publica academy

ADVANCED RETRİEVAL AUGMENTED GENERATİON (RAG) & EVALUATİON

A practical course for intending and professional AI Developers

Yusuf Anigilaje
AI Developer, Publica AI

publica academy

Course Objectives

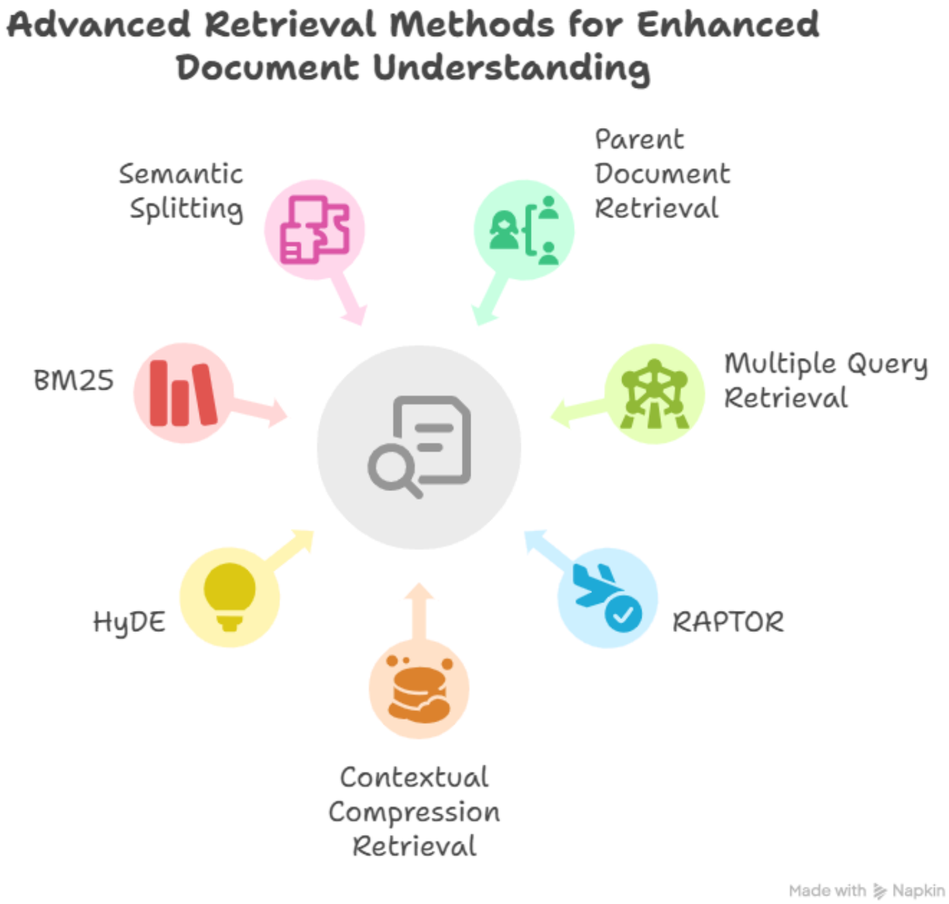
By the end of this section, you will be able to:

- Understand the limitations of basic retrievers in RAG systems
- Identify and explain at least 7 advanced retriever techniques used in modern RAG pipelines
- Compare the strengths and weaknesses of each technique
- Select the appropriate retriever(s) based on use case, data type, and complexity
- Apply advanced retriever techniques to build more accurate and efficient RAG systems

Why Go Beyond Basic Retrieval?

Limitations of basic retrievers:

- **Retrieve too much or too little**
- **Miss key context**
- **Struggle with complex queries**
- **Waste tokens in the LLM**



Advanced retrievers solve these by:

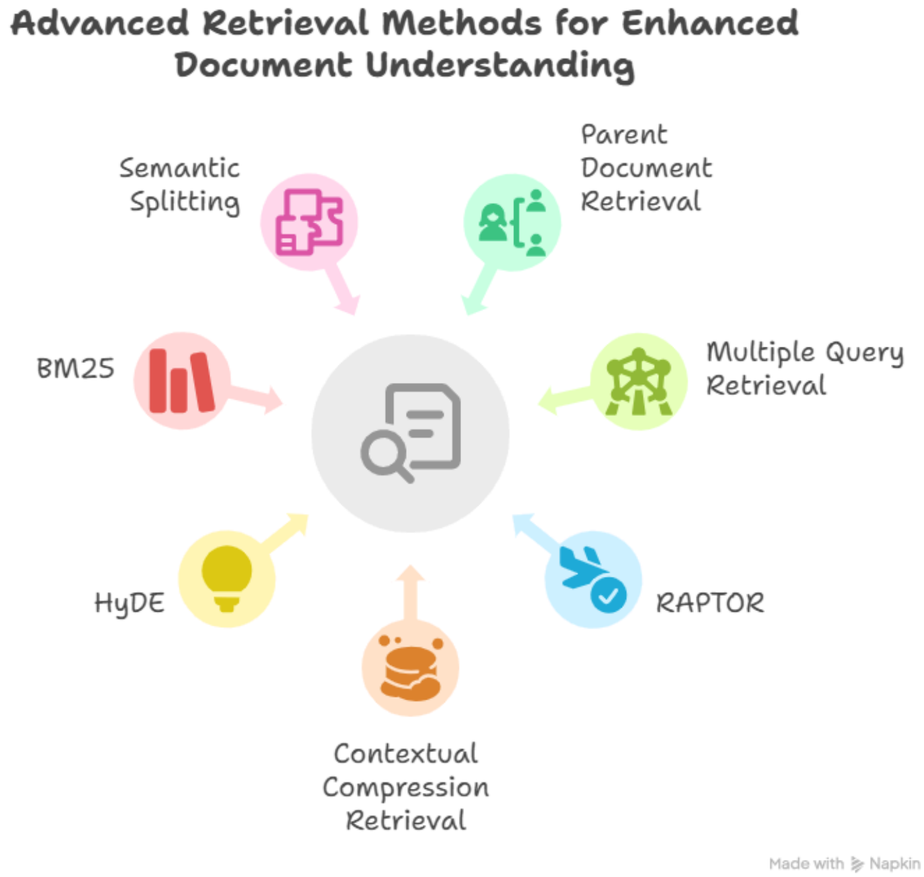
✅ **Improving relevance**
✅ **Compressing context**
✅ **Supporting deeper reasoning**
✅ **Expanding query interpretation**

publica academy

Here are some Advance Retriever techniques:

1

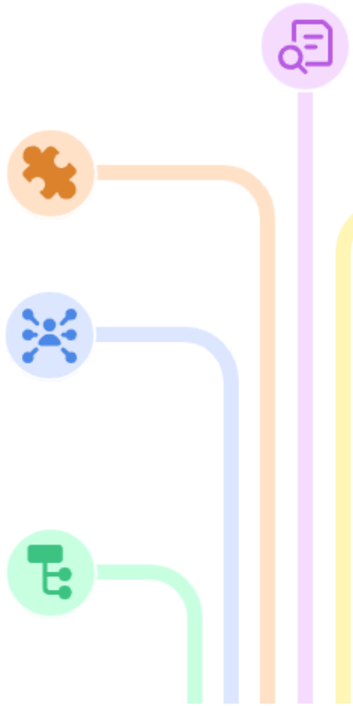
- Parent Document Retriever
- Multiple Query Retriever
- Raptor
- Contextual Compression Retrieve
- HyDE (Hypothetical Document Embeddings)
- BM25
- Semantic Splitting



publica academy

Contextual Compression Retrieval

Compresses documents while preserving context for efficient storage and retrieval.



RAPTOR

Orders passages for better coherence and relevance.

Multiple Query Retrieval

Uses multiple queries to cover diverse aspects of the information need.

Parent Document Retrieval

Retrieves entire documents to provide comprehensive context.

HyDE

Generates hypothetical documents to enhance semantic understanding.

BM25

Ranks documents based on term frequency and inverse document frequency.

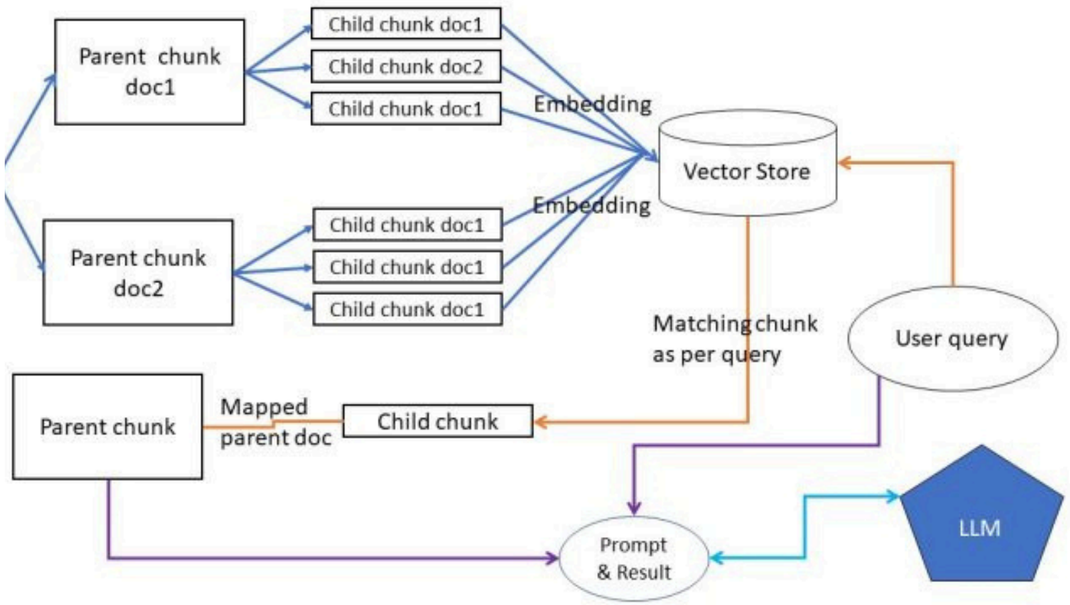
Semantic Splitting

Splits documents into semantic units for precise retrieval.

Parent Document Retriever

- **Many RAG systems split documents into small chunks for retrieval. Small chunks may miss important context needed to answer questions.**

- **The Parent Document Retriever embeds smaller chunks for efficient search but retrieves the full parent document for the LLM. Combines the benefits of concise retrieval with access to complete context for accurate generation.**



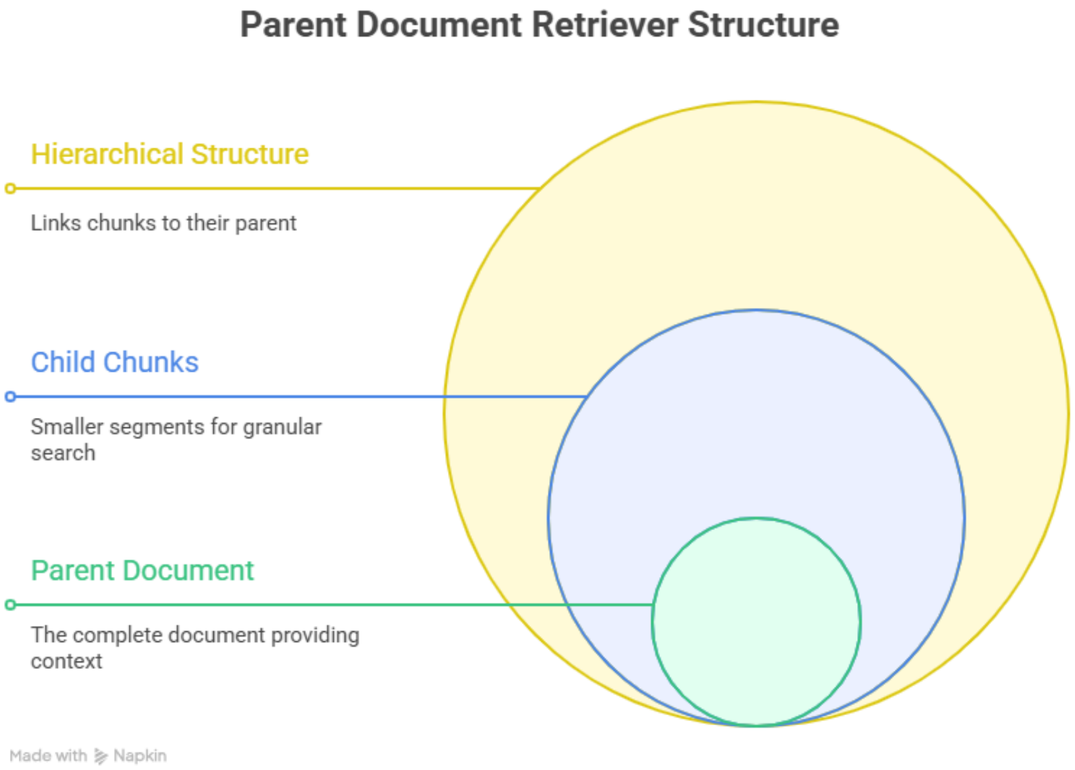
Parent Document Retriever

Benefits:

- **Improved Context: Provides more context than individual chunks, aiding in better understanding.**
- **Reduced False Positives: By considering the parent document, the retriever can filter out irrelevant chunks that might be semantically similar to the query but are not relevant in the larger context.**

Use Cases:

- **Legal Document Retrieval: Retrieving entire legal clauses or sections instead of isolated sentences.**



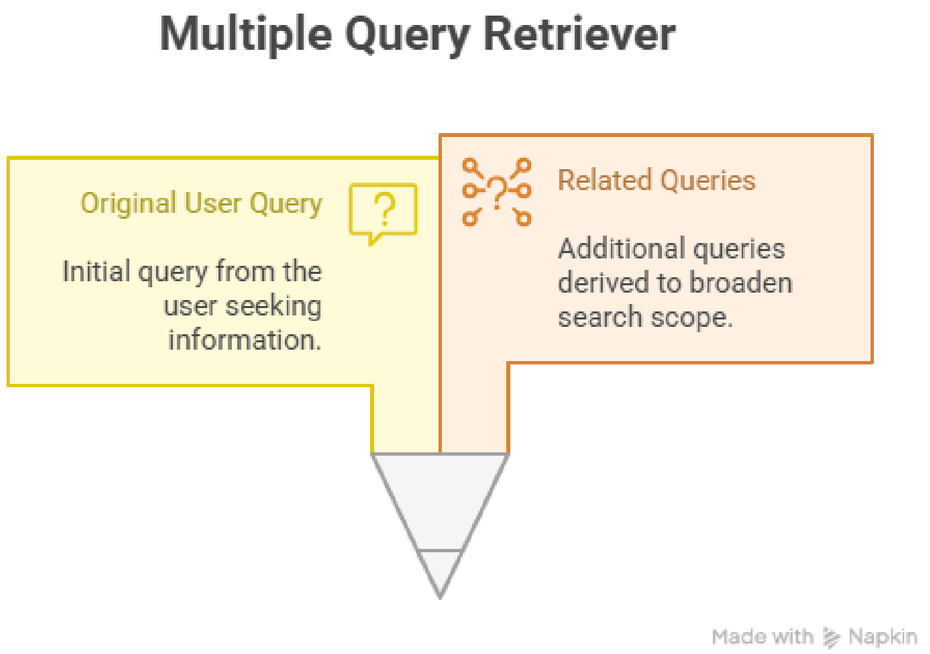
Multiple Query Retriever

Problem:

- **Traditional retrieval methods often miss relevant information by relying on a single query formulation.**
- **Important context can be lost when retrieving documents based on a single perspective.**

solution :

- **Multi-Query Approach Rewrites the original query from multiple perspectives.**
- **This method improves retrieval accuracy by capturing diverse aspects of the query, ensuring more comprehensive context.**



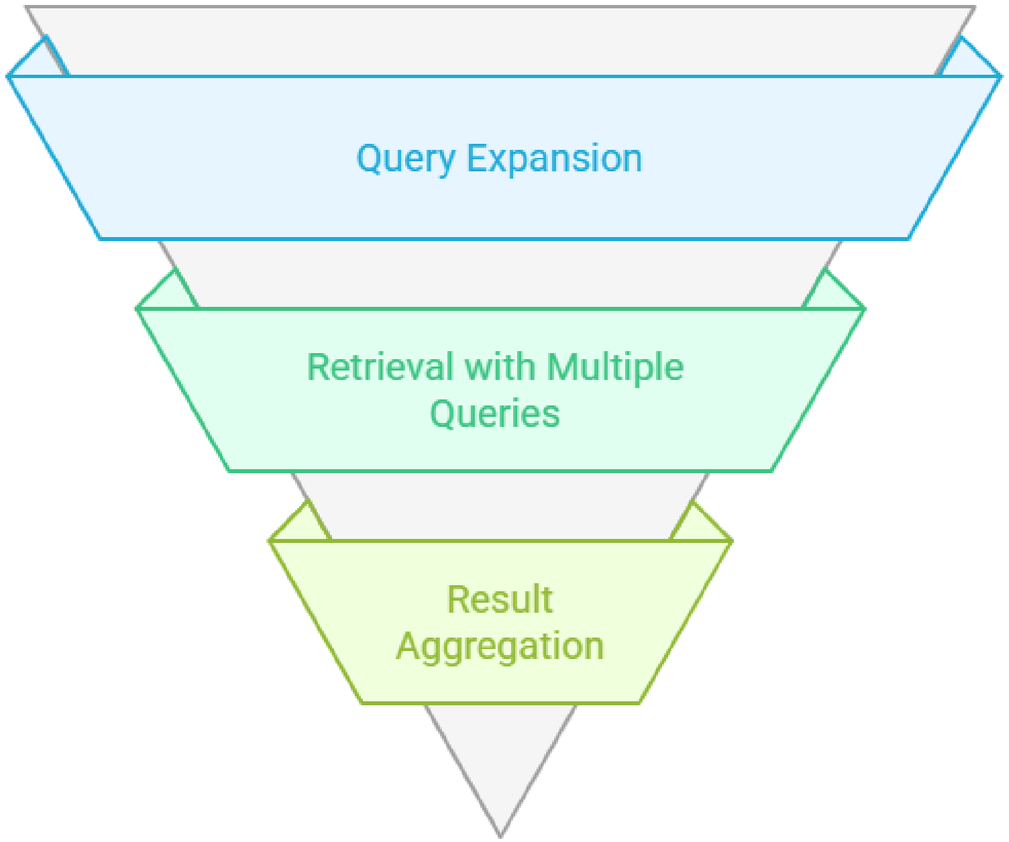
Multiple Query Retriever

How it Works:

1. **Query Expansion: The original query is expanded into multiple related queries using techniques like:**
    - **Rephrasing the query in different ways.**
    - **Generating queries that focus on different aspects of the original query.**

1. **Retrieval with Multiple Queries: Each generated query is used to retrieve relevant documents or chunks from the index.**

1. **Result Aggregation: The results from all queries are combined and ranked based on relevance scores.**



publica academy

RAPTOR

- RAG systems must handle specific (single-doc) and broad (multi-doc) questions that span many documents, which is challenging with typical retrieval methods.

- Raptor: Creates a tree of summaries by recursively clustering and summarizing documents. Captures both fine details and high-level concepts. Indexes summaries and original documents, providing coverage across all user questions.

RAPTOR solves the retrieval-scope mismatch problem — where basic RAG fails to handle both fine-detail questions and broad, cross-document questions. It does this by building a tree of summaries that provides:

- Local information
- Mid-level summaries
- Global, cross-document understanding

All in one structure.

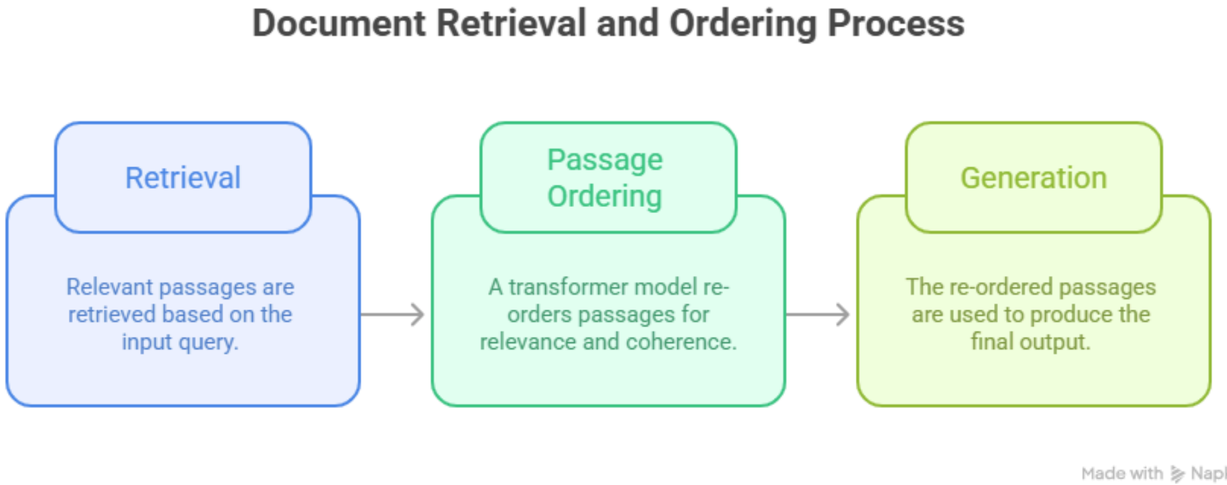
RAPTOR

Benefits:

- **Improved Coherence: Ensures that the retrieved passages are presented in a logical and coherent order.**
- **Enhanced Relevance: Prioritizes passages that are most relevant to the query.**

Use Cases:

- **Summarization: Generating coherent summaries from multiple documents.**
- **Question Answering: Providing comprehensive and well-structured answers to complex questions.**



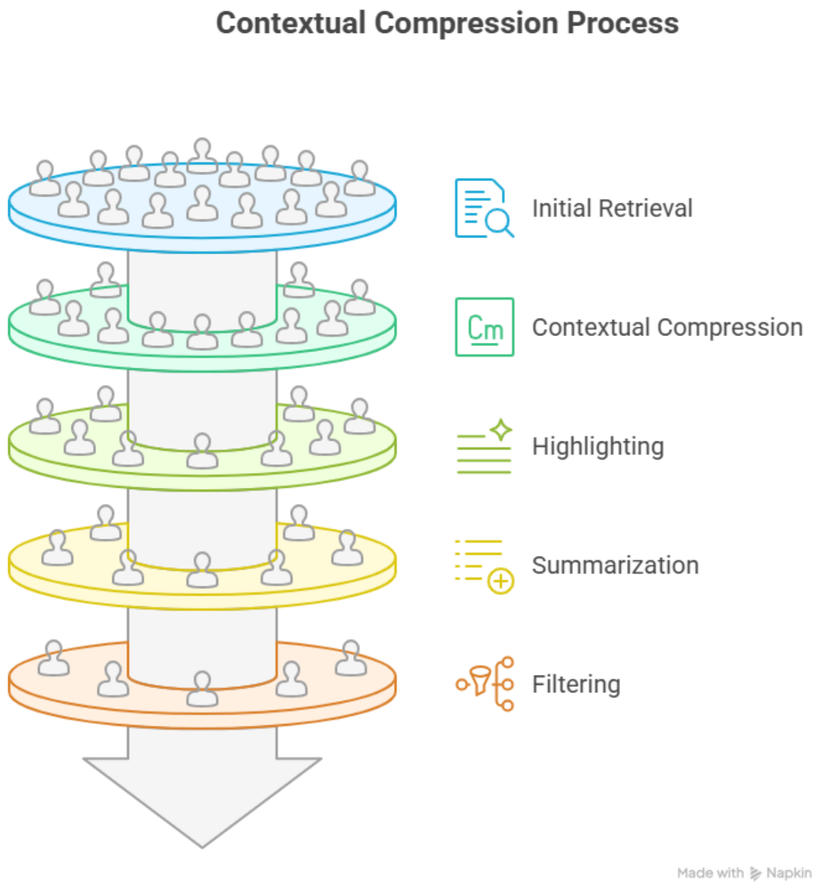
Contextual Compression

One of the biggest problems in RAG is retrieving entire documents or large chunks that contain irrelevant information.

Contextual Compression Approach:

- **Extracts only the most relevant parts of retrieved documents.**
- **Compresses information for more focused and efficient retrieval.**
- **Improves relevance while reducing unnecessary data.**

Contextual Compression



Benefits:

- **Reduced Noise: Filters out irrelevant information, improving the signal-to-noise ratio.**
- **Improved Efficiency: Reduces the amount of text that needs to be processed, saving computational resources.**

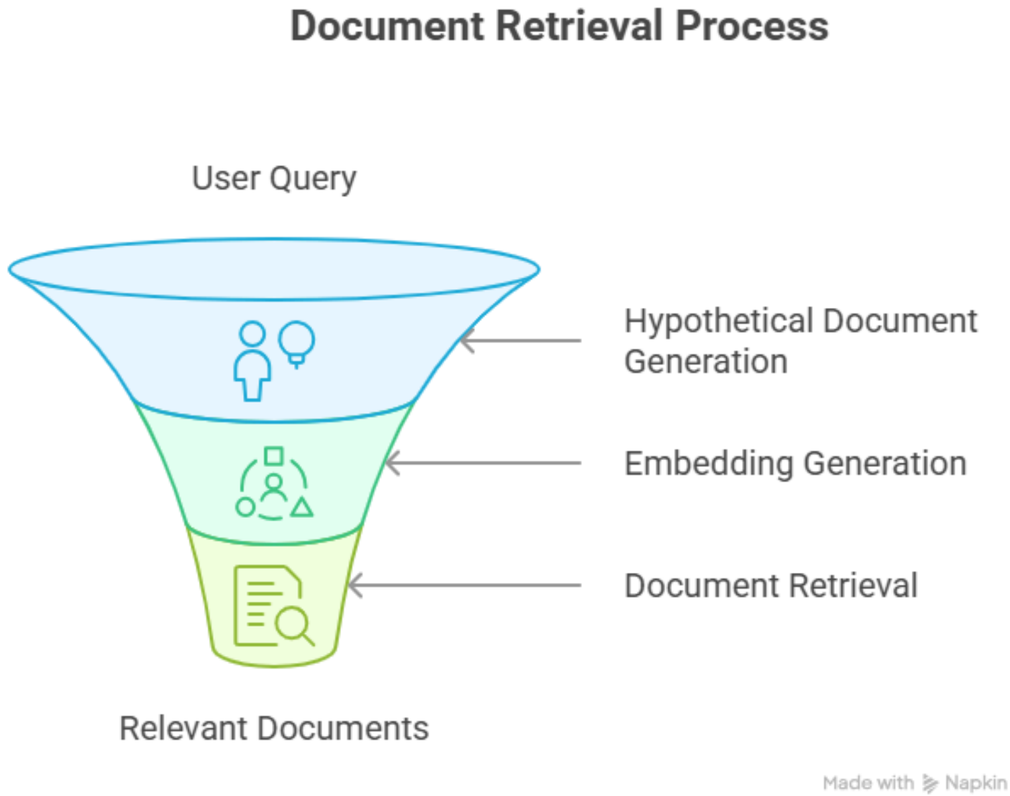
Use Cases:

- **Extracting the most relevant information from a large document to answer a specific question.**

- **Chatbots with Limited Context Windows**

HyDE (Hypothetical Document Embeddings)

- Some retrieval methods may not always find the most relevant documents due to limitations in keyword matching.
- Queries might not directly align with indexed documents, leading to suboptimal retrieval results.



publica academy

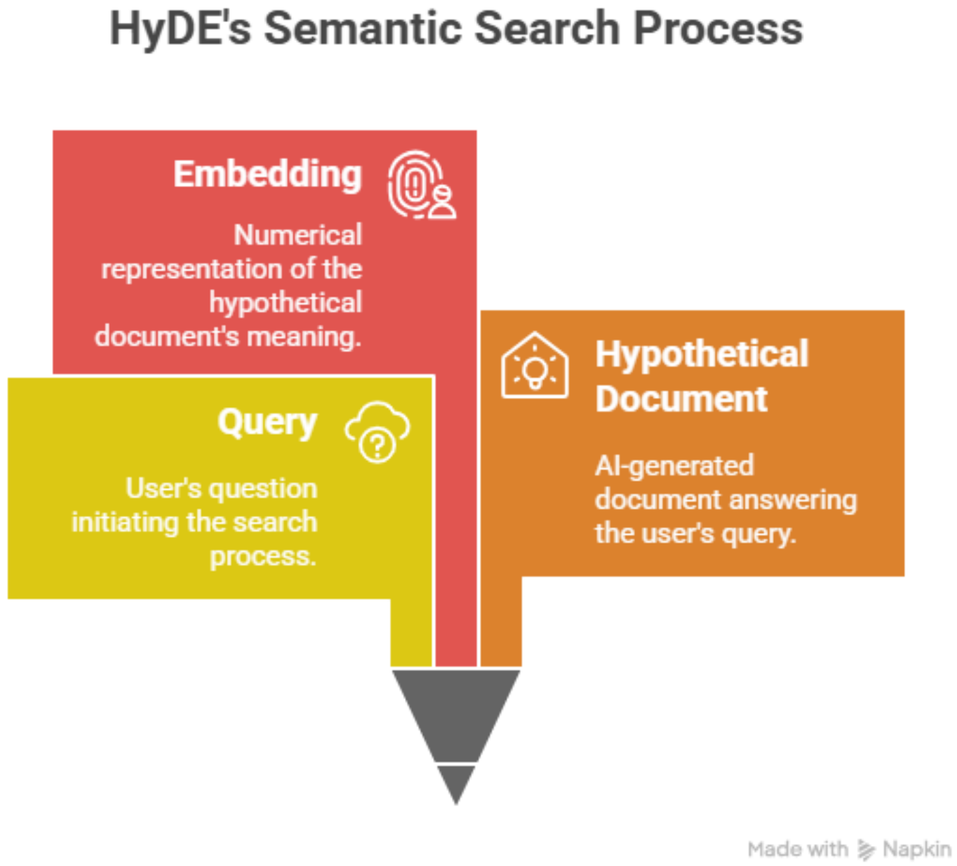
Solution: HyDE Approach:

- Uses LLMs to generate hypothetical documents based on the query.
- Converts these hypothetical documents into embeddings.
- Retrieves indexed documents that best align with these generated embeddings.

This method enhances retrieval quality by bridging the gap between user intent and stored information by generating an LLM-created “ideal answer” that better represents what the user is looking for, then using it to find the closest real documents.

Solution: HyDE Approach:

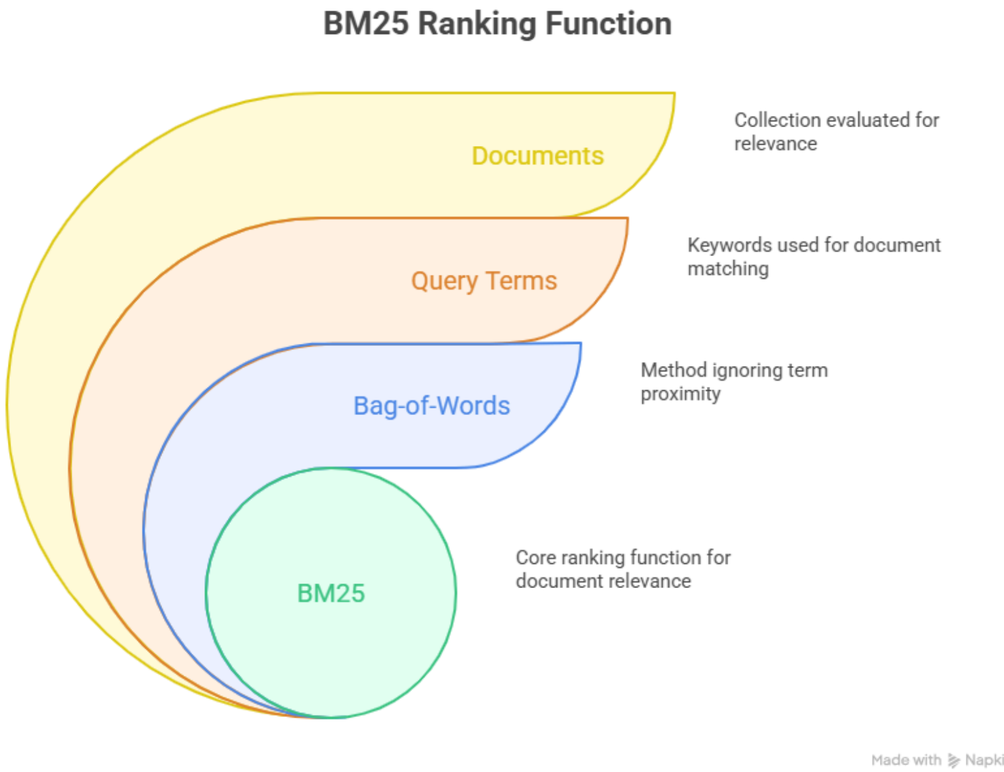
- **Use Cases:**
  - **Patent Search: Finding patents that are related to a specific invention, even if they use different terminology.**
  - **Scientific Literature Search: Discovering research papers that address a specific research question.**



BM25 (Best Matching 25)

- **Problem in RAG:**
  - **Many retrieval systems struggle with vague or under-specified queries — especially when embeddings don't capture the right keywords.**

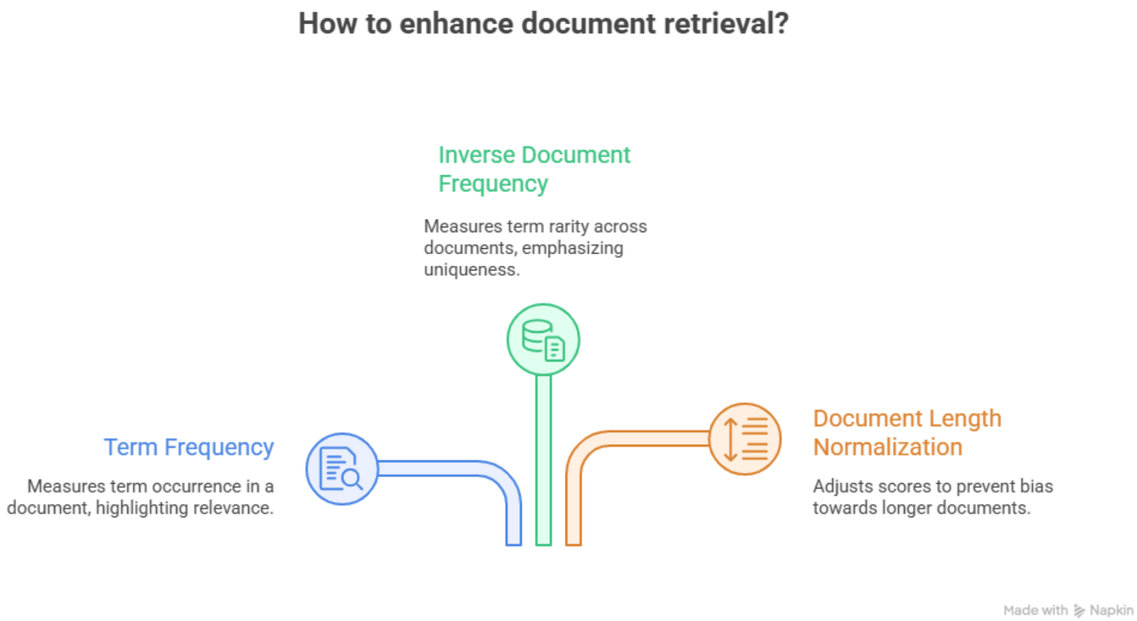
- **BM25 Solution:**
  - **Uses exact keyword matching and ranks based on term frequency and importance (TF-IDF)**
  - **Efficiently retrieves documents containing precise terms from the query**



publica academy

BM25 - How it Works:

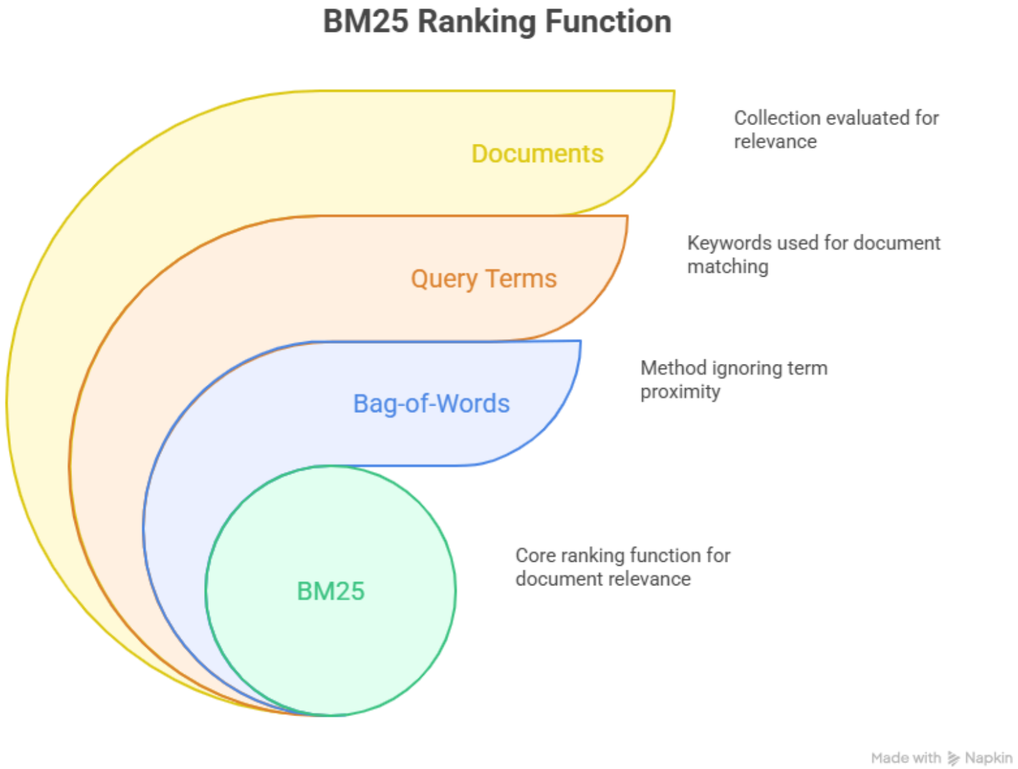
- **Term Frequency (TF): Measures how often a term appears in a document.**
- **Inverse Document Frequency (IDF): Measures how rare a term is across the entire document collection.**
- **Document Length Normalization: Adjusts the score based on the length of the document to prevent longer documents from being unfairly favored.**



publica academy

BM25 - Use Cases:

- Traditional Search Engines: Used as a core ranking function in many search engines.
- Information Retrieval Systems: Widely used in various information retrieval applications.



publica academy

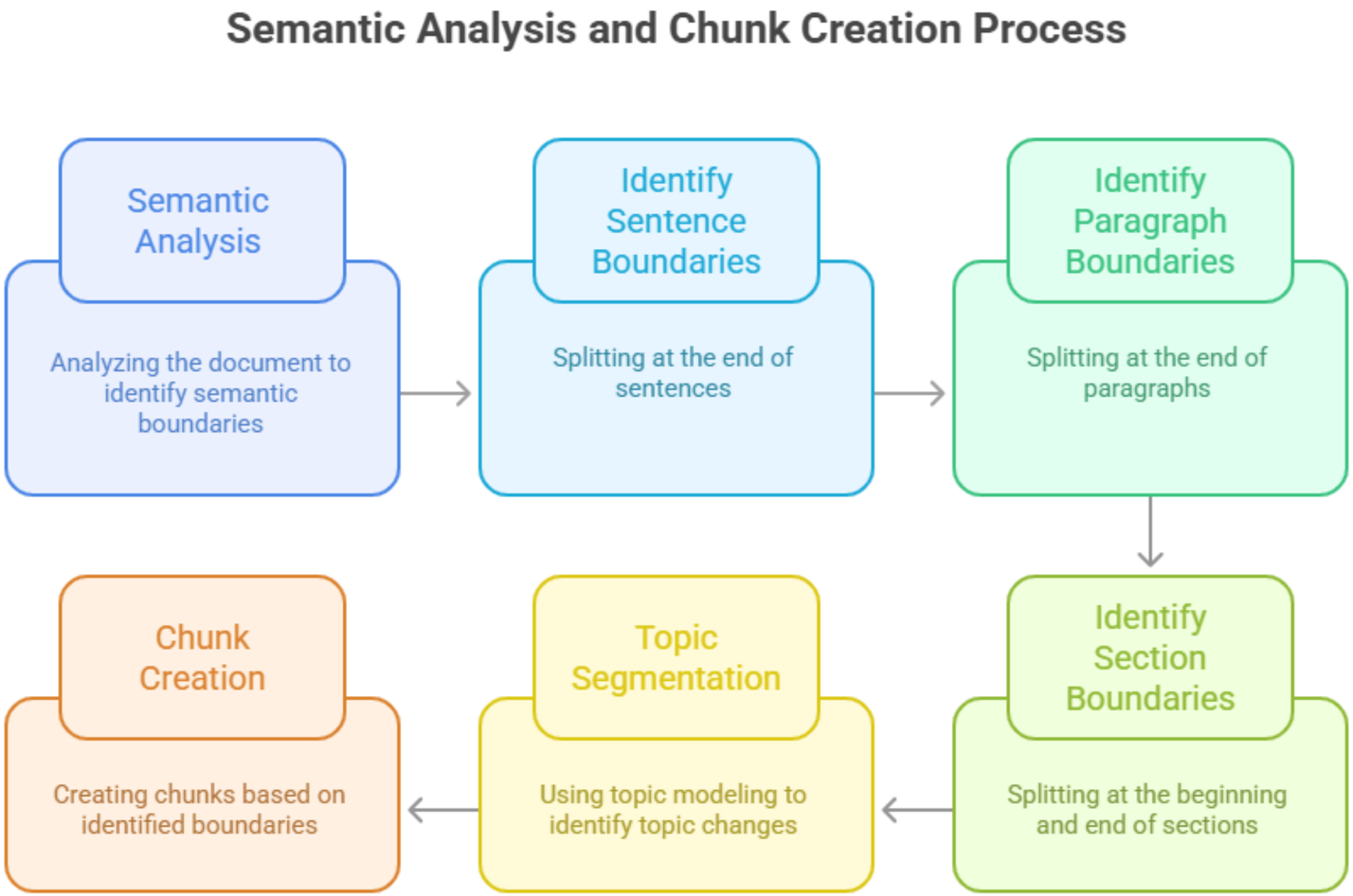
Semantic Splitting

- **Semantic splitting involves dividing documents into chunks based on semantic boundaries rather than fixed-size segments.**

- **This ensures that each chunk contains a complete and meaningful unit of information.**

publica academy

Semantic Splitting



Semantic Splitting

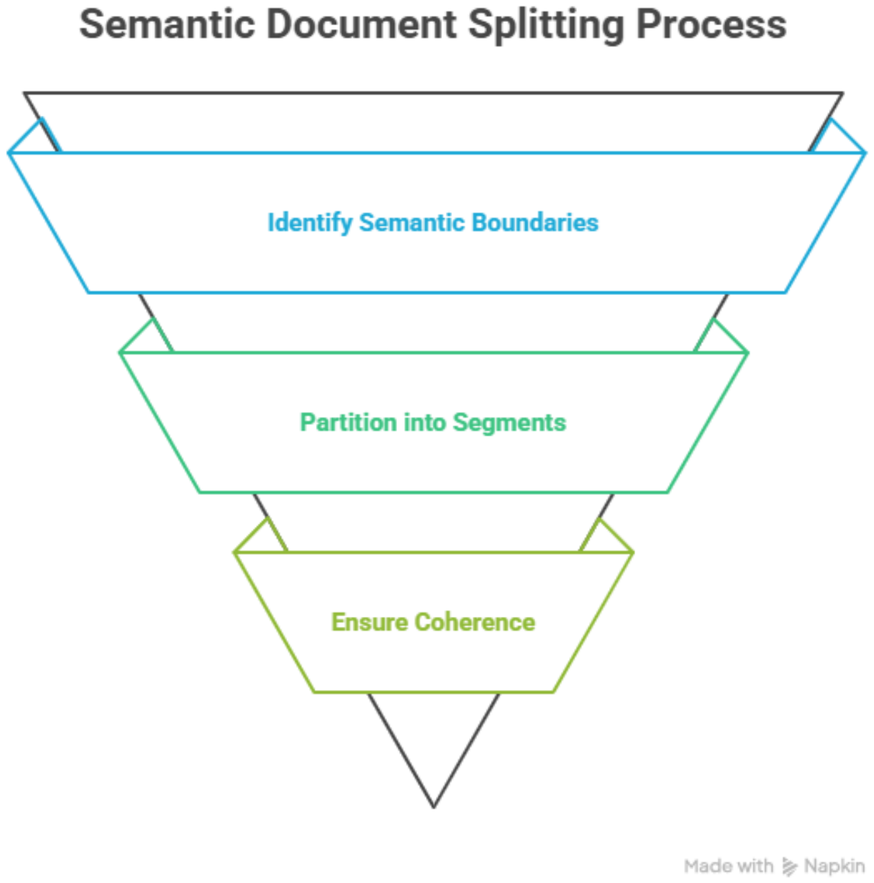
- **How it Works:**

- **Semantic Analysis: The document is analyzed to identify semantic boundaries, such as:**
  - **Sentence Boundaries: Splitting at the end of sentences.**
  - **Paragraph Boundaries: Splitting at the end of paragraphs.**
  - **Section Boundaries: Splitting at the beginning and end of sections.**
  - **Topic Segmentation: Using topic modelling techniques to identify changes in topic.**
  - **Chunk Creation: Chunks are created based on the identified semantic boundaries.**

publica academy

Semantic Splitting : Benefits:

- **Improved Context: Each chunk contains a complete and meaningful unit of information, providing better context for retrieval.**
- **Reduced Noise: Avoids splitting sentences or paragraphs.**



publica
academy

Glad to see you come this far

See you in the next section

1
2
3
4
5
6
7
8
9