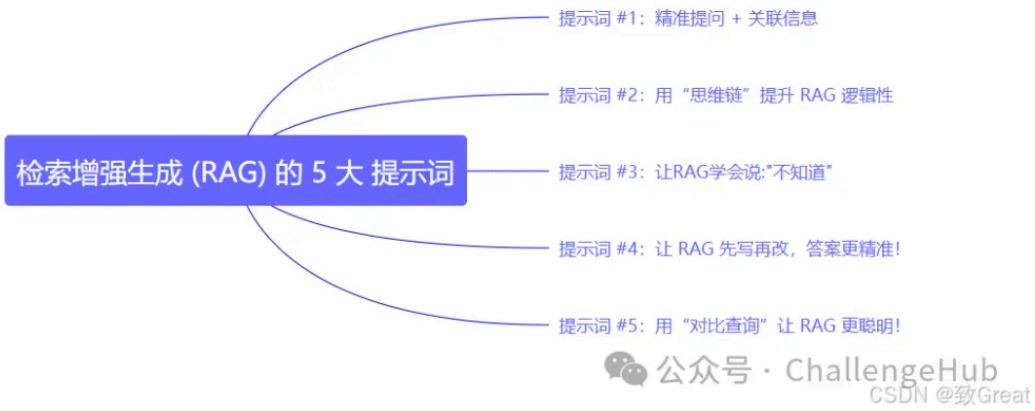
检索增强生成 (RAG) 的 5 大 提示词，非常实用!

致Great ChallengeHub 2025年03月14日 05:03 北京



> RAG 到底是啥？怎么用得更好？ （高手略过）

从去年到现在，检索增强生成（RAG）这套玩法越来越火。简单来说，它就是让大型语言模型（LLM）结合外部数据，确保回答更准确、不胡说八道（减少“幻觉”）。这样一来，RAG 系统不仅能给出更靠谱的答案，还能紧跟最新信息。

不过，光有 RAG 还不够，**怎么提问（也就是“提示词”）才是关键！** 你问得好，模型才能真正利用检索到的信息，给你想要的答案。

比如，在Stack Overflow 的看到过一篇文章就指出，**提示词太笼统，可能会让系统搜出一堆没用的信息，还浪费大量 token**（也就是处理能力）。笔者认为，优化提示词能带来巨大提升，包括自己之前参加过的一些评测，Prompt调整可以带来分数上的大幅度变动。

所以，问题来了：**怎么设计更高效的提示词？**本文就给大家分享 5 种实用的 RAG 提示词模板，帮你提升生成质量，减少无关信息，让 RAG 回答更精准！

为什么提示词对 RAG 这么重要？

你跟 RAG 交流的方式，直接决定了它的回答质量。**提示词就像是给大模型（LLM）下的“指令”**，告诉它该怎么理解你的问题、怎么用外部数据来回答。如果你只是随口一句“用外部数据回答”，那 RAG 可能还是答得不完整，甚至引用的内容可能都过时了。

所以，**想让 RAG 真的靠谱，你的提示词里得精准传达 3 件事：**

1. **检索到的信息怎么用**（RAG 不能只是拿到数据，还得理解怎么整合进回答里）；
2. **你的具体需求是什么**（RAG 不能靠猜，你得明确告诉它方向）；

3. 推理逻辑该怎么走（RAG 需要知道该怎么组织信息，避免胡编乱造）。

此外，LLM 处理文本是有“容量限制”的（它只能看一定数量的 token，也就是文本片段）。这意味着，你不能把一整个资料库丢给它，而是需要 RAG 系统先筛选出最相关的数据，再通过提示词引导 RAG 使用这些信息。

如果你的提示词不清晰，RAG 可能还是给出错误或不完整的答案。接下来，我们就看看5 种超实用的提示词模板，帮你让 RAG 生成的答案又稳又准！

提示词 #1：精准提问 + 关联信息

想让 RAG 给出靠谱的答案，先问好问题，再提供关键信息，这是一种简单但超实用的 RAG 提示策略。核心思路是：先把用户的问题提炼得更精准，再结合相关的知识库信息，让 RAG 生成最优答案。

示例提示词：

📌“用户想了解 [X]。请先提炼问题的核心意思，然后结合以下知识库中的内容，给出最清晰、最准确的答案。”

为什么有效？

✅避免 RAG 误解问题：先优化问题的表述，减少歧义，让 RAG 更聚焦。
✅让回答更精准：提示 RAG 必须基于检索到的信息回答，避免“脑补”或乱编。

如何操作？

1️⃣优化用户问题：用简短的 NLP 处理或摘要方法，提炼出最核心的意思。
2️⃣检索相关内容：从知识库中找到匹配的信息片段。
3️⃣组合提示词：把优化后的问题和检索到的内容放进提示词里。
4️⃣让 RAG 生成答案：确保 RAG 回答时紧扣检索内容，不添油加醋。

适用场景：

📍 用户问题比较笼统或模棱两可时，能帮 RAG 更好理解需求。
📍 你希望 RAG 直接抓住问题重点，而不是绕弯子。

这招适合各种开放式提问，特别是当用户的问题过于宽泛时，先“精炼问题 + 提供上下文”可以大大提升回答质量！

提示词 #2：用“思维链”提升 RAG 逻辑性

当问题复杂、信息量大时，RAG 容易给出混乱甚至错误的答案。这时候，“思维链”（ChRAGn-of-Thought，CoT）技巧就派上用场了！它的核心思路是让 RAG 按照清晰的步骤推理，而不是直接跳到答案，这样不仅逻辑更清楚，还能减少胡编乱造的情况。

怎么用？

可以让 RAG 按照下面的提示词来一步步思考：

📌“这是用户的问题和相关文本。请按照以下步骤来回答：
1️⃣用更简单的语言总结用户问题。
2️⃣挑选出最相关的文本片段。
3️⃣把这些片段整理成逻辑清晰的大纲。
4️⃣基于大纲撰写一个完整、连贯的答案。
🚀 **请展示你的推理过程，并提供最终精炼后的答案。”_

为什么这种方法有效？

✅强制 RAG 先分析，再回答，避免它“想当然”地生成错误内容。
✅先列大纲，保证结构清晰，让答案更易读、逻辑更顺。
✅推理过程透明可见，如果有错，你能轻松找到问题出在哪。

适用场景：

📍需要准确推理的任务，比如金融、法律、医学等领域，RAG 不能胡编。
📍检索信息量很大时，可以帮 RAG 过滤掉无关信息，专注于重要部分。

看过RAG论文的都知道，让 RAG 逐步推理，比直接生成答案更精准，特别是在多信息块组合的情况下！所以，下次遇到复杂问题，试试让 RAG 按步骤来想，答案质量可能会大幅提升 🚀！

提示词 #3：让RAG学会说:"不知道"

RAG 能帮大模型（LLM）检索外部数据，提高回答的准确性。但现实情况是，如果知识库里没有相关内容，RAG 可能还是会给出“不完整”甚至“瞎编”的答案。那怎么让它在“缺数据”的情况下做出更靠谱的回应？——引导 RAG 诚实地承认“我不确定”！

怎么用？

这个提示词可以让 RAG 在没有足够信息时，谨慎作答：

📌“以下是与用户问题最相关的内容。如果你发现其中有足够的信息来回答，就请据此作答；如果没有，请直接说：‘我没有关于此问题的完整信息。’

📌 摘要：
（插入检索到的相关文本或要点）

📌 现在你的最终答案是什么？”_

为什么这个方法有效？

✅减少“幻觉”问题：RAG 只会用现有数据回答，避免胡乱生成内容。
✅先检查再回答：提示 RAG 在回答前，先确认信息是否足够。
✅能发现数据缺口：如果 RAG 频繁回答“我不确定”，说明知识库可能需要补充新内容。

如何落地？

📍优化数据分块方式：确保 RAG 返回的是简明、有用的知识点。
📍定期更新知识库：如果某些问题 RAG 经常回答“我不知道”，可能是数据不足，需要补充新资料。

适用场景：

💬智能客服：避免 RAG 胡乱回答，而是礼貌地承认“没有完整信息”。
📊研究分析：确保 RAG 只在有足够依据的情况下给出答案，不随意推测。

RAG 的作用是增强信息获取，但它也不能凭空创造内容。与其误导用户，不如让它学会“坦诚不知”！

提示词 #4：让 RAG 先写再改，答案更精准！

有些任务，比如总结技术文档、改写政策文件、生成详细报告，单靠 RAG 一次性给出完美答案并不现实。这时候，多步骤修订的方法就特别有用——让 RAG 先写初稿，然后自己检查、修正，最后再输出完整答案，还能提供来源列表，增强可信度。

怎么用？

这个提示词能引导 RAG 进行“先写后改”：

📌第一步：“根据用户请求，生成一份完整的草稿，并结合下方 RAG 检索的所有相关段落。”

📌第二步：“现在重新检查初稿，看是否遗漏了任何有价值的上下文，并进行修订。”

📌第三步：“提供最终版本，确保内容连贯、精准。”

📌第四步：“标明引用的所有来源。”

为什么这种方法有效？

✅**自我审查，减少遗漏**：RAG 先写初稿，再进行自查修订，确保所有关键信息都被充分利用。
✅**多来源整合，提高准确性**：如果 RAG 检索到的内容较多，这种方式能帮助它全面整合，不遗漏重要细节。
✅**提供数据来源，增强可信度**：像研究论文一样，引用来源让读者更信任答案的可靠性。

适用场景：

📍**正式文档**：政策文件、人力资源指南、法律声明等，需要内容准确无误。
📍**多来源汇总**：比如营销文案，需要从多个产品页面提取信息并整合。
📍**复杂知识库**：如果你的数据库信息较多，单次生成可能会遗漏关键内容，多步骤审查能保证完整性。

让 RAG 先写后改，比一次性生成更靠谱！想要高质量内容，就别怕“多走一步”

提示词 #5：用“对比查询”让 RAG 更聪明！

想让 RAG 更精准地回答问题？**试试“对比查询”法！**这个方法不是简单地抛出一个问题，而是给 RAG**两个相关但有差异的问题**，让它在回答时学会分辨，并明确引用不同的信息来源。

怎么用？

你可以用这个结构化提示词来引导 RAG：

📌“查询 A：（用户的第一个问题）
📌查询 B：（一个相似但角度不同的问题）
📌检索到的文本：（插入相关内容片段）
📌要求：针对每个问题单独作答，确保每个答案都引用最匹配的文本。回答完成后，解释你是如何决定哪些内容适用于哪个问题的。”

为什么这个方法有效？

✅让 RAG 学会对比和归类：有时候，知识库的内容可能涵盖多个话题，这种方法能帮助 RAG 选取最合适的文本回答不同的问题。
✅减少“答案混淆”：指定每个答案必须基于不同的来源，防止 RAG 把多个问题的答案混在一起。
✅让 RAG 自我解释推理逻辑：这不仅能帮助调试，还能提高回答的透明度，让你知道它是如何选择答案的。

适用场景：

📍客户支持 & 销售：比如，一个客户问“这个产品多少钱？”，另一个问“这个产品支持哪些功能？”，RAG 需要从定价和技术文档中找出最匹配的内容，而不是混在一起回答。
📍内部培训 & 评测：用对比查询来测试 RAG 在不同问题上的表现，看看它是否真的能精准引用不同的文本来源。
📍多主题知识库：如果你的数据库里内容交叉较多，这种方法可以帮 RAG 识别哪些信息适合回答哪个问题。

“对比查询”是一种给 RAG 施加“压力测试”的方法，逼它更精准地匹配问题和答案！试试这个技巧，让你的 RAG 更聪明、更精准 🚀

如何优化 RAG 提示词？4 个关键技巧

想让 RAG 给出更精准、可靠的答案？除了设计合理的提示词，数据质量、格式选择、Token 限制等因素同样重要。以下四个实用技巧可以帮助你优化 RAG 提示词，提高整体生成效果。

1. 清理和整理 RAG 数据源

RAG 的输出质量，取决于它能检索到的内容。如果知识库中存在不相关或低质量的文档，模型可能会被误导，给出错误或冗余的回答。因此，定期清理数据源至关重要。可以设定规则，确保检索出的信息足够精准，并过滤掉无关内容，提高系统整体的准确性。

2. 控制 Token 限制，找到平衡点

LLM 处理的信息量是有限的，过长或过短的提示词都会影响效果。

- 过长的上下文可能会超出 Token 限制，让模型难以聚焦重点。
- 过短的上下文则可能遗漏关键信息，导致回答不完整。
  最佳做法是：使用摘要或预处理方式精简信息，确保 RAG 只接收最核心的数据。

3. 选择合适的提示词格式

不同的内容类型，适合不同的提示词格式。在某些情况下，调整提示词结构能显著提升 RAG 的回答质量。例如：

- 要点式总结：适用于技术性内容，让信息更清晰易读。
- 问答结构（Q&A）：适用于 FAQ 或知识库查询，便于模型精准匹配答案。
- 表格格式：适用于信息比对，比如产品参数、数据分析等场景。

4. 加入审核机制，确保答案可靠

在高风险场景（如法律、医疗、金融等领域），不能仅依赖 RAG 自动生成答案，而是需要增加审核机制。可以采用两种方式：

- 辅助模型审核：用另一个 AI 先检查回答质量，发现问题后优化。
- 人工复核：对于关键内容，增加人工审核流程，确保最终输出准确无误。

高质量的 RAG 提示词，不只是简单的指令设计，还涉及数据筛选、格式优化、Token 控制和审核机制等多个方面。通过合理运用这些技巧，可以让 RAG 生成的答案更精准、可靠，真正发挥出它的价值。

RAG模板

> 下面是笔者提供的一些模板样例，大家可以根据不同自行改造，这些模板从不同角度来尽量满足我们问答的需求。 更多模板可见：

https://github.com/gomate-community/TrustRAG/blob/main/trustrag/modules/prompt/templates.py

系统模板

```
SYSTEM_PROMPT = """你是一个专门用于回答中国电信运营商相关问题的AI助手。你的任务是基于提供的支撑信息，对用
1. 答案必须完全基于提供的支撑信息，不要添加任何不在支撑信息中的内容。
2. 尽可能使用支撑信息中的原文，保持答案的准确性。
3. 确保你的回答包含问题中要求的所有关键信息。
4. 保持回答简洁，尽量不要超过支撑信息的1.5倍长度。绝对不要超过2.5倍长度。
5. 如果问题涉及数字、日期或具体数据，务必在回答中准确包含这些信息。
6. 对于表格中的数据或需要综合多个段落的问题，请确保回答全面且准确。
7. 如果支撑信息不足以回答问题，请直接说明"根据提供的信息无法回答该问题"。
8. 不要使用"根据提供的信息"、"支撑信息显示"等前缀，直接给出答案。
9. 保持答案的连贯性和逻辑性，使用恰当的转折词和连接词。

记住，你的目标是提供一个既准确又简洁的回答，以获得最高的评分。"""
```

上下文模板

```
RAG_PROMPT_TEMPALTE="""使用以上下文来回答用户的问题。如果你不知道答案，就说你不知道。总是使用中文回答。
        问题: {question}
        可参考的上下文：
        ···
        {context}
        ···
```

如果给定的上下文无法让你做出回答，请回答数据库中没有这个内容，你不知道。

有用的回答:"""，

对话历史模板

GoGPT_PROMPT_TEMPALTE="""请基于所提供的支撑信息和对话历史，对给定的问题撰写一个全面且有条理的答复。

如果支撑信息或对话历史与当前问题无关或者提供信息不充分，请尝试自己回答问题或者无法回答问题。\n\n

对话历史：{context}\n\n

支撑信息：{concated_contents}\n\n

问题：{query}\n\n回答：:"""，

细化要求模板

Qwen_PROMPT_TEMPLATE="""作为一个精确的RAG系统助手，请严格按照以下指南回答用户问题：

1. 仔细分析问题，识别关键词和核心概念。

2. 从提供的上下文中精确定位相关信息，优先使用完全匹配的内容。

3. 构建回答时，确保包含所有必要的关键词，提高关键词评分(scoreikw)。

4. 保持回答与原文的语义相似度，以提高向量相似度评分(scoreies)。

5. 控制回答长度，理想情况下不超过参考上下文长度的1.5倍，最多不超过2.5倍。

6. 对于表格查询或需要多段落/多文档综合的问题，给予特别关注并提供更全面的回答。

7. 如果上下文信息不足，可以进行合理推理，但要明确指出推理部分。

8. 回答应简洁、准确、完整，直接解答问题，避免不必要的解释。

9. 不要输出“检索到的文本块”、“根据”，“信息”等前缀修饰句，直接输出答案即可

10. 不要使用"根据提供的信息"、"支撑信息显示"等前缀，直接给出答案。

问题：{question}

参考上下文：

...

{context}

...

```
请提供准确、相关且简洁的回答："""
```

结论：优化提示词，让 RAG 更智能

优化提示词的方式，直接决定了 RAG 的表现。从**精简查询**到**思维链推理**，每种策略都在解决同一个核心问题——如何精准检索上下文，让模型正确整合信息，并合理应对不确定性。

不断试验是关键。甚至微小的提示词调整，都可能对最终结果产生显著影响。因此，在实际应用中，持续优化提示词设计，观察模型的反馈，再根据效果调整，能让 RAG 更加精准和高效。

如果你正考虑搭建新的 RAG 应用，或想优化现有的 RAG 方案，那么一个能整合**检索、提示词优化和工作流管理**的平台会极大提高效率。笔者认为，统一管理这些环节，可以帮助你更方便地调整提示词，并从用户互动中提取有价值的反馈。

从小处着手。选择一个具体任务，尝试本文介绍的某种提示词策略，看看系统如何响应。通过不断迭代优化，你会逐步找到最适合自己业务场景的提示词方案。虽然没有“万能提示词”，但结合这些经过验证的方法，RAG 生成的答案质量会得到显著提升。

RAG 54 TrustRAG 14

RAG · 目录

上一篇 · 关于DeepResearch设计实现的碎碎念

个人观点，仅供参考

修改于2025年03月14日