# 检索器的应用

在高维空间中，距离型向量数据库检索工具通过将查询嵌入来找出相似的文档。然而，微妙的查询词汇改变或者嵌入不能很好地捕捉数据语义时，检索结果可能会有所不同。人们有时会进行提示工程或者调优来手动解决这些问题，但这样做可能很繁琐。

多查询检索器（MultiQueryRetriever）自动化了提示调优的过程，它使用大语言模型（LLM）从不同角度为给定的用户输入查询生成多个查询。对于每个查询，它都会检索一组相关文档，并取所有查询的唯一并集，以得到一组可能相关的更大的文档集。通过从多个角度生成对同一个问题的视角，多查询检索器可能能够克服基于距离的检索的一些局限性，得到更丰富的结果集。

当然，检索过程也面临一些挑战。通常在将数据录入系统时，你无法知道文档存储系统将面临的具体查询。这意味着与查询最相关的信息可能被埋在大量无关文本的文档中。将完整的文档传递给应用程序可能会导致更昂贵的LLM调用和较差的响应。

上下文压缩（Contextual compression）就是为了解决这个问题。其思想很简单：不是立即原样返回检索到的文档，而是可以使用给定查询的上下文来压缩它们，这样只有相关的信息会被返回。这里的“压缩”既指压缩单个文档的内容，也指整体过滤掉文档。

此外，我们还有自我查询检索器（Self-querying retriever），它能够自我查询。特别地，给定任何自然语言查询，检索器使用查询构造的LLM链来编写结构化查询，然后将该结构化查询应用到其底层的VectorStore。这使得检索器不仅可以使用用户输入的查询来与存储文档的内容进行语义相似性比较，还可以从用户查询中提取有关存储文档元数据的过滤器并执行这些过滤器。

然后是时间加权向量存储检索器（Time-weighted vector store retriever），它使用语义相似性和时间衰减的组合。

最后，我们还有向量存储支持的检索器（Vector store-backed retriever），它是一种使用向量存储来检索文档的检索器。它是围绕Vector Store类的轻量级封装，使其符合检索器接口。它使用由向量存储实现的搜索方法，如相似性搜索和MMR，来查询向量存储中的文本。

以上这些检索器都有各自独特的使用场景和优势，能够有效应对各种复杂的信息检索任务。