智能系统学报
CAAI TRANSACTIONS ON INTELLIGENT SYSTEMS

基于聚类重组和预解析的检索增强生成方法

王文博, 张志飞, 王睿智, 苗夺谦

引用本文:
王文博, 张志飞, 王睿智, 等. 基于聚类重组和预解析的检索增强生成方法[J]. 智能系统学报, 2026, 21(1): 236-244.
WANG Wenbo, ZHANG Zhifei, WANG Ruizhi, et al. Retrieval-augmented generation based on cluster reorganization and pre-parsing[J]. *CAAI Transactions on Intelligent Systems*, 2026, 21(1): 236-244.

在线阅读 View online: https://dx.doi.org/10.11992/tis.202506029

DOI: 10.11992/tis.202506029
网络出版地址: https://link.cnki.net/urlid/23.1538.TP.20251218.1329.002

基于聚类重组和预解析的检索增强生成方法

王文博[1], 张志飞[2], 王睿智[1], 苗夺谦[1]
(1. 同济大学 计算机科学与技术学院, 上海 201804; 2. 同济大学 国家海底科学观测系统项目办公室, 上海 200092)

摘 要: 检索增强生成 (retrieval-augmented generation, RAG) 技术因具有为大语言模型 (large language model, LLM) 提供模型外知识的能力而受到人们的关注, 然而绝大多数方法都难以同时兼顾局部的细节知识和原文中不连续的多跳知识。针对上述问题, 提出基于聚类重组和预解析的检索增强生成方法。在索引阶段, 首先通过聚类算法将不连续的相关知识组合成新分块, 以提高多跳知识的检索能力; 然后基于提示工程对各知识分块进行预解析生成更细粒度的新分块, 以提高检索阶段的召回率。在检索阶段, 将召回的所有新分块还原为原文分块, 并连同查询语句输入给大语言模型以得到最终答案。在数据集 QuALITY 上对所提出的方法进行了评估, 通过消融实验和开源基线对比实验验证了方法的有效性, 并在公开的评测排行榜上取得了最佳效果。本文分析结果可为 RAG 的索引和检索技术提供参考。

关键词: 深度学习; 自然语言处理; 大语言模型; 向量检索; 自动问答; 检索增强生成; 聚类算法; 提示工程
中图分类号: TP311.1 **文献标志码**: A **文章编号**: 1673−4785(2026)01−0236−09

中文引用格式: 王文博, 张志飞, 王睿智, 等. 基于聚类重组和预解析的检索增强生成方法 [J]. 智能系统学报, 2026, 21(1): 236–244.
英文引用格式: WANG Wenbo, ZHANG Zhifei, WANG Ruizhi, et al. Retrieval-augmented generation based on cluster reorganization and pre-parsing[J]. CAAI transactions on intelligent systems, 2026, 21(1): 236–244.

Retrieval-augmented generation based on cluster reorganization and pre-parsing

WANG Wenbo[1], ZHANG Zhifei[2], WANG Ruizhi[1], MIAO Duoqian[1]
(1. School of Computer Science and Technology, Tongji University, Shanghai 201804, China; 2. Project Management Office of China National Scientific Seafloor Observatory, Tongji University, Shanghai 200092, China)

Abstract: Retrieval-augmented generation(RAG) has garnered remarkable attention for its ability to provide external knowledge to large language models(LLM). However, existing RAG methods often struggle to simultaneously capture both local detailed knowledge and non-contiguous multi-hop knowledge within the original text. To address this issue, this study proposes a novel RAG method based on cluster reorganization and pre-parsing. In the indexing stage, clustering algorithms are used to group discontinuous but relevant knowledge into new chunks, enhancing the retrieval of multi-hop information. Furthermore, prompt engineering is applied to pre-parse these chunks, dividing them into finer-grained sub-units to improve recall during retrieval. In the retrieval stage, all retrieved chunks are restored to their original context blocks and, together with the query, are fed into the LLM to generate the final answer. Ablation and comparative experiments conducted on the QuALITY dataset demonstrate the effectiveness of the proposed method, achieving the best performance on the public leaderboard. The findings of this study provide valuable insights for improving indexing and retrieval technologies in RAG.

Keywords: deep learning; natural language processing; large language models; vector retrieval; question answering; retrieval-augmented generation; clustering algorithms; prompt engineering

随着大语言模型 (large language model, LLM) 的高速发展[1], ChatGPT[2-3]、Qwen[4-5]、DeepSeek[6] 等主流大模型已经在各种下游任务上展示出了良

收稿日期: 2025−06−25. **网络出版日期**: 2025−12−19.
基金项目: 国家重点研发计划项目 (2022YFB3104702); 上海市自然科学基金项目 (22ZR1466700).
通信作者: 张志飞. E-mail: zhifeizhang@tongji.edu.cn.

好的性能, 但是 LLM 仍面临着有限上下文窗口、模型幻觉和参数化知识无法更新等艰难挑战[7-8]。为了应对这些问题, 检索增强生成 (retrieval-augmented generation, RAG)[9] 技术逐渐受到关注。这种先检索后对话的方法可以对用户问题进行必要的知识补充, 从而使 LLM 在可控的知识范围内生成更加稳定、准确和可靠的回答。

RAG 的基本流程依次分为 3 个阶段[10]。索引阶段: 将原文文本以一定的规则分割成多个较小的文本块并作为候选知识存入向量数据库中; 检索阶段: 基于文本块在语义空间的向量表示, 从候选集中召回语义相似度最贴近用户查询的文本块作为检索结果, 也就是此次查询的参考知识; 生成阶段: 将用户查询与参考知识一起提交给 LLM 以得到合理、可靠的答案。目前围绕以上 3 个阶段已经有了大量的研究成果[10-12], 但是这些方案仍然存在两个具有挑战性的问题: 一是如何更准确地召回原文中不连续但信息相关的多跳知识; 二是如何更充分地利用文本块的细节信息以提升检索的召回率。

针对问题一, 分块聚类、循环检索等多种方法被提出, 但是这些方法都以原文分块为最小检索粒度, 容易忽视分块中的细节信息; 针对问题二, 主流的解决思路是针对每个分块从多个维度生成伪文档以提高召回率, 但是生成的伪文档仅仅基于原文的一个分块, 无法兼顾多跳知识的场景。为了更好地同时解决这两个问题, 本文提出了基于聚类重组和预解析的检索增强生成模型 (RAG based on cluster reorganization and pre-parsing, CAP-RAG)。具体而言, 在索引阶段, 先对原文分块的语义向量应用聚类算法, 将语义接近的分块合并为新的聚类分块, 增强模型对原文中不连续但信息相关知识的召回能力; 然后通过提示工程对各个原文分块和聚类分块分别进行预解析, 以产生大量兼顾分块原文和分块内细节的预解析分块, 进一步提高模型对分块内细节信息的检索能力。在检索阶段, 将召回的所有分块均还原为原文分块以避免预解析环节模型幻觉带来的影响。

本文在数据集 QuALITY[13] 上进行了消融实验和对比实验, 实验结果表明, 相比于各种基线方法, CAP-RAG 有效提高了相关知识分块的召回率, 同时在公开排行榜上取得了最佳效果。

1　RAG 相关工作

1.1　朴素 RAG

朴素 RAG[10,14] 是对 RAG 这 3 个阶段的最简实现。本文使用D表示原始文档, 朴素 RAG 里索引阶段通过原文切块、文本向量化得到候选集的过程可以表示为

$$d_1,d_2,\cdots,d_n = f_{\text{split}}(D)$$

$$\boldsymbol{v}_{d_i} = f_{\text{embed}}(d_i), i = 1,2,\cdots,n$$

式中: d_i为原文分块, 同时也是候选集的元素; f_{split}为文档分块函数; f_{embed}为向量化模型; $\boldsymbol{v}_{d_i}$为d_i的语义向量; $\boldsymbol{v}_d$为原文分块语义向量的集合。

检索阶段利用向量相似度检索的过程则可以表示为

$$\boldsymbol{v}_q = f_{\text{embed}}(q)$$

$$d'_1,d'_2,\cdots,d'_N = \text{Top}N\left(f_{\text{sim}}\left(\boldsymbol{v}_q,\boldsymbol{v}_{d_i}\right)\right), i = 1,2,\cdots,n$$

式中: q为输入的查询语句; $\boldsymbol{v}_q$为q的语义向量; f_{sim}为语义相似度计算函数; Top$N$为取语义相似度最高的N个元素; $d'_1,d'_2,\cdots,d'_N$为最终召回的N个原文分块。

生成阶段则表示为

$$d_{\text{answer}} = \text{LLM}\left(\text{Concat}\left(q,d'_1,d'_2,\cdots,d'_N\right)\right)$$

式中: Concat为字符串拼接函数, LLM为大语言模型推理过程, d_{answer}为该大语言模型对查询语句给出的最终答案。

1.2　索引算法

RAG 索引阶段里原始文档的解析入库对于后续的检索阶段、生成阶段都非常重要。划分合理、语义明确的文本分块既可以提高检索效率, 又可以改善生成阶段的幻觉问题。近年来, 相关的研究主要聚焦于文本分块的划分策略优化以及 Embedding 模型优化[15] 两个方向。除了朴素 RAG 里的固定长度分块方法, Sarthi 等[16] 又提出了 RAPTOR 方法, 该方法在索引阶段通过递归地执行两个步骤来构建多层摘要树: 先将文本分块聚类, 再对每个聚类簇生成抽象摘要作为上层节点。这种方案通过应用聚类算法很大程度上解决了不连续知识的检索问题, 但由于树的叶子节点依旧是原文分块, 忽略了原文分块的细节信息。Raina 等[17] 提出将文本分块进一步分解为固定个数原子问题的方法, 在检索召回率上得到了提升。在 Embedding 模型优化方面, Günther 等[18] 提出专门为检索器设计 Embedding 模型, 通过延迟分块的方法使模型可以基于文本块的上下文语境计算语义向量, 这种方法使稠密检索器的召回效果有了明显提升, 但由于模型输入窗口的有限性, 模型能够顾及的上下文仅限于连续上下文, 距离较远的不连续上下文信息则难以被处理。

1.3　检索算法

RAG 的核心思路是通过先检索后生成的方

式来应对 LLM 知识无法更新等问题，其研究重点在于检索算法的设计。早期的检索算法是以 BM25、TF-IDF 算法等为基础的稀疏检索算法，而朴素 RAG 方法采用了基于语义空间的文本向量相似度检索方式，即稠密检索[14]。Izacard 等[19] 通过对比学习训练出一种改进的稠密检索器；Ye 等[20] 研究指出稀疏检索和稠密检索两者融合的方式能得到更好的检索效果。

也有一些利用假设性文档的检索方式[21-22]，Wang 等[23] 利用提示工程将 query 扩展为语义信息更丰富的假设性文档，进而得到更好的检索结果；Gao 等[24] 直接利用 LLM 先为 query 生成可能的回答，再基于这个回答从向量数据库中召回语义相似的文本，在多个数据集和任务上取得了显著的效果提升。然而，这些方法都是从 query 出发，采用零样本问答的方式生成假设性文档，脱离了知识库本身的语义背景，存在假设性文档与 query 实际无关系的风险。此外，还有结合精排序与粗排序[25-26]、整合异构数据源[27-28] 等不同思路的优化方法。

2　CAP-RAG 模型

CAP-RAG 在检索时能够同时兼顾不连续的多跳知识和大文本分块内的局部细节信息，其模型结构如图 1 所示，主要包括 3 个部分：1) 聚类重组：通过聚类算法对原文分块聚类以得到新的包含不连续知识的新分块；2) 预解析：通过提示工程对各个分块生成兼顾分块原文和分块内细节的三元组 (query, summary, context)，其中，query 表示查询，summary 表示总结，context 表示原文或对原文的部分引用，从而构建出大量预解析分块；3) 检索与生成：将召回的所有分块均还原为原文分块，避免预解析环节模型幻觉带来的影响，然后连同查询语句一同输入给大语言模型以得到最终答案。

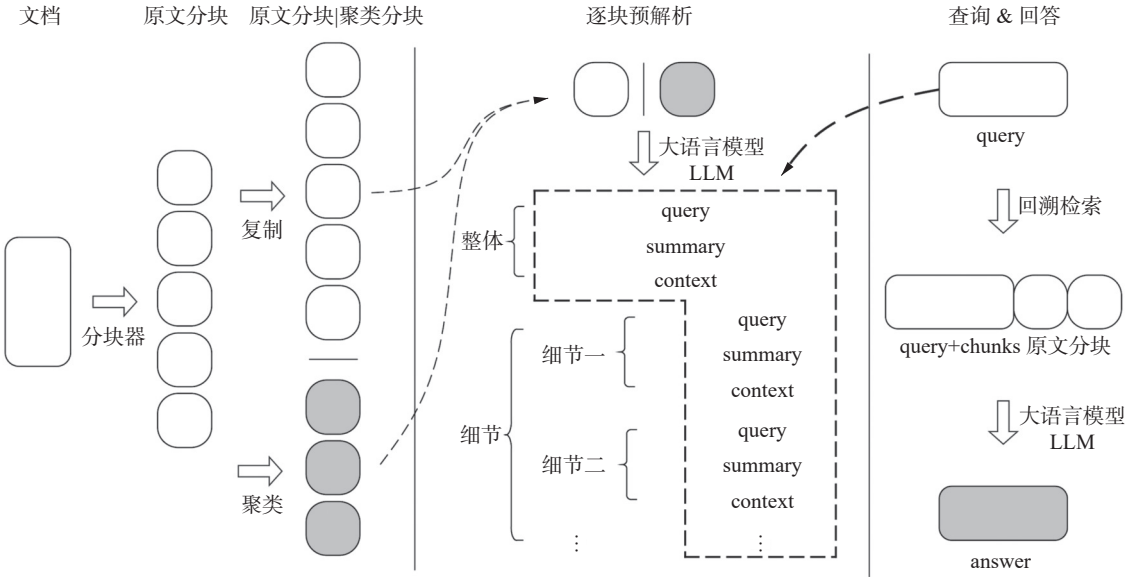

图 1　基于聚类重组和预解析的检索增强生成

Fig. 1　Retrieval-augmented generation based on cluster reorganization and pre-parsing

2.1　基于聚类的分块重组方法

为了能同时采样不连续的上下文信息，本文在语义空间上使用聚类算法对语义相似的原文分块进行重组，基本过程如图 2 所示。索引阶段利用 Embedding 模型对所有原文分块向量化得到 $\boldsymbol{v}_d$ 后，首先使用统一流形近似与投影 (uniform manifold approximation and projection, UMAP) 算法[29] 对 $\boldsymbol{v}_d$ 降维得到 $\boldsymbol{v}'_d$；然后利用贝叶斯信息准则 (Bayesian information criterion, BIC) 对 $\boldsymbol{v}'_d$ 预测聚类的个数 k；最后使用高斯混合模型 (Gaussian mixture models, GMMs) 对 $\boldsymbol{v}'_d$ 聚类得到 k 个聚类分组，即 k 个聚类分块 $d_{n+1}, d_{n+2}, \cdots, d_{n+k}$。上述过程可以表示为

$$\boldsymbol{v}'_d = \mathrm{UMAP}(\boldsymbol{v}_d)$$

$$k = \mathrm{BIC}(\boldsymbol{v}'_d)$$

$$d_{n+1}, d_{n+2}, \cdots, d_{n+k} = \mathrm{GMMs}(\boldsymbol{v}'_{d_1}, \boldsymbol{v}'_{d_2}, \cdots, \boldsymbol{v}'_{d_n}, k)$$

这里使用 UMAP 算法降维旨在降低 BIC 预测的难度以及避免聚类算法对高维向量的聚类结果不佳，使用 GMMs 则是为了实现软聚类算法。该方法最终得到的聚类分块 $d_{n+1}, d_{n+2}, \cdots, d_{n+k}$ 包含了不连续但语义相关的上下文信息，若将这些聚类分块和原文分块一同作为检索目标，即将 $d_1, d_2, \cdots, d_n, d_{n+1}, \cdots, d_{n+k}$ 全部作为检索的候选集，可以在检索阶段对于一些复杂问题有更高的检索效果。但是这些聚类分块在 token 长度上必然会

远大于原文分块, 向量化得到的稠密向量也很难兼顾到其中细节的语义, 因此在 2.2 节进一步提出了预解析算法。

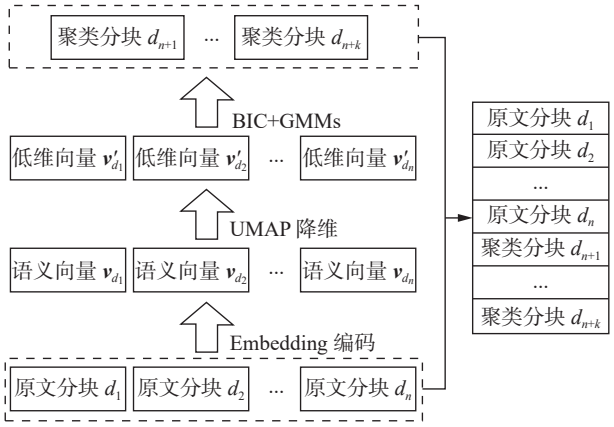

图 2　基于聚类的分块重组方法

Fig. 2　Chunk reorganization method based on clustering

2.2　基于提示工程的文本预解析算法

朴素 RAG 的稠密检索方法是将文本分块和查询分块都转化为语义向量, 再借助空间向量距离召回相似文本。这种方法对向量化模型的语义空间表示能力有很高的要求。一个文本分块的语义向量虽然可以表示整个分块的语义, 但当分块较大、细节信息较多时, 该语义向量很难兼顾到细节的语义。

因此, 本文提出基于提示工程的预解析算法, 基本思路是借助 LLM 的推理能力, 在向量化前对每个文本分块进行预解析, 包括信息抽取和假设性文档生成。信息抽取是将目标分块作为一个整体, 单独抽取其中有价值的细节片段; 假设性文档生成则包括预测用户可能提出的查询语句以及对目标分块进行文本摘要。通过预解析得到的内容都可以作为独立的检索目标, 相比朴素 RAG 可以更加突出细节知识, 提高对细节知识的召回效果。

具体实现如图 1 的逐块预解析部分所示, 本文通过设计提示词模板和 LLM 的结构化输出特性, 要求 LLM 在预解析原文分块或聚类分块时, 不仅要从整体层面上解析出一组可能的查询 query、对整体的总结 summary 以及原文 context; 还要从细节层面上解析生成 l 组可能的查询 query、对应的总结 summary 以及相关的原文 context, l 的值不固定, 由 LLM 基于输入的文本内容自行决定。于是, 每一个目标分块的解析结果都可以用三元组 (z_q, z_s, z_c) 表示, 该算法可以用符号表示为

$$(z_{0q}, z_{0s}, z_{0c}), (z_{1q}, z_{1s}, z_{1c}), \cdots, (z_{lq}, z_{ls}, z_{lc}) = \mathrm{LLM}(d_{\mathrm{prompt}}, d)$$

式中: d_{prompt} 为提示词, d 为输入的文本分块, (z_{0q}, z_{0s}, z_{0c}) 为从整体角度解析出的三元组, $(z_{1q}, z_{1s}, z_{1c}), \cdots, (z_{lq}, z_{ls}, z_{lc})$ 为从细节角度解析出的 l 个三元组。所有解析出的三元组元素都将作为检索阶段的候选集元素, 参与到检索过程。

2.3　回溯检索与答案生成

CAP-RAG 相比于朴素 RAG, 增加了聚类重组和预解析的过程, 产生了聚类分块和预解析分块。当在检索阶段命中这些新增的分块时, 本文采取的策略是将它们回溯到原文分块, 以避免生成阶段 LLM 推理答案时使用到原文文本以外的任何信息, 降低了预解析过程中模型幻觉对生成阶段的影响。

算法 1　回溯检索算法

输入　查询语句 q, 候选集 $Z = \{z_{0q}, z_{0s}, z_{0c}, z_{1q}, z_{1s}, z_{1c}, \cdots, z_{lq}, z_{ls}, z_{lc}\}$;

输出　结果集 D'

1) 初始化: D' 为空集合;

2) 召回与 q 语义相似度最高的前 $M(M > N)$ 个候选集元素 $Z' = \{z_1, z_2, \cdots, z_M\}$;

3) for each z in Z' do

4) if z 来自原文分块 d then;

5) add d to D';

6) else if z 来自聚类分块 $d_{n+j} \subseteq \{d_1, d_2, \cdots, d_n\}$, $1 \leqslant j \leqslant k$ then;

7) add d_{n+j} 中的原文分块元素 to D';

8) End if;

9) 对 D' 中的原文分块元素去重;

10) if D' 中的元素个数大于 N then;

11) break;

12) End for;

13) $D' \leftarrow$sort by 每个分块在原文中出现的顺序。

回溯检索算法如算法 1 所示, 包括稠密检索和回溯去重两个部分。步骤 1)~2) 为稠密检索, 即计算查询语句与 2.2 节里生成的所有三元组元素 Z 的欧氏距离, 召回前 $M(M > N)$ 个语义相似度最高的样本。步骤 3)~13) 为回溯去重, 首先将所有召回条目按照相似度从高到低排序, 逐个将其回溯为原文分块, 并将这些原文分块不重复的加入到结果集中; 如果检索到聚类分块, 就将该聚类分块关联的所有原文分块都加入结果集; 接着, 当结果集中原文分块的个数超过 N 时, 对结果集中的原文分块按照原文中出现的顺序排序, 返回作为检索结果。算法中 $M(M > N)$ 是为了避免因为回溯和去重导致候选集总数不足 N。

CAP-RAG 的答案生成过程与朴素 RAG 保持一致。

3　CAP-RAG 实验设置

3.1　数据集

本文采用 QuALITY 数据集[13] 验证模型的性能。QuALITY 数据集是一个高质量的英文单项选择问答数据集，其中所有文章的平均长度超过 5 000 个 token，涵盖小说、杂志、文学作品等多种体裁。每篇文章都附有由专业标注员编写的多道单项选择题，共计 6 737 道题目。本实验将这些单项选择题改写为问答题并要求 LLM 预测正确的选项。此外，该数据集还依据人工做题的出错样本，标注了一个更具挑战性的困难子集以进一步测试模型的性能。

实验使用该数据集中的训练集和测试集，前者包含 150 篇文章和 2 523 道题目，后者包含 116 篇文章和 2 128 道题目，详细的数据分布信息见表 1。

表 1　QuALITY 数据集统计信息

Table 1　QuALITY dataset statistics

数据集	类型	文章/篇	题目/道	困难子集/道
训练集	小说	118	2 000	1 056
	杂志	22	355	142
	文学作品	10	168	53
	总计	150	2 523	1 251
测试集	小说	81	1 486	828
	杂志	25	450	170
	文学作品	10	192	46
	总计	116	2 128	1 044

3.2　实验设计

为了验证方法的有效性，本文在训练集上进行了消融实验，在测试集上进行了对比实验。代码已开源 (https://github.com/yyTraveler/CAP-RAG)。

1) 实验细节

本文所有实验的索引阶段均在本地 Ubuntu 工作站上进行，该工作站配置有 32 核 i9-12 代 CPU 和两张 40 系 16 GB 显卡，并通过 VLLM 框架[30] 部署了 Qwen2.5-7B-Instruct 模型的 OpenAI API 服务。选择 Qwen2.5-7B-Instruct 作为 LLM 主要有两方面原因：一方面，本文的实验环境可以支持离线部署该模型，便于控制实验成本；另一方面，该模型支持了结构化输出的特性，即可以通过提示词来确保模型以 JSON 形式输出结构化、可解析的数据。对于本文提出的文本预解析算法而言，该特性允许在一次问答内完成对一个分块的预解析工作，显著提高实验效率。

实验索引阶段的 Embedding 模型使用开源的 SBERT 模型，即 multi-qa-mpnet-base-cos-v1，因为该模型更适用于非对称语义的文本向量检索任务。预解析算法使用的提示词参见代码仓库。

2) 消融实验

在消融实验的索引阶段，以 100 个 token 为单位完成原文分块的切分工作，并在生成阶段同样使用 Qwen2.5-7B-Instruct 作为 LLM，实验中使用 API 调用 LLM 模型的参数见表 2。在索引阶段，为了能尽可能多地生成预解析条目，没有限制模型输出的最大 token 个数，而在检索生成阶段，控制最大 token 个数为 30 来避免模型输出选项以外的冗余内容。

表 2　消融实验和对比实验中与 LLM 交互的参数

Table 2　Parameters in ablation and comparative experiments

实验	参数	索引阶段	生成阶段
消融实验	temperature	0.7	0.7
	top_p	0.8	0.8
	max_tokens	not_given	30.0
对比实验	temperature	0.7	1.0
	top_p	0.8	not_given
	max_tokens	not_given	30.0

基于本文提到的聚类重组和预解析方法，消融实验设计了朴素 RAG[10]、C-RAG(含聚类分块的 RAG)、P-RAG(含预解析分块的 RAG) 和 CAP-RAG(含聚类分块和预解析分块)4 种方法。

本文比较了每种方法在召回 Top1、2、5 个分块下的性能表现，并进一步分析了在召回更多分块时，各方法之间的性能差异。

3) 对比实验

本文针对 QuALITY 数据集组织了开源基线实验、公榜基线实验两组对比实验。

对于开源基线实验，本文在 QuALITY 的训练集上横向对比了 BM25、DPR[14]、Contriever-MS MARCO[19]、RAPTOR[16]、CAP-RAG 这 5 种算法作为检索阶段算法的实验效果。实验的索引阶段以 100 个 token 为单位完成原文分块的切分工作，并在检索阶段只召回 Top5 个分块，其他条件与消融实验保持一致。

对于公榜基线实验，本文在对测试集使用 CAP-RAG 方法评测后，将结果文件按照官方要求提交给公开的线上评测系统以获取官方公布的评测结果。为了能充分利用 LLM 的有效上下文窗口，该实验将索引阶段的原文分块大小调整为 150 个 token，在检索阶段召回 30 个分块，其他参数则与消融实验保持一致。对于检索阶段的 LLM

选择, 虽然官方排行榜上排行靠前的实验都选择了 GPT-4、GPT-4o, 但是由于网络环境的限制, 本文最终选择 DeepSeek-V3, 该模型和 GPT-4o 具有相当的性能表现[6], 相关的请求参数如表 2 所示, 基于 DeepSeek-V3 的研究报告, 实验中参数 temperature 被设置为 1.0。CAP-RAG 的评测结果已经发布在 QuALITY 官方的排行榜上。

3.3　评价指标

由于 QuALITY 数据集的题目都是单项选择题, 而利用大模型的结构化输出能力, 可以确保输出的结果都是确切的选项, 所以直接使用准确率 (accuracy) 来评测模型的性能表现。

此外, 由于本文在检索与生成阶段都将检索结果全部回溯为原文分块, 因此准确率也直接反应了预解析对相关上下文的召回效果。

4　实验结果与分析

4.1　消融实验

消融实验中 4 种方法的实验结果如表 3 所示。

表 3　消融实验的实验结果
Table 3　Ablation study results　　%

方法	准确率		
	Top1	Top2	Top5
朴素RAG[10]	32.84	41.66	52.68
C-RAG	38.20	47.60	55.25
P-RAG	36.42	43.12	54.78
CAP-RAG	**41.10**	**48.16**	**55.45**

注: 加黑代表每种检索条件下的最优结果。

可以看出:

1) 相比于朴素 RAG, 3 种优化方法在召回 Top1、Top2 和 Top5 个分块时都有明显的性能提升。在召回 Top1 个分块时, 朴素 RAG 表现出的准确率只有 32.84%, 而 C-RAG、P-RAG 以及 CAP-RAG 的准确率分别增长了 5.36%、3.58% 和 8.26%; 在召回 Top2 和 Top5 个分块时, 3 种方法分别增长了 5.94%、1.46%、5.04% 和 2.57%、2.1%、2.77%。这部分实验结果表明, 无论是聚类分块、预解析分块还是基于聚类分块得到的预解析分块, 这些新生成的“伪文本分块”确实使与问题相关的原文文本更“容易”被召回。

2) 直接使用聚类分块的效果比直接使用预解析分块的效果更好。表 3 中 C-RAG 在召回 Top1、Top2 和 Top5 个分块时, 准确率比 P-RAG 分别高出 1.78%、4.48% 和 0.47%。原因在于: 一方面是本文选择的向量化模型适用于非对称语义, 即该模型对查询语句较短而候选文本较长的文本也有很好的性能表现, 非常适合聚类分块这种多个原文分块拼合的情况; 另一方面是聚类分块本身就对不连续的中长文本有较好的信息抽取能力, 相比于预解析这种细化每个分块内信息的方法, 聚类分块更加适合 QuALITY 这样的中长文本数据集。

此外, 为了进一步确认聚类分块、预解析分块是否在检索中发挥了作用, 针对检索结果中各个分块的类型进行了统计分析, 分析结果如图 3 所示。由于 CAP-RAG 检索的候选集里同时包含了“聚类分块方法”和“预解析分块方法”得到的文本分块, 所以直接使用 CAP-RAG 方法, 可以针对召回 Top1、Top2 和 Top5 个分块的情况分别统计出召回分块的类型分布 (只统计正确的样本)。图 3 中的“预解析分块”是针对原文分块应用预解析方法得到的文本分块, 而“聚类后预解析分块”表示针对聚类分块上应用预解析方法得到的文本分块。从图 3 中可以看出, 基于预解析算法得到的“预解析分块”和“聚类后预解析分块”分别相比于“原文分块”和“聚类分块”有明显更高的占比, 在 Top1、Top2 和 Top5 这 3 种召回条件下, “预解析分块”的占比分别为 65.38%、73.41% 和 77.26%, 而“聚类后预解析分块”的占比分别为 32.11%、22.85% 和 17.67%, 说明预解析算法确实提高了聚类分块和原文分块的召回率, 充分说明了预解析算法的有效性; 此外, 由于检索阶段召回的“聚类分块”和“聚类后预解析分块”实质上都先被回溯为聚类分块, 之后再进一步被回溯为原文分块, 因此检索召回的各个分块也可以按照是否源自聚类分块划分为两类, 其中源自聚类分块的分块数量在 Top1、Top2、Top5 这 3 种召回条件下分别有 32.50%、23.57%、18.73% 的占比, 同样验证了聚类算法的作用。

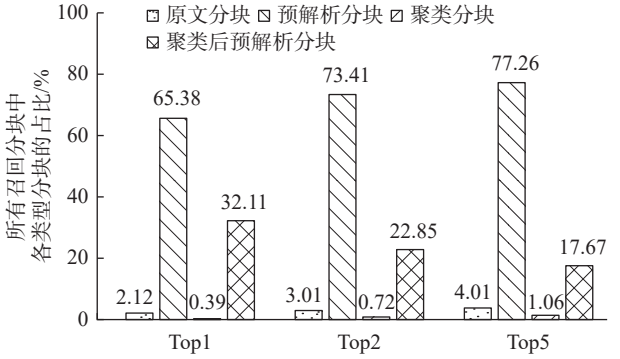

图 3　CAP-RAG 在召回 Top1、Top2 和 Top5 分块时的召回结果分布图

Fig. 3　Distribution of CAP-RAG's recall results for Top1, Top2 and Top5 chunks

4.2　对比实验

4.2.1　开源基线实验

表 4 给出了 CAP-RAG 与其他 4 条开源技术路线的效果对比。

表 4　QuALITY 数据集的开源基线实验结果
Table 4　Experimental results of open-source baselines on QuALITY　%

方法	准确率
BM25	50.22
DPR[14]	50.57
Contriever-MS MARCO[19]	52.87
RAPTOR[16]	54.58
CAP-RAG	**55.45**

注: 加黑代表最优结果。

从表 4 中可以看出:

1) 在 QuALITY 数据集上, 稠密检索方法普遍比稀疏检索方法效果好。本实验中, 除了 BM25 属于经典稀疏检索方法, 其他方法均属于稠密检索方法。可见在比较依赖语义的问答场景下, 稠密检索的效果明显比稀疏检索效果更好。

2) 聚类重组等分块优化技巧能有效提升 RAG 的检索效果。表 4 中 DPR 是经典稠密检索算法, Contriever-MS MARCO 是 DPR 的变种, 采用训练 Embedding 模型的方式取得了 2.30% 的提升; RAPTOR 是在 DPR 基础上结合使用了聚类和摘要任务, 相比于二者分别提升了 4.01% 和 1.71%; CAP-RAG 也使用到了聚类的技巧, 相比于二者分别提升了 3.88% 和 2.58%。可见聚类算法的应用有效提升了 RAG 的检索效果, 更容易召回有价值的文本分块。

3) 预解析方法对挖掘文档中的细节知识有明显的增益效果。RAPTOR 算法通过循环应用聚类和摘要生成任务将候选文档构建成一棵树形结构, 在预解析阶段输出了大量靠摘要生成任务生成的检索条目, 可以认为这些摘要任务是一次次知识压缩的过程, 而压缩的最小粒度对标本文提到的原文分块; CAP-RAG 相比于 RAPTOR, 只对原文档应用了一次聚类算法, 之后则通过预解析方法深度解析了原文分块和聚类分块, 取得了 0.87% 的检索效果增长。可见预解析方法在最大程度上保留住聚类算法聚集不连续多跳知识优点的同时, 能够进一步提高对细节知识的召回精度。

4.2.2　公开榜单基线实验

在经过数据集官方对测试集的黑盒评测后, 数据集的前 5 位排行如表 5 所示 (https://nyu-mll.github.io/quality/)。从表 5 中可以看出, 本文提出的 CAP-RAG 方法在该数据集上取得了最佳结果, 并在完整测试集、困难子集上相较其他方法都有显著的性能提升。相比于 RAPTOR(RAPTOR (collapsed tree) + GPT-4) 及其改进方法 (RAPTOR + gpt-4o w/ query intent & entity understanding), 本方法在测试集上的表现分别提升了 5.4% 和 4.9%, 在困难子集上的表现分别提升了 5.6% 和 4.6%, 提升效果明显。

表 5　QuALITY 数据集公开排行榜前 5 的方法和准确率
Table 5　Top 5 methods and accuracy on the QuALITY dataset public leaderboard　%

方法	测试集	困难子集
人工标注[13]	93.5	89.1
CAP-RAG + DeepSeek-V3	**88.0**	**81.9**
RAPTOR + gpt-4o w/ query intent & entity understanding	83.1	77.3
RAPTOR (collapsed tree) + GPT-4[16]	82.6	76.2
Long-context GPT-3.5 (gpt-3.5-turbo-16k)	74.7	64.3

注: 加黑代表非人工标注情况下的最优结果。

4.3　案例分析

为了更直观地给出 CAP-RAG 的工作原理, 从召回 Top5 个检索结果中取出一例样本作为示例, 如表 6 和表 7 所示。对于问题“What happens to drafted workers?”, 表 6 给出了该问题的 4 个选项, 表 7 则是 CAP-RAG 在检索阶段召回的 Top5 结果。从表 7 中可以看出, 得分最高的第一个项就是基于聚类分块得到的预解析分块, 该分块恰好也是查询语句, 语义同问题基本一致, 基于该召回的分块, 检索阶段经过原文回溯, 可以直接得到最准确的原文证据, 进而选出最佳答案 A。而传统 RAG 方法在该问题上则因为没有召回有效的相关文本而选择了错误的选项 D。

表 6　案例问题的 4 个选项
Table 6　Four options for the question

选项	内容
A	They train and work for a time, then retire with extra funds.
B	They receive no pay, and have to undergo training and work for some time
C	They are called upon throughout their life for periods of work.
D	They work a short period of time, then return to normal life.

表 7　CAP-RAG 针对案例问题在检索阶段的 Top5 召回结果

Table 7　Top5 recall results of CAP-RAG for case questions in the search stage

得分排序	聚类分块	预解析分块	预解析标签	分块内容
1	是	是	query(细节分块)	What happens to those who are drafted into the labor force?
2	是	是	summary(细节分块)	A few workers put in a reasonable number of hours, while others receive Inalienable Basic stock as unemployment insurance, and can be drafted for labor if needed.
3	否	是	context(细节分块)	When new employees were needed, a draft lottery was held. All persons registered in the labor force participated. If you were drawn, you must need serve.
4	否	是	summary(细节分块)	The narrator was surprised to be drafted for labor work.
5	否	是	context(细节分块)	He is now legally eligible for retirement. He was drafted into the working force reserves, served his time, and is now free from toil for the balance of his life.

5　结束语

本文提出了一种基于聚类重组和预解析的检索增强生成方法。在检索增强生成系统中, 利用聚类重组和预解析相结合的方式实现在一定量的信息里召回更多的有用信息, 并在 QuALITY 数据集的开源基线对比实验和黑盒评测中得到最佳结果。此外, 本文介绍的很多实现细节可以针对实际场景灵活调整, 如聚类算法、文本预解析的输出结构等, 以期得到更好的性能表现。

本文的工作仍存在一些值得继续深入研究的问题, 比如: 如何在生成阶段也利用索引阶段生成的伪文本、降低推理难度; 如何使用更好的方法来针对聚类分块进行信息压缩和信息提取等。

参考文献:

[1] 吴国栋, 秦辉, 胡全兴, 等. 大语言模型及其个性化推荐研究[J]. 智能系统学报, 2024, 19(6): 1351−1365.
WU Guodong, QIN Hui, HU Quanxing, et al. Research on large language models and personalized recommendation[J]. CAAI transactions on intelligent systems, 2024, 19(6): 1351−1365.

[2] BROWN T, MANN B, RYDER N, et al. Language models are few-shot learners[J]. Advances in neural information processing systems, 2020, 33: 1877−1901.

[3] BAHRINI A, KHAMOSHIFAR M, ABBASIMEHR H, et al. ChatGPT: applications, opportunities, and threats[C]// 2023 Systems and Information Engineering Design Symposium. Charlottesville: IEEE, 2023.

[4] BAI Jinze, BAI Shuai, CHU Yunfei, et al. Qwen technical report[EB/OL]. (2023−09−28)[2025−02−23]. http://arxiv.org/abs/2309.16609.

[5] YANG An, YANG Baosong, ZHANG Beichen et al. Qwen2.5 technical report[EB/OL]. (2025−01−03)[2025−02−22]. http://arxiv.org/abs/2412.15115.

[6] GUO Daya, YANG Dejian, ZHANG Haowei, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning[J]. Nature, 2025, 645(8081): 633−638.

[7] RAFFEL C, SHAZEER N, ROBERTS A, et al. Exploring the limits of transfer learning with a unified text-to-text transformer[J]. Journal of machine learning research, 2020, 21(140): 1−67.

[8] HUANG Lei, YU Weijiang, MA Weitao, et al. A survey on hallucination in large language models: principles, taxonomy, challenges, and open questions[J]. ACM transactions on information systems, 2025, 43(2): 1−55.

[9] LEWIS P, PEREZ E, PIKTUS A, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks[J]. Advances in neural information processing systems, 2020, 33: 9459−9474.

[10] GAO Yunfan, XIONG Yun, GAO Xinyu, et al. Retrieval-augmented generation for large language models: a survey[EB/OL]. (2023−12−29)[2024−01−02]. http://arxiv.org/abs/2312.10997.

[11] 邹佰翰, 汪莹, 彭鑫, 等. 重新审视代码补全中的检索增强策略[J]. 软件学报, 2025, 36(6): 2747−2773.
ZOU Baihan, WANG Ying, PENG Xin, et al. Revisiting retrieval-augmentation strategy in code completion[J]. Journal of software, 2025, 36(6): 2747−2773.

[12] 田萱, 吴志超. 基于信息检索的知识库问答综述[J]. 计算机研究与发展, 2025, 62(2): 314−335.
TIAN Xuan, WU Zhichao. Review of knowledge base question answering based on information retrieval[J]. Journal of computer research and development, 2025, 62(2): 314−335.

[13] PANG R Y, PARRISH A, JOSHI N, et al. QuALITY: question answering with long input texts, yes![C]//Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Seattle: USAACL, 2022.

[14] KARPUKHIN V, OGUZ B, MIN S, et al. Dense passage retrieval for open-domain question answering[C]//Proceedings of the 2020 Conference on Empirical Methods

in Natural Language Processing. Online: SIGDAT, 2020.

[15] 邸剑, 刘骏华, 曹锦纲. 利用 BERT 和覆盖率机制改进的 HiNT 文本检索模型[J]. 智能系统学报, 2024, 19(3): 719−727.
DI Jian, LIU Junhua, CAO Jingang. An improved HiNT text retrieval model using BERT and coverage mechanism[J]. CAAI transactions on intelligent systems, 2024, 19(3): 719−727.

[16] SARTHI P, ABDULLAH S, TULI A, et al. RAPTOR: recursive abstractive processing for tree-organized retrieval[C]// The Twelfth International Conference on Learning Representations. Vienna: ICLR, 2024.

[17] RAINA V, GALES M. Question-based retrieval using atomic units for enterprise RAG[C]//Proceedings of the Seventh Fact Extraction and VERification Workshop. Miami: ACL, 2024.

[18] GÜNTHER M, MOHR I, WILLIAMS D J, et al. Late chunking: contextual chunk embeddings using long-context embedding models[EB/OL]. (2024−10−02) [2024−10−10]. http://arxiv.org/abs/2409.04701.

[19] IZACARD G, CARON M, HOSSEINI L, et al. Unsupervised dense information retrieval with contrastive learning[J]. Transactions on machine learning research, 2022, 2022: 1−6.

[20] YE Qinyuan, BELTAGY I, PETERS M, et al. FiD-ICL: a fusion-in-decoder approach for efficient in-context learning[C]//Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics. Toronto: USAACL, 2023.

[21] MA Xinbei, GONG Yeyun, HE Pengcheng, et al. Query rewriting in retrieval-augmented large language models[C]// Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Singapore: USAACL, 2023.

[22] SHAO Zhihong, GONG Yeyun, SHEN Yelong, et al. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy[C]//Conference on Empirical Methods in Natural Language Processing(Findings). Singapor: ACL, 2023.

[23] WANG Liang, YANG Nan, WEI Furu. Query2doc: query expansion with large language models[C]//Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Singapore: SIGDAT, 2023.

[24] GAO Luyu, MA Xueguang, LIN J, et al. Precise zero-shot dense retrieval without relevance labels[C]//Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics. Toronto: USAACL, 2023.

[25] GLASS M, ROSSIELLO G, CHOWDHURY M F M, et al. Re2G: retrieve, rerank, generate[C]//Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Seattle: USAACL, 2022.

[26] ZHUANG Shengyao, LIU Bing, KOOPMAN B, et al. Open-source large language models are strong zero-shot query likelihood models for document ranking[C]//Conference on Empirical Methods in Natural Language Processing(Findings). Singapore: ACL, 2023.

[27] 余润杰, 阳羽凡, 周健, 等. 面向海量数据的高效流水化检索增强生成系统[J]. 中国科学: 信息科学, 2025, 55(3): 542−558.
YU Runjie, YANG Yufan, ZHOU Jian, et al. Efficient pipeline for retrieval-augmented generation system under big data[J]. Scientia sinica informationis, 2025, 55(3): 542−558.

[28] 吴文隆, 尹海莲, 王宁, 等. 大语言模型和知识图谱协同的跨域异质数据查询框架[J]. 计算机研究与发展, 2025, 62(3): 605−619.
WU Wenlong, YIN Hailian, WANG Ning, et al. A synergetic LLM-KG framework for cross-domain heterogeneous data query[J]. Journal of computer research and development, 2025, 62(3): 605−619.

[29] HEALY J, MCINNES L. Uniform manifold approximation and projection[J]. Nature reviews methods primers, 2024, 4: 82.

[30] KWON W, LI Zhuohan, ZHUANG Siyuan, et al. Efficient memory management for large language model serving with PagedAttention[C]//Proceedings of the 29th Symposium on Operating Systems Principles. Koblenz: ACM, 2023.

作者简介：

王文博，硕士研究生，主要研究方向为深度学习与向量检索。E-mail: wang.wenbo.top@qq.com。

张志飞，博士，博士生导师，中国人工智能学会粒计算与知识发现专业委员会委员，上海市计算机学会计算机视觉专业委员会秘书长，主要研究方向为模式识别与大数据挖掘。主持国家自然科学基金、上海市自然科学基金等项目，获吴文俊人工智能自然科学奖二等奖。发表学术论文 30 余篇。E-mail: zhifeizhang@tongji.edu.cn。

王睿智，副教授，博士生导师，中国人工智能学会粒计算与知识发现专业委员会委员，主要研究方向为深度学习与粒计算。获吴文俊人工智能自然科学奖二等奖。发表学术论文 50 余篇。E-mail: ruizhiwang@tongji.edu.cn。