MASTERING LLM PRESENTS

COFFEE BREAK CONCEPTS

A System design case study

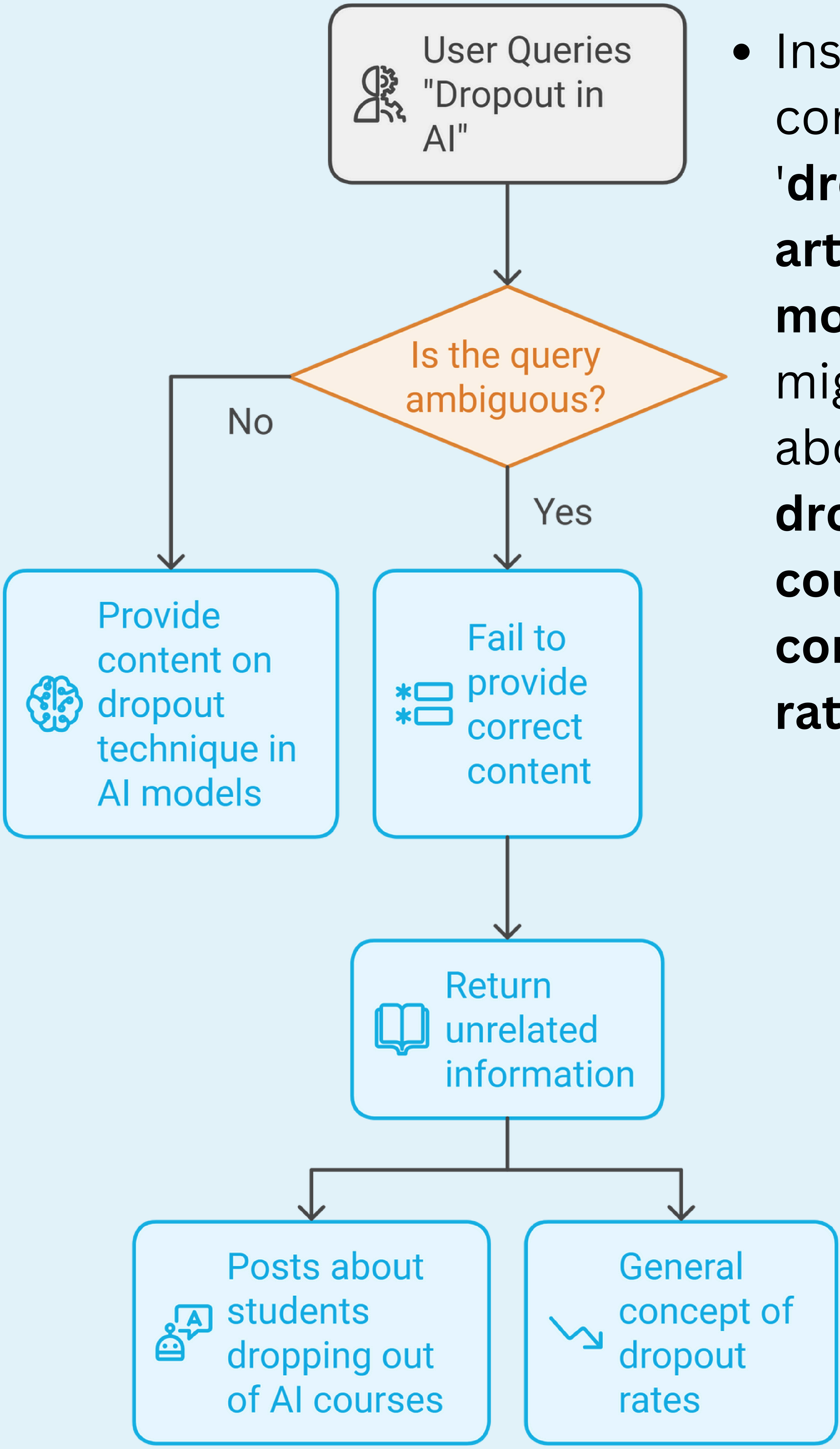
LLM System Design - LinkedIn Search

Follow us on

Linkedin’s Search Problem

- **Complex Queries:** Increase in search queries using natural language and specific concepts.
- **Inadequate Results:** Existing search engine often failed to return relevant posts due to lack of conceptual understanding.
- **User Engagement:** Need to improve user engagement metrics such as on-topic rate and long-dwells.
- **Content Relevance:** Requirement for a search engine capable of semantically understanding and matching the intent behind user queries.

An Example of the problem



- Instead of providing content related to the '**dropout**' technique in **artificial intelligence models**, the search might return posts about **students dropping out of AI courses** or the **general concept of dropout rates**.

Key objective for improvement

Improve On-topic Rate (Quality metric)

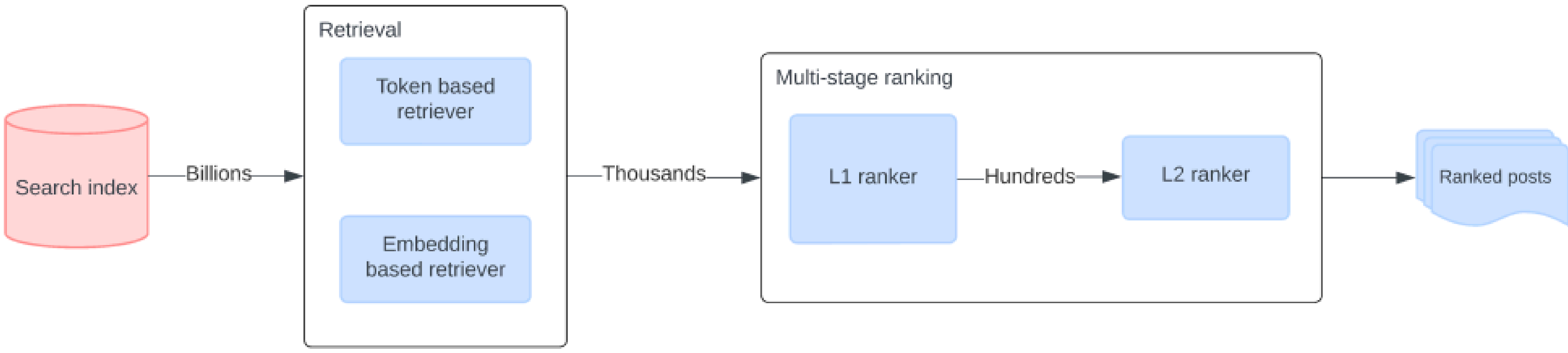
- **Implement AI:** Use AI to check if the posts are relevant and directly answer the user's query.
- **Upgrade Search Algorithms:** Improve the search system to better understand what users are asking for.

Boost Long-dwells (Engagement metric)

- **Track Reading Time:** Measure how long users read the posts to gauge their interest.
- **Focus on Reading, Not Just Clicks:** Prioritize posts that keep users reading longer, even if they don't interact with likes or comments.

These goals focus on making sure the search results are more relevant and engaging for users.

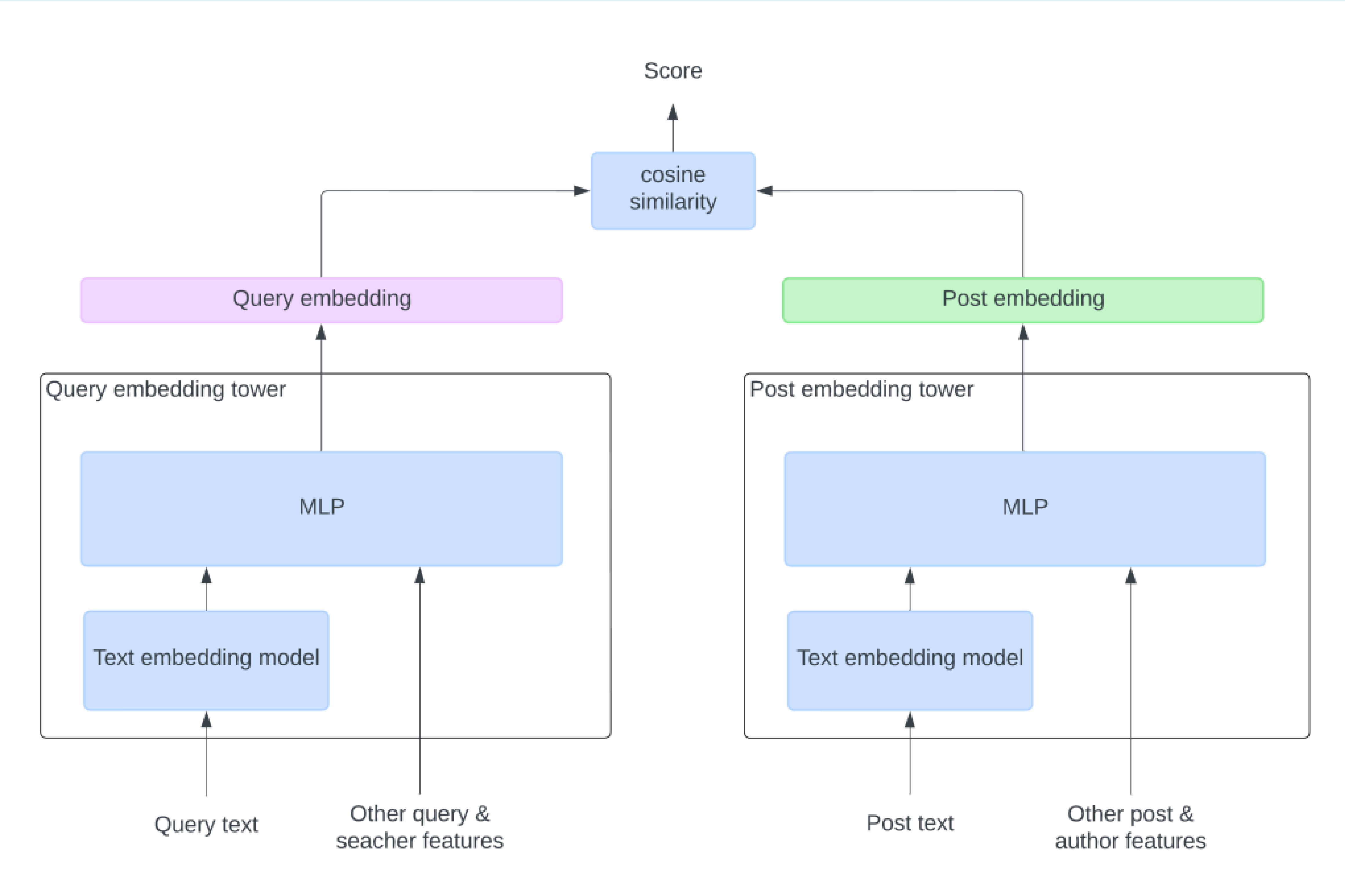
High-level System Design



- **Retrieval Layer:**
  - **Function:** Selects a few thousand candidate posts from the vast pool of billions, based on the initial query.
- **Multi-Stage Ranking Layer:**
  - **Function:** Scores these candidate posts in two stages, refining the selection with each step.
  - **Result:** Outputs a ranked list of posts that best match the query criteria.

This design ensures that the system efficiently manages and processes large datasets while accurately responding to complex search queries.

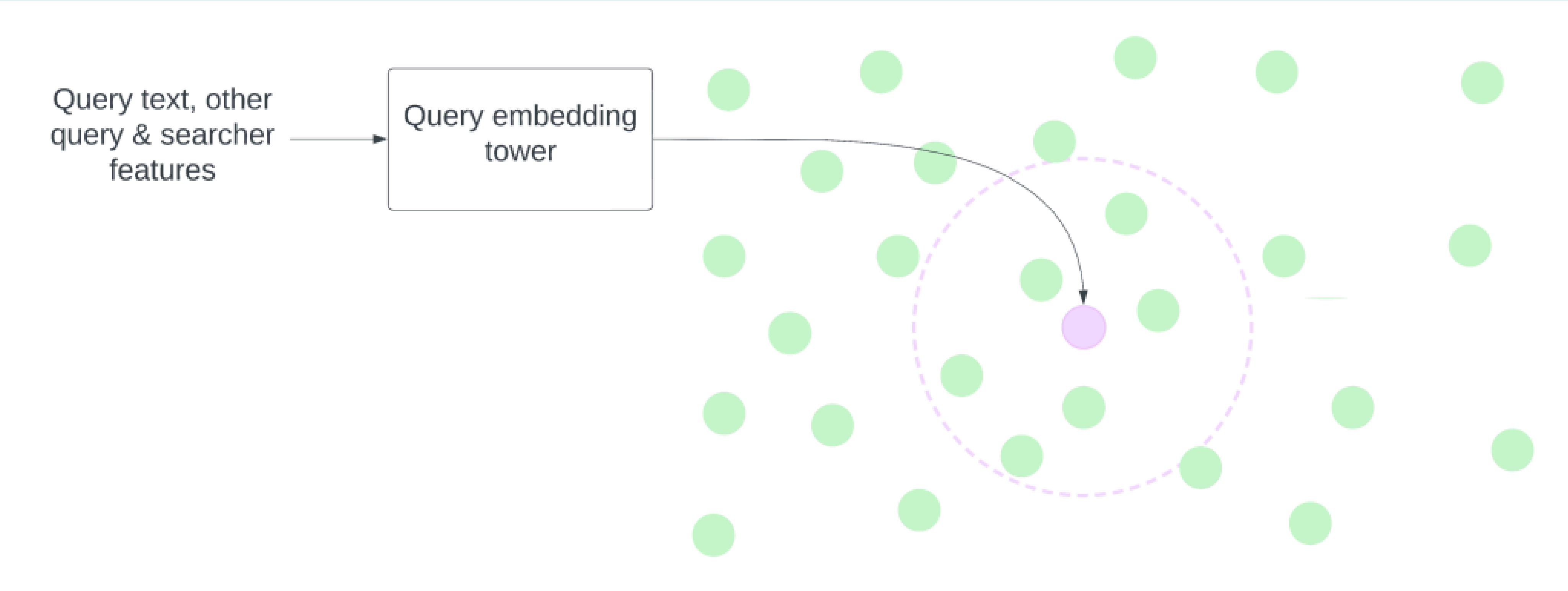
Retrieval layer



1. **Token-Based Retriever (TBR):**
   - Uses an inverted index for exact keyword matching.
   - Intersects post lists from keywords to find matches containing all keywords.
2. **Embedding-Based Retriever (EBR):**
   - Employs a two-tower model for creating query and post embeddings.
   - Uses multilingual-e5 for text embedding and MLP for additional processing.
   - Calculates post relevancy using cosine similarity between embeddings.

This approach ensures accurate and context-aware search results through both keyword and semantic analysis.

Efficient Two-Tower Model Implementation



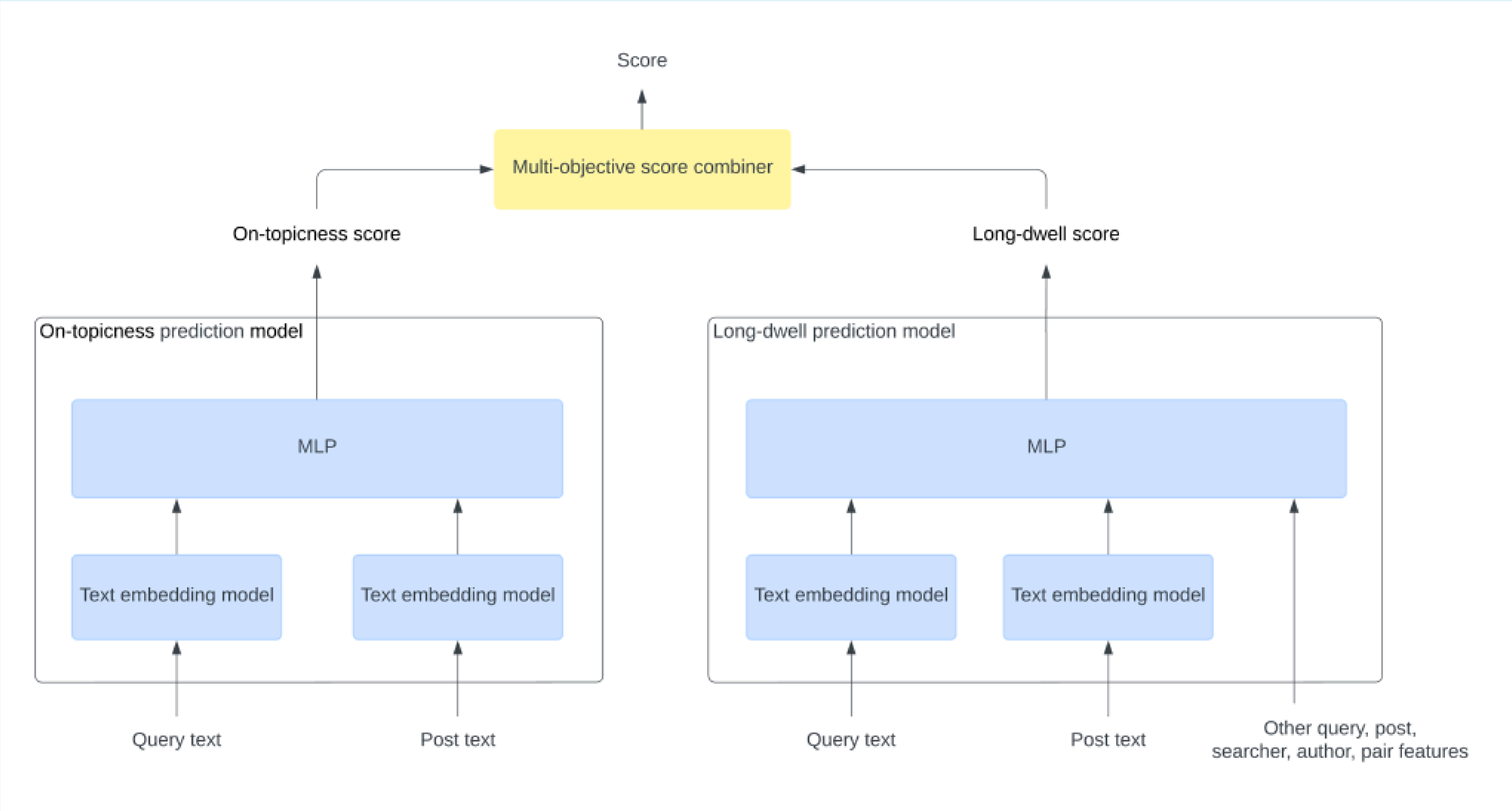
1. **Training:** Utilizes historical query, post, and label data to train query and post embedding towers.
2. **Model Advantages:**
   - **Pre-computed Embeddings:** Post embeddings are pre-stored, bypassing real-time computation for each query.
   - **Top-k Selection:** Identifies top-k posts by comparing pre-computed post embeddings to real-time query embeddings.
3. **Operational Process:**
   - **Offline and Nearline Jobs:** Batch processes compute and update post embeddings for storage.
   - **Efficient Query Handling:** Uses nearest neighbor search to quickly find the most relevant posts during a query.

This streamlined approach allows for rapid and scalable search capabilities within LinkedIn's vast content network.

Why Token-Based Retriever (TBR)

1. **Exact Keyword Matching:**
    - TBR is crucial for scenarios where precision in **keyword alignment is necessary**, ensuring that results directly contain queried terms.
2. **Navigational Queries:**
    - Ideal for searches aimed at **finding specific posts or content**, such as named reports or titled documents.
    - **Example:** Searching for "**Introducing Semantic Capability in LinkedIn's Content Search Engine**" to find a specific blog post.
3. **Complementing Semantic Search:**
    - Balances the semantic depth of EBR by handling straightforward, keyword-specific queries that require direct matching rather than conceptual interpretation.
    - **Example**: A query for "**Python coding tips**" would pull up posts explicitly mentioning these exact keywords, even if they don't delve into broader coding advice.

Multi-stage ranking layer



- **Real-Time Scoring:** Feasible due to fewer posts; allows detailed feature interaction.
- **Optimization Goals:** Enhances on-topic rate and long-dwells, considering factors like content quality, searcher intent, and post freshness.
- **Ranking Process:**
  - **L1 Stage:** Scores thousands of posts using a simple model, narrowing down to hundreds.
  - **L2 Stage:** Applies a complex model to these hundreds, finely tuning the final rankings.
- **Model Differences:** L1 and L2 use similar architectures but vary in complexity and feature depth.

Architecture For Ranking Models

Two Separate Models:

- **On-topicness Model:** Predicts relevance of posts to the query.
- **Long-dwell Model:** Estimates user engagement duration with posts.

Input Features:

- **Common to both:** Query text, post text.
- **On-topicness Model:**
  - Uses text embeddings from **multilingual-e5**.
  - Embeddings are processed through an **MLP to generate a score**.
- **Long-dwell Model:**
  - Includes additional features: **BM25 score, job titles in query, post popularity, searcher's job-seeking intent, author's popularity, and connection status** between searcher and author.
  - Text embeddings combined with these features are processed by an **MLP for scoring.**

Text Embedding:

- Both models use multilingual-e5 due to its effectiveness in semantic matching and high performance on MTEB leaderboard.

Training:

- Models trained using historical query-post interactions and their labels (on-topicness and long-dwells).

Scoring Combination:

- Scores from both models are combined to determine the final post ranking.

Outcome of LinkedIn's Enhanced Content Search Engine

- **Enhanced Query Resolution:** Successfully handles complex queries such as "**how to ask for a raise?**".
- **Improved Metrics:** On-topic rate and long-dwells have both increased by over **10%**.
- **Boosted User Engagement:** Improved search results have led to more member interaction and longer site-wide sessions on LinkedIn.

Read Complete blog here 

www.masteringllm.com

50% OFF

LLM Interview Course

Want to Prepare yourself for an LLM Interview?

- 100+ Questions spanning 14 categories with Real Case Studies
- Curated 100+ assessments for each category
- Well-researched real-world interview questions based on FAANG & Fortune 500 companies
- Focus on Visual learning
- Certification

HOT SALE!

Coupon Code - LLM50

Coupon is valid till 30th Jan 2024

www.masteringllm.com

AgenticRAG with LlamaIndex

Want to learn why AgenticRAG is future of RAG?

- Master **RAG fundamentals** through practical case studies
- Understand how to overcome **limitations of RAG**
- Introduction to **AgenticRAG** & techniques like **Routing Agents, Query planning agents, Structure planning agents, and React agents with human in loop**.
- **5** real-time **case studies with code walkthroughs**