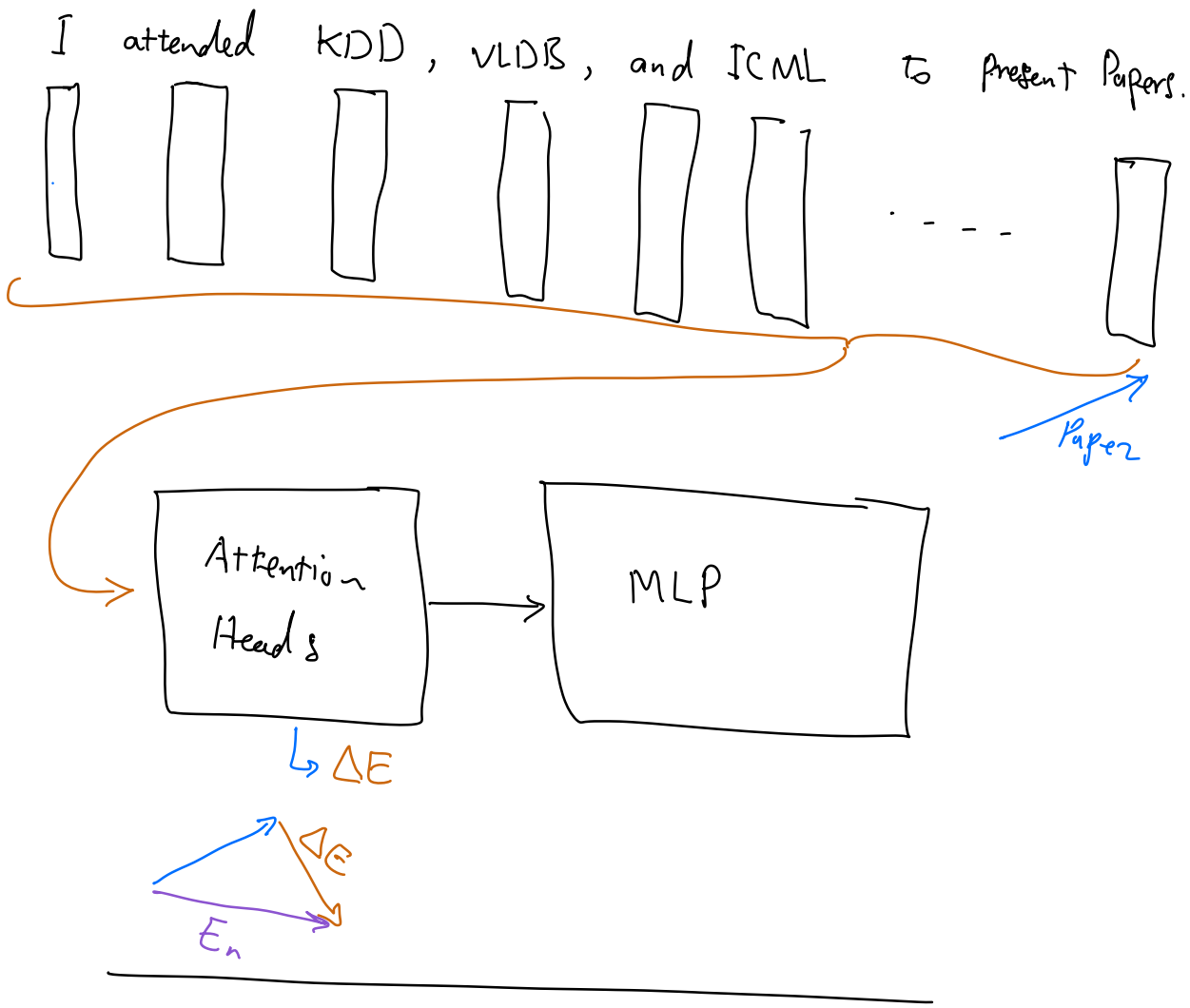


Databases
↓
Non-Parametrized
Retrieval-Only Systems

- Always returns facts ✓
- Does not use the Collective Information in the world

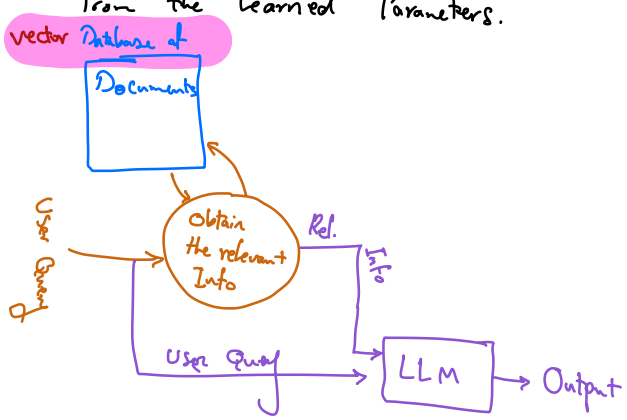
Retrieval-Augmented Generation (RAG)

- RAG -

LLMs
↓
Data Generation only based on the Learned Parameters
Parametrized

- May return wrong Info.
- Generated based on Rich parameters learned by the model ✓

RAG: Seperates the factual Statements from the learned Parameters.



Challenges:

- Retrieval Quality
  - High Precision: The retrived documents are highly relevant to the Query
    + Fine-Tune Embeddings based on the Context.
    + Use an Ensemble of Fine-Tuned Embedders
  - High Recall: The relevant documents are not missed
    + Optimize the Chunking
    + Increase the value of K
      - → will lead to lower Precision
      - → A lot of noise (irrelevant Documents) will be retrieved

Challenge 2:

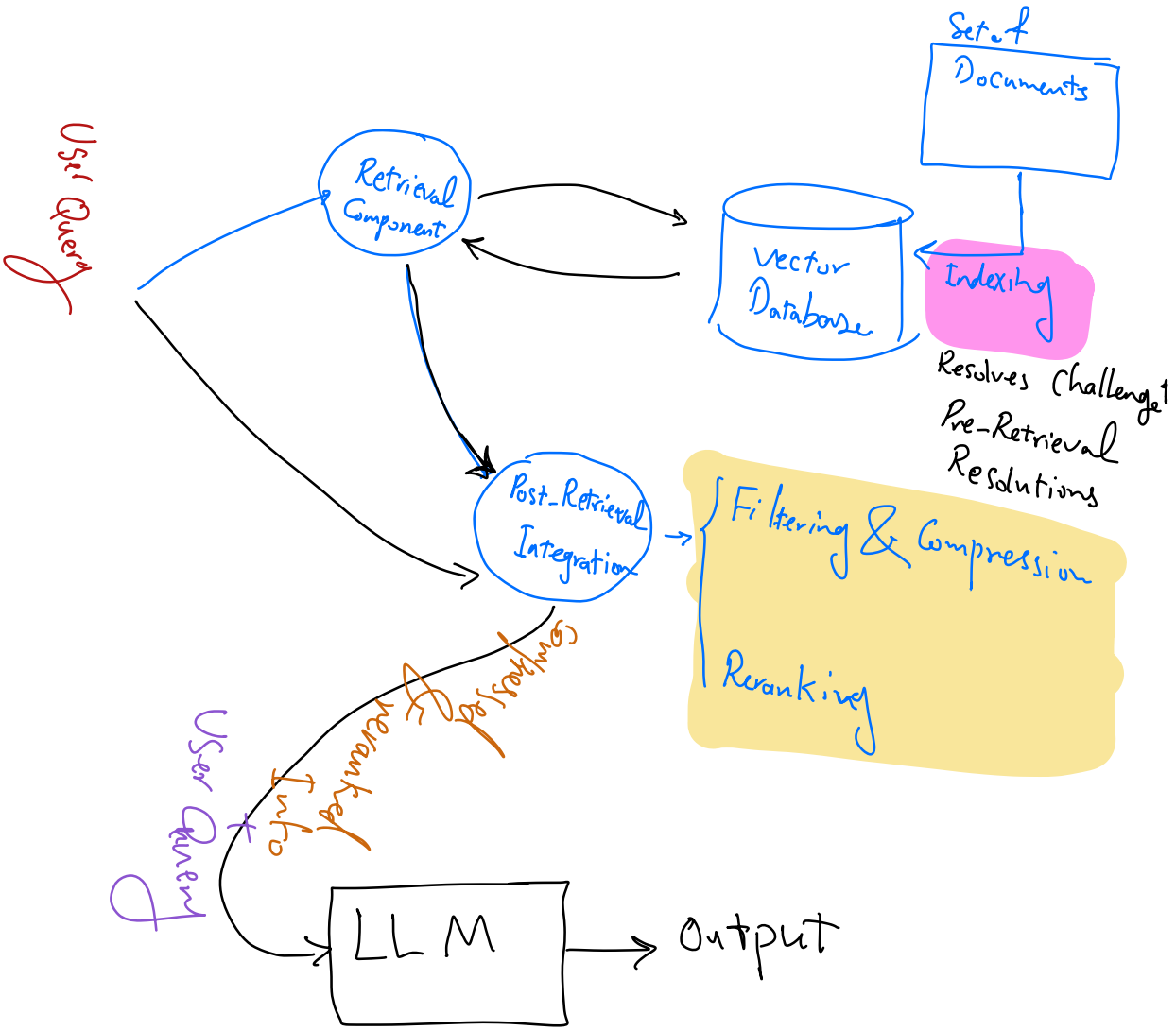
Integrating the retrieved documents Into the Prompt.

- The noise in the retrieved documents Can mislead the LLM
- Limited Context window Size
- Lost In The Middle issue

Challenge 3:

The LLM may still put a high weight on the Learned Parameters (MLP)

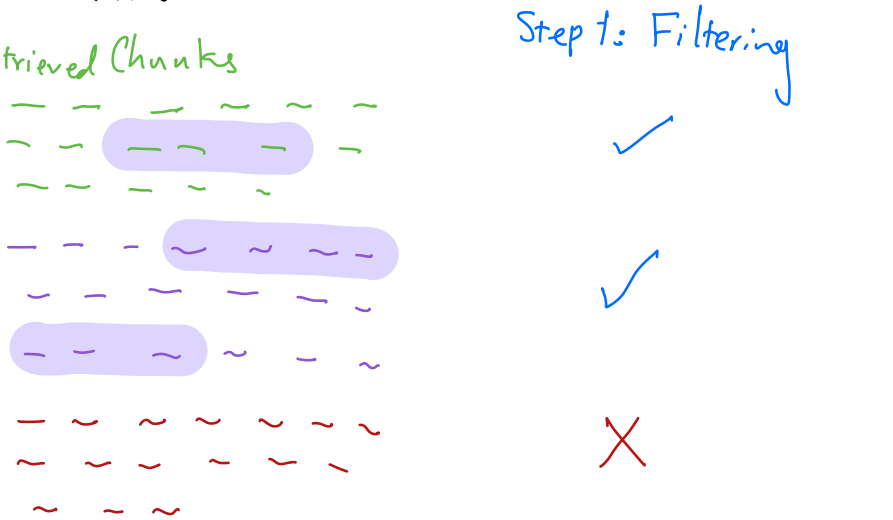
+ Retrain / Fine-Tune the LLM for RAG


Set of Documents
User Query
Retrieval Component
Vector Database
Indexing
Resolves Challenges
Pre-Retrieval Resolutions
Post-Retrieval Integration
Filtering & Compression
Reranking
User Query + Retrieved Info + Instructions
LLM
Output

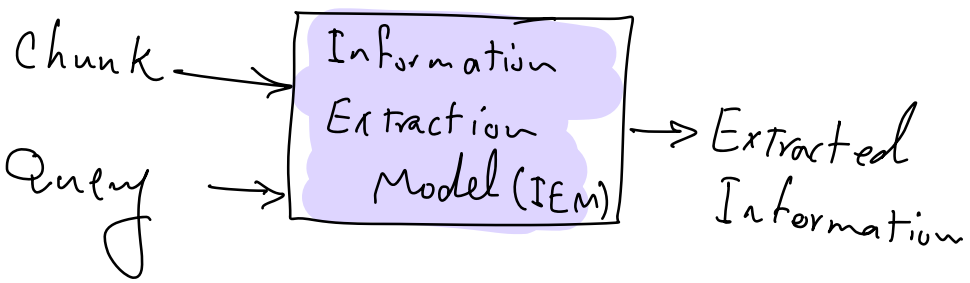
Filtering & Compression:

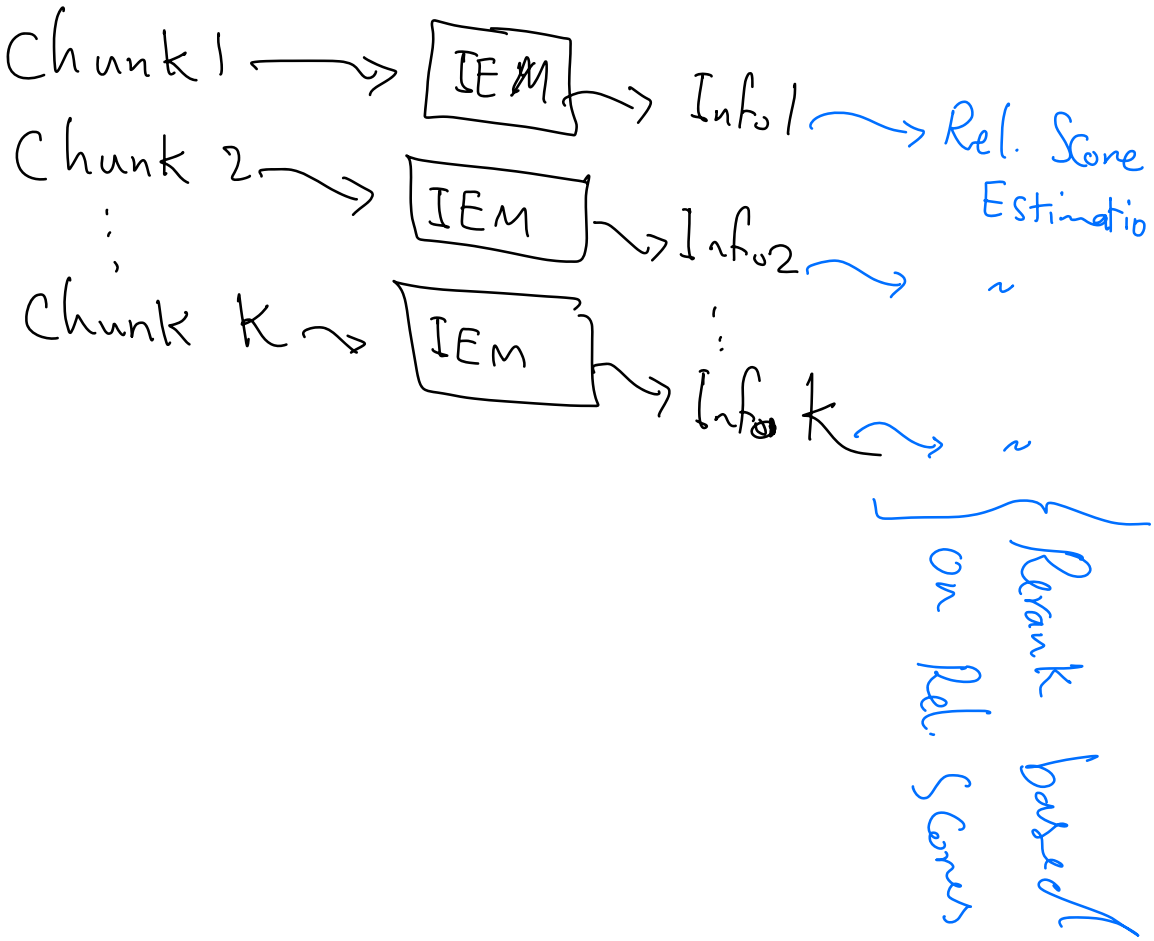
to take the retrived Sequence of Documents & remove the noise & extract the relevant Information

e.g.,



Step 2: Extract Relevant Parts of each Chunk




Chunk 1
IEM
Info 1
Chunk 2
IEM
Info 2
Chunk K
IEM
Info K
Rel. Score Estimatio
Rerank based on Rel. Scores