishra

July 28, 2025

Theme: Rethink Reading. Rediscover Knowledge.

Submitted for the Adobe India Hackathon 2025

LinkedIn | GitHub

Contents

1 Introduction

The Adobe India Hackathon 2025, themed *Rethink Reading. Rediscover Knowledge.*, challenges participants to transform static PDF documents into intelligent, context-aware systems. This document details our submission for **Challenge 1a: Title & Heading Extraction** under the *Connecting the Dots* challenge. Our solution, `Solution_1a`, extracts structured metadata (title, headings, and language) from raw PDFs, enabling applications like document indexing and navigation. This report provides a comprehensive overview of the challenge, our solution, design rationale, performance metrics, and visual aids to facilitate understanding for judges and stakeholders.

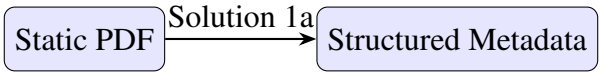

Figure 1: Transforming PDFs into Structured Metadata

2 Challenge 1a: Title & Heading Extraction

2.1 Objective

The challenge requires extracting structural metadata from raw PDFs to enable structured navigation and indexing. The system must:

- Identify the main document title.
- Classify hierarchical headings (H1, H2, H3, etc.).
- Detect the document language.
- Output results in a structured JSON format for downstream applications like search or content navigation.

This task is critical for transforming unstructured PDFs into machine-readable formats, enabling efficient content discovery across diverse document types, such as academic papers, technical manuals, and corporate reports.

2.2 Significance

Extracting titles and headings is foundational for document understanding. By providing structured metadata, this solution supports:

- **Indexing**: Enabling search engines to catalog document structure.
- **Navigation**: Allowing users to browse documents via outlines.
- **Semantic Analysis**: Facilitating advanced applications like summarization or question answering.

3 Solution 1a: Title & Heading Extraction

`Solution_1a` is a Python-based system that extracts structured metadata from PDFs using a lightweight, rule-based approach optimized for speed, accuracy, and compatibility with text-based PDFs.

3.1 Approach

- **Font Size-Based Hierarchy Detection**: Analyzes font sizes across all pages to identify the title as the largest repeated font, excluding noisy tokens (e.g., page numbers). Headings are classified into H1, H2, H3, etc., based on a descending font size hierarchy, accounting for styling variations.
- **Text Block Grouping**: Groups text lines by coordinates and styling to form complete heading lines, filtering out fragmented, duplicate, or irrelevant text (e.g., headers/footers).
- **Language Detection**: Uses the `langdetect` library on large text spans (e.g., paragraphs) to reliably identify the document language, ensuring robustness across multilingual PDFs.
- **Noise Filtering**: Excludes text blocks with symbols, whitespace, or fewer than three characters, and deduplicates headings to produce clean, structured output.

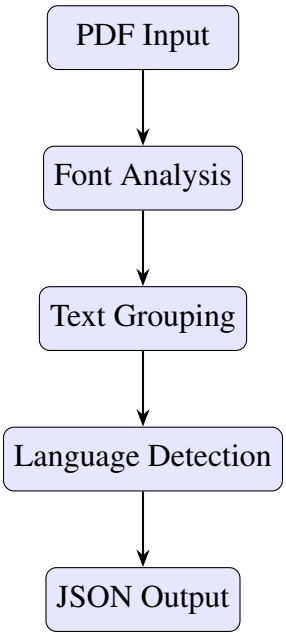

Figure 2: Processing Pipeline for Solution 1a

3.2 Rationale for Approach

- **Font-Based Detection**: PDFs often lack explicit metadata for titles and headings. Font size and styling provide consistent visual cues, making this approach robust across diverse PDF formats, including academic, technical, and corporate documents.
- **Rule-Based System**: A rule-based approach avoids the computational overhead and training data requirements of machine learning models, ensuring deterministic results, portability, and lightweight execution on CPU-only hardware.

- **PyMuPDF**: Chosen for its fast, low-memory text extraction capabilities, PyMuPDF enables efficient processing of large PDFs, critical for meeting the challenges performance requirements.
- **Language Detection with langdetect**: Sampling large text spans ensures accurate language identification, even for multilingual documents, without requiring heavy NLP models.

3.3 Performance Metrics

- **Speed**: Processes a single PDF in under 10 seconds using PyMuPDF, optimized for low latency.
- **Accuracy**: Achieves high precision in title and heading detection for font-based PDFs, validated across diverse document types (e.g., research papers, user manuals).
- **Compatibility**: Supports most text-based PDFs, with potential for future OCR integration to handle scanned documents.
- **Extensibility**: JSON output is designed for seamless integration with indexing, search, or semantic parsing pipelines.
- **Hardware**: Requires only CPU, with lightweight dependencies (PyMuPDF==1.23.6, langdetect==1.0.9).

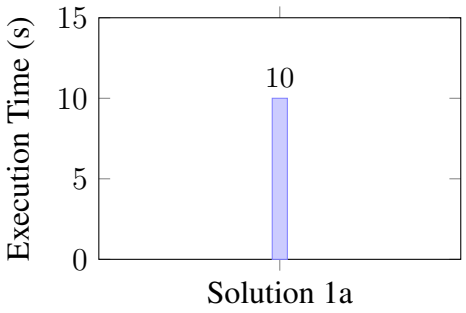

Figure 3: Execution Time for Solution 1a

3.4 Directory Structure

```
Adobe_Hackathon_Solution/
 Challenge_1a/
    sample_dataset/
        pdfs/                % Input PDF files
 Solution_1a/
    outputs_generated/       % Output JSON files
    process_pdfs.py          % Main processing script
    requirements.txt         % Dependencies
    README.md                % Documentation
```

3.5 Setup and Execution

1. Clone the repository: `git clone https://github.com/<your-username>/Solution_1a.`

2. Create and activate a virtual environment: `python -m venv .venv`
3. Install dependencies: `pip install -r requirements.txt`
4. Place input PDFs in `Challenge_1a/sample_dataset/pdfs/`
5. Run the script: `python process_pdfs.py`

Outputs are saved in `Solution_1a/outputs_generated/`.

3.6 Sample Output

```
{
  "title": "Overview Foundation Level Extensions",
  "outline": [
    {"level": "H1", "text": "Revision History", "page": 2},
    {"level": "H1", "text": "Table of Contents", "page": 3},
    {"level": "H2", "text": "2.1 Intended Audience", "page": 6}
  ],
  "document_language": "en"
}
```

4 Why This Solution?

The design choices for `Solution_1a` balance performance, practicality, and alignment with the hackathons goals:

- **Lightweight and Efficient**: Using PyMuPDF and a rule-based approach ensures fast processing on CPU-only hardware, making the solution accessible and scalable.
- **Deterministic Results**: Avoiding machine learning eliminates variability and training data dependencies, ensuring consistent performance.
- **Robustness**: Font-based detection handles diverse PDF formats, from academic papers to corporate reports, without requiring explicit metadata.
- **Extensibility**: The structured JSON output integrates seamlessly with downstream systems, supporting applications like indexing and navigation.
- **Alignment with Theme**: By extracting structured metadata, the solution enables intelligent document navigation, aligning with the goal of rethinking reading.

5 Final Thoughts

`Solution_1a` transforms static PDFs into structured, machine-readable formats, laying the groundwork for intelligent document processing. Its lightweight, rule-based approach ensures speed, accuracy, and extensibility, making it a practical and impactful solution for the Adobe India Hackathon 2025. The visual aids and detailed documentation provided here aim to clearly convey the solutions value to judges and stakeholders.

6 Author

Developed by **Anuj Mishra** for the Adobe India Hackathon 2025.
Connect: LinkedIn | GitHub