GRADIO AND ADVANCED RAG TECHNIQUES -01

GRADIO

Gradio is an open-source Python library that lets you quickly build UIs (web apps) for your ML or LLM models, right inside Colab or Jupyter.

basically streamlit but for gc

DENSE RAG

Da one we did yest

Retrieval by meaning (semantic similarity)

Uses embeddings — text → vectors → cosine similarity

Pros

Understands paraphrases

Robust to vocabulary differences

Perfect for natural language queries

Cons

Needs embedding model

BM25 RAG

Retrieval by exact keyword overlap
Based on TF-IDF and bag-of-words (classic search engine logic)

Pros

- Great for proper nouns, numbers, IDs
- No embeddings needed
-

Cons

- Fails for synonyms or paraphrases
- Can miss meaning-based queries

BM25 RAG

Document Preprocessing and Indexing

1. Document Chunking: The knowledge base documents are preprocessed and split into manageable chunks to create a searchable corpus.
2. Tokenization and Indexing: Each chunk is tokenized, and an inverted index is created. The BM25 algorithm calculates term frequencies and inverse document frequencies.

BM25 RAG

BM25 Retrieval-Augmented Generation Workflow

- Query Input: A user provides a query that needs to be answered.
- Retrieval Step: The query is tokenized, and relevant documents are retrieved using the BM25 scoring algorithm. This step considers term frequency, inverse document frequency, and document length to find the most relevant chunks.
- Generation Step: The retrieved document chunks are passed to a large language model as additional context. The model uses this context to generate a more accurate and relevant response.

WHAT IS TF-IDF?

TERM FREQUENCY INVERSE DOCUMENT FREQUENCY

WHEN IS IT BETTER?

Domain-specific jargon
Queries use technical, rare, or domain-specific words (e.g., “attendance condonation policy 2024”)
Embedding models may not know rare/new terms; BM25 still matches exact tokens
Noisy datasets
Sparse retrieval tends to outperform dense models on small corpora since embeddings need lots of training data to generalize

HYBRID RAG

Best of both worlds

Combine Dense + BM25 scores

Retrieve results by combining semantic + keyword relevance

Pros

- Handles both “meaning” and “exact match”
- Used in real-world production RAG systems

Example

Query: “Penalty for late returns ₹5?”

→ Dense RAG finds “late fee per day”

→ BM25 ensures “₹5” also appears

→ Combined score ranks correctly