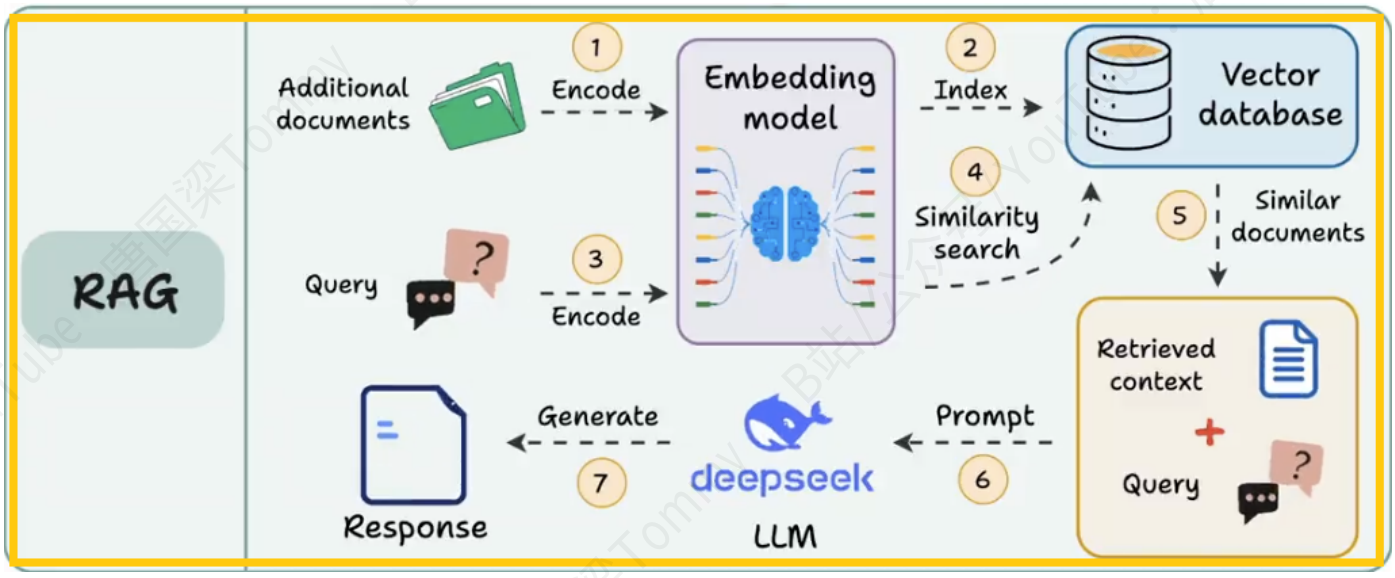
特征	LLM（大模型）	LLM With RAG
准确性	容易生成未经证实或幻觉内容，不依赖可靠来源	响应基于外部来源的可验证引用
时效性	依赖静态的预训练数据，可能过时或与当前事件无关	通过整合来自外部源的实时、最新信息增强静态预训练数据
上下文清晰度	经常难以解释歧义查询，导致模糊或不完整的答案	检索特定上下文的信息，提高响应的清晰度
定制化	无法访问或利用用户特定数据集或私有源，导致通用响应	集成公共和私有数据集，实现高度定制化和相关的输出
搜索范围	限于预训练知识库；无法扩展到新信息或外部信息	能够在多个数据库或在线源上进行广泛的、按需搜索
可靠性	由于依赖静态和预生成知识，错误的可能性高	通过实时交叉引用多个受信任的来源确保可靠性
用例	适用于通用任务，但对于动态或数据密集型应用效果较差	适用于需要实时更新、研究或自定义数据集成的任务
透明度	提供的信息没有明确的参考或引用，难以验证	提供引用或链接到来源，确保透明度和可信度

2. RAG 工作原理与挑战

RAG系统架构分为两个主要部分：1️⃣ **数据索引** 2️⃣ **搜索与生成**



2.1 数据索引

这部分侧重于在检索过程中准备和管理知识库。

- **步骤1：加载**：系统获取不同类型的数据（例如，文本文件、PDF、URL和JSON文件）。数据可以来自不同的来源，确保知识库的全面性。
- **步骤2：分割**：数据被分成更小、有意义的块（chunks）。这一步确保检索高效工作，允许系统获取文档中精确相关的部分，而不是检索整个文件。
- **步骤3：嵌入：**数据的每个块使用嵌入模型转换为向量表示。这些嵌入捕获文本的语义含义，使系统能够执行基于相似度的搜索。
- **步骤4：存储**：嵌入和相应的数据存储在向量数据库中。向量数据库针对快速准确的相似度搜索进行了优化，这对检索至关重要。

2.2 搜索与生成

这描述了结合检索和生成来产生答案的整个过程。

- **步骤1：问题输入**：用户提供查询或问题。系统首先分析此问题以了解上下文和意图。
- **步骤2：检索**：系统查询已索引的知识库（检索系统）以收集最相关的文档或信息片段。这些文档作为生成答案的支持证据或上下文。
- **步骤3：提示创建：**检索到的文档被构建成一个用于LLM的提示。提示包括原始问题和检索到的信息，指导LLM生成一个上下文感知的响应。
- **步骤4：生成**：LLM处理提示，利用其生成能力创建连贯精确的响应。响应结合了检索到的文档中的见解和LLM的预训练知识。
- **步骤5：答案输出**：最终答案呈现给用户，融合了检索到的知识和LLM的生成能力。

2.3 RAG面临的挑战

尽管RAG在克服LLM局限性方面有效，尤其适用于基础任务（如问答场景），但它在复杂用例中仍然面临挑战。

挑战类别	具体挑战	解决方案
1️⃣ 检索与生成融合	- 检索结果不准确或评分不高导致生成错误 - 多文档融合难，影响上下文利用 - LLM提取答案受上下文噪声影响	- 优化检索算法，提升相关性 - 设计有效的融合策略 - 采用Self-RAG等自反机制提升生成准确性
2️⃣ 文档切分粒度	- 切块过粗导致语义丢失 - 切块过细带来噪声和上下文不完整	- 基于语义和语言模型辅助分块 - 通过测试调整最佳分块粒度
3️⃣ 系统扩展性与成本	- API调用成本高 - 向量数据库延迟和吞吐瓶颈 - 维护成本高	- 精简索引，删除冗余 - 分层索引检索减少计算 - 结合本地模型与云端API分配资源
4️⃣ 隐私与数据安全	- 私有数据传输风险 - 向量数据库存储敏感信息安全问题	- 本地处理或加密存储敏感数据 - 设计访问控制和隐私保护机制
5️⃣ 检索-生成语义不对称	- 查询与文档语义不匹配 - LLM对矛盾或无关信息鲁棒性差	- 使用假设文档嵌入（HyDE）提升语义对称性 - 利用LLM优化查询 - 引入自然语言推理过滤上下文
7️⃣ 垂直领域适应性	- 通用模型在专业领域表现差 - 文档拆分和问答分句效果有限	- 领域微调embedding和LLM - 改进拆分和问答分句技术
8️⃣ 系统有效性验证	- 缺乏统一评估标准 - 需要大量测试和调优	- 建立代表性测试集 - 持续监控和调整模型与检索配置

3. Agentic RAG 概述

Agentic RAG（智能体驱动的检索增强生成）是对传统 RAG 架构的深度演进。通过引入自主 AI 智能体，Agentic RAG 实现了对检索策略的动态管理、上下文理解的迭代优化，以及任务流程的自适应调整，从而突破了传统 RAG 在处理复杂任务和多步推理方面的局限性。

- **核心理念**

  Agentic RAG 的核心在于将智能体引入 RAG 流程，使系统具备以下能力：

  - **动态检索策略**：智能体根据任务需求，灵活选择检索工具（如向量数据库、网络搜索、API 接口等），并优化查询方式，以获取最相关的信息。
  - **上下文理解与迭代优化**：通过反思机制，智能体能够评估生成结果的质量，必要时重新检索信息或调整生成策略，提升回答的准确性和相关性。
  - **自适应工作流程**：智能体根据任务的复杂度和类型，动态调整工作流程，实现从顺序执行到多智能体协作的灵活转变。