

## Book-Based Q&A System with Hybrid RAG and Prompt Engineering

This project builds an intelligent question-answering system that takes in a PDF book, processes its contents, and allows users to ask questions.

### Features

- Upload and process any PDF book
- Extracts and chunks text for efficient retrieval
- Hybrid Retrieval using FAISS (Dense) + BM25 (Sparse)
- Advanced prompt templates for different question types:
  - What
  - Explain
  - Summarize
  - Types/List
  - Steps/Process
- Real-time Q&A using Hugging Face t5-base pipeline
- Gradio web interface with sharing enabled

### Technologies Used

- Model: t5-base (text2text-generation)
- Embeddings: sentence-transformers/all-MiniLM-L6-v2
- Sparse Retrieval: rank\_bm25
- Dense Retrieval: FAISS (Flat L2 Index)
- PDF Parsing: pypdf
- Web Interface: gradio
- Hugging Face Hub: huggingface\_hub

### Installation

```
pip install torch transformers gradio pypdf huggingface_hub faiss-cpu rank_bm25 sentence-transformers
```

### Hugging Face Authentication

```
from huggingface_hub import notebook_login
notebook_login()
```

### Uploading and Processing PDFs

1. Upload your PDF file.
2. The file is processed by:
  - Extracting text with pypdf
  - Chunking text into 512-character passages
  - Creating embeddings for dense retrieval (FAISS)
  - Tokenizing for BM25 ranking

### Asking Questions

The system detects the type of question and uses different prompts:

- "What is..." -> Returns a precise definition
- "Explain..." -> Returns a detailed explanation
- "Summarize..." -> Returns a concise summary
- "Types of..." / "List..." -> Returns categorized items
- "Steps..." / "Process..." -> Returns procedural steps

- Other -> Uses a general-purpose Q&A prompt

## Running the Interface

`iface.launch(share=True)`

## Example Questions

- What is Reinforcement Learning?
- Explain how backpropagation works.
- Summarize the key ideas in Chapter 2.
- List the types of neural networks.
- What are the steps in training a model?

## File Structure

`book_qa/`

■■■ <code>app.ipynb</code> / <code>main.py</code>	# Main notebook or script
■■■ <code>README.md</code>	# Documentation
■■■ <code>requirements.txt</code>	# (Optional) Dependencies

## Acknowledgements

- Hugging Face Transformers
- FAISS: Facebook AI Similarity Search
- Sentence Transformers
- Gradio

## License

This project is open-source under the MIT License.