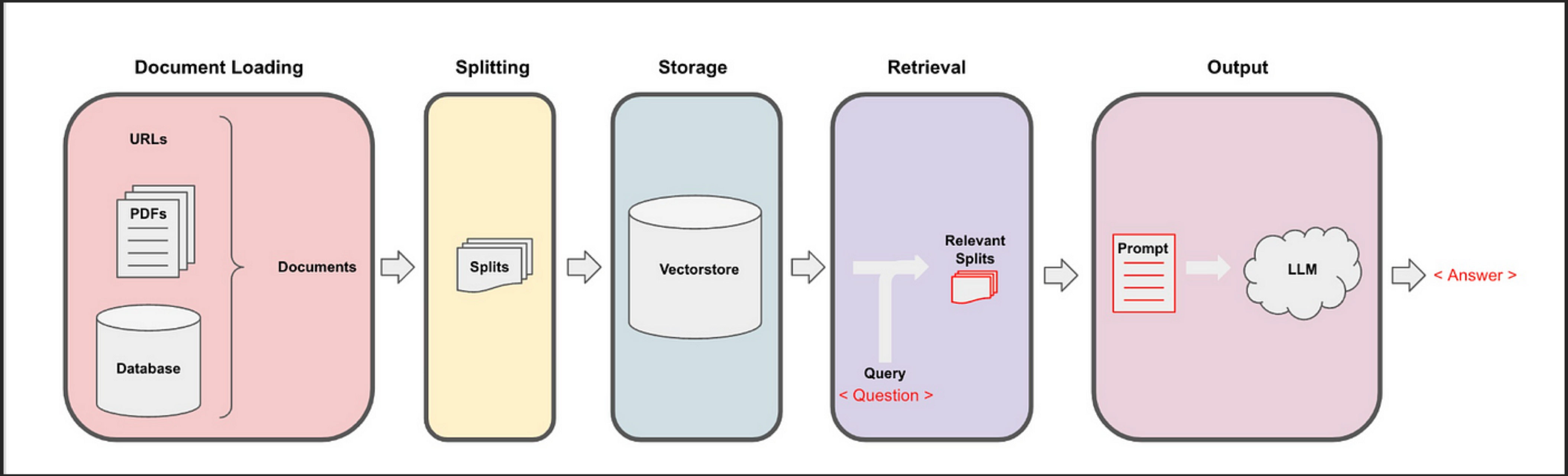
如何提升RAG应用的性能
？？？

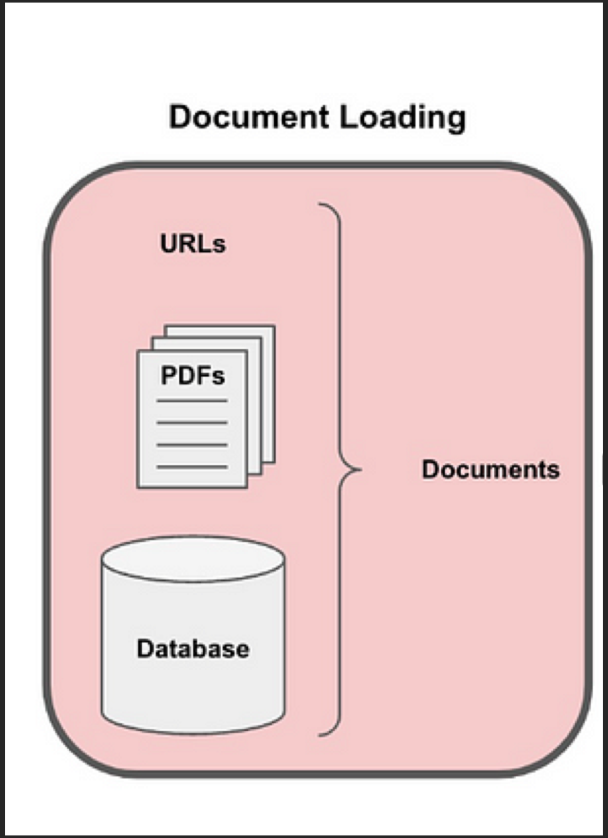
创建一个简单的RAG原型应用　　难度系数：★ ★

实现生产级或让客户满意的RAG　　难度系数：★ ★ ★ ★ ★

Retrieval Augmented Generation Pipeline检索增强生成管道（RAG Pipeline ）



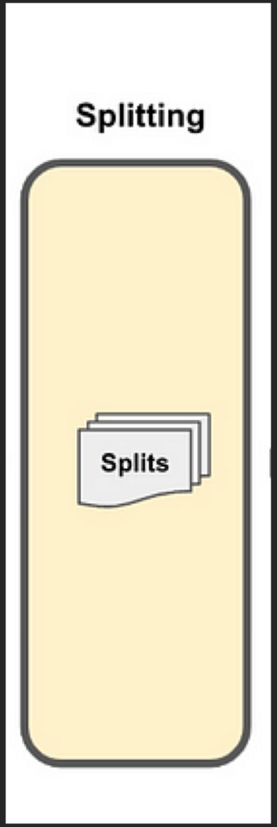
提升: 指的是可以用于RAG管道中的任何环节



清理数据-保证内容和形式逻辑清晰

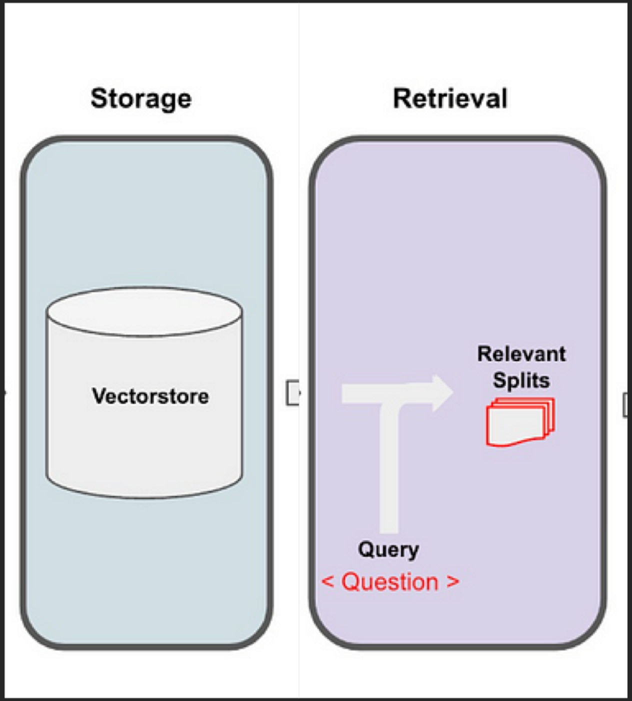
信息是否冗余和冲突
文件是否按逻辑分开
涉及的主题是集中在一处还是分散在各处
是否有重复的文件

文章推荐：
[改进您的 RAG Pipeline]

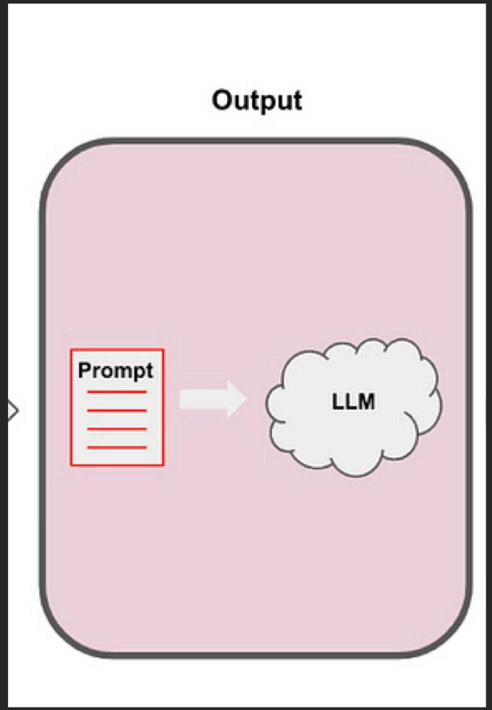


1. **调chunk size, top_k, chunk overlap**
2. **尝试不同的分块策略**

文章推荐：
为什么你的 RAG 在生产环境中不可靠？
LLM应用的分块策略
通过 LangChain 尝试不同的分块策略
利用 LLMs优化 RAG：探索分块技术和重新排名以提高性能
按上下文分割文件
获取详细答案的最佳分块策略。



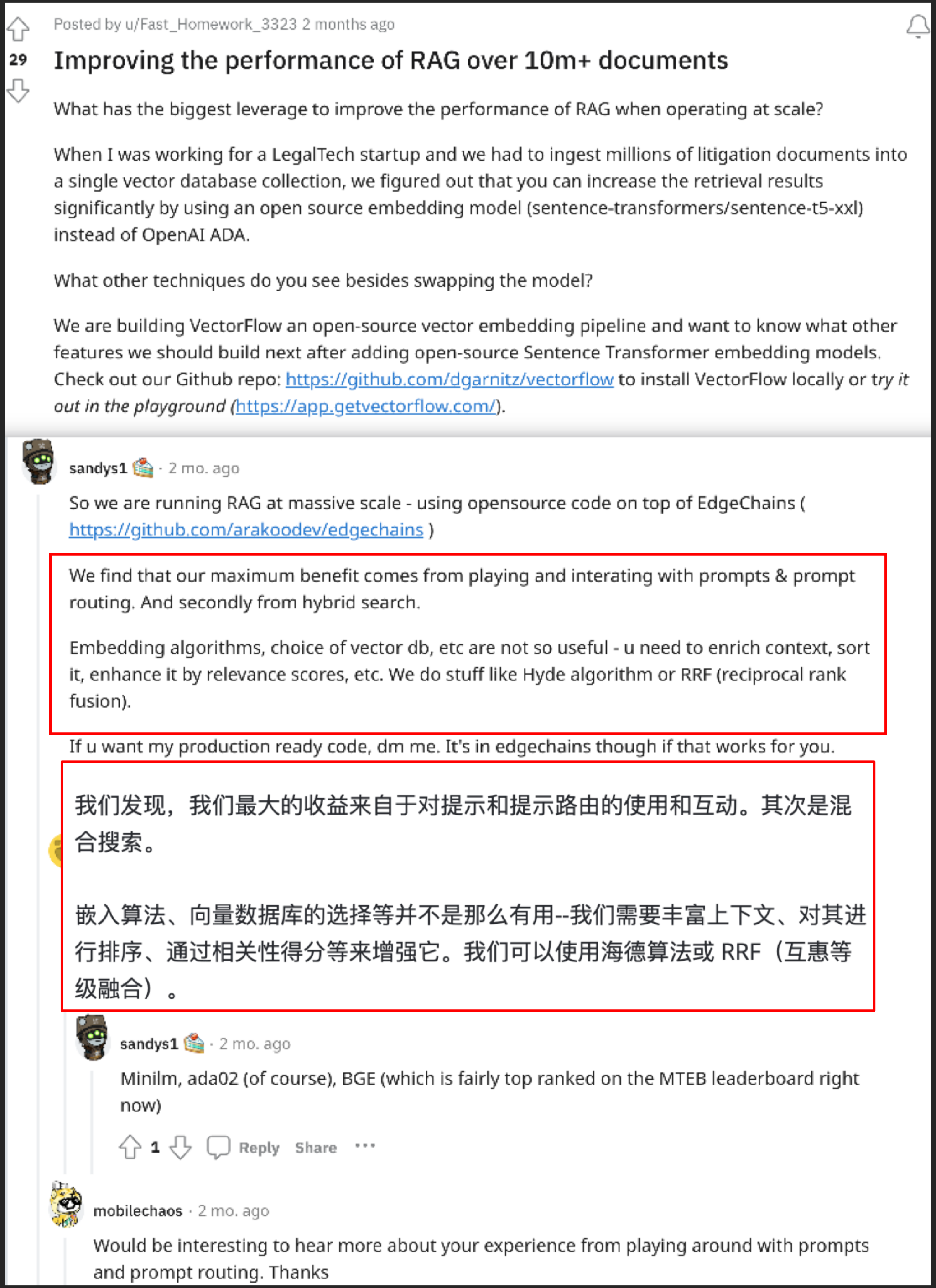
尝试不同的索引类型，如LlamaIndex 和 LangChain
尝试不同的vector store，常见的有Faiss, Chroma, Pinecone以及Weaviate
尝试不同的检索器，如LangChain提供的混合搜索Ensemble Retriever，HyDE: **Improve document indexing with HyDE**
尝试不同的Embedding，如从OpenAIEmbedding换成HuggingFace的BGE



LLM部分

更改问题表达
迭代提示
尝试不同LLM

**不同API所需模板不一样，可以优先参考官方给的提示模板


Posted by u/Fast_Homework_3323 2 months ago

29

Improving the performance of RAG over 10m+ documents

What has the biggest leverage to improve the performance of RAG when operating at scale?

When I was working for a LegalTech startup and we had to ingest millions of litigation documents into a single vector database collection, we figured out that you can increase the retrieval results significantly by using an open source embedding model (sentence-transformers/sentence-t5-xxl) instead of OpenAI ADA.

What other techniques do you see besides swapping the model?

We are building VectorFlow an open-source vector embedding pipeline and want to know what other features we should build next after adding open-source Sentence Transformer embedding models. Check out our Github repo: https://github.com/dgarnitz/vectorflow to install VectorFlow locally or *try it out in the playground* (https://app.getvectorflow.com/).

sandys1 · 2 mo. ago

So we are running RAG at massive scale - using opensource code on top of EdgeChains (https://github.com/arakoodev/edgechains)

We find that our maximum benefit comes from playing and interating with prompts & prompt routing. And secondly from hybrid search.

Embedding algorithms, choice of vector db, etc are not so useful - u need to enrich context, sort it, enhance it by relevance scores, etc. We do stuff like Hyde algorithm or RRF (reciprocal rank fusion).

If u want my production ready code, dm me. It's in edgechains though if that works for you.

我们发现，我们最大的收益来自于对提示和提示路由的使用和互动。其次是混合搜索。

嵌入算法、向量数据库的选择等并不是那么有用--我们需要丰富上下文、对其进行排序、通过相关性得分等来增强它。我们可以使用海德算法或 RRF（互惠等级融合）。

sandys1 · 2 mo. ago

Minilm, ada02 (of course), BGE (which is fairly top ranked on the MTEB leaderboard right now)

1 Reply Share ···

mobilechaos · 2 mo. ago

Would be interesting to hear more about your experience from playing around with prompts and prompt routing. Thanks

我的经验（基于我们自己的用例）

1. 按照上下文的逻辑进行分块可以明显提升检索器的准确率
2. LangChain的混合搜索Ensemble Retriever (BM25+Faiss)性能比单纯使用Faiss相似性搜索的性能更好， HyDE和上下文压缩等性能不如Faiss的相似性搜索。
3. 对于Faiss的搜索类型，相似性搜索优于MMR, Similarity score threshold
4. 就Emedding实验来看，BGE优于Amazon Titan， OpenAI ADA02.