Two techniques for optimizing the RAG model developed in Task 1.

1. Dynamic Query Expansion for Better Document Retrieval

Problem: The quality of the retrieved documents plays a crucial role in generating accurate answers in the RAG model. If the query is too short or lacks specificity, relevant documents might not be retrieved, leading to degrade of the quality of the final response generated by the AI model.

Solution: Implement dynamic query expansion, where the query is enhanced with additional terms or keywords before performing the document retrieval step. This can be done by:

- Using synonyms or related terms based on the context of the query.
- Expanding the query with keywords from initial GPT predictions.
- Extracting important terms from the initial retrieval results to re-query Pinecone for more relevant documents.
-

Steps:
Step 1: Embed and retrieve documents with the initial user query.
Step 2: Analyze the retrieved documents to identify important terms.
Step 3: Expand the query using the newly identified terms.
Step 4: Perform a second document retrieval step with the expanded query to refine the results.
Example: If a user asks: "My head hurts, I have high temperature and have red eyes, what disease do I have?" The query can be expanded to include terms like “headache”, "fever" or "sick" to retrieve more targeted and relevant documents.

Benefits:
Improved Retrieval Accuracy: More comprehensive retrieval of relevant documents.
Enhanced Context: AI model will have access to richer context, leading to better answers.
Implementation Insight: The dynamic expansion can be automated by analyzing keyword importance using term frequency-inverse document frequency (TF-IDF) scores, word embedding, or context-aware techniques like BERT.

2. Hierarchical Document Retrieval for Multi-Stage RAG

Problem: Retrieving documents in a single stage may miss highly relevant information if the initial search isn’t precise or if the corpus is large. It can also result in too many irrelevant documents, increasing noise in the context provided to GPT for answer generation.

Solution: Use hierarchical document retrieval, a multi-stage process where documents are filtered through increasingly specific criteria. This hierarchical approach ensures only the most relevant documents are used in the final context for generation.

Steps:
Stage 1: Perform an initial, broad document retrieval using a lower threshold for similarity (fewer constraints).
Stage 2: Re-embed the top retrieved documents or perform a more targeted similarity search using additional constraints or fine-tuned embeddings (like a domain-specific model).
Stage 3: Use only the top results from this second stage to form the final context for GPT.

Example: If the user asks, "What are the latest advancements in AI for healthcare?", the system might:

Stage 1: Retrieve all documents related to AI and healthcare broadly.
Stage 2: From this set, perform a more focused retrieval looking for documents specifically mentioning "latest advancements", "2024 innovations" or “current research”

Benefits:
Reduced Noise: Removes less relevant documents that might dilute the context for GPT.
Focused Answer Generation: GPT focuses on the most specific and relevant data, increasing answer precision.
Implementation Insight: The second stage can use a fine-tuned domain-specific model for embedding (e.g., a healthcare-specific BERT model) or apply custom filtering logic based on metadata such as publication date, source credibility, etc.