1. Adaptive context choice through reinforcement learning

Overview**:** In conventional RAG models, the context for feedback typically involves a collection of documents that have been received. However, by incorporating reinforcement learning (RL) into the context selection process, you can enhance the ability to choose the most suitable document based on user interaction.

How it functions:

- Reward Mechanism: Establish a reward mechanism that assesses the quality of the model's responses, based on feedback from users. Positive feedback (such as the number of users or engagement levels) can reinforce the selection of certain documents as references.
- Training of the Agent: To create an RL agent capable of selecting the optimal documents from the database. This agent is rewarded for how closely its generated responses align with the user's satisfaction.
- Exploration vs Exploitation: The RL agent is designed to explore documents (exploitation) and experiment with new combinations to enhance future responses (exploration). This approach allows the model to evolve and adapt to shifts in user preferences and questions. Interest:
- Personalized Learning: The system becomes increasingly tailored over time, as it identifies documents that respond most effectively to specific user traits or types of queries.
- Enhanced Responsiveness: Regularly updating pertinent information based on user feedback boosts responsiveness, which in turn increases user satisfaction.

2. Mixed Embedding Strategies for Better Document Search

Introduction: Rather than depending on just one type of OpenAI embeddings for searching documents, mix different types of embeddings or models (like semantic embeddings and embeddings based on keywords) to develop a mixed embedding strategy. This method boosts the accuracy of searching and enables the system to grasp more complex queries.

How It Functions:

- Diverse Embedding Sources: Employ various embedding models, such as:
- Semantic Models: Models like text-embedding-ada-002 that capture the essence of the text.
- Keyword Models: Conventional TF-IDF or BM25 models that highlight key words in documents.
- Weighted Merging: When a search query is issued, generate embeddings from both semantic and keyword-based models. Merge these embeddings with weights to represent the query and the documents. These weights can be tweaked according to the context (for instance, give more weight to semantic relevance in open-ended questions and more to keyword relevance in specific queries).
- Multi-Vector Indexing: Keep both kinds of embeddings in the vector database (Pinecone) and use them together during the search process.

Advantages:

- Enhanced Search Accuracy: By incorporating both semantic and keyword data, the system can more accurately match queries to relevant documents, thereby improving both recall and precision.
- Versatility in Query Handling: The hybrid strategy enables the system to handle a wide range of query types, making it effective in understanding both complex and simple queries.