DocuSearch AI: Semantic PDF Search with Visual and Tabular Insights

📋 Overview

DocuSearch AI is an advanced PDF search engine that goes beyond simple text matching. It leverages natural language processing and semantic search to understand the meaning behind your queries, then returns relevant content with complete context - including associated images and tables.

✨ Key Features

- **Semantic Search**: Find content based on meaning, not just keywords
- **Visual Context**: Automatically extracts and displays images relevant to search results
- **Table Recognition**: Identifies and preserves tabular data in search results
- **Interactive Interface**: Simple command-line interface for easy searching
- **Efficient Indexing**: Fast search performance using FAISS vector indexing

📊 How It Works

DocuSearch AI processes PDFs in four key steps:

1. **Text Extraction**: Pulls raw text content from the PDF
2. **Media Recognition**: Identifies and extracts images and tables
3. **Semantic Indexing**: Creates dense vector embeddings of text chunks using transformer models
4. **Similarity Matching**: Uses FAISS to efficiently find the most relevant content for your query

🔧 Installation

Prerequisites

- Python 3.7 or higher
- pip package manager

Setup

1. Clone this repository:
2. git clone https://github.com/yourusername/docusearch-ai.git

3. cd docusearch-ai
4. Create and activate a virtual environment (optional but recommended):
5. python -m venv venv
6. source venv/bin/activate # On Windows: venv\Scripts\activate
7. Install the required dependencies:
8. pip install -r requirements.txt

📚 Usage

Basic Usage

Run the main script and follow the prompts:

python docusearch.py

The program will ask you to:

1. Enter the path to your PDF file
2. Enter search queries

Example

Enter the path to your PDF file: research_paper.pdf

Processing PDF: research_paper.pdf

Processed 45 text chunks from 10 pages

Creating embeddings...

Embeddings created and indexed

Enter your search query (or 'exit' to quit): machine learning applications

[Displays search results with relevant text, images, and tables]

Enter your search query (or 'exit' to quit): exit

In Python Code

You can also use DocuSearch AI programmatically:

```
from docusearch import PDFSearchEngine

# Initialize the engine with your PDF
search_engine = PDFSearchEngine("path/to/document.pdf")

# Search for content
results_html = search_engine.search("your search query", top_k=5)

# In Jupyter/IPython environments:
from IPython.display import display, HTML
display(HTML(results_html))
```

🔍 Advanced Usage

Customizing Search Results

You can adjust the number of results returned:

```
# Get more results
results = search_engine.search("machine learning", top_k=5)

# Get fewer, more precise results
results = search_engine.search("machine learning", top_k=1)
```

📄 Requirements

The project requires the following Python packages:

- PyMuPDF (fitz)
- NLTK
- NumPy
- FAISS
- PyTorch

- Transformers
- IPython (for display in notebook environments)

A complete list with versions is available in requirements.txt.

🤝 Contributing

Contributions are welcome! Please feel free to submit a Pull Request.

1. Fork the repository
2. Create your feature branch (git checkout -b feature/amazing-feature)
3. Commit your changes (git commit -m 'Add some amazing feature')
4. Push to the branch (git push origin feature/amazing-feature)
5. Open a Pull Request

🙏 Acknowledgments

- Sentence-Transformers for the semantic embeddings model
- PyMuPDF for PDF processing capabilities
- FAISS for efficient similarity search

Made by [**GANGI JATHIN**]