

Hybrid Search in Apache Doris: New Choices in the AI Era

李昊鹏

飞轮科技资深研发

Apache Doris PMC Member





目录

01 什么是混合搜索

02 混合搜索对 Doris 的挑战

03 Doris 混合搜索的实现方案

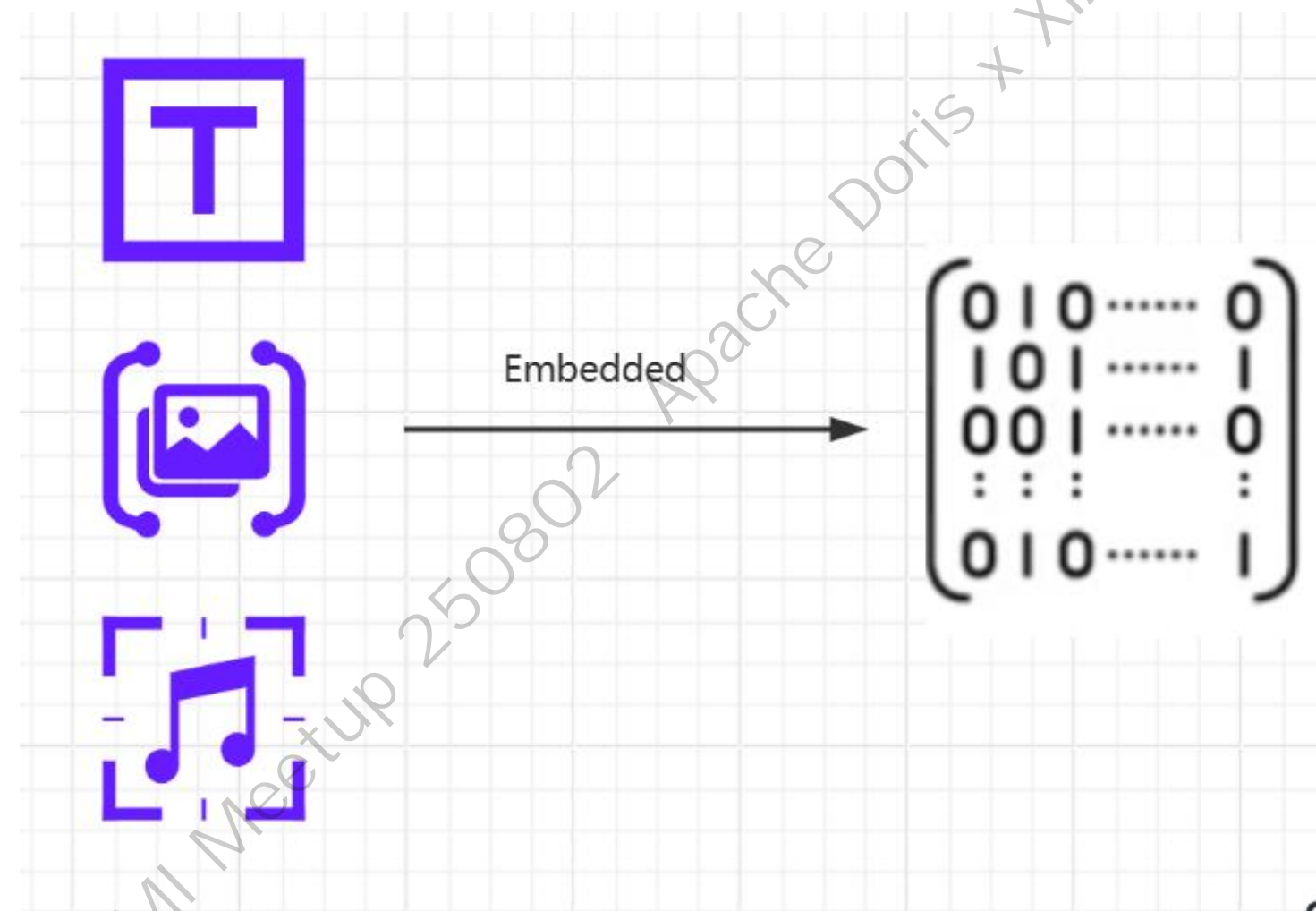
04 混合搜索的应用场景

05 特别致谢

1

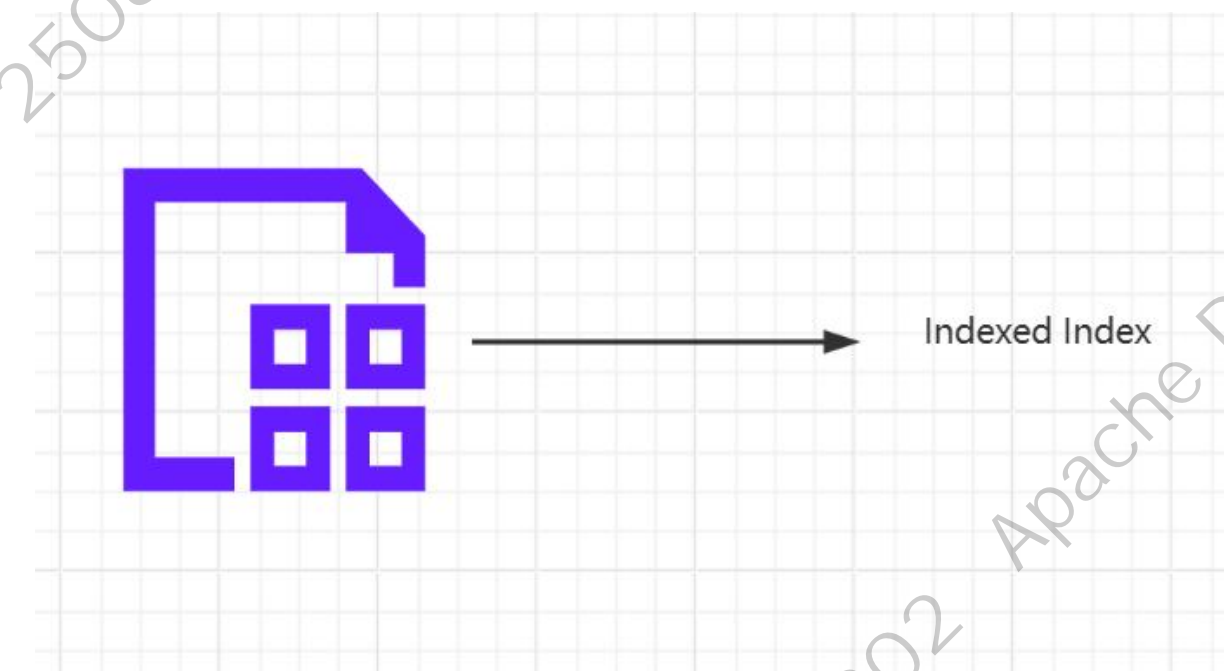
什么是混合搜索

向量搜索 全文搜索



向量搜索

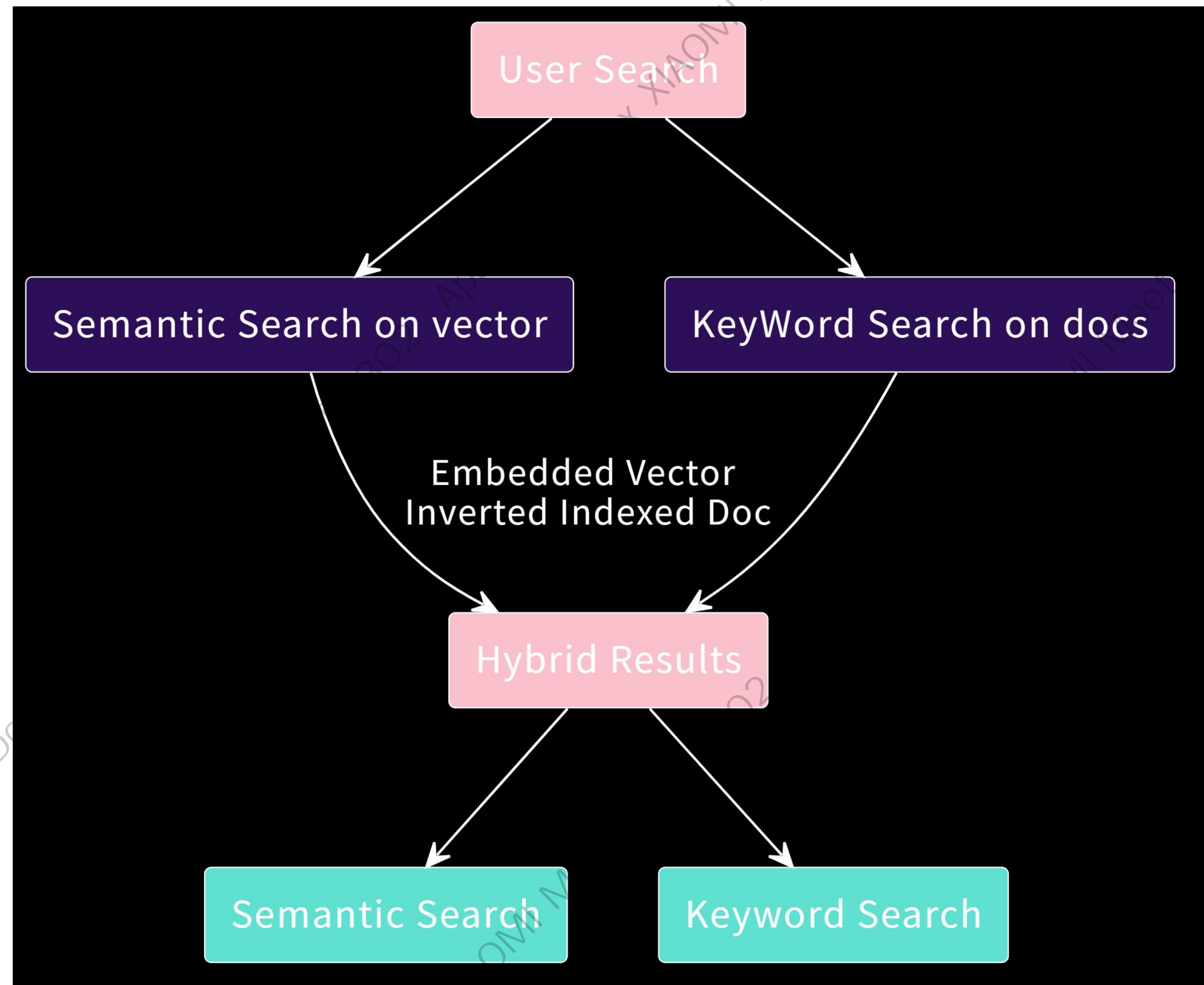
通过语义相似度进行的搜索，需要对原始的非结构化数据进行 Embedded。对向量进行相似度的搜索



全文搜索

通过倒排索引进行的关键字，短语的搜索，对与文档的搜索能力

混合两类的搜索



Hybrid search outperforms vector or keyword search alone

Cortex Search combines the strengths of vector search, keyword search and semantic reranking into a single search interface. Our internal research shows that this approach yields higher quality search results across a variety of RAG-oriented search workloads than a vector search or a keyword search alone. This means that you get an out-of-the-box quality boost over standalone vector databases, which typically provide only vector search without lexical search or reranking. In fact, on a sampled set of public and proprietary “question-and-answer”-style benchmarks, we found that Cortex Search’s hybrid retrieval approach achieved a more than 12% retrieval boost, as compared to simple vector search alone, and drastically outperforms simple keyword search (see Figure 3 below).

This complex retrieval and reranking stack is fully managed, saving you from having to stitch together and tune hyperparameters for multiple retrieval and reranking services. More details on the research behind the Cortex Search retrieval stack will be shared on our [Snowflake Engineering Blog](#).

基于 Snowflake 的分析数据：混合搜索通过将精确匹配与语义理解相结合，提供高质量的结果。

混合搜索的优点

要点	优点
提升搜索准确性和相关性	混合搜索通过将精确匹配与语义理解相结合，提供高质量的结果
提升用户体验	即使用户输入不准确的术语或模糊的关键词，该系统也能提供有意义的内容。
个性化	混合搜索系统可以配置为动态调整关键词权重和语义相关性，或者让用户能够对其进行控制

混合搜索的难点



索引的管理

- 倒排索引
- 向量索引



高效存储

- 存储空间
- 读写效率



高性能分析

- 查询
- 排序

2

混合搜索对 Doris 的挑战

混合搜索对 Doris 的挑战

要点	优势	问题
索引的管理	存在高效稳定实现的倒排索引	需要添加向量索引，并搭建通用的索引管理框架
高效存储	列式存储压缩率高	可能存在 IO 读放大
高性能分析	向量化计算引擎，计算能力优秀	需要进行特定混合场景的优化

3 Doris 混合搜索的实现方案

Doris 混合搜索的实现工作

工作	现状	工作
BM25 的关键词搜索	支持倒排索引	在倒排索引的基础上支持，BM25 的打分能力
ANN 的向量索引搜索	1.支持 array<float> 类型 2. 支持 KNN 的暴力搜索	支持向量类型的索引，引入向量索引库
混合搜索的框架适配	无	支持一个通用的虚拟列的计算框架，兼容上述索引的能力
全局延迟物化	单表的能力	适配上述工作

BM25 的使用介绍

```
CREATE TABLE IF NOT EXISTS bm25_test_mow_multi_segment (  
  `id` BIGINT NOT NULL,  
  `c` TEXT NULL,  
  INDEX c_idx(`c`) USING INVERTED PROPERTIES("parser" = "chinese", "parser_mode" = "coarse_grained",  
  "support_phrase" = "true")  
)  
UNIQUE KEY(`id`)  
DISTRIBUTED BY HASH(`id`) BUCKETS 1  
PROPERTIES(  
  "replication_allocation" = "tag.location.default: 1"  
);  
  
SELECT id, bm25() as score FROM bm25_test_mow_multi_segment WHERE c MATCH 'Doris' ORDER BY score;
```

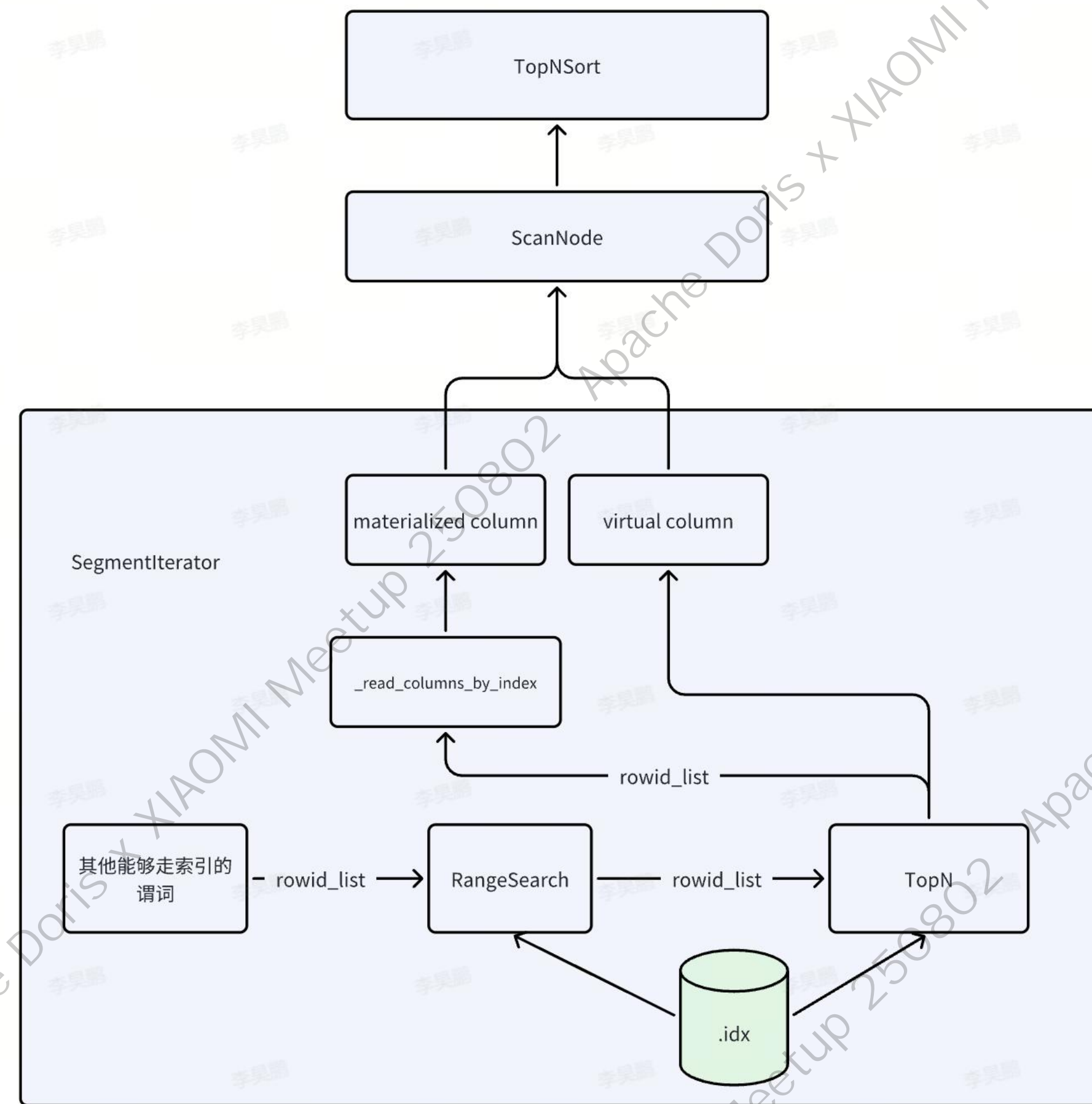

ANN 向量索引的使用介绍

```
CREATE TABLE `vector_test` (  
  `id` bigint NOT NULL,  
  `embedding` array<float> NOT NULL,  
  INDEX idx_test_ann (`embedding`) USING ANN PROPERTIES("dim" = "3", "index_type" = "hns",  
"metric_type" = "l2_distance")  
) ENGINE=OLAP  
DUPLICATE KEY(`id`)  
DISTRIBUTED BY HASH(`id`) BUCKETS 8  
PROPERTIES (  
  "replication_allocation" = "tag.location.default: 1"  
);  
  
SELECT id, l2_distance([1.0,2.0,3.0], y) as score FROM vector_test  
WHERE id = 'xxx' ORDER BY score DESC LIMIT 10
```

- dim: 维度，入库数据维度不符合会报错
- index_type: 支持hns的ann索引
- metric_type: l2_distance/cosine_similarity 这两类，只能设置一种

支持 where 混合标量的等值查询和向量的索引的 topn 查询

混合搜索的框架支持



BM25 和向量索引类似，这里都引入了一个虚拟列的框架

在 scan node 结束之后，将 where/order by 子句之中的
打分信息物化

后续的排序的 topn，基于物化的分数列进行排序

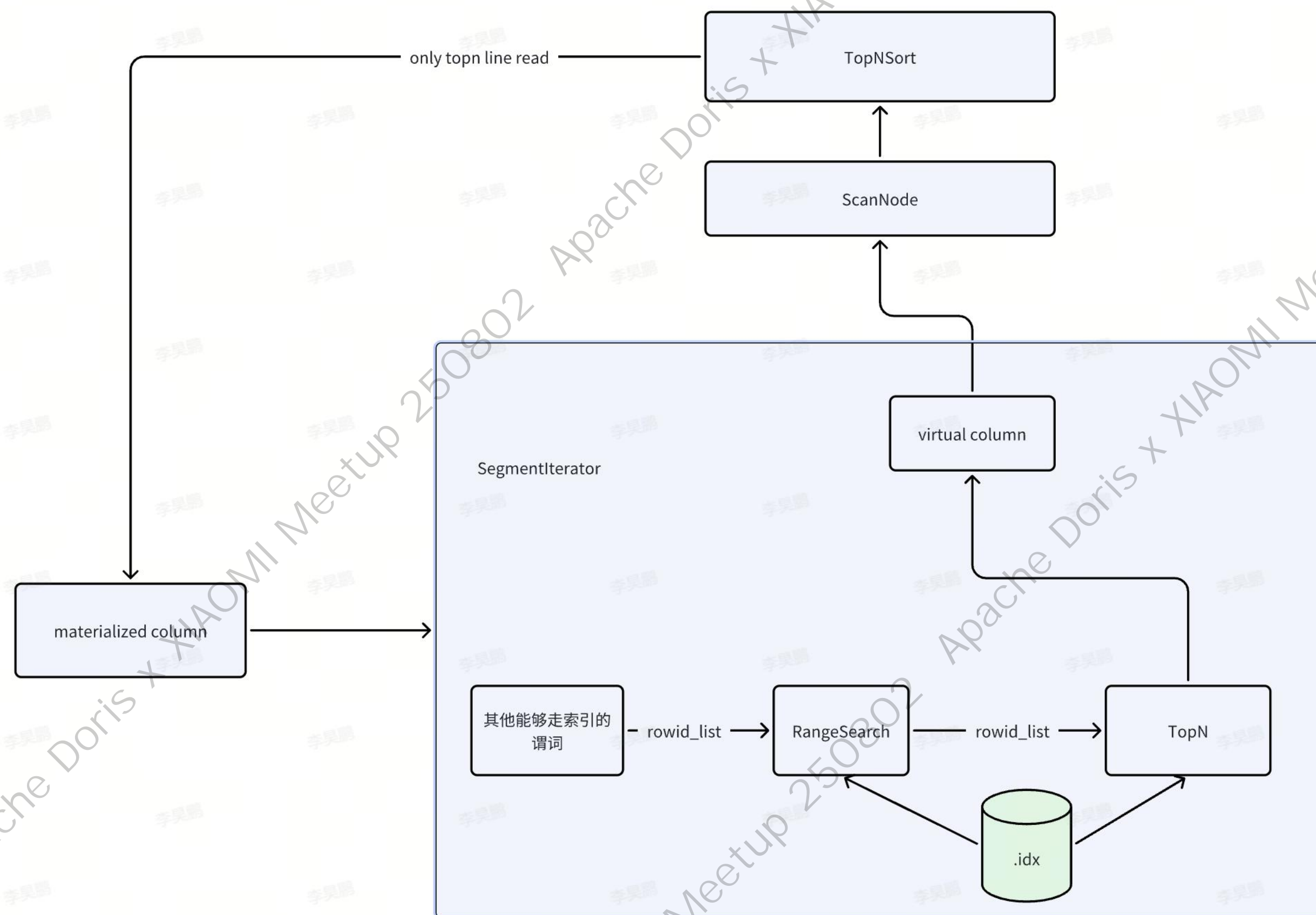
混合搜索的使用介绍

```
WITH channel_1 AS (  
    SELECT id, c, bm25() as score  
    FROM my_table  
    WHERE c MATCH_ANY 'doris' AND x = 'xxx'  
    ORDER BY score DESC  
    LIMIT 10  
) , channel_2 AS (  
    SELECT id, c, L2_distance([1.0,2.0,3.0], y) as score  
    FROM my_table  
    WHERE x = 'xxx'  
    ORDER BY score DESC  
    LIMIT 10  
)  
SELECT * FROM channel_1  
UNION ALL  
SELECT * FROM channel_2
```

```
SELECT *  
FROM (  
    SELECT * FROM channel_1  
    UNION ALL  
    SELECT * FROM channel_2  
) t0  
ORDER BY  
CAST(AI_QUERY( -- 让 LLM 打分重排序  
            '评价上下文相关性分数，输出 0-10。上下文：', c)  
        ) AS DOUBLE) DESC  
LIMIT 10
```

1. 直接输出，业务层面自己来重新排序
2. 利用 LLM 大模型 or 业务自己 UDF 的嵌入的排序模型来排序

全局延迟物化



```
SELECT  doc
        Cosine_distance([1, 2], children) AS dist
FROM    hackernews_1m
ORDER BY dist limit 10;
```

通过全局延迟物化 在 topn 之后再去读取真正需要的结果
这减少了 IO 和排序计算的内存拷贝，得到更好的性能表现

4

混合搜索的应用场景

应用场景1 – RAG

RAG 场景的向量搜索

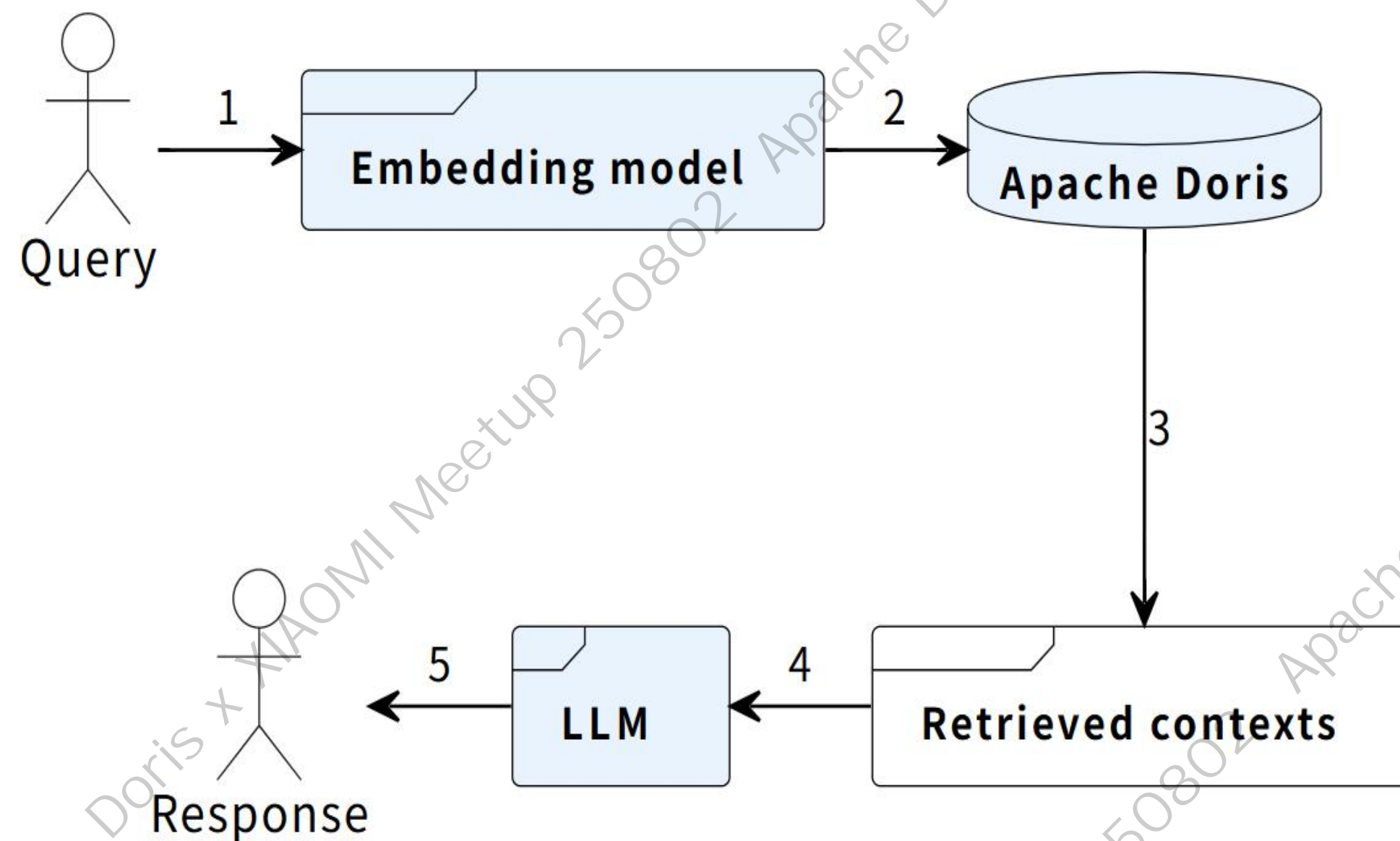
优势

支持语义检索

- 混合检索则能找出语义相关的内容，即使词不完全一样，如“富含维生素 C 的水果”也能被匹配。

解决 LLM 的记忆问题

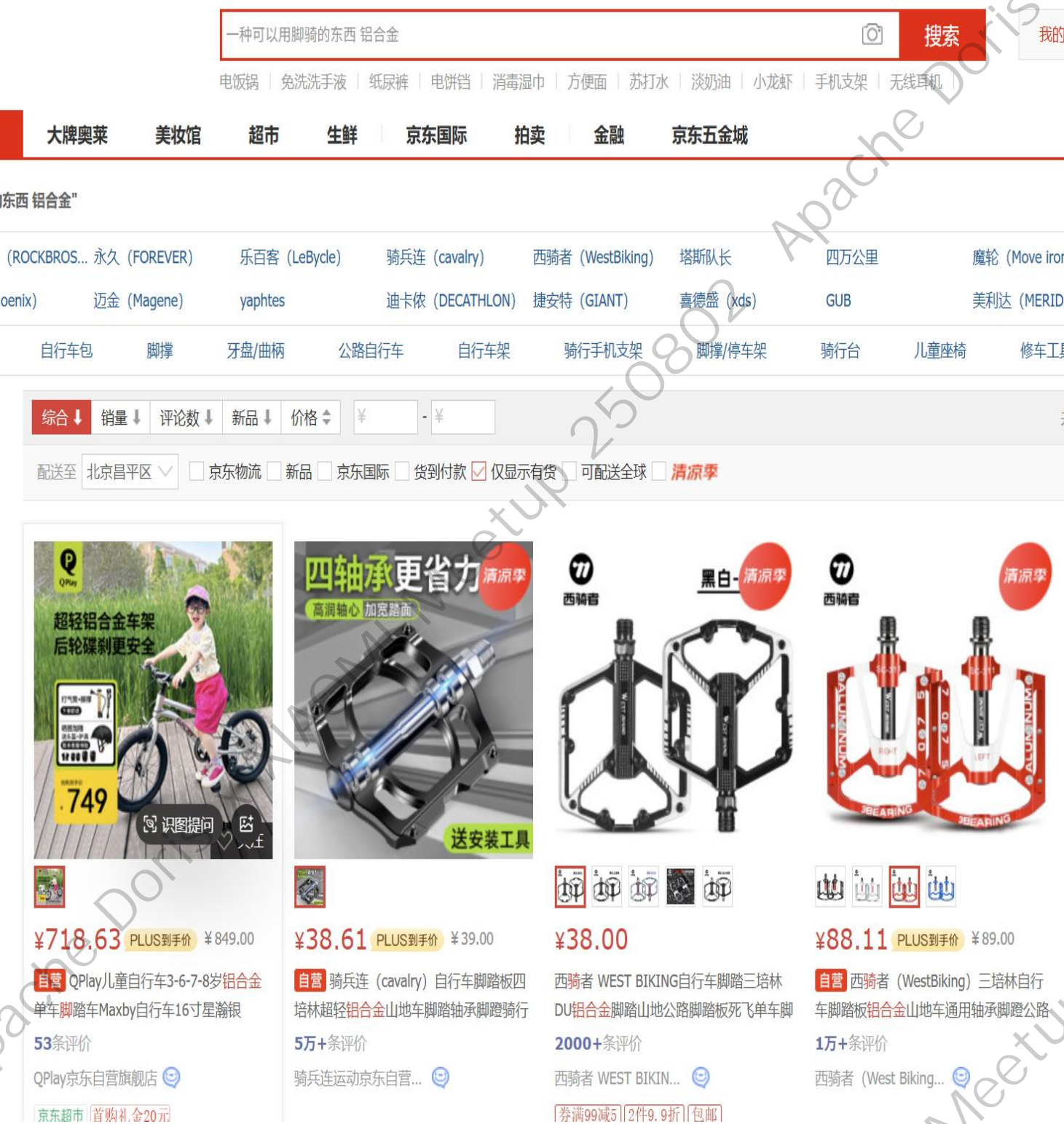
- 大语言模型是基于历史知识进行训练的，它的知识库是定格的。而 Doris 充当向量数据库之后，可以不断的通过向量的方式更新知识



应用场景2 – 电商推荐

建表和查询

优势



```
CREATE TABLE items (  
  `id` datetime(6) NULL,  
  `desc` text NULL,  
  `tag` text NULL,  
  `item_vec` array<float> not NULL)
```

```
SELECT  
  id.... bm25() as score  
FROM items  
WHERE desc match 'xxx' ORDER BY score DESC LIMIT  
50
```

union all

```
SELECT  
  id.... l2_distance() as score  
FROM items  
ORDER BY score DESC LIMIT 50
```

或者是

```
SELECT  
  id.... l2_distance() as score  
FROM items  
WHERE desc match 'xxx' ORDER BY score DESC LIMIT
```

- 当顾客输入模糊查询时，系统会利用关键词匹配和语义分析来展示相关产品。

5

特别致谢

感谢社区的同学开源贡献

- Doris 混合搜索的能力基底由@chenlinzhong、@guozhehui、@lixin 贡献
- 在字节跳动内部 RAG 场景有大面积的推广应用
- 友好的社区合作进行了框架整合，将在 4.0 的 preview 版本和大家正式见面

Thanks !

doris.apache.org
doris-forum.org.cn



活动已结束

关注 SelectDB 公众号解锁更多技术资讯!



解锁更多技术资讯



加入社区