构建 RAG 应用程序：入门

备忘单：使用 LlamaIndex 构建 RAG 应用程序

LlamaIndex 和 LangChain 一样，是一个用于 LLM 驱动的上下文增强框架。典型的用例包括聊天机器人、检索增强生成（RAG）应用程序和各种类型的 AI 助手。

- LlamaIndex 的 Document 类封装或存储来自文档存储的整个文档。除了存储文档的文本外，Document 类还可以存储嵌入、文档的元数据（例如文档创建时间或来源目录）以及与其他文档的关系。
- 可以使用 `SimpleDocumentReader` 加载器加载各种文档，该加载器可以加载单个文件或整个目录。此外，LlamaHub.ai 提供了额外的加载器和连接器。
- 文档使用文档分割器进一步分块为 `Nodes`。LlamaIndex 的 `Node` 在结构上类似于 LlamaIndex 的 `Document`，也存储元数据、关系和嵌入。
- LlamaIndex 提供了多种文本分割器，例如 `SentenceSplitter`，它递归地根据列表中的字符分割文档，同时确保每个块不超过最大令牌长度。此外，除了提供本地文本分割器外，LlamaIndex 还提供了一个 `LangChainNodeParser` 类，能够包装并使用任何 LangChain 文本分割器。
- 在 LlamaIndex 中，嵌入是在一步中生成并存储的。这通常使用 LlamaIndex 的 `VectorStoreIndex` 类完成。`VectorStoreIndex` 类默认情况下将节点存储在内存中，但也可以用于包装外部向量数据库，例如 Chroma DB、FAISS 或 Milvus。
- 嵌入用户的提示并检索相关块通常在 LlamaIndex 中一步完成。这是通过从 `VectorStoreIndex` 实例创建的检索器实现的，方法是调用 `as_retriever()` 方法。使用从向量存储索引生成的检索器确保了用于嵌入节点的相同嵌入模型也用于嵌入用户的提示。
- LlamaIndex 提供了额外的高级检索器。
- 提示增强以及查询 LLM 以获得响应在 LlamaIndex 中发生在同一步骤中。
  - 如果已经检索到相关节点，可以通过使用“响应合成器”来完成，该合成器以用户的原始提示和检索到的节点作为输入，并生成 LLM 的响应作为输出。
  - 另外，用户不必将检索到的节点作为中间步骤存储。在这种情况下，使用 `as_query_engine()` 方法从向量存储索引实例创建的“查询引擎”将用户的原始提示作为输入，并输出 LLM 的响应。
  - LlamaIndex 的提示增强由可自定义的提示模板控制。提示模板是预定义的文本，具有用户原始提示和检索文本块的占位符。
  - LlamaIndex 因其简单性、开发便捷性以及强大的本地工具而表现优异。另一方面，LlamaIndex 经常与之进行比较的 LangChain 具有更多的外部集成、更模块化的设计以及其各种组件的更易定制性。

作者

Wojciech "Victor" Fulmyk

更新日志

日期	版本	更改者	更改描述
2025-04-15	0.1	P.Kravitz	初始版本完成。SME 内容转换为 markdown 格式
2025-04-15	0.2	Leah Hanson	QA 审核