Document QA System Project Explanation

This is a multimodal document question-answering system that allows users to upload documents (PDFs or images), ask questions about them, and receive accurate answers with supporting evidence. The system combines text processing, image analysis, and natural language generation to create a comprehensive document understanding application.
Core Components Overview

1. Document Processing (document_processor.py)

Document Parsing: Uses **PyMuPDF (fitz)** to extract text and images from PDFs
OCR: Employs **pytesseract** for extracting text from images
Text Feature Extraction: Uses **SentenceTransformer ('all-MiniLM-L6-v2')** to create embeddings from text chunks.
Image Feature Extraction: Employs **ResNet50** for visual feature extraction
Image Captioning: Implements a **ViT-GPT2 model** to generate descriptive captions for images

2. Question Processing (question_processor.py)

Question Analysis: Uses **BERT** ('bert-base-uncased') to understand questions
Dimension Adaptation: Custom neural network adapter to match BERT's 768-dim embeddings to SentenceTransformer's 384-dim space
Question Classification: Determines if the question requires text, images, or both

3. Retrieval Engine (retrieval_engine.py)

Semantic Search: Leverages **cosine similarity** to find relevant text chunks
Image Retrieval: Matches questions to image captions using embedding similarity
Hybrid Retrieval: Returns both text and image results based on relevance

4. Answer Generation (answer_generator.py)

Model: Uses **Facebook's OPT-1.3B model** (a GPT alternative)
Context Building: Intelligently builds context from most relevant text and image captions
Parameter Tuning: Uses specific generation parameters (temperature=0.3, top_p=0.9) for factual responses
Post-processing: Cleans generated answers by removing repetition

5. Audio Processing (audio_processor.py)

Speech-to-Text: Uses SpeechRecognition with Google's API for transcribing audio questions
Text-to-Speech: Employs gTTS (Google Text-to-Speech) to generate audio responses

6. Web Application (app.py and index.html)

Backend: Flask web server with REST API endpoints
Frontend: Bootstrap-based responsive UI with document upload, text and voice input
Cache: In-memory document cache (would use a database in production)

Detailed Model Breakdown

1. Text Processing Models

SentenceTransformer ('all-MiniLM-L6-v2')
Used for: Creating text embeddings for retrieval
Files: document_processor.py, question_processor.py
Feature: Creates 384-dimensional embeddings for text chunks and questions

BERT ('bert-base-uncased')
Used for: Question understanding
File: question_processor.py
Feature: Creates 768-dimensional embeddings for deep question analysis

OPT-1.3B (facebook/opt-1.3b)
Used for: Answer generation
File: answer_generator.py
Feature: Generates coherent, factual answers based on retrieved information

2. Image Processing Models

ResNet50
Used for: Image feature extraction
File: document_processor.py
Feature: Extracts visual features from images in documents

ViT-GPT2 (nlpconnect/vit-gpt2-image-captioning)
Used for: Image captioning
File: image_captioning.py
Feature: Generates descriptive captions for images to enable text-based retrieval

3. Audio Processing

SpeechRecognition (with Google's API)
Used for: Transcribing voice questions
File: audio_processor.py
Feature: Converts speech to text for processing

gTTS (Google Text-to-Speech)
Used for: Generating spoken answers
File: audio_processor.py
Feature: Converts text answers to speech for audio output

Data Flow

1. Document Upload & Processing:
User uploads PDF/image via web interface
Document is temporarily saved
Text and images are extracted
Text is chunked and embedded
Images get feature extraction and captioning
Features are stored in memory cache with unique document ID

2. Question Processing:
User types or speaks a question
Question is analyzed with BERT
Question embeddings are created
Question is classified (text/image/both)

3. Information Retrieval:
Most relevant text chunks are retrieved via embedding similarity
Relevant images are retrieved via caption similarity
Results are ranked and filtered

4. Answer Generation:
Context is built from top text chunks and image captions
OPT-1.3B generates an answer based on context
Answer is post-processed for clarity and conciseness

5. Response Delivery:
Answer is displayed with supporting evidence
Supporting text chunks are shown with page references
Image captions are displayed to support the answer
Optional: Answer is spoken via text-to-speech

Advanced Features

1. Multimodal Understanding

The system integrates both text and image understanding, allowing for questions about visual content such as charts, graphs, and photos embedded in documents.

2. Voice Interface

Supports both speech-to-text for questions and text-to-speech for answers, enabling a hands-free interaction experience.

3. Semantic Search

Rather than keyword matching, the system uses deep semantic understanding to match questions with relevant information.

4. Evidence-Based Answers

All answers include supporting evidence from the document, showing users exactly where the information came from.

5. Advanced Text Generation

Uses modern language models to generate coherent, concise, and accurate answers based on retrieved information.

6. Evaluation System

Includes a testing framework (evaluate_system.py) to assess answer quality against ground truth when available.

Technical Implementation Notes

Memory Management: Careful handling of temporary files with proper cleanup
Error Handling: Comprehensive try/except blocks with logging
Modular Design: Clear separation of concerns between components
Responsiveness: Asynchronous processing on the frontend to keep UI responsive
Efficiency: Text chunking to handle large documents efficiently
Security: Input validation and secure temporary file handling

The system showcases a complete AI application that integrates multiple machine learning models to solve a complex problem of understanding and answering questions about documents containing both text and visual information.