基于ChatGPT查询改写的文档检索方法

李澳，涂新辉*，熊英豪
华中师范大学，计算机学院，湖北 武汉
aoleeccnu@mails.ccnu.edu.cn, tuxinhui@mail.ccnu.edu.cn,
yhxiong@mails.ccnu.edu.cn

摘要

查询改写是一种通过优化查询从而提高检索结果质量的技术。传统的基于伪相关反馈的方法受限于伪相关文档的质量。本文提出了一种基于ChatGPT查询改写的文档检索方法。这种方法不依赖伪相关文档，可以避免伪相关文档质量不高的问题。首先，利用BM25模型进行检索，获得初次检索结果集；同时借助ChatGPT生成新查询；然后分别将原始查询和新查询作为输入，利用重排模型对初次检索结果集进行重排，得到各自的文档相关性得分；最后，将两个查询的文档相关性得分进行融合，得到最终的文档得分。在多个检索测试集上的实验结果表明，相比于基准模型，基于ChatGPT查询改写的文档检索方法在nDCG@10指标上平均提升了约4.5个百分点。

关键词： 查询改写 ； ChatGPT ； 信息检索

Document Retrieval Method Based on ChatGPT Query Rewriting

LI Ao, TU Xinhui*, XIONG Yinghao
School of Computer Science, Central China Normal University, Wuhan Hubei
aoleeccnu@mails.ccnu.edu.cn, tuxinhui@mail.ccnu.edu.cn,
yhxiong@mails.ccnu.edu.cn

Abstract

Query rewriting is a technique aimed at enhancing the quality of search results through the optimization of queries. Traditional methods based on pseudo-relevance feedback are constrained by the quality of pseudo-relevant documents. This paper introduces a document retrieval method based on query rewriting with ChatGPT. This approach does not rely on pseudo-relevant documents, thus avoiding the issue of low-quality pseudo-relevant documents. Initially, the BM25 model is employed for retrieval to obtain an initial retrieval results. Concurrently, ChatGPT is utilized to generate a new query. Subsequently, both the original and the new queries are used as inputs to re-rank the initial retrieval results with a re-ranking model, producing document relevance scores for each query. Finally, the document relevance scores from both queries are integrated to obtain the final document scores. Experimental results on multiple retrieval test sets indicate that compared to baseline models, this approach achieves an average improvement of approximately 4.5 percentage points in the nDCG@10 metric.

* 通讯作者
基金项目:国家语委重点项目（ZDI145-22）

Keywords: Query Rewriting , ChatGPT , Information Retrieval

1 引言

信息检索旨在从大规模语料库中根据用户提供的查询来检索相关文档。当用户输入的查询较短时，这种检索过程往往面临着诸多挑战。简短的查询往往不足以准确地反映用户的信息需求，从而导致词汇不匹配问题频繁出现。此外，简短的查询通常在语义丰富度上有所欠缺，这进一步降低了词汇匹配的准确性。

为解决上述挑战，查询改写技术应运而生。它的核心目的是为了弥补信息检索过程中的“语义鸿沟”，即缓解因查询词与文档内容之间可能存在的语义差异所造成的影响。通过对原始查询进行扩展、替换或重组，查询改写技术能够将用户的信息需求转化为一个更具描述性、更准确地反映其潜在意图的查询，从而提高对相关文档的准确匹配率。

查询改写一直是一个广泛应用且具有长期历史的技术 (Azad and Deepak, 2019)。传统的查询改写方法大多数基于伪相关反馈机制 (Cao et al., 2008)，尽管这类基于伪相关反馈的方法在很多情境下均显示出良好的效果，但它们受限于初次检索结果中的质量。若初次检索结果集的质量较低，则伪相关反馈方法的性能亦会显著受损。

随着ChatGPT (Wu et al., 2023)引发的广泛关注，大语言模型(LLM) (Zhao et al., 2023)成为众多学者研究的焦点。大语言模型凭借强大的自然语言理解能力、文本生成能力和知识记忆能力展现了巨大的应用潜力，使得它们被视为有力的查询改写工具。因此，众多研究者尝试将大语言模型与查询改写技术相结合，研究出如生成式相关性反馈 (Mackie et al., 2023)、Query2doc (Wang et al., 2023a)等新型的查询改写方法。然而，目前大部分的研究工作更多地侧重于使用大语言模型进行查询与文档间的相关性推断，而较少深入挖掘其对于查询语义的深层次理解能力。

本文提出了一种基于ChatGPT查询改写的文档检索方法，该方法利用ChatGPT的文本生成能力和对查询的语义理解能力实现对原始查询的改写。首先，我们利用原始查询进行初次检索，从而获得一组初次检索的结果集，并将其输入检索模型；同时借助ChatGPT生成新查询；然后分别将原始查询和新查询作为检索模型的输入，检索模型利用两个查询分别给初次检索的结果集进行重排，得到各自的文档相关性得分；最后，将两个查询的文档相关性得分进行融合，得到最终的文档得分。

在基于ChatGPT的查询改写中，本文试图解答不同长度的查询是否会对检索模型的检索性能带来影响。实验数据显示，确实存在一个影响检索模型性能的查询长度最佳区间。位于这一区间内的查询能够更为精确地呈现其核心语义信息，从而进一步优化检索的精准度。在深入研究ChatGPT的文本生成能力时，我们还观察到模型在生成过程中偶尔存在内容不可控性。这种不可控性可能导致生成的文本内容中的语义信息存在混乱或缺乏明确的指向，从而影响到后续文本检索任务的性能。为了有效应对这一挑战，本研究针对性地设计了一种文档得分融合机制。核心思路是将原始查询的文档相关性得分与通过ChatGPT改写后的新查询的文档相关性得分进行结合，从而解决由ChatGPT不可控性所引发的问题，并优化信息检索的精确度和效率。原始查询中的关键词往往包含用户的核心检索需求，其重要性不言而喻。而经ChatGPT改写后的查询，可能会增加一些新的语义维度或对原始信息进行拓展和变换。通过将这两部分相关文档的得分信息进行融合，可以在一定程度上提高文档相关性排名的精确度。

最后，为了深入验证我们所提出的查询改写方法在检索场景下的泛化性及其对检索性能的实际影响，我们选择了不同领域的实验数据集和不同的检索模型。对于数据集，我们选择TREC-DL19 (Craswell et al., 2020)、TREC-DL20 (Craswell et al., 2021)作为通用领域的数据集，TREC-COVID (Voorhees et al., 2021)作为专业领域的数据集。对于检索模型，我们选择ColBERT (Khattab and Zaharia, 2020)、MonoT5 (Pradeep et al., 2021)以及RankGPT (Sun et al., 2023)作为检索模型。这些模型具有不同的语义理解与生成能力，因此结合这三个模型可以为我们提供一个综合的、多层次的评估视角。

2 相关工作

2.1 传统的查询改写方法

查询改写一直是信息检索领域的研究焦点。传统的查询改写方法大多数基于伪相关反馈机

制。其核心思路是将初次检索到的高排名文档视为与查询内容“伪相关”，并基于这些文档的内容来优化和改写用户的原始查询，从而期望得到更好的检索结果。传统的基于伪相关反馈的查询改写方法主要包括RM3 (Abdul-Jaleel et al., 2004)、Bo1 (Amati and Van Rijsbergen, 2002)以及KL扩展 (Amati and Van Rijsbergen, 2002)等。此外，伪相关反馈机制常与语言模型相结合，如ColBERT-PRF (Wang et al., 2023b)、BERT-QE (Zheng et al., 2020)、ANCE-PRF (Li et al., 2022)等。这类模型将伪相关反馈机制融入了语言模型的结构中，允许语言模型能够对检索返回的文档进行深入分析、聚类和排序。通过这种方式，模型可以生成一组与原查询高度相关的候选文档集合，进一步实现查询的改写和优化，从而更好地满足用户的检索需求。

2.2 大语言模型

目前的大语言模型普遍基于Transformer 架构 (Vaswani et al., 2017)来构建。与传统基于Transformer的模型显著不同的是，它们的模型规模远超前者，其神经网络的参数数量高达数亿甚至数十亿，并在海量文本数据上进行训练。这些特点使得大语言模型展现出卓越的自然语言处理能力，能够适应并完成一系列复杂的自然语言任务。

OpenAI发布的GPT系列逐渐成为自然语言处理领域的研究热点。从GPT-3 (Brown et al., 2020)开始，OpenAI正式迈入了大语言模型时代。GPT-3的模型参数高达175B，并引入了语境学习 (Dong et al., 2022)机制，从而使其能够有效地处理少样本乃至零样本的推理任务。2022年11月，ChatGPT正式上线，OpenAI使用RLHF (Ouyang et al., 2022)算法和PPO (Schulman et al., 2017)算法进一步提高训练数据集的质量，从而确保ChatGPT在多种任务上都能够呈现出卓越的推理性能。2023年3月，GPT-4 (Achiam et al., 2023)正式发布，标志着GPT系列从纯文本任务领域跨入多模态任务领域，相较于ChatGPT，GPT-4在处理复杂任务上展现出更为卓越的能力，输出的内容更为可靠，并且在降低模型产生的幻觉方面也取得了明显的进展。

Meta AI发布了LLaMA系列模型，成为目前最流行的开源语言模型。其中，LLaMA-13B (Touvron et al., 2023a)的模型参数规模仅为GPT-3的十分之一，但是在数学推理、阅读理解、代码生成等大多数自然语言处理任务中的性能却优于GPT-3；另外，Llama 2-70B (Touvron et al., 2023b)在多项自然语言处理任务中也展现出卓越的性能，并超越了至今为止的大多数开源模型。但与GPT-4相比，Llama 2-70B仍存在一定的性能差距。

2.3 基于大语言模型的查询改写

随着大语言模型的广泛应用，与大语言模型结合的查询改写工作正日益增多。Mackie等人提出了一种新的查询扩展方法，被称为生成式相关性反馈(Generative Relevance Feedback) (Mackie et al., 2023)。相较于传统的伪相关反馈方法，GRF的优势在于它不再仅仅依赖于初始检索所得文档的质量。其核心思想是利用大语言模型直接生成相关的文本内容，进而用作伪相关反馈中的相关文档集，进一步对查询进行优化。Liang Wang等人提出了名为query2doc (Wang et al., 2023a)的查询改写方法。该方法的核心机制是结合大语言模型与少样本提示策略，生成“伪文档”，再与原始查询进行结合，从而构建一个更为全面的新查询。Jagerman et al. (2023)的研究则更加注重大语言模型在查询改写任务中的文本生成。他们对大语言模型如何响应不同提示方式进行了深入研究，旨在揭示不同提示方法对模型查询改写效果的具体影响。为此，他们选定了Flan-T5 (Raffel et al., 2020)与Flan-UL2 (Tay et al., 2022)两个模型，并设计了八种独特的提示策略，以深度探讨这两个模型在查询扩展任务上的性能差异。

不同于上述研究，我们的工作不仅使用大语言模型对查询进行改写，而且着重研究了不同查询长度对检索性能的影响。另外，我们采用了一种文档得分融合机制，以增强模型的检索性能。

3 方法

本文提出的方法是利用ChatGPT的文本生成能力和对查询的语义理解能力实现查询的改写，并使用改写后的新查询提升文档检索的性能。其主要涵盖三大核心模块：查询生成、文档相关性计算以及原始查询和新查询的文档得分融合。

考虑到实验设备的限制和模型的运行时间，我们采用了一个三阶段的检索策略，如图 1所示。在第一阶段，我们利用了稀疏检索模型作为预筛选手段，对于每一个原始查询 q ，该

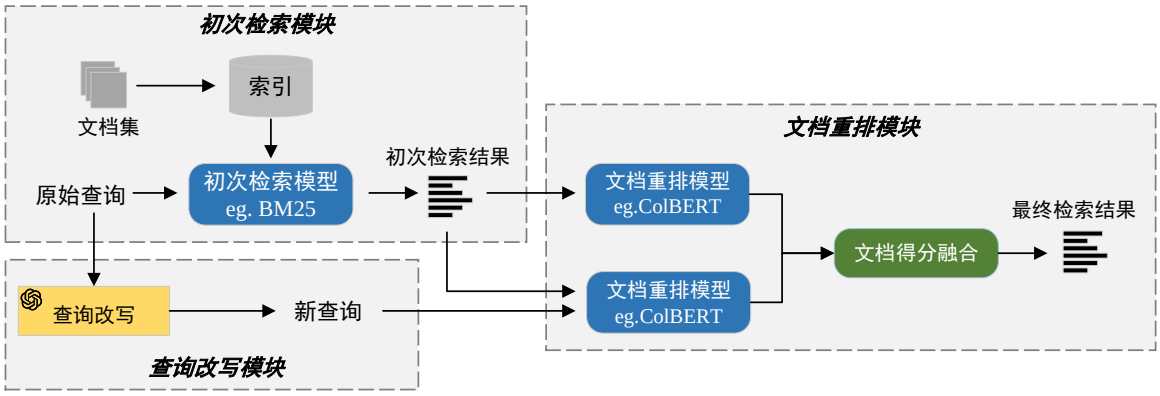


Figure 1: 基于ChatGPT查询改写的文档检索方法

模型会对整个文档集进行初次检索，从大量的文档集中快速检索出与原始查询 q 相关的前100篇文档。目的是缩小文档重排操作处理的文档范围并减轻计算负担。第二阶段，我们引入ChatGPT对原始查询 q 进行改写，生成新的查询 q' 。改写查询的意图在于捕捉原始查询的潜在语义信息。然后，我们将原始查询 q 及新查询 q' 均输入检索模型，分别对初次检索的文档集进行相关性评分，为后续的文档得分融合机制奠定基础。第三阶段，通过融合原始查询 q 和新查询 q' 的文档得分信息对相关文档进行重排，确保最终呈现给用户的文档是与当前查询最为相关的结果。

为了在检索系统中更好地利用查询的语义信息并提升检索质量，我们不会使用改写后的新查询进行初次检索，而是将新查询仅用于文档重排阶段。因为在初次检索的实验过程中，我们发现采用改写后的查询进行基于BM25模型的初次检索时，效果并不理想。具体来说，相比于使用原查询进行初次检索，BM25模型的精度下降了5%左右。因此我们认为经过ChatGPT改写后的新查询并不适合于BM25模型，但是有利于文档重排模型的检索结果。

3.1 基于ChatGPT的查询生成

System:
You are RewriteGPT, an intelligent assistant that rewrites content in COVID-19 fields.

User:
First, I will provide you with a query text for information retrieval delimited by ```.

Then, please rewrite this text according to the expertise related to the text. And the rewritten text will serve as a query.

Finally, the rewritten text has at least {{ length }} words.

Query Text: ``` {{query}} ```

Figure 2: 用于查询改写的Prompt示例（Covid-19数据集）

给定一个原始查询 q 作为ChatGPT的输入，借助提示方法，让ChatGPT对原始查询 q 进行进一步的优化和改写。图 2 展示了在Covid-19数据集上的使用的Prompt示例，其中 *query* 和 *length* 是变量，分别代表查询内容和查询长度。表 1 展示了由ChatGPT进行查询改写后的示例，以TREC-DL19数据集中的“what are the three percenters?”为例，其中的“three percenters”可能会造成混淆，因为其在语义上的表达不够明确，实际上指代美国和加拿大的极右反政府民兵组织。ChatGPT能够识别到其潜在的语义模糊性，并对其进行改写，得到一个更

Origin query	Rewritten query by ChatGPT
what are the three percenters?	The Three Percenters, often abbreviated as 3%ers, are a loosely organized American patriot movement. Originating in 2008, they advocate for limited government and resist perceived infringements on constitutional rights. Can you explain their ideological spectrum?
what is the un fao?	The UN FAO, or United Nations Food and Agriculture Organization, is a specialized agency of the United Nations that focuses on global food security, agriculture, and rural development.
what carvedilol used for?	Carvedilol is a medication primarily used for treating high blood pressure and heart failure. It belongs to the class of beta-blockers, helping to lower blood pressure and improve heart function.

Table 1: ChatGPT查询改写示例

为准确和明确的新查询。这种查询改写方法在处理语义模糊的查询时表现出了强大的适应性，确保查询意图更为直接和具体，从而在后续的检索任务中实现更为准确的结果匹配。

此外，我们发现查询的长度可能会影响语义信息的丰富度。为此，我们利用ChatGPT生成不同长度的新查询，旨在探究不同长度的查询如何影响检索模型的检索性能和输出质量。

3.2 文档相关性计算

我们选择RankGPT (Sun et al., 2023)、ColBERT (Khattab and Zaharia, 2020)以及MonoT5 (Pradeep et al., 2021)作为检索模型，即计算文档相关性得分。其中，我们通过调用OpenAI的第三方接口实现RankGPT模型的使用，而ColBERT和MonoT5模型属于开源模型。

RankGPT是一个基于ChatGPT(GPT-3.5-turbo)和GPT-4文档重排模型。RankGPT通过为每个文档添加唯一标识，加强了模型对每个文档的识别能力，使得ChatGPT能够根据当前查询和文档相关性对文档进行降序排列。此外，还提出一种滑动窗口策略来解决ChatGPT每次输入的最大长度限制。该策略旨在通过分段输入文本，并利用窗口重叠的方式，确保模型能够连贯地理解文本，从而突破了原有模型在长文本处理上的限制。

ColBERT通过利用BERT (Devlin et al., 2018)强大的语言处理能力，实现对查询和各类文档内容的高度精准编码。首先，ColBERT模型对查询和文档的每个部分进行独立编码。这种编码方法利用了BERT的语境化编码能力，可以捕捉文本中的细微差异和深层含义。在编码完成后，ColBERT采用一种被称为“后期交互”的方法，这是一种计算效率较高的相似度评分策略。不同于传统的立即交互模式，后期交互允许模型分阶段处理信息，先分别对查询和文档进行深度编码，然后再在一个精简的向量空间中计算二者的相似性。这种方法减少了在大规模数据集上的计算负担，优化了检索时间，提高了排序的准确性和效率。

MonoT5是基于T5 (Raffel et al., 2020)模型的一个衍生模型，专门用于处理文档排名任务。该模型的设计思想源于 Pradeep et al. (2021)提出的“Expando-Mono-Duo”设计模式，该模式采用预训练的序列到序列(Seq2Seq)模型进行文本排序。在这种设计模式中，“Mono”代表的是一种基于逐点方法(point-wise approach) (Li, 2022)的文档重排策略。逐点方法是一种评价文档相关性的方法，它独立地对每个文档进行评分，而不是将其与其他文档对比。MonoT5作为这一设计模式下的一个具体实现，展示了T5模型在处理文档排名问题上的高效性和准确性。

3.3 文档相关性得分融合

在多次实验观测中，我们注意到ChatGPT在某些情况下展现出的内容生成的不可控性。这种不可控性可能导致生成的新查询的语义信息存在混乱或缺乏明确的指向，从而影响到后续文本检索任务的性能。

针对这一问题，我们提出了一种文档得分融合机制，该机制会给原始查询和新查询分配合适的权重，从而有效地结合两者的信息。当新生成的查询 q' 的文档相关性得分较低时，我们将

更多地依赖原始查询 q 。在这种情境下，原始查询 q 的文档得分信息将被赋予较大的权重，而新查询 q' 的文档得分则作为一种辅助信息进行参考。相较之下，当新生成的查询 q' 的文档得分较高时，它将成为主导的检索依据。在这种场景下，新查询 q' 的文档得分信息会被赋予较大的权重，而原始查询 q 则被视作一个补充或辅助的查询来源。

我们采取了一种基于检索模型打分的权重分配策略，即使用检索模型给原始查询 q 和新查询 q' 进行文档相关性评分。权重分配的公式如下：

$$S_{final}(q,d) = \lambda \cdot S(q,d) + (1-\lambda) \cdot S(q',d) \tag{1}$$

其中，$S_{final}(q,d)$ 表示文档得分融合后的总得分；$S(q,d)$ 表示原始查询的文档得分；$S(q',d)$ 表示经过ChatGPT改写后的新查询的文档得分；λ 表示原始查询的权重分配比例，是一个超参数，该参数在不同的领域存在一定的泛化能力，本文的 5.3 节会对这个参数进行进一步的分析。

此文档得分融合机制能够充分利用原始查询与新查询中的有效信息，同时避免因ChatGPT的不可控性导致的检索误差，从而提高整体的检索质量和效率。

4 实验配置

本文的实验在以下三个数据集上进行：TREC-DL19、TREC-DL20和TREC-COVID。其中，TREC-DL19和TREC-DL20属于通用领域数据集，而TREC-COVID属于专业领域数据集。

表 2 统计了这三个数据集的查询(queries)数量、查询平均长度(len(q)，以单词为单位)、文档(docs)数量和查询相关性判断(qrels)数量信息。这三个数据集在信息检索领域具有较高的权威性，并涵盖了多种不同的文本内容与查询特点，有助于我们从多角度全面评估改写方法的效果。

Dataset	queries	len(q)	docs	qrels
TREC-DL19	43	5	8.8M	9.3K
TREC-DL20	54	6	8.8M	11K
TREC-COVID	50	11	193K	69K

Table 2: 数据集统计信息

TREC数据集指的是“Text Retrieval Conference”（文本检索会议）数据集，它是信息检索领域中一个重要的基准数据集。该数据集包含数百万甚至上亿的文本文档，涵盖了不同主题和领域。此外，该数据集涵盖了多个不同的任务和子任务，这些任务涵盖了信息检索领域的不同方面，例如文本分类、文本聚类、问答系统等。本文聚焦于TREC数据集中的段落检索任务，其中所包含的数据集涵盖了大量的查询以及与之相关的段落文档。本研究使用的TREC-DL19和TREC-DL20两个数据集，它们分别对应于2019年和2020年TREC发布的测试集。这两个数据集在构成上存在着一定差异，主要表现在TREC-DL19涵盖了43个查询，而TREC-DL20则涵盖了54个查询。需要注意的是，两个数据集都包括880万个段落，每个段落文本的字数都在100到200字之间。

TREC-COVID数据集是由文本检索与信息检索技术共同会议（TREC）组织联合构建的一个重要数据资源，专注于支持与COVID-19（新冠病毒）相关的信息检索和文本挖掘研究。该数据集汇集来自多个来源（包括科学文献、新闻报道、预印本等）的文本数据，并涵盖了广泛的主题，从病毒本身的特征到传播途径、临床症状、治疗方法，乃至公共卫生政策等各个领域的内容。TREC-COVID数据集涵盖了50个查询。每个查询都关联着与COVID-19有关的医学研究文章、新闻报道以及临床研究文章。此外，每篇文本文档的平均篇幅约为300字。

4.1 评价指标

本文将使用nDCG@K作为评价指标。nDCG@K衡量了系统在排名前K个结果中的文档相关性以及排名情况，提供了一个更综合的评价，考虑了文档的相关性和排名的影响。其中K代表

排名前K个检索结果。其计算公式如公式 2 所示：

$$nDCG@K = \frac{DCG@K}{IDCG@K} \tag{2}$$

其中，nDCG@K 的计算方式涉及两个关键概念：DCG 和IDCG。DCG衡量了在排名前K个结果中，每个文档的相关性对系统性能的贡献。DCG 的计算方式是对每个排名位置上的文档，将其相关性除以对数的排名，然后将所有文档的贡献相加得到DCG 值。IDCG表示在理想情况下，排名前K个文档的最大可能DCG值，即假设所有相关文档都在前K个位置上的DCG值。

4.2 对比模型

本文涉及的实验部分均采用PyTerrier框架 (Macdonald et al., 2021)进行实施与验证。PyTerrier是一种基于Python语言开发的信息检索框架，它支持开发者以声明性语法构建从简单到复杂的信息检索处理流程。PyTerrier框架不仅继承了Terrier的稳定性和效率，同时引入了更高级别的抽象和灵活性。

我们使用OpenAI的API接口实现对gpt-3.5-turbo模型的调用，使用该模型对查询进行改写。由于RankGPT是一个基于ChatGPT的检索模型，因此我们同样通过调用gpt-3.5-turbo使用RankGPT模型。有关gpt-3.5-turbo的重要参数设置如表 3 所示。

“Max_tokens”参数定义了模型输入与输出的最大长度总和，我们将此参数设定为“Infinite”以确保模型输出的完整性，避免因为超出最大令牌数而产生截断的情况。“Temperature”参数则负责调节输出结果的确定性，其取值在 0 到 1 之间。在此参数设定上，较低的“Temperature”值会使得模型生成的回答具有高度确定性和可预测性，即使多次以相同的提示向模型发起请求，也能获得高度一致或相同的答案；而较高的“Temperature”值则使得模型生成的回答更为多元化，但同时也可能增加结果偏离预期或主题的风险。基于上述考量，我们选择将“Temperature”设为 0.5，以平衡确定性和创新性。

ColBERT和MonoT5模型属于开源模型，可以下载到本地使用。ColBERT和MonoT5的模型参数大小分别为110M和220M。超参数的设定对模型性能有着直接影响，适当的调整超参数可以提升模型的准确度与效率。我们将ColBERT和MonoT5的BatchSize大小分别设置为 32 和 4 。

本文将RankGPT、ColBERT和MonoT5视为基准模型，并将与查询改写和文档得分融合机制结合后的模型视为它们相应的模型变体。

5 实验分析

5.1 整体性能分析

表 4 展示了在TREC-DL19、TREC-DL20以及TREC-COVID数据集上，不同检索模型在采用查询改写和文档得分融合机制前后的检索性能表现。其中，标有“w/o rewrite”的数据行展示了未进行查询改写和未采用文档得分融合机制时的检索性能；标有“w/ rewrite”的数据行展示了只采用查询改写不采用文档得分融合机制时的检索性能；标有“w/ fusion”的数据行展示了采用查询改写和文档得分融合机制后的检索性能；此外，n@k表示nDCG@k评价指标。为了直观地呈现本文提出的方法对检索性能的影响，表 4 仅列举了表现最佳的结果。至于不同长度的查询以及不同的文档得分融合情况对检索模型性能的影响将在 5.2 节和 5.3 节具体分析。

Important Parameters	Parameter Value
OpenAI API	gpt-3.5-turbo
Max_tokens	Infinite
Temperature	0.5

Table 3: gpt-3.5-turbo模型的重要参数设置

从表 4 中的实验数据得知，无论是通用领域的数据集还是专有领域的数据集，与查询改写和文档得分融合机制结合后的模型变体，其检索性能均优于相应的基准模型。

	TREC-DL19			TREC-DL20			TREC-COVID		
	n@1	n@5	n@10	n@1	n@5	n@10	n@1	n@5	n@10
BM25	50.77	49.02	47.95	56.48	49.65	49.36	75.00	67.51	62.32
ColBERT									
w/o rewrite	74.41	71.54	68.78	76.23	70.78	65.91	73.00	69.41	66.62
w/ rewrite	**74.81**	70.05	66.17	73.77	68.76	65.46	71.00	**73.82**	**71.99**
w/ fusion	**79.85**	**74.01**	**71.50**	**79.32**	**72.87**	**69.08**	**76.00**	**75.22**	**72.62**
MonoT5									
w/o rewrite	77.13	73.02	69.89	78.39	70.13	67.39	78.00	74.57	72.16
w/ rewrite	75.97	72.64	69.44	75.31	68.95	65.58	75.00	**78.13**	**75.95**
w/ fusion	**79.06**	**76.19**	**71.68**	**80.86**	**72.84**	**68.34**	**81.00**	**79.93**	**78.00**
RankGPT									
w/o rewrite	82.17	71.15	65.80	79.32	66.76	62.91	-	-	76.67
w/ rewrite	79.07	70.66	**67.49**	78.70	**68.64**	**63.74**	-	-	**77.19**

Table 4: 查询改写对不同检索模型的性能影响（n@k表示nDCG@k评价指标）

在通用领域数据集（DL19和DL20）上，我们发现当检索模型只采用查询改写不采用文档得分融合机制时，ColBERT和MonoT5模型的检索性能出现了下降，但RankGPT模型展现出了更好的检索性能。我们从不同检索模型架构的角度对这一现象进行了分析：(1) ColBERT是基于BERT的排序方法，主要关注查询与文档的精确匹配。改写后的查询虽更自然，但可能带来语义微调或额外信息，干扰原查询的匹配，导致检索性能下滑。(2) MonoT5是基于“Seq2Seq”的T5模型，预测查询与文档的匹配度。当改写后的查询偏离原意时，捕捉完整语义变的困难，从而导致检索性能减弱。(3) RankGPT使用ChatGPT理解查询的上下文和意图，并检索与之相关的文档。与前两种模型不同，RankGPT更能够适应和理解由ChatGPT改写的查询。这是因为改写后的查询往往更具有上下文信息和语言流畅性，这使得RankGPT能够更好地抓住查询的整体意图并进行有效的文档匹配。综上所述，对于更依赖于查询和文档之间精确匹配的模型（如ColBERT和MonoT5），ChatGPT的改写可能会降低它们的检索性能。但对于能够理解和适应自然语言查询的模型（如RankGPT），改写可能增强它们的检索能力，因为改写带来了更丰富的语境和语义信息。此外，这一实验结论也证明了文档得分融合机制的可行性和泛化性。

相比于通用领域数据集，在专业领域数据集（TREC-COVID）上，我们发现当检索模型只采用查询改写不采用文档得分融合机制时，ColBERT、MonoT5和RankGPT模型的检索性能都有显著提升。对于通用领域的数据集，我们认为一些查询包含的关键词可能拥有多重语义解释。因此，当ChatGPT对查询进行改写时，它们可能会基于和检索文档不相关的语义信息进行改写，导致了检索模型的性能下降。而TREC-COVID只关注新冠病毒的主题，这种单一和专业化的主题特点，使得ChatGPT在进行查询改写时，可以更为准确地捕捉到相关的语义信息，避免了因为语义的多样性而带来的改写误区。因此，我们推测对于具有明确主题和范围的数据集，ChatGPT在查询改写方面具有更好的表现，从而更有可能提升检索任务的性能。

5.2 不同长度查询的对比分析

表 5 中，“Len”一列记录了新查询的查询长度，kx 表示新查询长度是原始长度的 k 倍；n@k表示nDCG@k评价指标。根据表 5 的实验数据得知，当检索模型采用查询改写和文档得分融合机制时，不同的查询长度对检索模型的性能产生了显著的影响。从表 5 中可以看出，过于简短或冗长的查询都无法使得检索模型达到其最佳的性能水平。实验数据显示，一般情况下，查询长度达到原始长度的 5 倍或 7 倍时，检索模型的性能往往呈现出最佳表现。

这一实验结论可以从语义信息的丰富度来解释：对于长度较短的查询，其主要问题在于语义信息的匮乏。当查询词汇量有限时，其语义信息难以涵盖广泛的语义空间，导致检索模型难以准确把握用户的检索意图，也就增加了相关文档被遗漏的风险。相反，过度冗长的查询可能包含大量的非关键性信息，这不仅会增加信息处理的计算成本，还可能导致关键语义的丢失。此外，过长的查询往往导致文本中核心的语义信息被边缘化或模糊化，这使得检索模型难以识

		TREC-DL19			TREC-DL20			TREC-COVID		
	Len	n@1	n@5	n@10	n@1	n@5	n@10	n@1	n@5	n@10
BM25	-	50.77	49.02	47.95	56.48	49.65	49.36	75.00	67.51	62.32
ColBERT										
w/o rewrite	-	74.41	71.54	68.78	76.23	70.78	65.91	73.00	69.41	66.62
	1x	75.19	71.58	69.43	77.78	71.11	67.68	77.00	72.48	71.56
	3x	79.06	72.41	70.40	**79.32**	**72.91**	68.39	74.00	75.13	71.04
w/ fusion	5x	**79.85**	73.74	70.63	**79.32**	72.87	**69.08**	**78.00**	75.06	71.28
	7x	**79.85**	**74.01**	**71.50**	76.23	70.32	67.25	76.00	**75.22**	**72.62**
	9x	74.41	72.40	70.42	79.01	71.66	69.06	73.00	70.91	68.60
MonoT5										
w/o rewrite	-	77.13	73.02	69.89	78.39	70.13	67.39	78.00	74.57	72.16
	1x	77.51	73.27	70.67	**82.09**	**71.37**	**68.63**	79.00	78.20	77.41
	3x	78.29	74.32	71.36	80.86	**72.84**	68.34	76.00	78.80	76.97
w/ fusion	5x	78.29	75.67	**71.90**	78.09	69.83	67.27	**81.00**	**79.93**	**78.00**
	7x	**79.06**	**76.19**	71.68	76.23	69.46	66.16	77.00	75.55	73.10
	9x	75.96	72.21	69.45	73.14	68.33	65.48	71.00	69.31	66.37

Table 5: 不同长度的查询对检索性能的影响（n@k表示nDCG@k评价指标）

别文本的重点，从而降低了检索模型在相关文档评分上的准确性。综上所述，为了优化检索模型的性能，合适的查询长度是至关重要的。在对查询进行改写优化时，关键在于找到一个平衡点，确保查询既不失去必要的语义信息，也不包含过多的冗余信息。

5.3 不同组合权重的对比分析

检索模型首先独立地评估与原始查询和新查询相关的每个文档的相关性。然后引入一个权重分配策略，该策略更倾向于给予得分较高的查询更高的权重，即认为这些查询与查询的真实意图更为接近，因此应在最终排名中占据更重要的位置。在进行文档得分融合时，我们会对原始查询的文档得分和新查询的文档得分分配不同的权重，即设置超参数 λ 的值，如公式 1 所示。

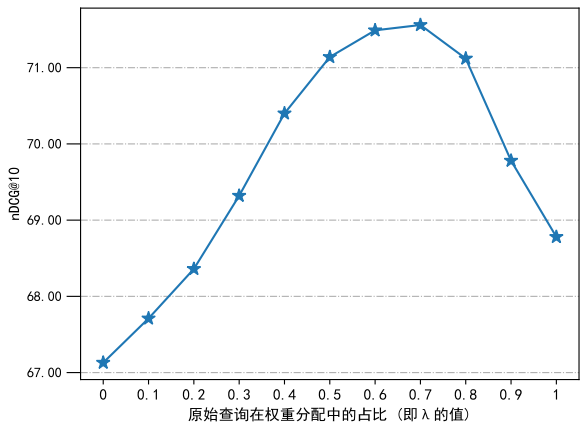
图 3 展示了在进行文档得分融合时，不同的权重分配对检索性能的影响。当 λ 值为 0 时，表示新查询的文档相关性得分；当 λ 值为 1 时，表示原始查询的文档相关性得分。图 3(a) 和图 3(c) 分别展示了ColBERT模型在DL19数据集和DL20数据集上进行文档得分融合后的检索性能，在这两个通用领域数据集上，当原始查询的权重占比为 0.7 左右时，检索模型的性能达到最高。图 3(b) 和图 3(d) 分别展示了MonoT5模型在DL19数据集和DL20数据集上进行文档得分融合后的检索性能，同样当原始查询的权重占比为 0.7 左右时，检索模型的性能达到最高。图 3(e) 和图 3(f) 分别展示了ColBERT模型和MonoT5模型在Covid-19数据集上进行文档得分融合后的检索性能，在这个专有领域的数据集上，当原始查询的权重占比为 0.4 左右时，两个检索模型的性能达到最高。

文档得分融合机制的实验结果表明，在通用领域的数据集中，超参数 λ 会存在一定的泛化能力；而在专有领域（比如医学领域）的数据集中，超参数 λ 也表现出一定的泛化能力。综上所述，超参数 λ 在相同领域的数据集中存在一定的泛化能力。因此超参数 λ 在不同领域的数据集中需要设置不同的值，但是在相同领域的数据集中可以设置相似的 λ 值。另外，图 3 中的实验结果也证明了我们的权重分配策略是合理的，即更倾向于给予得分较高的查询更高的权重。

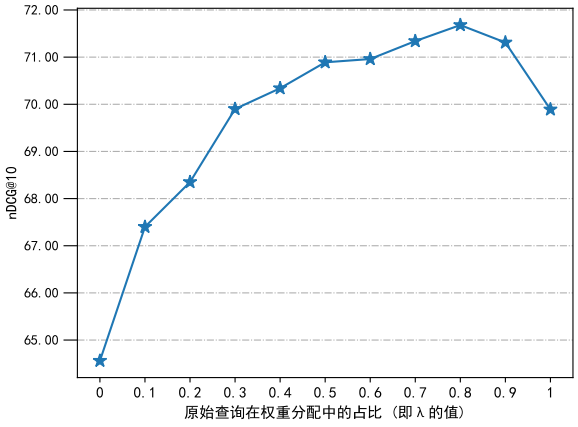
6 结语

本文提出了一种基于ChatGPT查询改写的文档检索方法。面对大语言模型生成内容的不可控性，我们设计了一种文档得分融合机制，这一机制在实验中已被证明能够显著增强检索模型的性能。针对文本长度对于检索模型性能的影响，本文证明了存在一个最佳的查询长度区间，该区间内的查询文本能够带来最佳的检索性能。此外，在两个通用领域的数据集和一个专业领

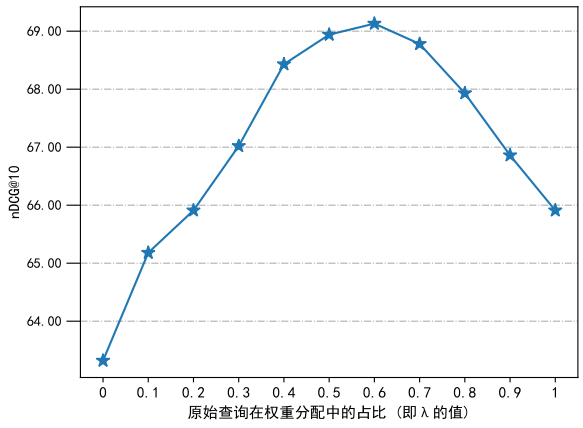
域的数据集上的大量实验表明，我们提出的方法有较好的可行性和泛化性。最后，我们未来的工作会考虑如何利用检索增强技术结合权威知识库的信息进一步完善和提高查询改写的效果。



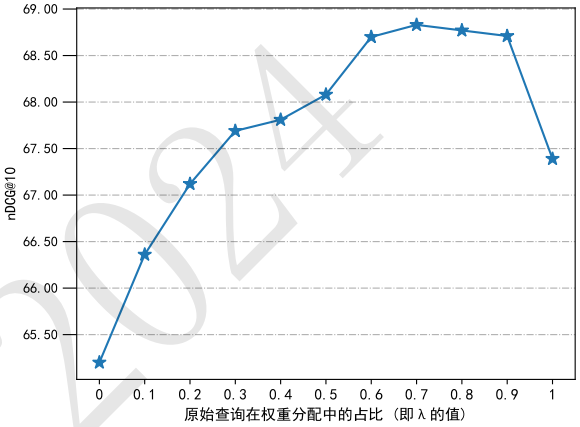
(a) ColBERT在DL19上进行文档得分融合



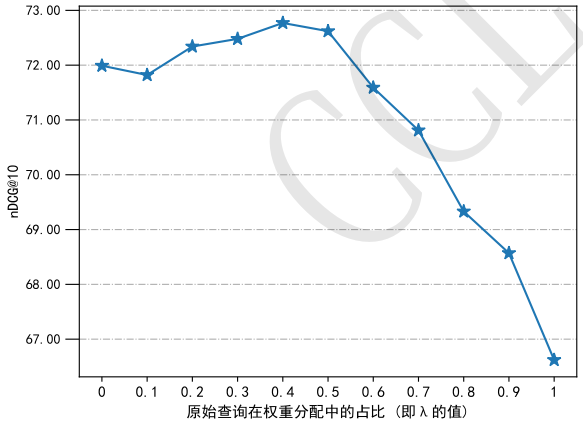
(b) MonoT5在DL19上进行文档得分融合



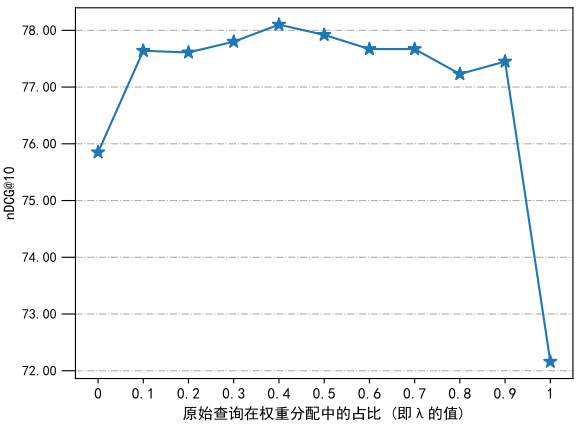
(c) ColBERT在DL20上进行文档得分融合



(d) MonoT5在DL20上进行文档得分融合



(e) ColBERT在Covid19上进行文档得分融合



(f) MonoT5在Covid19上进行文档得分融合

Figure 3: 文档得分融合时，不同的权重分配对检索性能的影响

参考文献

Nasreen Abdul-Jaleel, James Allan, W Bruce Croft, Fernando Diaz, Leah Larkey, Xiaoyan Li, Mark D Smucker, and Courtney Wade. 2004. Umass at trec 2004: Novelty and hard. *Computer Science Department Faculty Publication Series*, page 189.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Gianni Amati and Cornelis Joost Van Rijsbergen. 2002. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems (TOIS)*, 20(4):357–389.

Hiteshwar Kumar Azad and Akshay Deepak. 2019. Query expansion techniques for information retrieval: a survey. *Information Processing & Management*, 56(5):1698–1735.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. 2008. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 243–250.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2019 deep learning track. *CoRR*, abs/2003.07820.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the TREC 2020 deep learning track. *CoRR*, abs/2102.07662.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Rolf Jagerman, Honglei Zhuang, Zhen Qin, Xuanhui Wang, and Michael Bendersky. 2023. Query expansion by prompting large language models. *arXiv preprint arXiv:2305.03653*.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Hang Li, Shengyao Zhuang, Ahmed Mourad, Xueguang Ma, Jimmy Lin, and Guido Zuccon. 2022. Improving query representations for dense retrieval with pseudo relevance feedback: A reproducibility study. In *European Conference on Information Retrieval*, pages 599–612. Springer.

Hang Li. 2022. *Learning to rank for information retrieval and natural language processing*. Springer Nature.

Craig Macdonald, Nicola Tonellotto, Sean MacAvaney, and Iadh Ounis. 2021. Pyterrier: Declarative experimentation in python from bm25 to dense retrieval. In *Proceedings of the 30th acm international conference on information & knowledge management*, pages 4526–4533.

Iain Mackie, Shubham Chatterjee, and Jeffrey Dalton. 2023. Generative relevance feedback with large language models. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2026–2031.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Ronak Pradeep, Rodrigo Nogueira, and Jimmy Lin. 2021. The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models. *arXiv preprint arXiv:2101.05667*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Pengjie Ren, Dawei Yin, and Zhaochun Ren. 2023. Is chatgpt good at search? investigating large language models as re-ranking agent. *arXiv preprint arXiv:2304.09542*.

Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Siamak Shakeri, Dara Bahri, Tal Schuster, et al. 2022. Ul2: Unifying language learning paradigms. *arXiv preprint arXiv:2205.05131*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Ellen Voorhees, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, William R Hersh, Kyle Lo, Kirk Roberts, Ian Soboroff, and Lucy Lu Wang. 2021. Trec-covid: constructing a pandemic information retrieval test collection. In *ACM SIGIR Forum*, volume 54, pages 1–12. ACM New York, NY, USA.

Liang Wang, Nan Yang, and Furu Wei. 2023a. Query2doc: Query expansion with large language models. *arXiv preprint arXiv:2303.07678*.

Xiao Wang, Craig Macdonald, Nicola Tonellotto, and Iadh Ounis. 2023b. Colbert-prf: Semantic pseudo-relevance feedback for dense passage and document retrieval. *ACM Transactions on the Web*, 17(1):1–39.

Tianyu Wu, Shizhu He, Jingping Liu, Siqi Sun, Kang Liu, Qing-Long Han, and Yang Tang. 2023. A brief overview of chatgpt: The history, status quo and potential future development. *IEEE/CAA Journal of Automatica Sinica*, 10(5):1122–1136.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Zhi Zheng, Kai Hui, Ben He, Xianpei Han, Le Sun, and Andrew Yates. 2020. Bert-qe: contextualized query expansion for document re-ranking. *arXiv preprint arXiv:2009.07258*.