1. Conceptual Understanding and Learning Phase

1. **Learn About Large Language Models (LVLMs):**
    - Understand transformer architectures, attention mechanisms, and training techniques.
    - Explore widely used LVLMs (e.g., GPT-4o, Qwen2VL, LLaVA, etc.) and investigate their strengths and limitations for document QA tasks.

study training data and training process of various LVLMs

Study the basic architecture of LVLM (using playlist shared by sir)

2. **Retrieve-and-Generate (RAG) Pipelines:**
    - Study RAG architectures, which integrate retrieval and generation processes to answer questions using external knowledge sources.
    - Familiarize with RAG configurations in OpenAI, Hugging Face, and Meta's AI models, understanding the mechanics of retrieval models (e.g., dense retrievers, sparse retrievers) and their integration with LLMs.
3. **Indexing and Retrieval Techniques:**
    - Learn techniques for building scalable document indexing and retrieval systems.
    - Explore FAISS, Elasticsearch, and vector search libraries to manage and query large datasets of PDF content.
4. **Visual In-Context Learning:**
    - Study visual in-context learning, allowing models to answer questions with contextual understanding, especially for domain-specific knowledge.
    - Learn multimodal integration techniques if aiming to incorporate visual elements, like graphs or images, from educational PDFs.

---

2. Dataset Acquisition and Preparation

1. **Gather Educational Books and PDFs:**
    - Identify and download 300+ educational PDFs covering diverse topics. Ensure content spans across multiple educational levels and fields (sciences, humanities, etc.).
    - Consider using open-access sources, libraries, or APIs for educational materials (e.g., Project Gutenberg, Open Library, or datasets available through academic institutions).
2. **Data Storage and Retrieval Structure:**
    - **Vector Database:** Set up a vector database (e.g., FAISS, Weaviate, or Pinecone) for scalable embedding storage and retrieval.
    - **Embedding Generation:** Use LLMs or pre-trained sentence transformers to convert document segments into vector embeddings.

---

3. Development of the QA System

1. **Prototype a Multi-Modal Retrieval-Augmented QA Pipeline:**

Learn about multi-model rag

- Implement a basic multimodal-RAG pipeline where the system retrieves relevant passages and generates answers using a generative model.
- Integrate InternVL2 or Qwen2VL to handle retrieval and response generation.

2. **Implement Scalable Indexing & Search:**
   - Design and optimize the search pipeline to return the most relevant sections from large datasets.
   - Use vector similarity search for embedding-based retrieval and configure parameters (e.g., number of retrieved passages) for accuracy.
3. **Build an API for the System:**
   - Develop a RESTful API to handle user queries, route them through the retrieval-generation pipeline, and return responses.
   - Document the API endpoints, request structures, and response formats.

---

4. Research and Improvement Phase

1. **Conduct Literature Review and Benchmarking:**
   - Research state-of-the-art open-domain QA systems, paying attention to benchmark datasets (e.g., MMLongBench, DocMatix, M-LongDoc, MP-DocVQA, SP-DocVQA).
   - Benchmark initial prototype performance against existing models and datasets.
2. **Identify Model Limitations and Enhance Architecture:**
   - Observe failure cases where the model struggles with complex queries or specific document types.
   - Explore architectural improvements, such as knowledge distillation, specialized training on educational corpora, or hybrid models combining retrieval with graph-based reasoning.
3. **Develop Your State-of-the-Art Model:**
   - Design a custom model architecture that addresses gaps identified in the benchmarking phase.
   - Implement improvements, such as:
     - **Multimodal RAG:** Combine multimodality with the RAG pipeline.
     - **Answer generation with VLMs**: After getting the relevant documents, we will apply the LVLMs again to generate the answer.
     - **Multi-step Reasoning:** Integrate multi-step or chain-of-thought prompting to handle complex questions that require multiple reasoning steps.
     - **Fine-tuning for Educational Domains:** Fine-tuning on QA datasets relevant to education to improve model accuracy in this domain.

---

5. Testing, Evaluation, and Iteration

1. **Quality and Performance Testing:**
   - Test the system across different educational subjects and difficulty levels.

    - Implement qualitative and quantitative metrics (accuracy, relevance, response time) to evaluate QA performance.
2. **Refine Model and API Based on Feedback:**
    - Use feedback from users and SMEs (subject matter experts) to iteratively improve the model's retrieval accuracy, comprehension, and response clarity.
3. **Continuous Research and Publication:**
    - Document findings, challenges, and breakthroughs. Publish research in conferences or journals to share new insights with the AI community and potentially establish the solution as a state-of-the-art model.

Summary of Tasks

- **Learning Phase:** LVLMs, RAGs, multimodal learning, retrieval techniques.
- **Data Preparation:** Dataset acquisition, text extraction, segmentation, and vector embedding.
- **Development Phase:** Basic and MultiModal RAG pipeline, scalable indexing, API setup.
- **Research and Improvement:** Literature survey, model refinement, new architecture development.
- **Testing and Iteration:** Quality assurance, model fine-tuning, and iterative updates based on feedback.

By following these tasks, you'll create a comprehensive roadmap for developing a cutting-edge Open Domain Educational Document QA system with potential applications in self-learning, education platforms, and more.