Enhanced OCR for Scanned PDF Text Extraction

Implementation Summary - October 16, 2025

■ Problem Identified

• The "Bank Details Form.pdf" was returning 0 text length during extraction
• Scanned PDFs require OCR capabilities that weren't properly configured
• Need for a robust fallback system when primary extraction methods fail

■ Solution Architecture: 3-Tier Extraction Pipeline

Tier 1: Azure Document Intelligence (Primary)

• Uses Azure Cognitive Services Document Intelligence API
• Best for text-based PDFs and high-quality scanned documents
• Handles both text extraction and layout analysis

Tier 2: PyPDF2 Fallback

• Traditional PDF text extraction for text-based PDFs
• Backup when Azure DI fails or returns insufficient content
• Fast but doesn't work with scanned/image-based PDFs

Tier 3: OCR Pipeline (Ultimate Fallback)

• **pdf2image**: Converts PDF pages to PIL images
• **pytesseract**: Python wrapper for Tesseract OCR engine
• **Automatic path detection**: Finds Tesseract installation automatically
• Processes each page as an image and extracts text via OCR

■ Dependencies Installed

```
# OCR-related packages added to requirements.txt
pytesseract==0.3.10
pdf2image==1.17.0
# Pillow (already included for image processing)
```

■ Windows Environment Setup

Poppler Installation (Required for pdf2image)

1. Downloaded Poppler for Windows from GitHub releases
2. Extracted to C:\poppler\poppler-23.01.0\Library\bin
3. Added to system PATH permanently
4. Verified with `pdftoppm -h` command

Tesseract OCR (Already available)

• Automatic detection of Tesseract installation paths
• Supports multiple common Windows installation locations

■ Enhanced Code Implementation

The process_with_document_intelligence function now includes:

```
def process_with_document_intelligence(file_path):
    """
    Enhanced PDF processing with 3-tier fallback:
    1. Azure Document Intelligence (primary)
    2. PyPDF2 (fallback for text-based PDFs)
    3. OCR with pytesseract (ultimate fallback for scanned PDFs)
    """

    # Tier 1: Azure DI
    try:
        # Azure Document Intelligence processing
        if extracted_text and len(extracted_text.strip()) > 50:
            return extracted_text, entities
    except Exception as e:
        print(f"[PDF] Azure DI failed: {e}")

    # Tier 2: PyPDF2 fallback
    try:
        # Traditional PDF text extraction
        if text and len(text.strip()) > 50:
            return text, []
    except Exception as e:
        print(f"[PDF] PyPDF2 failed: {e}")

    # Tier 3: OCR fallback
    try:
        # Convert PDF to images and OCR each page
        text = ocr_pdf_pages(file_path)
        if text and len(text.strip()) > 50:
            return text, []
    except Exception as e:
        print(f"[PDF] OCR failed: {e}")
```

■ Results Achieved

Before Enhancement:

• "Bank Details Form.pdf": 0 characters extracted
• Failed to process scanned documents
• No fallback mechanisms

After Enhancement:

• "Bank Details Form.pdf": 849 characters extracted ■
• All PDF files processing successfully
• Robust fallback system operational
• Multilingual support maintained (Arabic PDFs → Arabic collections)

■ Key Features Added

1. **Automatic Path Detection**: Finds Tesseract installation automatically
2. **Comprehensive Error Handling**: Each tier fails gracefully to the next
3. **Content Validation**: Minimum 50-character threshold for meaningful extraction
4. **Detailed Logging**: Clear debug output showing which method succeeded
5. **Performance Optimization**: Primary methods tried first, OCR only as last resort
6. **Cross-Platform Compatibility**: Works on Windows with proper dependency management

■ Final Outcome

The enhanced system now successfully:
■ Extracts text from scanned PDFs using OCR
■ Maintains high performance for text-based PDFs
■ Provides robust fallback mechanisms
■ Supports both English and Arabic documents
■ Integrates seamlessly with the existing RAG pipeline

The "Bank Details Form.pdf" that was previously failing now extracts 849 characters successfully, demonstrating the effectiveness of the OCR enhancement!