20240530Meeting

312707003 黃鈺婷

Retrieval-Augmented Generation(RAG)

- 檢索式模型+生成式模型
- 利用外部知識來增強生成的文本，使生成的文本更加準確
- 不需要重新訓練模型
- 適用於有大量資料，但多數資料未分類或標記
- 開放領域的問答任務、資訊檢索

RAG vs. Fine-tuning

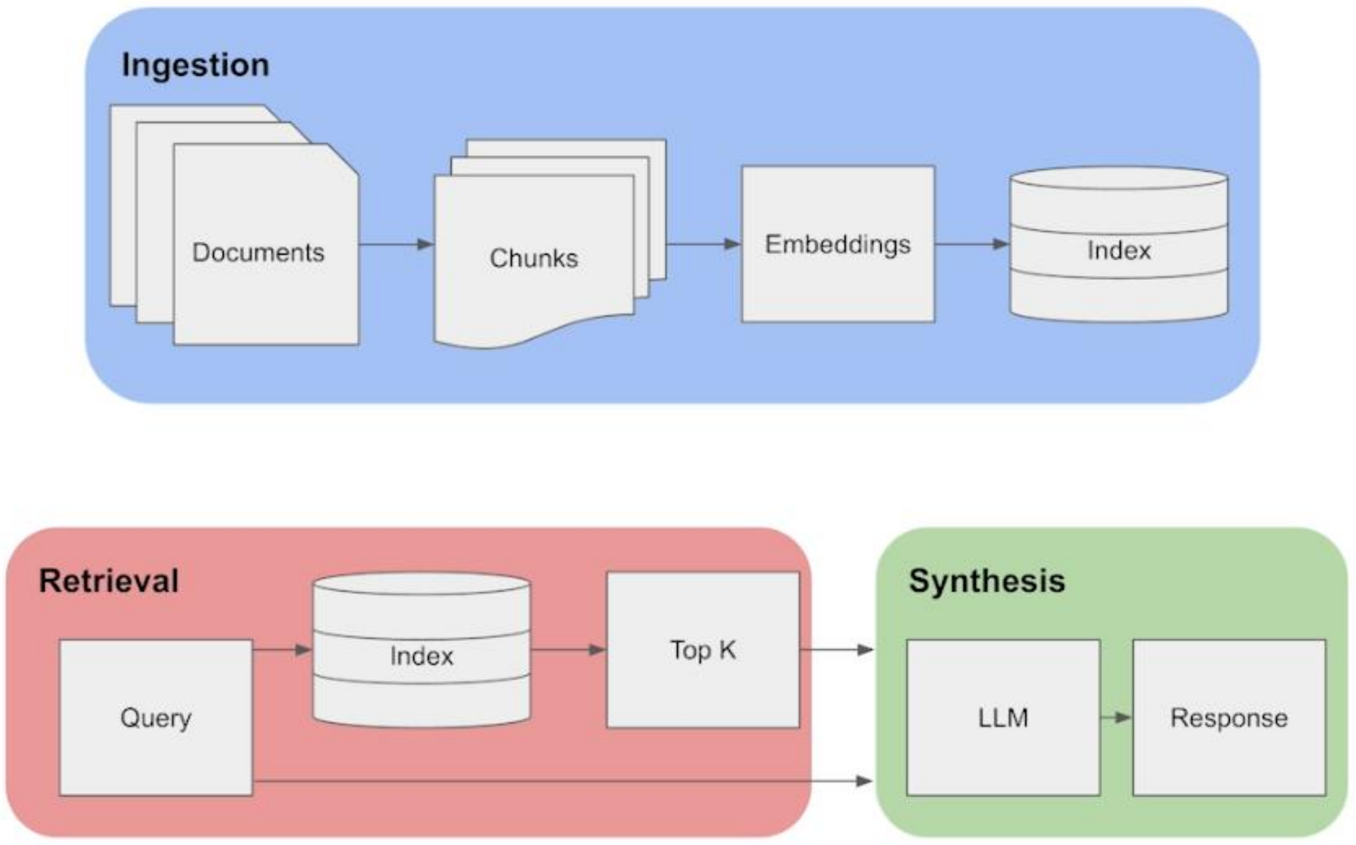
RAG	Fine tuning
直接更新檢索知識庫	需要重新訓練
需要少的數據處理	有限的資料集無法有顯著效能的提升
答案可以追溯到特定的資料來源	像黑盒子，無法知道模型為何以某種方式做出反應
無法自訂模型的行為或風格	允許根據特定語氣或術語調整模型的風格

Neural Retrieval

- Encode the query and documents into dense vector representations
  → compute cosine similarity
  → determine which documents are most relevant to a query
- Advantage:
  adept at dealing with long and complex queries
- Challenge:
  performance depends on the data they are trained on

Process of RAG

1. Vector Database Creation
2. User Input
3. Information Retrieval
4. Combining Data
5. Generating Text



Embedding

- 高維離散的特徵映射到相對低維的連續向量空間中的表示方式
- 將原始數據轉換成一種特別的數據格式，以便 AI 或機器學習演算法能夠處理這些數據，並加入了距離的概念
- Sparse embedding
  - TF-IDF
  - lexical matching the prompt with the documents
- Semantic embedding
  - BERT
    - 擷取document和query中上下文的細微差別
  - SentenceBERT

Sentence Embedding vs Token-Level Embedding

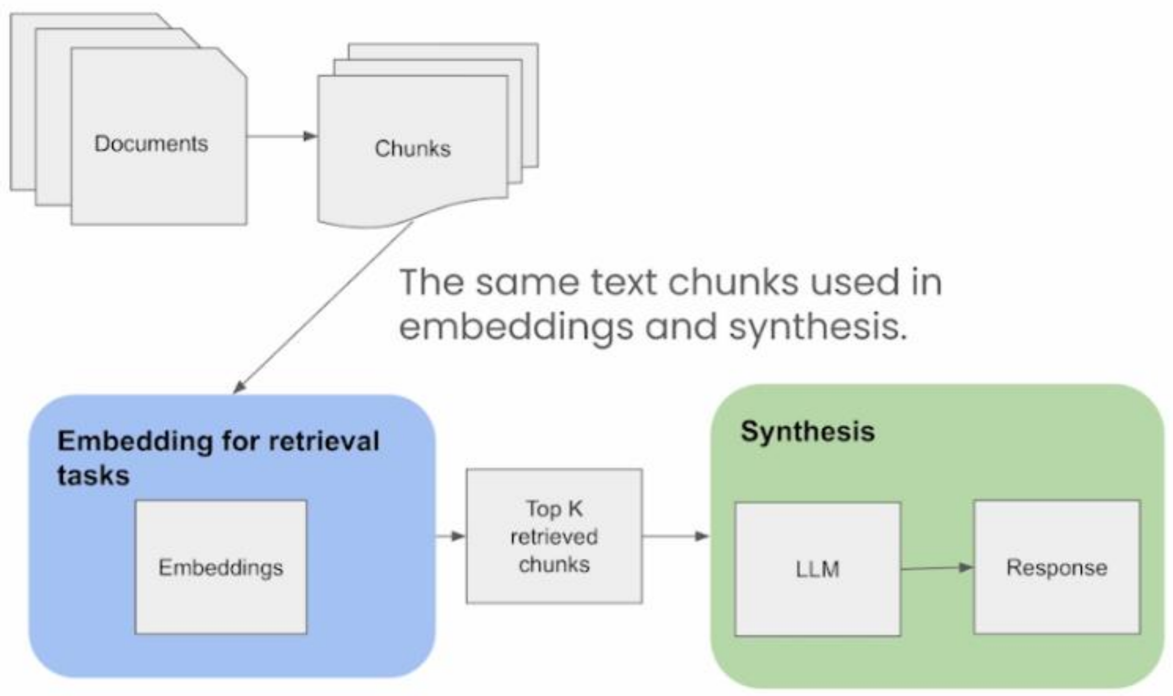
- Modification of the traditional BERT model
- Are trained specifically to understand the meaning of entire sentences
- Generate embeddings where sentences with similar meanings are close in the embedding space
- Provide a single embedding for the entire sentence
- Are more suited for tasks that rely on sentence-level understanding (like semantic search, sentence similarity)

Retrieval

- Standard/Naive Approach
- Sentence-Window Retrieval
- Auto-merging Retriever

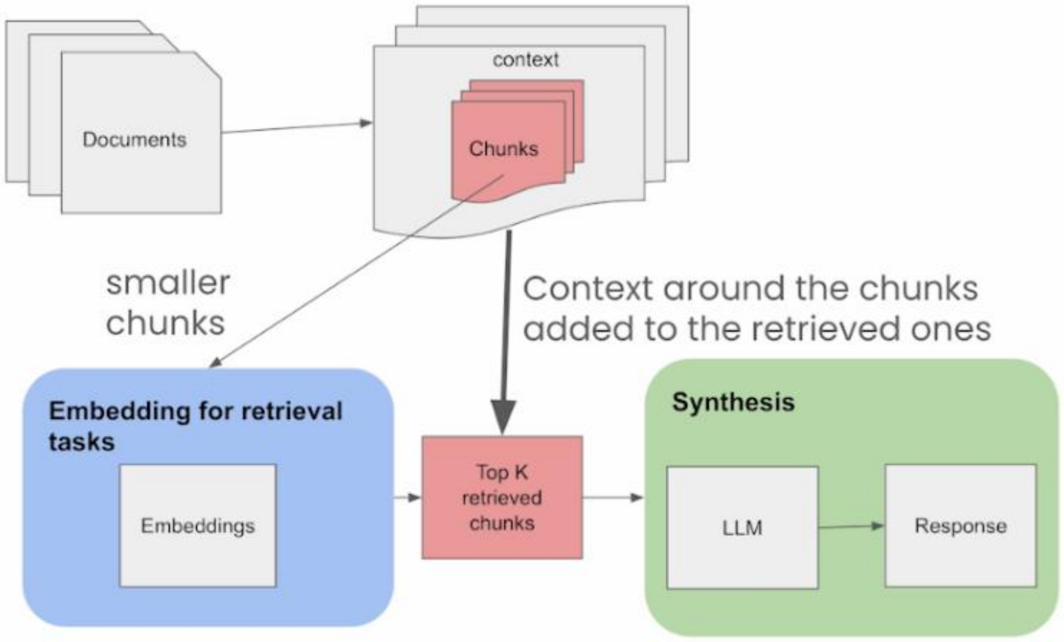
Standard/Naive Approach

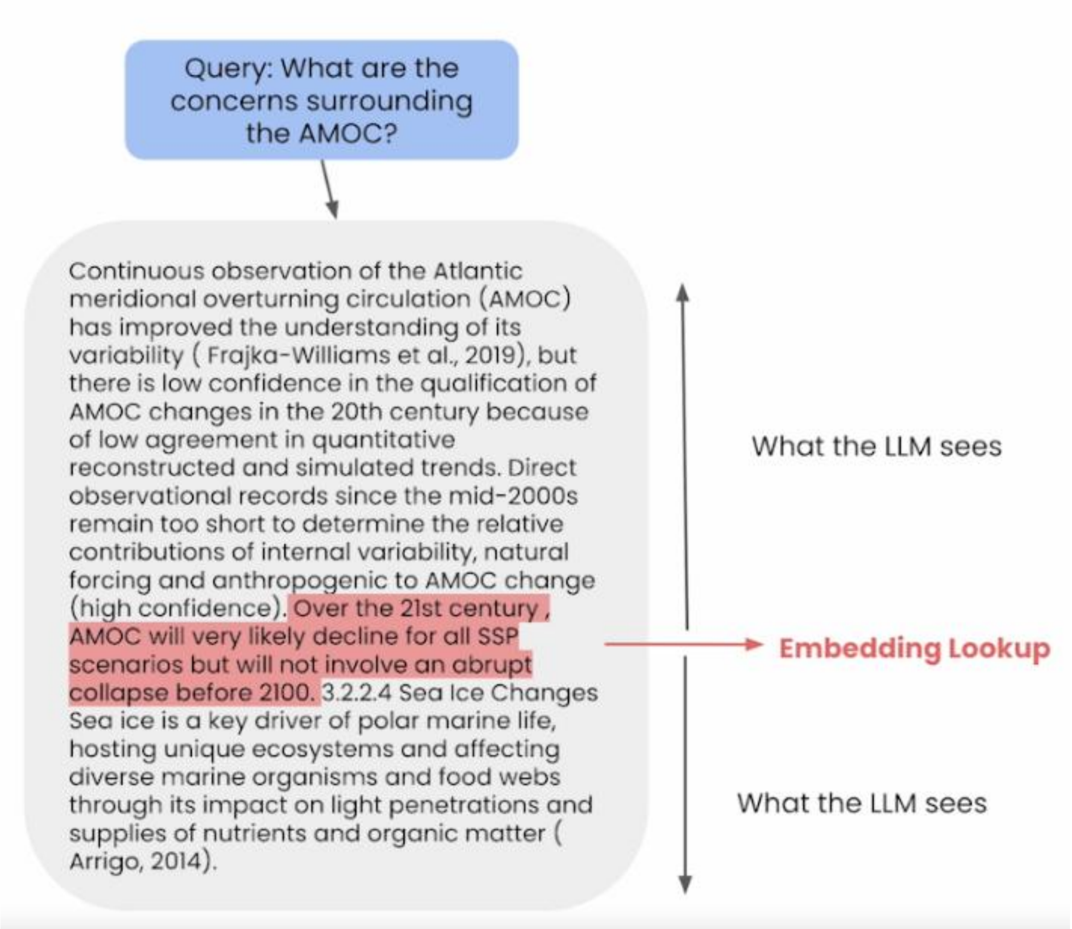
- Using the same text chunk for both embedding and synthesis, simplifying the retrieval process.
- Maintains consistency in the data used across both retrieval and synthesis phases



Sentence-Window Retrieval

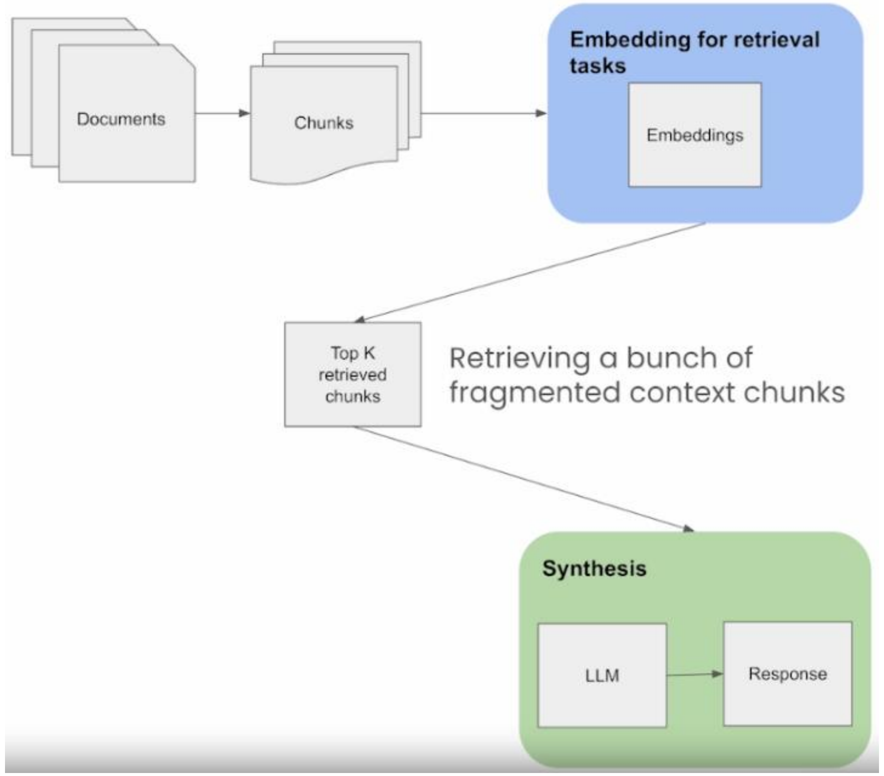
- Breaks down documents into smaller units, such as sentences or small groups of sentences





Auto-merging Retriever

- Aims to combine information from multiple sources or segments of text to create a more comprehensive response to a query



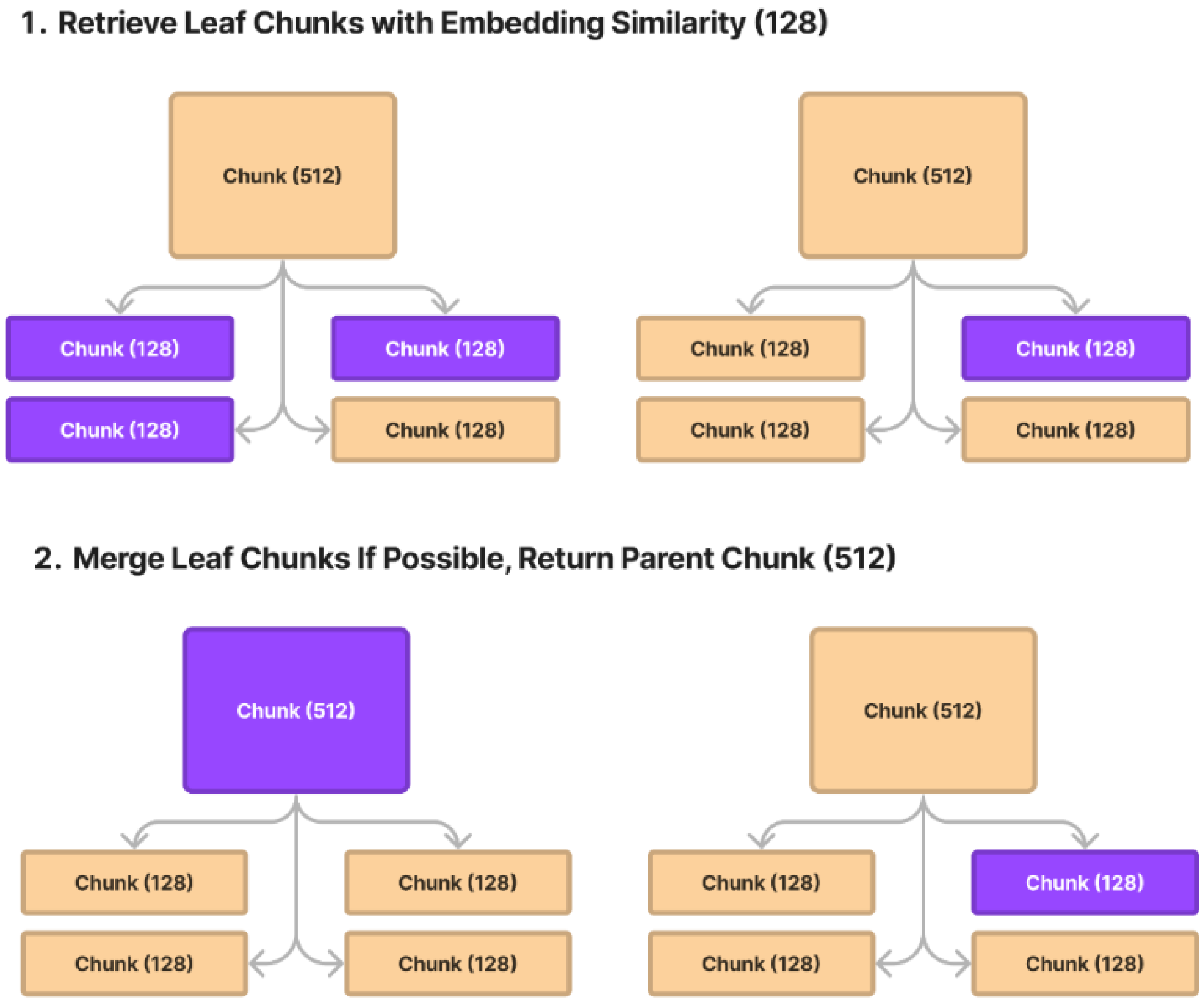
Auto-merging Retriever

- 定義切割的層次結構為[2048, 512,128]
- 頂層chunk size=2048
- 中間層chunk size=512
- 底層的子節點chunk size=128
- 檢索時只拿底層的子節點和問題進行匹配，當某個父節點下的多數子節點都與問題匹配，則將父節點作為context返還給LLM

```
from llama_index.node_parser import HierarchicalNodeParser

node_parser = HierarchicalNodeParser.from_defaults(
    chunk_sizes=[2048, 512, 128]
)
```

Auto-merging Retriever



Response Generation / Synthesis

- 模型將檢索到的信息和預訓練的知識相結合，產生連貫且上下文相關的回覆
- 此過程涉及整合從各種來源收集的見解，確保準確性和相關性
- 在input sequence的開頭或結尾策略性地放置重要資訊可以增強RAG的有效性，從而提高RAG的效率

Retrieval Metrics

- Context Relevance- measures the alignment of retrieved information with the user’s query
- Context Recall- evaluates how well the system retrieves all relevant parts of the context
- Context Precision- focuses on the system’s ability to rank relevant items higher
- These metrics ensure the effectiveness of the retrieval system in providing the most relevant and complete context for generating accurate responses.

Generation Metrics

- Faithfulness- focuses on the factual accuracy of the answer
- Answer Relevance- assesses how well the answer addresses the original question
- These metrics ensure the generation component produces contextually appropriate and semantically relevant answers.