RAG落地必备！七大常见问题及应对策略

这篇论文介绍了如何设计一个检索增强生成系统（RAG），该系统通过将查询与文档进行语义匹配，并使用大型语言模型（LLM）提取正确答案，旨在减少基于语言模型的回答出现幻觉的问题、链接来源和参考文献以及消除对元数据注释的需求。然而，RAG系统存在信息检索系统的固有局限性和对LLM的依赖性等问题。作者通过对三个不同领域的案例研究，总结了七个失败点并提出了相关建议。此外，作者还指出了验证RAG系统只能在运行期间完成以及其鲁棒性随时间推移而不断发展的两个关键要点。最后，作者列出了关于RAG系统的研究方向，以供软件工程社区参考。

RAG常见问题

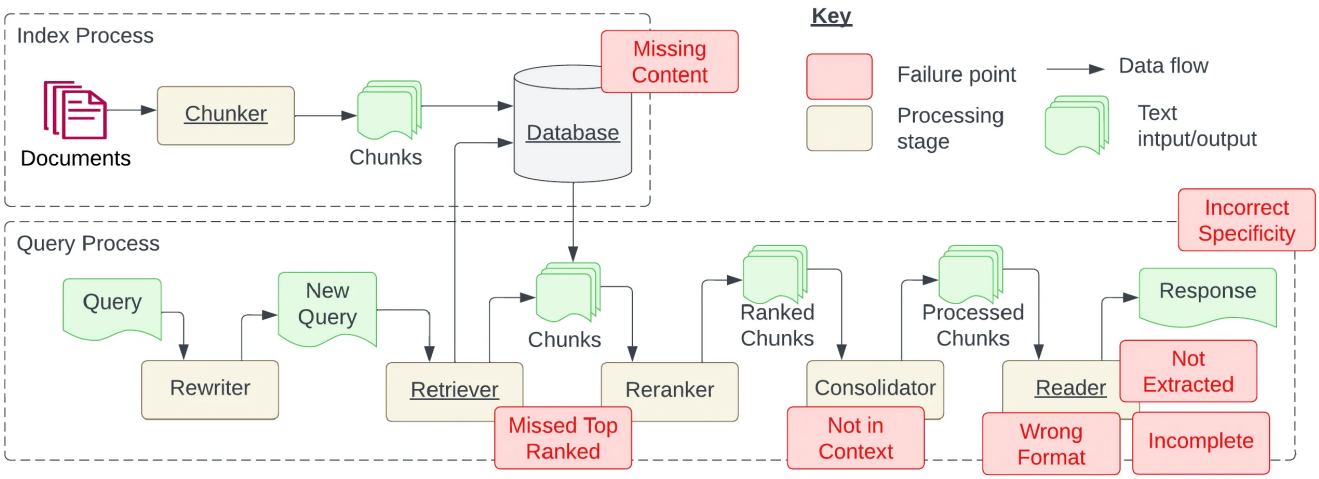

Figure 1: Indexing and Query processes required for creating a Retrieval Augmented Generation (RAG) system. The indexing process is typically done at development time and queries at runtime. Failure points identified in this study are shown in red boxes. All required stages are underlined. Figure expanded from [19].

- Missing Content：用户问题超出知识库内容范围，例如水利知识库，用户咨询文学知识。
  - 解决方案：思维链提示工程；知识库扩充；
- Missed Top Ranked：正确内容未被检索到。问题的答案在文档中，但无法返回给用户，在实践中，返回前K个文档，其中K是根据性能选择的值。
  - 解决方案：检索器进行多路召回；Embedding微调；

- Not in Context：带有答案的文档是从数据库中检索的，但没有放入生成答案的上下文中。多路召回或进行重排操作时，容易发生该问题。例如返回K个文档，有m(m<K)个文档放入上下文传递给大模型。
    - 解决方案：大模型支持更长上下文；上下文内容压缩；
- Wrong Format：格式错误，常发生于工具调用的情形下，要求模型以特定格式，如json、list返回，模型忽略或生成错误内容。
    - 解决方案： 大模型微调
- Not Extracted：这里答案存在于上下文中，但是大型语言模型未能提取出正确的答案。通常，当上下文中有太多噪声或矛盾信息时，就会发生这种情况。
    - 解决方案： 大模型微调
- Incomplete：不完整的答案并不是不正确，而是错过了一些信息，即使这些信息在上下文中并且可以提取。
    - 解决方案： 大模型微调
- Incorrect Specificity：模型生成的答案不适合，无法满足用户的需求。当用户不确定如何提问并且过于笼统时，也会发生不正确的问题。
    - 解决方案： 大模型微调

在实际场景中，应用RAG解决问题时，面临的开放性问题包括：

- chunk size：文本切块方式和块大小，直接影响问答的准确性。
- top k：受限于chunk size和大模型的上下文窗口限制。
- multi-hop Q&A：一个问题中提及多个知识点的情况，例如帮我对比苹果和华为手机的优缺点。
- World Knowledge：检索到的知识，会覆盖或干扰模型已有的知识，是否使用检索内容的边界难以确定。

文章优点

- 本文提出了一种新的信息检索方式——Retrieval-Augmented Generation（RAG）系统，并对其在软件工程中的应用进行了深入研究。
- 研究人员通过三个案例研究和实验数据，提出了构建RAG系统的挑战和解决方案，为实践者提供了宝贵的指导和经验教训。
- 该研究还探讨了RAG系统未来的研究方向，包括嵌入式技术和与Fine-tuning的比较等。

方法创新点

- 本文提出的RAG系统是一种将信息检索机制与大型语言模型（LLM）的生成能力相结合的方法，可以合成与用户查询相关的准确且最新的信息。
- 该研究重点介绍了如何预处理领域知识、存储处理后的信息、实现或整合适当的查询–文本文档匹配策略、排名匹配文档以及调用LLM的API以传递给用户查询和上下文文档等方面的挑战和解决方案。

未来展望

- 本文为软件工程师提供了一个关于如何构建鲁棒的RAG系统的指南，同时也指出了未来需要进一步探索的方向。
- 例如，研究人员建议关注嵌入式技术、与Fine–tuning的比较等方面的问题，以便更好地理解RAG系统的性能和局限性。
- 此外，随着LLM技术的不断发展，研究人员需要不断更新和改进RAG系统的构建方法，以适应不同的应用场景。

> **原文**：Seven Failure Points When Engineering a Retrieval Augmented Generation System