Two innovative techniques for optimising the RAG model developed in Task 1 -

1. Dynamic Context Selection

Problem: Different queries demand different amounts of context. Simple queries may only need a few documents, while more complex ones require additional context to provide a detailed response.

Solution: Adjust the number of documents retrieved dynamically based on the complexity of the query. For example:

- For straightforward queries like "What is the refund policy?" retrieve a smaller number of relevant documents (e.g., `top_k=3`).
- For more complex or multi-part queries, increase the number of retrieved documents (e.g., `top_k=5` or `top_k=10`).

Implementation Strategy: To determine how complex a query is, use a heuristic or scoring system that evaluates factors such as:

- **Query length:** Longer queries might indicate a need for more context.
- **Keywords:** Queries containing terms like "history" or "background" suggest that more documents are needed for a comprehensive answer.
- **Query type:** If the query asks for detailed information, a list, or explanation, more documents should be retrieved.

2. Hybrid Retrieval Mechanism

Problem: Dense vector retrieval methods like Pinecone excel at capturing the semantic meaning of a query, but they can struggle with rare or out-of-vocabulary terms. On the other hand, sparse retrieval methods (such as BM25) are great at handling rare terms but lack the ability to understand semantic similarity in the same way.

Solution: To address this, combine **dense vector retrieval** (using Pinecone) with **sparse retrieval** (using a technique like BM25 from Elasticsearch). This hybrid approach allows us to leverage the strengths of both methods—semantic understanding from dense retrieval and precise keyword matching from sparse retrieval.

Implementation Strategy:

1. Start by querying the sparse retrieval system (e.g., Elasticsearch with BM25) to find documents based on keyword matching.
2. Then, use Pinecone to retrieve documents that are semantically relevant to the query.

3. Combine the results from both systems, and prioritize documents by scoring. We can blend BM25 scores with cosine similarity from Pinecone to rank documents according to both their keyword relevance and semantic meaning.

This approach ensures that the retrieval process accounts for both rare terms and semantic context, improving the overall quality and accuracy of the retrieved documents.