腾讯云开发者社区 | 腾讯云向量数据库 Tencent Cloud VectorDB | Andon

腾讯云向量数据库

RAG七天入门课 第四节

RAG的核心
—— 结果召回和重排序

腾讯云高级算法工程师
赵九州

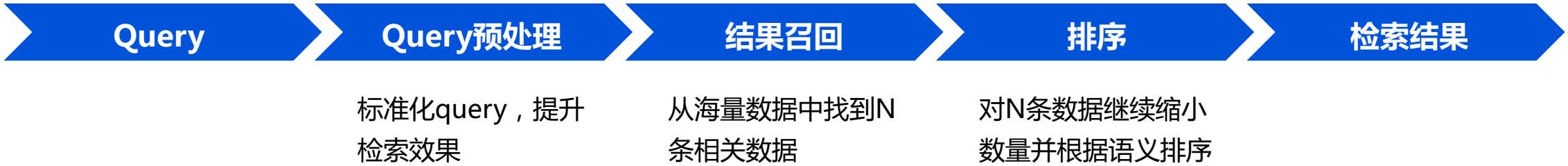
BBTalk

第四节

RAG的核心

- 结果召回和重排序

完整RAG应用的检索流程



Query预处理

1、意图识别

判断query问的是什么类型的问题，从而决定是否走RAG链路

示例1：

深圳有什么好玩的 闲聊问题

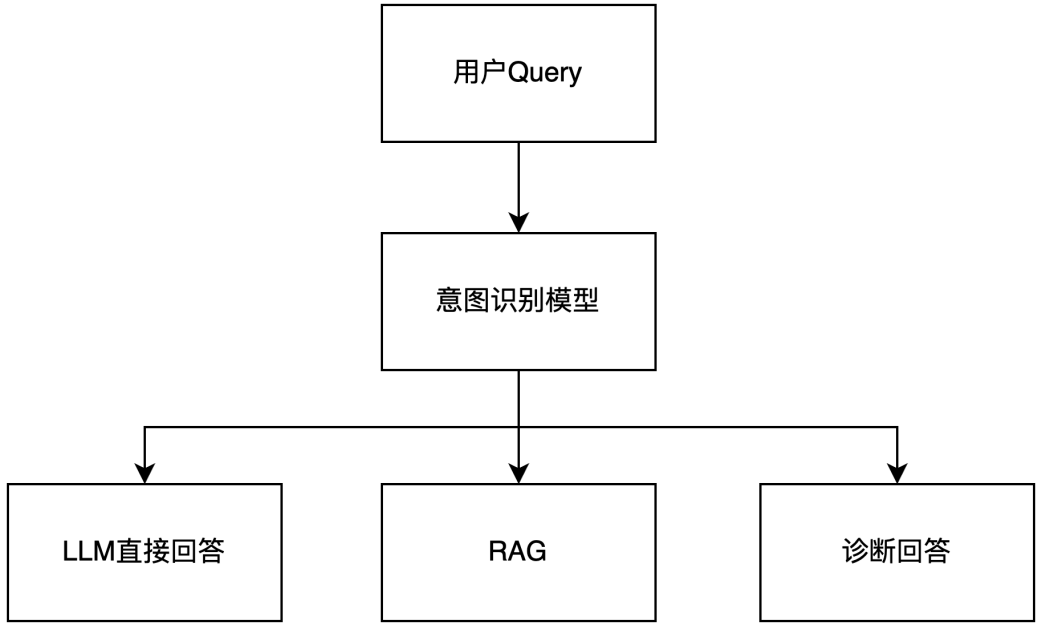
VDB支持哪些检索算法 产品常见问题

示例2：

云Redis如何扩容 产品常见问题

为什么某个MongoDB实例内存占用过高 检查类问题

流程图：



Query预处理

2、生成同义query

针对query生成同义句，不同问法提高召回，检索结果做合并

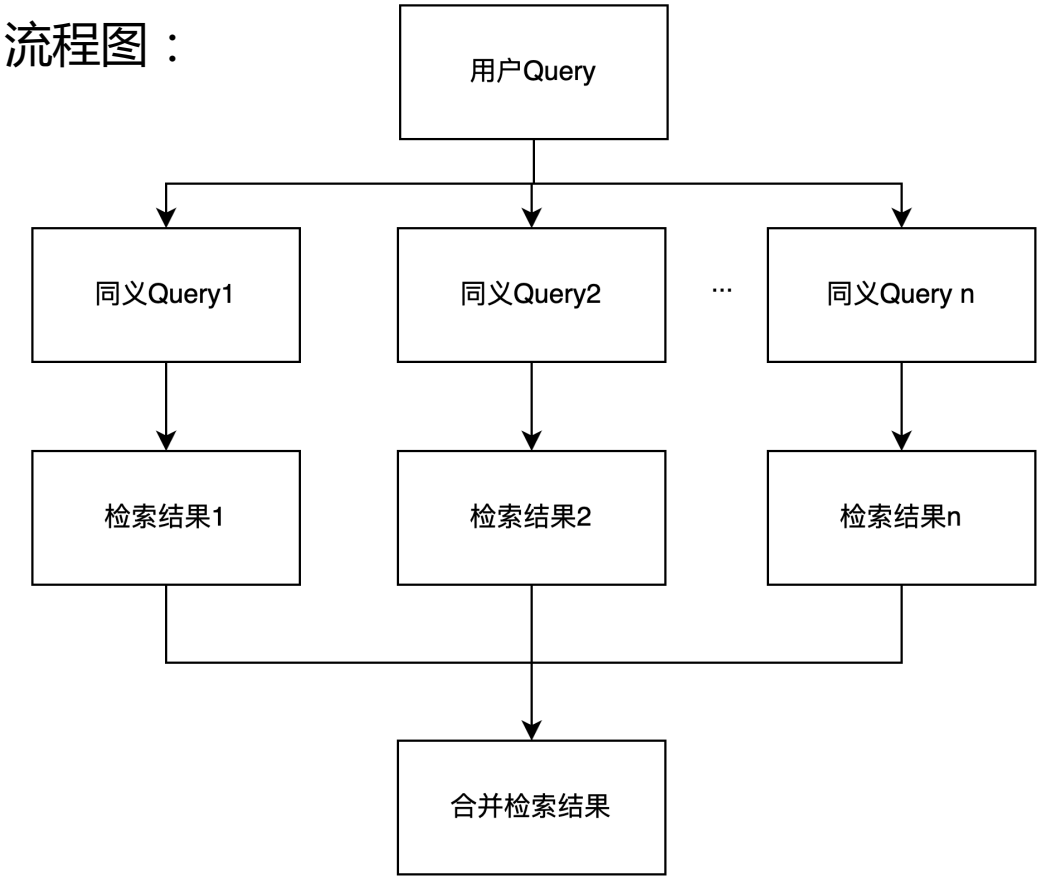
示例1：**VDB支持哪些检索算法**

- VDB有哪些可用的检索算法
- 列举一下VDB所支持的检索算法

示例2：**腾讯云向量数据库的优势是什么**

- 腾讯云向量数据库有哪些主要优点
- 腾讯云向量数据库的核心竞争力是什么

流程图：



Query预处理

3、query标准化

针对query中的专有名词、简写、英文做标准化处理

示例1：VDB支持哪些检索算法

↓

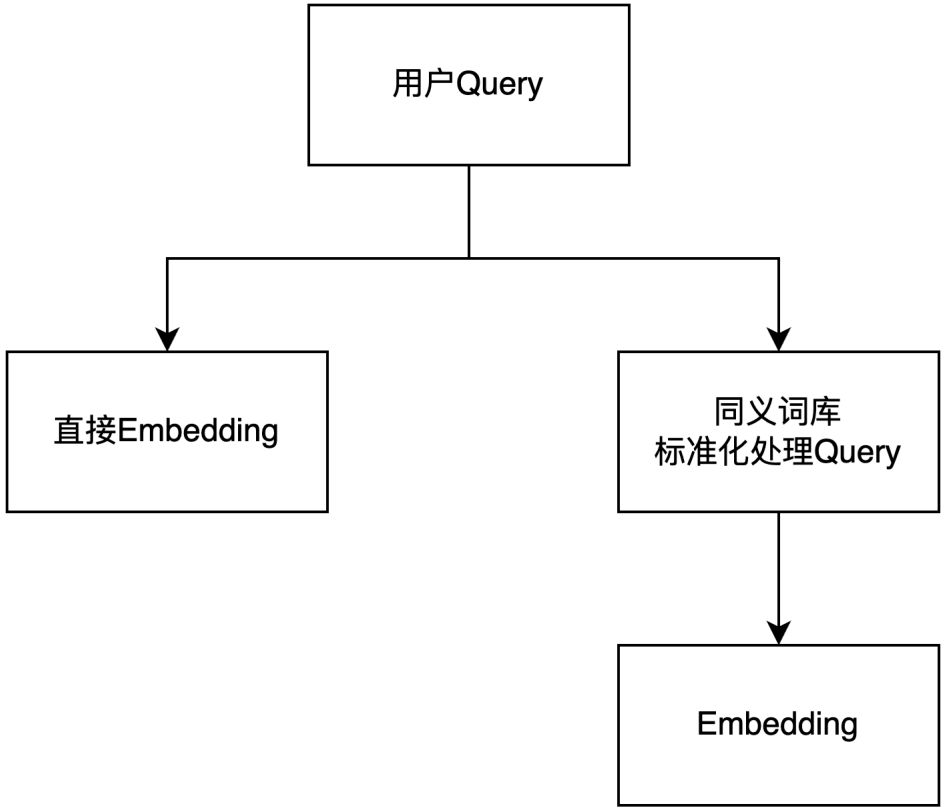
腾讯云向量数据库支持哪些检索算法

示例2：COS如何上传对象

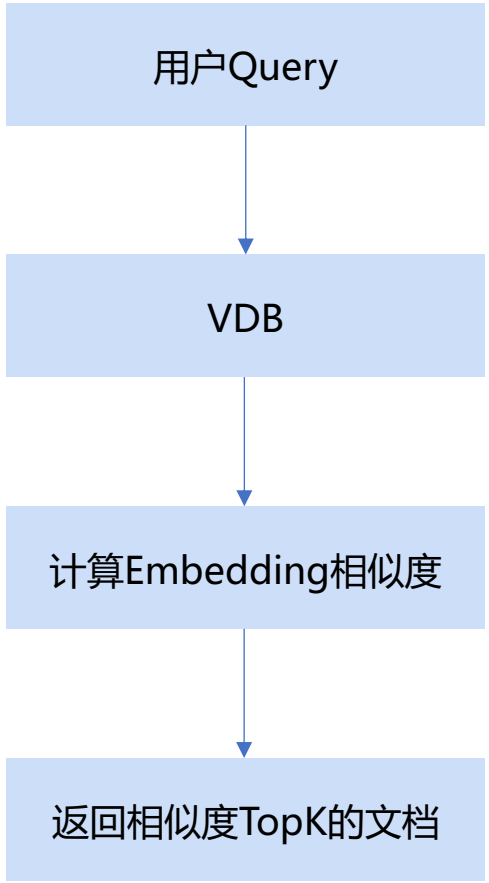
↓

腾讯云对象存储如何上传对象

流程图：



检索召回



VDB提供了多种场景下的相似度计算方法

索引类型	使用场景	适用向量规模	召回率
FLAT	暴力检索，召回率100%，但检索效率低。	10万以内	最高，可保证100%召回率
HNSW	• 基于图算法构建索引，可通过调整检索参数提升召回率。具体信息，请参见 配置索引参数 。 • 检索效率高，但数据量大后写入效率会变低。具体测试数据，请参见性能白皮书的 测试结果 。	10万-1亿	95%+，可根据参数调整
IVF系列	基于聚类算法构建的索引，可通过参数调整召回率，适用于上亿规模的数据集，检索效率高，内存占用低，写入效率高。	1亿以上	95%+，可根据参数调整

What's more?

每次召回时，如何提升结果的排序效果，使与Query更相关的结果更靠前

Query预处理中，做了生成同义Query，最终应该如何合并检索结果？

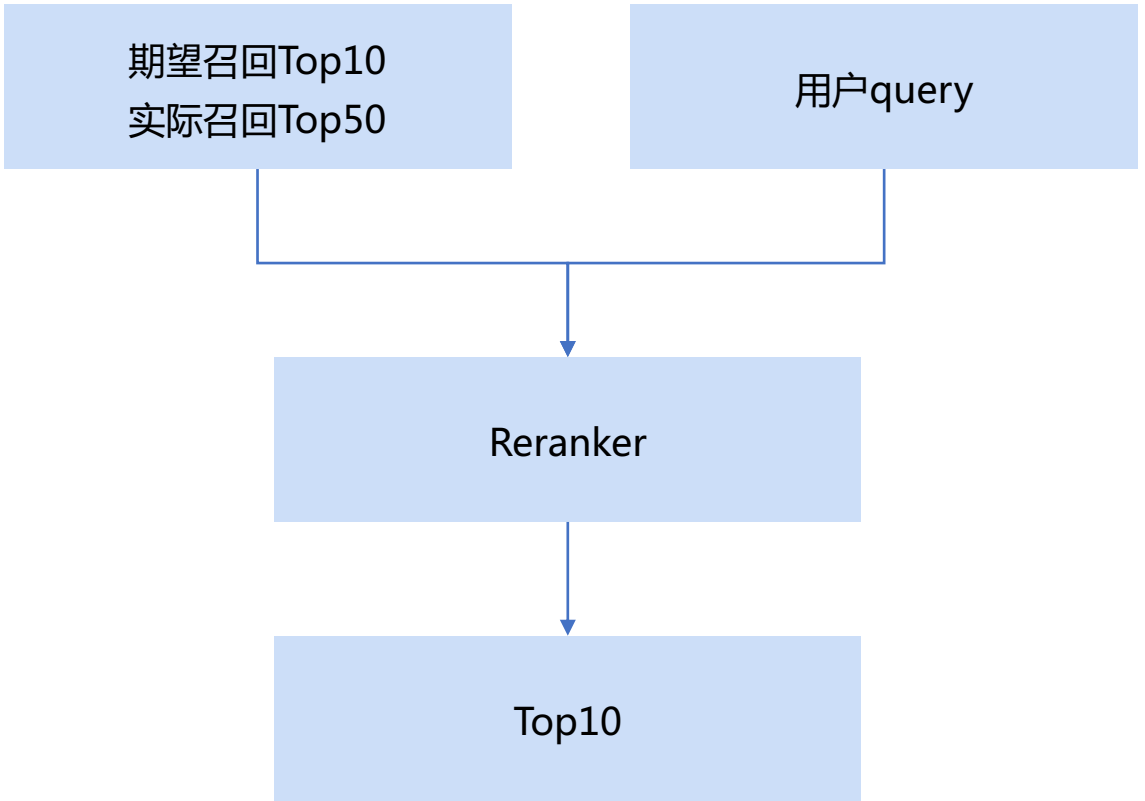
如何在召回阶段，将召回的结果效果做得更优质，减少干扰信息对LLM的影响

为什么要做排序（Rerank）

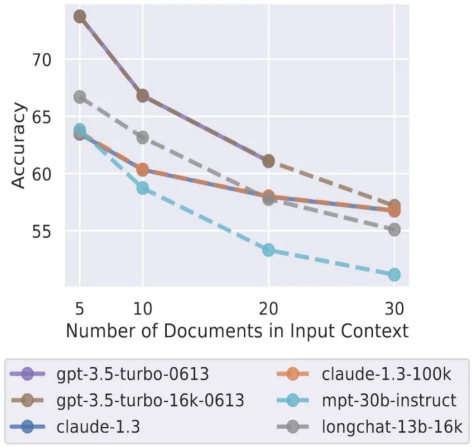
Embedding模型存在一定的局限性

在实际召回结果中，embedding没办法完全反应出语义的相似性，至少这K个文件的排名并不是我们认为的从高分到低分排序的。

Rerank：RAG中百尺竿头更进一步



排序



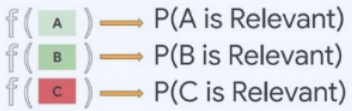
排序模型的目的在于对召回内容有一个更合理的排序结果，
减少提供给模型的上下文长度，长度越长，对模型来说压力越大

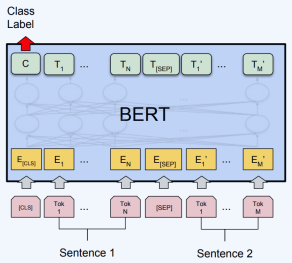
基于Learning2Rank的思路提升文本语义排序效果

1. Pointwise

Documents are considered **independently** of each other

$f \rightarrow$ **score** $\rightarrow$ order $\rightarrow$ metric





2. Pairwise

Look at **a pair of documents** at a time

$f \rightarrow$ **partial order** $\rightarrow$ order $\rightarrow$ metric

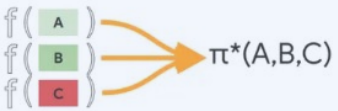
$$\mathcal{L}_{\text{ranking}} = -\log(\sigma(r_\theta(x, y_c) - r_\theta(x, y_r) - m(r)))$$

	Significantly Better	Better	Slightly Better	Negligibly Better / Unsure
Margin Small	1	2/3	1/3	0
Margin Large	3	2	1	0

3. Listwise

Consider the ordering of the entire list

$f \rightarrow$ **order** $\rightarrow$ metric



$$a, b, c \in X$$

$$\left.\begin{matrix} abc, acb \\ bac, bca \\ cab, cba \end{matrix}\right\} \in Y$$

三条文本对应了六种排列方式，一个6分类

Listwise的优化

排序的文本有n条，那么排序结果就是n!种，时间复杂度高

优化方案：ApproxNDCG，把排序指标NDCG作为loss

$$\mathrm{NDCG} = N_n^{-1} \sum_{x \in \mathcal{X}} \frac{2^{r(x)} - 1}{\log_2 (1 + \pi(x))}$$

$$\pi(x) = 1 + \sum_{y \in \mathcal{X}, y \neq x} 1\{s_{x,y} < 0\}$$

$$s_{x,y} = s_x - s_y$$

$\pi(x)$表示排序结果的位置，NDCG的结果不连续无法BP

指示函数做近似

$$\frac{\exp(-\alpha s_{x,y})}{1 + \exp(-\alpha s_{x,y})}$$

document	s_x	$\pi(x)$	$\hat{\pi}(x)$ ($\alpha = 100$)
x_1	4.20074	2	2.00118
x_2	3.12378	4	4.00000
x_3	4.40918	1	1.00000
x_4	1.55258	5	5.00000
x_5	4.13330	3	2.99882

最终loss function

$$\hat{\pi}(x) = 1 + \sum_{y \in \mathcal{X}, y \neq x} \frac{\exp(-\alpha s_{x,y})}{1 + \exp(-\alpha s_{x,y})}$$

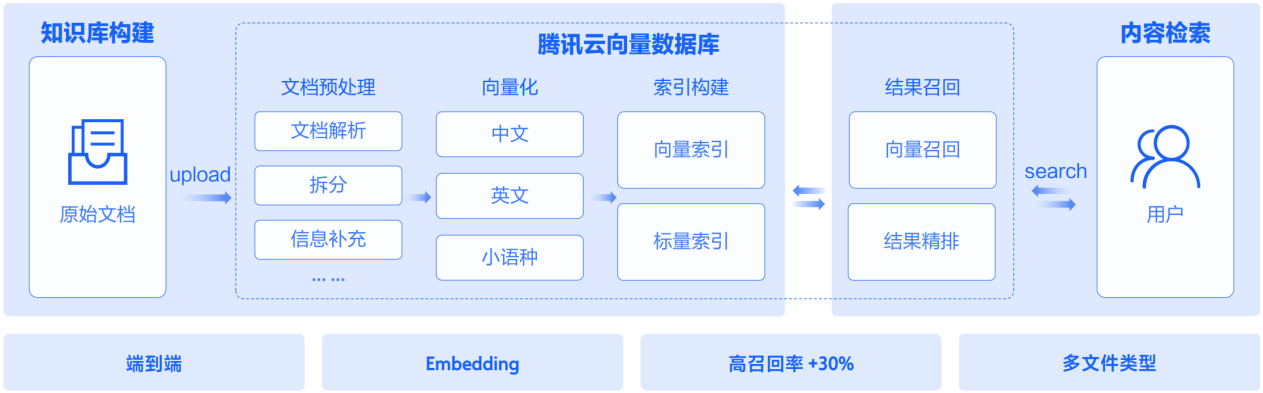
$$\widehat{\mathrm{NDCG}} = N_n^{-1} \sum_{x \in \mathcal{X}} \frac{2^{r(x)} - 1}{\log_2 (1 + \hat{\pi}(x))}$$

效果对比	model	FAQ ACC@5	文档 ACC@5	混合数据 ACC@5
	bge-reranker-large(开源SOTA)	90.26	75.98	77.17
	Our Model	93.76	83.20	81.21

腾讯云向量数据库：消除大模型幻觉，加速大模型在企业落地

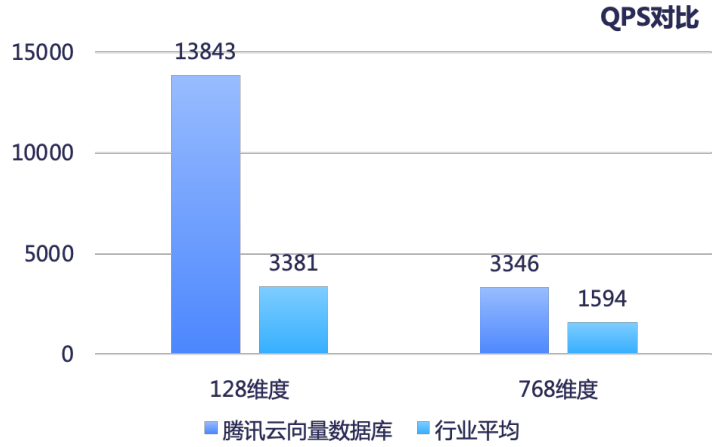
端到端AI套件，AGI时代的知识库解决方案

提供**一站式**知识检索方案，实现业界内**最高召回率、大幅降低开发门槛**，帮助企业快速搭建RAG应用，解决大模型幻觉问题



源自集团多年积累，产品能力行业领先

源自腾讯自研向量检索引擎OLAMA，集团内部**40+**业务线上使用，日均处理**1600亿次**检索请求



首家

『首家』通过中国信通院向量数据库标准测试

千亿

单索引支持最高**千亿级**超大数据规模

500万

单实例最高可达**500万 QPS**