Hybrid Search Strategies

10 recorder

RAG pipeline

2) Re-Ranking Techniques

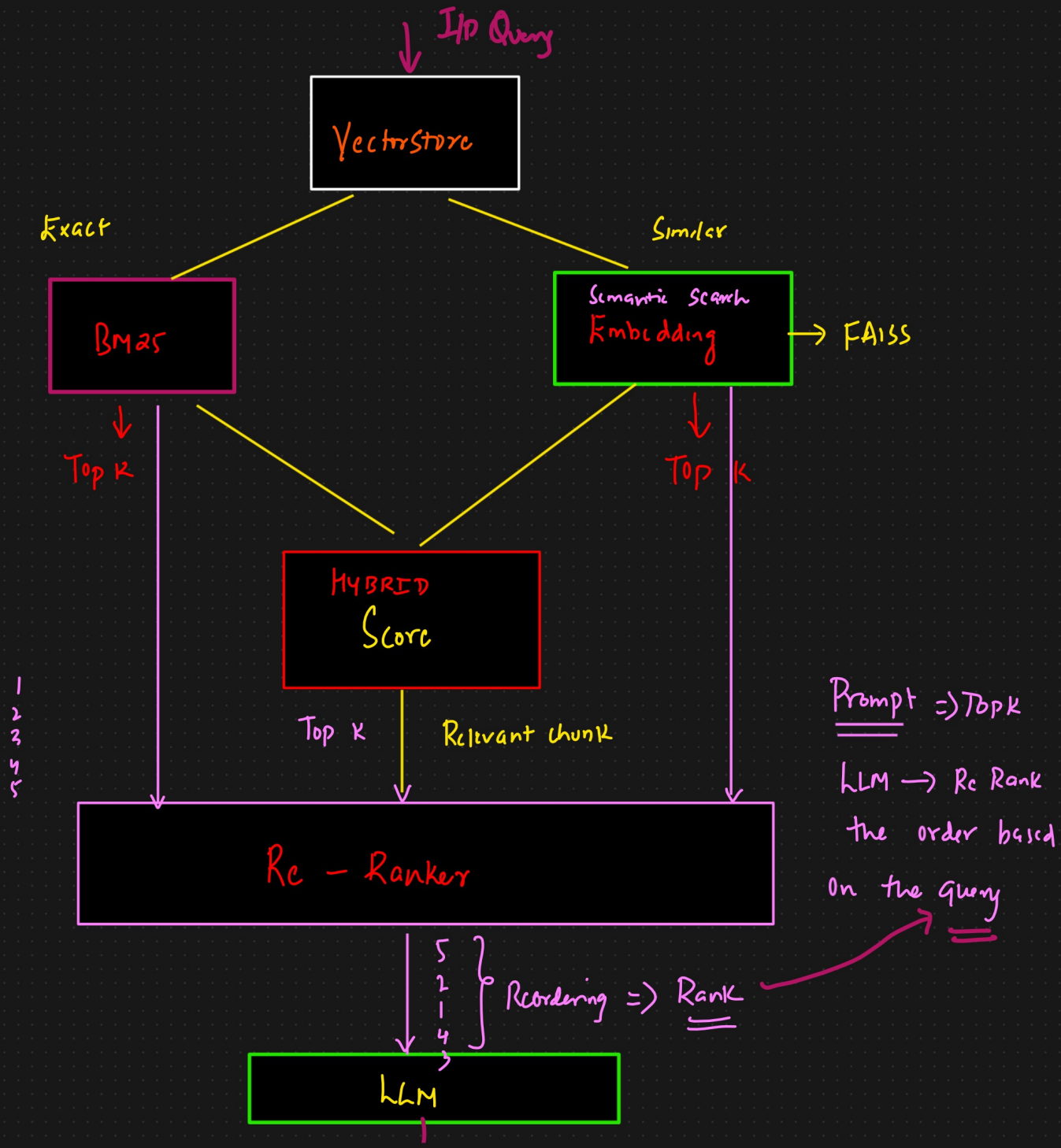
1)
2)
3)

Re-ranking is a second-stage filtering process in retrieval systems, especially in RAG pipelines, where we:

1. First use a fast retriever (like BM25, FAISS, hybrid) to fetch top-k documents quickly.

2. Then use a more accurate but slower model (like a cross-encoder or LLM) to re-score and reorder those documents by relevance to the query.

👉 It ensures that the most relevant documents appear at the top, improving the final answer from the LLM.



Stages

1) Retrieval
2) Re-Ranking
3) Generation

↓

O/p => Accurate

Practical

📊 Why Use Rerankers in a RAG Pipeline

🔍 Reason	💡 Without Reranker	✅ With Reranker
1. Relevance of Context	Top-k documents may be loosely or partially related	Top-k documents are re-scored and reordered for maximum relevance
2. Factual Accuracy	LLM may hallucinate if poor context is retrieved	Irrelevant docs are filtered out → grounded, factual answers
3. Handling Ambiguity	Retriever lacks deep understanding of query intent	Reranker evaluates full query–doc pair → better intent alignment
4. Semantic Matching	Dense retrievers may miss low-similarity but relevant docs	Reranker uses deeper models (cross-encoders / LLMs)
5. Keyword vs Meaning	BM25 may favor exact match even if not meaningful	Reranker balances semantic and lexical relevance
6. Prioritization of Evidence	All retrieved docs treated equally	Most relevant documents float to the top
7. Long-Tail Queries	Weak retrievers struggle with uncommon queries	Rerankers better capture rare but meaningful matches
8. LLM Efficiency	Poor context leads to verbose or incorrect answers	High-quality context improves precision and conciseness
9. Noise Reduction	Noisy docs (ads, unrelated text) may slip in	Reranker pushes noisy docs to the bottom or filters them out
10. Flexible Scoring Strategies	Fixed retriever scoring	Reranker can include metadata, recency, user preferences