Adobe India Hackathon 2025

"Connecting the Dots" - Round 1A

Document Outline Extraction - Technical Approach

🎯 Primary Objective

Extract a structured outline from PDF documents including:

- Document Title
- Hierarchical Headings (H1, H2, H3) with exact text content
- Corresponding page numbers for each heading
- Return data in specified JSON format

📋 Technical Solution Architecture

1. Advanced PDF Parsing Strategy

Primary Technology: PyMuPDF (fitz)

Core Capabilities:

- **High-Performance Parsing:** Direct access to PDF structure without rendering overhead
- **Font Metadata Extraction:** Font size, family, weight, and style information
- **Spatial Layout Analysis:** Bounding box coordinates for text positioning
- **Multi-Page Processing:** Efficient page-by-page text block extraction

Key Advantage: Offline processing with sub-second per-page performance, enabling real-time document analysis without external dependencies.

2. Intelligent Text Block Processing

Normalization Pipeline:

- **Whitespace Filtering:** Remove empty blocks and normalize spacing
- **Block Consolidation:** Group fragmented text lines using coordinate proximity
- **Metadata Preservation:** Maintain font properties and positional data
- **Character Encoding:** Handle special characters and Unicode properly

Data Structure for Each Block:

- Text content (cleaned and normalized)

- Font size (in points)
- Font family and weight indicators
- Bounding box coordinates (x, y, width, height)
- Page number reference

3. Font-Based Hierarchy Detection

Statistical Font Analysis Approach

Step 3.1: Font Size Distribution Analysis

- Collect all unique font sizes across the document
- Calculate frequency distribution of each font size
- Apply clustering algorithms (K-means with k=4-5) to identify distinct size groups
- Rank clusters by size to establish hierarchy

Step 3.2: Hierarchical Level Assignment

- **Title Level:** Largest font size, typically on first page
- **H1 Level:** Second largest size group, often bold
- **H2 Level:** Third largest size group
- **H3 Level:** Fourth largest or bold variants of smaller sizes

4. Advanced Heading Classification Rules

Multi-Criteria Heading Detection:

Criteria Set A: Textual Characteristics

- **Length Filter:** ≤ 15 words (configurable threshold)
- **Sentence Structure:** No ending punctuation (period/comma)
- **Capitalization Patterns:** Title case or ALL CAPS detection
- **Numeric Prefixes:** Section numbering (1., 1.1, A., etc.)

Criteria Set B: Spatial Layout

- **Alignment Detection:** Left-aligned, center-aligned, or indented positioning
- **Isolation Check:** Surrounded by whitespace (not inline with paragraphs)
- **Page Position:** Top 80% of page preferred for headings
- **Margin Analysis:** Consistent left margins for same-level headings

Criteria Set C: Typographic Features

- **Font Weight:** Bold, semi-bold, or heavy weight detection
- **Font Style:** Italic variants for sub-headings
- **Color Analysis:** Different colors for hierarchy (if available)
- **Underline/Formatting:** Additional formatting indicators

🎯 Title Extraction Methodology

Multi-Phase Title Detection

Phase 1: First Page Analysis

- Focus on top 30% of first page
- Identify largest font size in this region
- Check for center alignment or prominent positioning
- Verify isolation from other text blocks

Phase 2: Validation Checks

- **Length Validation:** Reasonable title length (5-50 words)
- **Content Analysis:** Avoid headers/footers, page numbers
- **Contextual Relevance:** Should relate to document content
- **Format Consistency:** Matches expected title formatting

Phase 3: Fallback Strategies

- Search in document metadata if available
- Look for title patterns in first few pages
- Use filename as last resort (cleaned and formatted)

📊 Output Format Specification

```json
```

```
{
  "title": "Annual Financial Report 2024",
  "outline": [
    {
      "level": "H1",
      "text": "Executive Summary",
      "page": 3
    },
    {
      "level": "H2",
      "text": "Key Financial Highlights",
      "page": 4
    },
    {
      "level": "H3",
      "text": "Revenue Growth Analysis",
      "page": 5
    },
    {
      "level": "H1",
      "text": "Market Performance",
      "page": 8
    },
    {
      "level": "H2",
      "text": "Quarterly Results",
      "page": 9
    }
  ]
}
```


Output Quality Assurance

- **Hierarchical Consistency:** Ensure logical H1 → H2 → H3 flow
- **Page Number Accuracy:** Verify page numbers match actual heading locations
- **Text Cleaning:** Remove extra whitespace, special characters, line breaks
- **Duplicate Detection:** Handle repeated headings across pages

---

🐋 Docker Implementation Strategy

Container Architecture

Base Image: python:3.9-slim (lightweight, fast startup)

Directory Structure:

- `/app/input/` - PDF input files
- `/app/output/` - JSON output files
- `/app/src/` - Application source code
- `/app/models/` - Any ML models (if used)

Key Dependencies:

- **PyMuPDF:** Core PDF processing library
- **NumPy:** Numerical operations for font clustering
- **scikit-learn:** Clustering algorithms
- **Pandas:** Data manipulation and analysis

Performance Optimizations:

- Multi-layer caching for dependencies
- Minimal system packages installation
- Memory-efficient PDF processing
- Parallel processing for multi-document scenarios

⚡ Performance Optimization

Speed & Efficiency Targets

- **Processing Speed:** ≤ 10 seconds for 50-page documents
- **Memory Usage:** < 512MB RAM for large documents
- **Accuracy Target:** > 95% heading detection accuracy
- **Scalability:** Handle documents up to 200 pages

Optimization Techniques

- **Lazy Loading:** Process pages on-demand
- **Text Block Filtering:** Early elimination of non-heading blocks
- **Font Caching:** Cache font analysis results
- **Bounding Box Optimization:** Spatial indexing for faster lookups

---

💡 Innovation & Competitive Advantages

- **Confidence Scoring:** Assign confidence levels to each detected heading for quality assessment
- **Multi-Language Support:** Unicode handling for international documents

- **Format Adaptability:** Dynamic adjustment to different document styles
- **Error Recovery:** Graceful handling of malformed or encrypted PDFs
- **Modular Design:** Reusable components for Round 1B integration

---

🔧 Error Handling & Edge Cases

Robust Error Management

- **Corrupted PDFs:** Validation and repair attempts
- **Password-Protected Files:** Graceful failure with informative messages
- **Scanned Documents:** Detection and appropriate handling
- **Non-Standard Fonts:** Fallback font analysis methods
- **Empty Documents:** Proper handling of blank or image-only PDFs

Quality Assurance Measures

- **Output Validation:** JSON schema compliance checking
- **Consistency Checks:** Logical heading hierarchy validation
- **Performance Monitoring:** Processing time and memory usage tracking
- **Accuracy Metrics:** Automated testing against known document structures

---

Adobe India Hackathon 2025 - Round 1A Technical Approach Document Prepared for "Connecting the Dots" Challenge