Basics of RAG

Ben Clavie (Answer.AI)

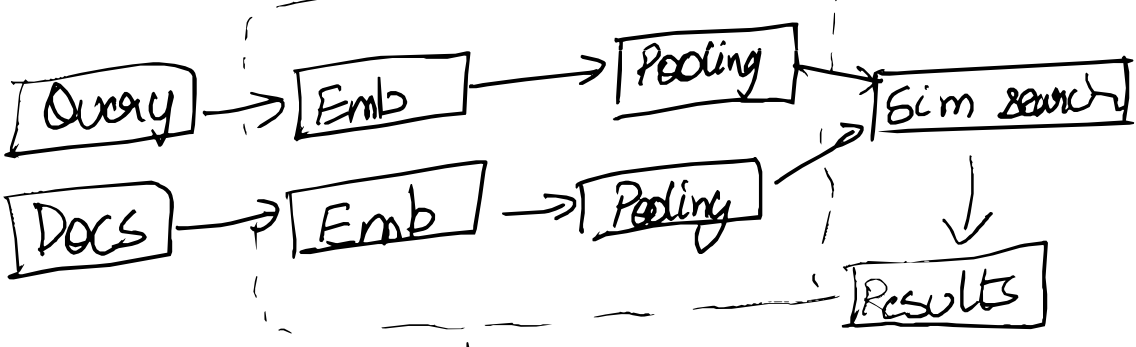
RAGatouille → Lib to use ColBERT easily

Rerankers → Lib for rerankers

MVP → Bi-encoder single vec emb & cosine similarity are all you need.

Metadata filtering → Important

Good RAG:
- → Good retrieval pipeline
- → Good generative model
- → Good way to link them up.



'Bi-encoder' approach

Why 'bi-encoders'

→ Generally used to create 'single vector' representations. They precompute doc representations

→ Docs & queries are computed seperately Not aware of each other

→ Efficient but has some drawbacks i.e retrieval perf tradeoffs

Reranking: The power of cross encoders

→ Encode both query & docs at the same time

→ Not computationally realistic to compute query aware document representations for every single document pair.

→ Its effectively a binary classifier. The prob of positive class represents the relevance of doc to query, & is taken as the similarity score

→ Reranking :- Use a expensive model to score only a subset of your retrieved documents

→ Use Keyword search
- → Embeddings represent info that is useful to their training queries
- → It'll never be fully representative of your queries
- → To capture signals, use keyword search.

→ BM25 is too strong a baseline

→ Powerful when you have lot of domain-specific jargon.

Metadata filtering
- → Remove docs that are not relevant for query.
- → Use GLiNER to extract entities from your text & use in filtering