

2024.4.10 带香豆

RAG: 一种结合检索和生成的技术, 旨在通过利用外部知识库来增强大语言模型的性能, 它通过检索与用户输入相关的信息片段, 并结合这些信息来生成更准确、更丰富的回答。

OpenMMLab Bilibili

## RAG 技术概述

### 定义

RAG (Retrieval Augmented Generation) 是一种结合了检索 (Retrieval) 和生成 (Generation) 的技术, 旨在通过利用外部知识库来增强大型语言模型 (LLMs) 的性能。它通过检索与用户输入相关的信息片段, 并结合这些信息来生成更准确、更丰富的回答。



解决 LLMs 在处理知识密集型任务时可能遇到的挑战。提供更准确的回答、降低成本、实现外部记忆。



- 生成幻觉 (hallucination)
- 过时知识
- 缺乏透明和可追溯的推理过程

应用



问答系统



文本生成



信息检索



图片描述

应用领域

问答系统、文本生成、信息检索、  
图片描述

工作原理：

索引 Indexing：将知识源，如文档，分割成 chunk，编码成向量，并存储在数据库中。



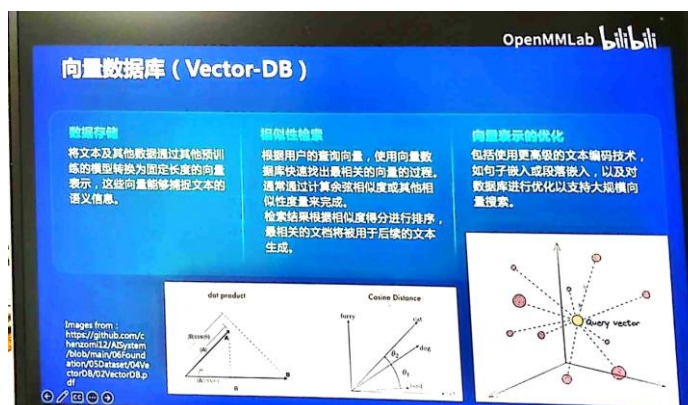
检索 Retrieval：接收到用户的问题后，也编码成向量，并在向量数据库中找到与之最相关的文档块 (top-k chunks)



生成 Generation：将检索到的文档块与原始问题

一起作为提示 (prompt)，输入到 LLM 中，生成最终的答案

数据库向量：长度固定

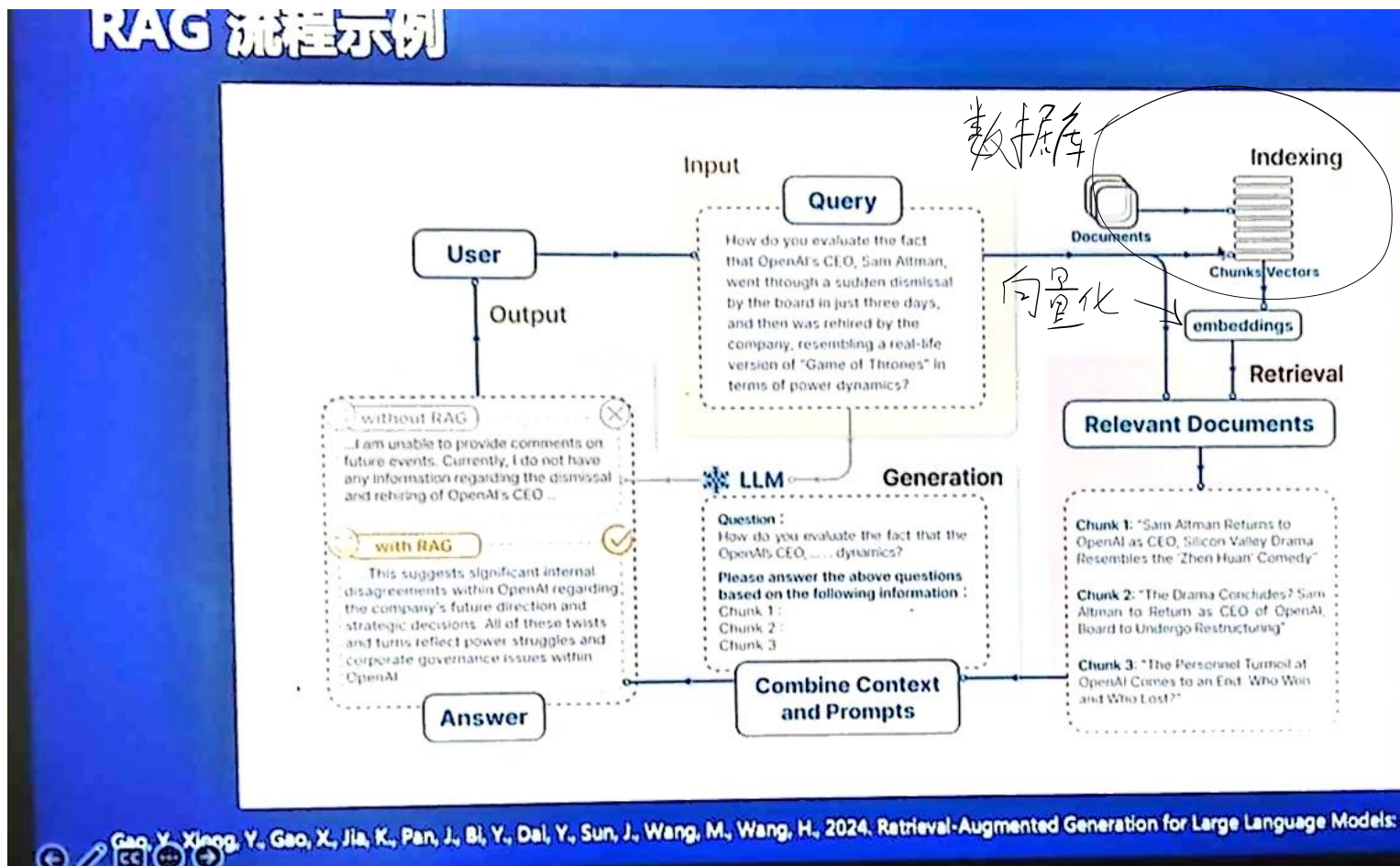


高效的相似性检索

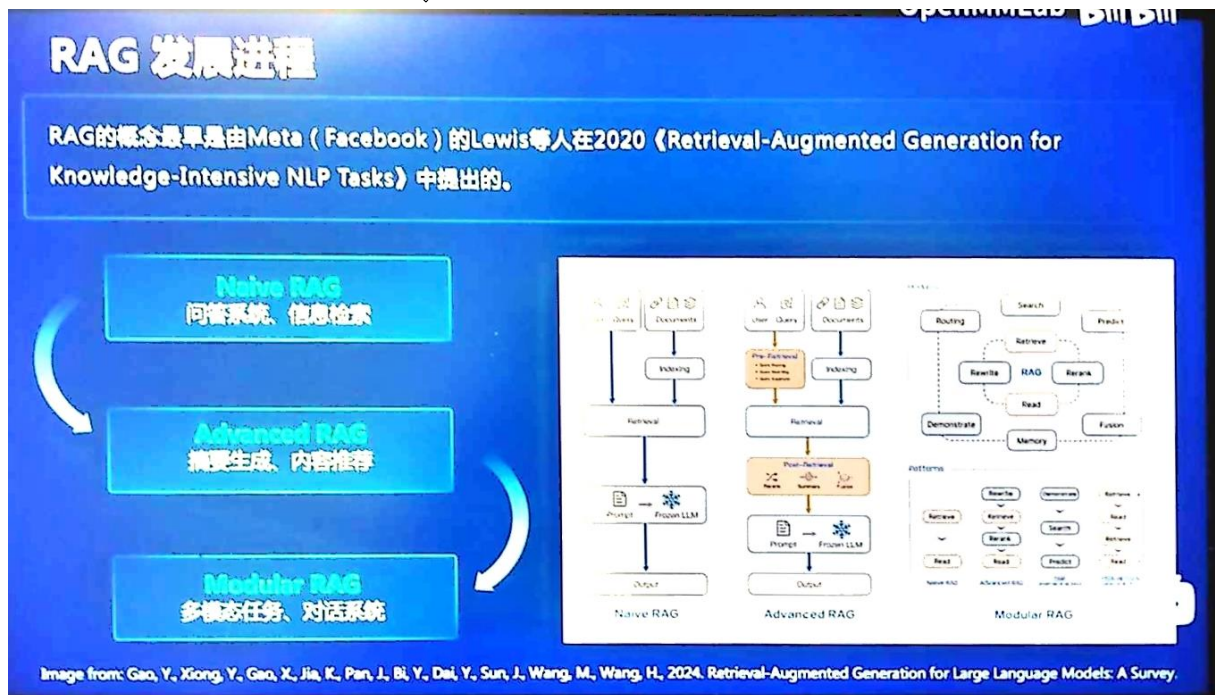
向量表示的优化：更高级的文本编码技术，更好的预训练模型

↓  
影响 RAG 的好坏

RAG 流程：



RAG 三种范式：



RAG：优化：

提DB质量

① 嵌入优化

② 索引优化

1) 结合稀疏编码器和  
密集检索器

1) 细粒度分割

2) 多任务

1) 添加元数据

③ 查询优化：

④ 上下文管理

1) 查询扩展、转换

1) 重排

2) 多查询

2) 上下文选择/压缩



## RAG 常见优化方法

### 嵌入优化 Embedding Optimization

- ✓ 结合稀疏和密集检索
- ✓ 多任务

### 索引优化 Indexing Optimization

- ✓ 细粒度分割 (Chunk)
- ✓ 元数据

### 查询优化 Query Optimization

- ✓ 查询扩展、转换
- ✓ 多查询

### 上下文管理 Context Curation

- ✓ 重排 (rerank)
- ✓ 上下文选择/压缩

### 迭代检索 Iterative Retrieval

- ✓ 根据初始查询和迄今为止生成的文本进行重复搜索

### 递归检索 Recursive Retrieval

- ✓ 迭代细化搜索查询
- ✓ 链式推理 (Chain-of-Thought) 指导检索过程

### 自适应检索 Adaptive Retrieval

- ✓ Flare, Self-RAG
- ✓ 使用LLMs主动决定检索的最佳时机和内容

### LLM微调 LLM Fine-tuning

- ✓ 检索微调
- ✓ 生成微调
- ✓ 双重微调

