embedding model

RAG本身的原理并不难理解，但要将其推广到生产环境中则会面临多方面的挑战。这主要是因为 RAG 系统涉及多个不同的组件，每个组件都需要精心设计和优化。本文讨论的就是其中embedding的模块，也就是相关doc召回时的粗排阶段。

在本文中，将主要讨论以下几个问题：

- embedding模型的架构
- embedding模型的评测基准 MTEB
- 如何选择合适的embedding模型

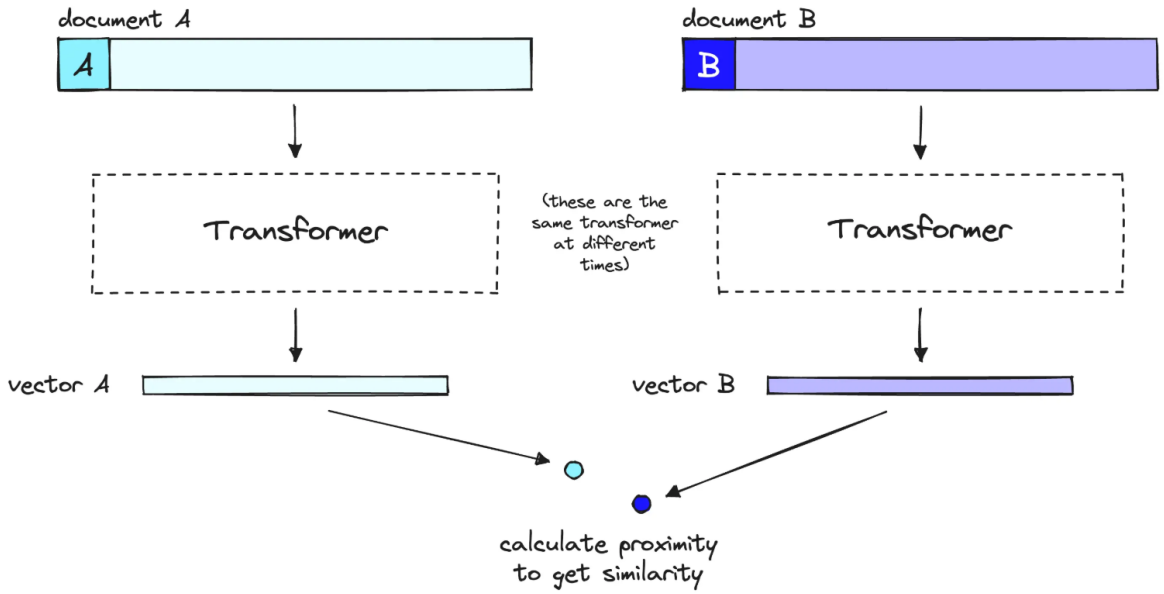
embedding模型的架构

在RAG框架中，常见的两种用于doc召回的embedding模型是双编码器（Bi-Encoder）和稀疏嵌入模型（Sparse Embedding Models）。

双编码器（Bi-Encoder）

双编码器的基本思想是使用两个独立的encoder来分别处理query和doc（或候选doc），然后将它们嵌入到相同的向量空间中。在检索阶段，query和doc会被转化为固定长度的向量表示，然后通过计算query向量和doc向量之间的相似度来进行匹配。

- **工作方式**：query和doc分别通过两个相同的encoder处理，每个编码器将输入转化为一个embedding。这两个embedding向量在同一个向量空间中表示它们的语义信息，之后根据相似度（例如余弦相似度）来判断查询与文档之间的相关性。
- **优点**：这种方法的优势在于它具有较高的计算效率，因为查询和文档的编码是独立进行的，适合用于大规模数据集。通常，使用双编码器进行检索时，检索过程会非常快速。



稀疏嵌入模型（Sparse Embedding Model）

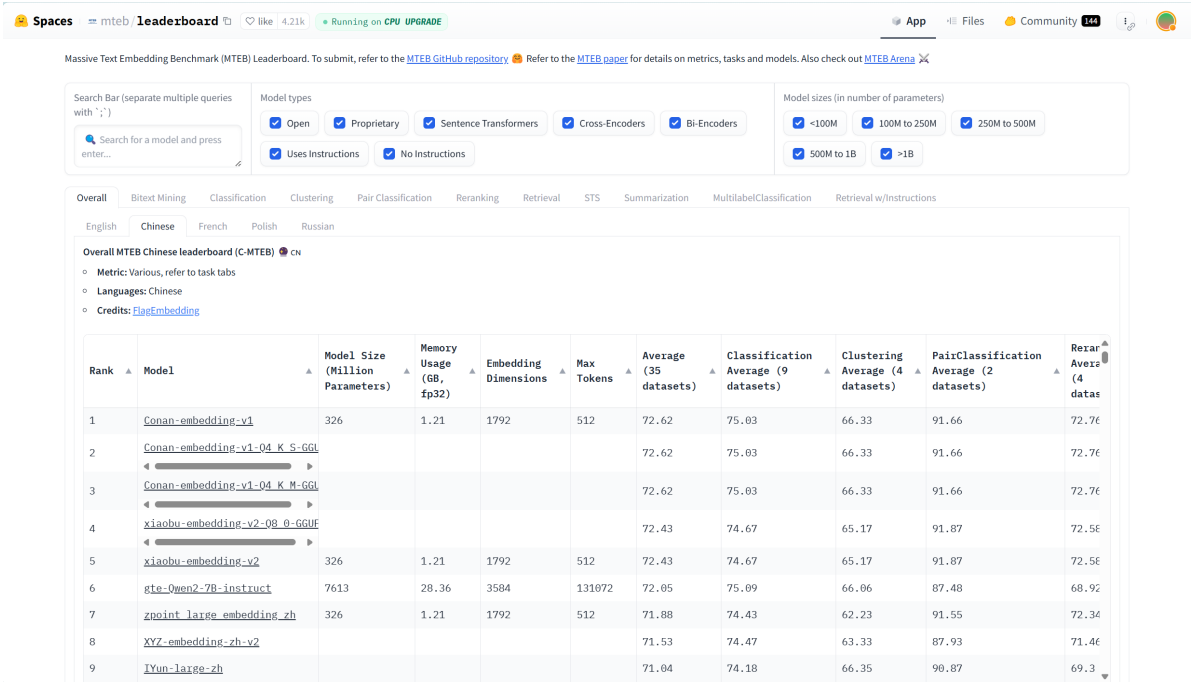
稀疏嵌入模型则是一种不同于密集嵌入（dense embedding）的模型，通常基于传统的词袋模型（如TF-IDF）或稀疏编码技术。这些模型生成的嵌入是稀疏的，意味着嵌入向量中大多数元素的值是零，仅有少量非零元素。

- **工作方式**：在稀疏嵌入模型中，文本的表示通常不是通过密集的向量（如BERT生成的嵌入向量）来表示，而是通过一种稀疏表示，其中很多维度的值为零，只在少数维度上有较高的值。这种稀疏表示通常是通过词频或其他特征的权重计算得到的，常见的实现包括基于词频的向量化方法（如TF-IDF、bm25）和一些稀疏编码方法（如LDA等）。
- **优点**：稀疏嵌入模型往往计算效率较高，并且可以避免高维密集向量所带来的计算开销，特别是在大型文档库的检索中。此外，稀疏表示有时能捕捉到更加显著的词汇特征，适用于特定的检索任务，如关键词匹配等。

embedding模型的评测基准 MTEB

MTEB（Massive Text Embedding Benchmark）是衡量文本嵌入模型（Embedding模型）的评估指标的合集，是目前业内评测文本向量模型性能的重要参考。

可以在huggingface上找到对应的leaderboard：



C-MTEB(Chinese Massive Text Embedding Benchmark)则是专门针对中文文本向量的评测基准，被认为是目前业界最全面、最权威的中文语义向量评测基准之一，涵盖了分类、聚类、检索、排序、文本相似度、STS等7个经典任务，共计35个数据集，为深度测试中文语义向量的全面性和可靠性提供了可靠的实验平台。

对于国内开发者而言，我们更加会专注C-MTEB。不过但也只能作为一个参考，这些模型在公开数据集上的 benchmark 在垂直领域、企业自身的业务领域不一定成立，具体选择哪个向量模型还需结合业务特点进行综合比较、权衡。

如何选择合适的embedding模型

可以从以下几个角度考虑：

1. **语言支持和性能**：大部分开源向量模型只支持单一或者有限的文本语言，所以需要确保 Embedding 模型支持的语言种类。多语言模型如 OpenAI Embedding 和 bge-m3 等模型能够处理多种语言。bge-m3 支持 100 多种语言，适合多语言需求的场景。另外，某些模型在主要语言（如中文）中的表现较好，但在处理较少使用的语言时可能会表现不佳。因此，需要评估模型在所有必需语言中的准确性，以确保一致的性能。
2. **处理长文本的能力**：切分的文本片段后续需要通过 Embedding 模型进行向量化，所以必须考虑向量模型对输入文本块的 tokens 长度限制，超出这个限制则会导致模型对文本进行截断，从而丢失

信息，影响下游任务的性能。不同的 Embedding 模型对文本块长度的支持能力不同。比如，BERT 及其变体通常支持最多 512 个tokens，处理长文本时则需要将文本分成更小的块，意味着需要更加精细化的分块策略。而 Jina AI 的 Embedding 模型和 bge-m3 模型则支持 8K 的 tokens 输入，适合处理长文本块。

3. **模型在特定领域的表现**：通用 Embedding 模型在特定垂直领域（如医学、法律和金融等）可能不如专用模型有效。这些领域通常需要专门训练 Embedding 模型来捕捉特定的专业术语和语境。为特定业务需求优化的 Embedding 模型能够显著提升检索和生成的质量。例如，通过结合向量检索和重排序（reranking）技术，可以进一步优化结果。
4. **存储和内存等资源需求**：高维向量需要更多的存储空间，这可能会带来长期成本。例如，较高维度的模型如 text-embedding-ada-002 需要更多的存储资源。另外，较大的模型可能会占用更多内存，因此不适合内存有限的设备。
5. **模型响应时间**：Embedding 模型的处理速度在实时应用中尤为关键。例如，intfloat/e5-base-v2 模型在处理速度上表现优异，但需要在 GPU上 运行以达到最佳性能。在选择模型时，需要评估其在嵌入和检索过程中的延迟。例如，OpenAI 的 Embedding 模型在许多基准测试中显示出较高的性能和较低的延迟。

通用的 Embedding 模型通常是在大规模、多样化的数据集上训练的，可能不完全适合特定领域的任务，比如医学、法律等专业领域，它们无法很好的理解一些专有词汇。如果模型在业务数据集上表现不能满足预期，可以通过微调，让模型学习到特定领域的词汇和概念，使其在特定应用场景中表现更佳。

你了解哪些embedding模型

bge

BGE，全称BAAI General Embedding，是智源研究院提出的开源通用向量模型，在过去短短一年时间内，在huggingface上总下载量已超数亿次，是目前下载量最多的国产AI系列模型。

Conan-Embedding

最近在C_MTEB霸榜的embedding模型，该工作来自腾讯。论文：Conan-embedding: General Text Embedding with More and Better Negative Samples