大语言模型和知识图谱协同的查询扩展方法

张旷，涂新辉*，刘晗
华中师范大学 计算机学院，湖北 武汉，430079
kuangzhang@mails.ccnu.edu.cn, tuxinhui@ccnu.edu.cn,
liuhan@mails.ccnu.edu.cn

摘要

查询扩展旨在通过丰富查询来提升检索效果。在大语言模型结合伪相关反馈的查询扩展方法中，伪相关文档中的噪声及不连贯信息严重影响了大语言模型的扩展质量。为此，本文提出一种大语言模型和知识图谱协同的查询扩展方法（LKQE）。LKQE 首先检索出相关文档并提取关键句，然后利用大语言模型从中抽取知识三元组，并补全实体关系构建出知识图谱，最终在知识图谱指导下生成高质量扩展文本。实验结果表明，与基线模型相比，LKQE 在 DL19 与 DL20 数据集上的表现具有显著优势。

关键词： 信息检索 ； 查询扩展 ； 大语言模型 ； 知识图谱

Query Expansion via Collaboration between Large Language Models and Knowledge Graphs

ZHANG Kuang, TU Xinhui*, LIU Han
School of Computer Science, Central China Normal University, Wuhan Hubei, 430079
kuangzhang@mails.ccnu.edu.cn, tuxinhui@ccnu.edu.cn,
liuhan@mails.ccnu.edu.cn

Abstract

Query expansion aims to improve retrieval performance by enriching the original query. However, in query expansion methods based on large language models(LLMs) that incorporate pseudo-relevant feedback, the presence of noise and incoherent information in feedback documents often degrades the quality of the generated expansions. To address this issue, we propose LKQE, a query expansion via collaboration between large language models and knowledge graphs. Specifically, LKQE first retrieves relevant documents and extracts key sentences, then leverages LLMs to extract knowledge triples and supplement entities and relations to construct a complete knowledge graphs. Finally, guided by the knowledge graphs, the LLMs generates high-quality expanded queries. Experimental results on the DL19 and DL20 datasets demonstrate that LKQE shows a clear performance advantage compared to existing baseline methods.

Keywords: Information Retrieval , Query Expansion , Large Language Models , Knowledge Graphs

* 通讯作者

1 引言

信息检索旨在根据用户提交的查询，从大规模文档库中检索出相关文档。然而，由于用户提交的查询往往较为简短甚至模糊，缺乏充足的上下文语义信息，导致检索系统难以准确理解用户真实的检索意图，因此系统返回的结果往往与用户期望存在较大偏差。为了解决这一问题，查询扩展技术被广泛研究并应用。查询扩展的核心思想是对用户提交的原始查询进行语义丰富化，具体表现为在原始查询的基础上补充或替换某些词汇或短语，以缩短查询与文档间的语义距离。扩展后的查询能够提高查询的语义丰富度，显著增强了检索系统对潜在相关文档的识别能力。

近年来，随着大语言模型(Zhao et al., 2023)技术的快速发展，一些大语言模型结合伪相关反馈(Cao et al., 2008)的查询扩展方法取得了不错的效果。借助伪相关反馈框架，基于大语言模型的查询扩展方法能够结合相关文档中的潜在信息，进一步提升扩展的质量和相关性。具体而言，首先利用原始查询对文档集合进行初步检索，从返回结果中选取前若干个高置信度文档，作为伪相关文档；随后利用这些伪相关文档，作为提示信息，用于引导大语言模型生成更具针对性和上下文关联性的扩展关键词或文本。例如，Query2Term+PRF (Jagerman et al., 2023)通过伪相关文档提示大语言模型生成若干关键词，以丰富原始查询的表达；Query2Doc+PRF (Wang et al., 2023)则通过伪相关文档引导模型生成一段针对原始查询的完整回答，从而补充查询语义；而 MILL (Jia et al., 2023)则是通过伪相关文档和大语言模型生成的扩展文档进行相互验证，最终筛选出最优的扩展结果。

尽管伪相关文档一定程度上增强了大语言模型生成扩展文本的语义一致性，但由于这些伪相关文档本身往往来自不同文本片段，存在语义不连贯、冗余甚至噪声的问题，这些因素不可避免地限制了大语言模型的生成质量。此外，这类方法主要依赖语言模型对上下文文本的隐式语义建模，而缺乏明确的结构化知识支撑，导致其在处理查询中的实体关系识别、语义歧义消解以及深层语义推理方面存在明显不足。

为了进一步提高基于大语言模型的查询扩展方法的语义准确性和内容可控性，本文引入了知识图谱(Fensel et al., 2020)作为辅助知识源。知识图谱是一种结构化的知识表示工具，通过有向标记图的形式清晰地描述实体及其相互之间的语义关系，能够以更加细粒度、紧凑且高效的方式组织信息(Kau et al., 2024)。与直接使用原始文本信息进行知识注入相比，知识图谱经过了数据清洗、去重和语义整合过程，提供了更为纯净且高质量的知识单元。因此，知识图谱的引入可以有效缓解大语言模型在生成过程中易产生的幻觉现象，减少生成文本中的噪声，并使生成内容的语义更加明确且易于解释。

基于此，本文提出了一种大语言模型和知识图谱协同的查询扩展方法（LKQE），通过五个步骤实现高质量的查询扩展。首先，使用传统的 BM25 检索模型从大规模语料库中获取与原始查询高度相关的初步文档集合；随后，从该初步结果中提取与查询最密切相关的关键句子，并进行聚合，以确保获得的上下文信息更具代表性；第三，利用大语言模型对聚合后的文本进行深度处理，抽取其中的核心实体与实体之间的关系，从而形成规范化的三元组表示；第四，基于原始查询及已抽取的三元组信息，进一步补充和完善实体及关系，构建出全局的结构化知识图谱。最后，利用构建好的知识图谱引导大语言模型生成更加精确、语义连贯且具备可解释性的扩展查询文本。实验结果表明，在 DL19 和 DL20 两个公共数据集上的性能评估中，本文提出的 LKQE 方法相较于现有的主流方法表现出显著优势，有效提高了查询扩展的质量和检索系统的整体性能。

2 相关工作

2.1 查询扩展

查询扩展是信息检索领域的经典技术，其核心思想是通过向原始查询中添加与之语义相近的词汇，扩充语义信息，从根本上提升检索系统的性能。

传统的查询扩展方法大多基于伪相关反馈的扩展。该方法假设初次检索结果中排名靠前的若干文档与原始查询具有高度相关性，并以此为基础抽取出频率或权重较高的词汇扩展原查询例如，RM3 (Abdul-Jaleel et al., 2004)通过分析排名前几的文档来估计词项的相关性分布，并利用该分布对查询进行扩展；KL 扩展(Amati and Van Rijsbergen, 2002)通过计算查询与文档之间的KL散度来度量查询与文档模型的差异，从而选择性地扩展查询。然而，由于初次检索结果

的质量难以保证，一旦前期结果中包含大量噪声文档，将严重干扰查询扩展过程，导致扩展查询内容偏离用户最初的检索意图，从而降低检索性能。

近年来，基于大语言模型的查询扩展方法已成为信息检索领域的研究热点。大语言模型凭借卓越的语言理解与生成能力，能够生成更契合用户意图的扩展查询。Query2Doc(Wang et al., 2023)提出了一种结合少样本提示策略的框架，利用大语言模型生成语义相关的“伪文档”，并据此扩展原始查询。与传统方法不同，该框架无需依赖初步检索结果，而是依赖于大语言模型固有的知识生成语义丰富的伪文档，增强查询表达，显著改善检索效果。

此外，将大语言模型与伪相关反馈机制相结合，已成为提升查询扩展效果的重要途径。例如，MILL (Jia et al., 2023)将检索得到的文档与大语言模型生成的文档进行相互验证，选取最相关的信息从而实现最佳扩展。这类方法利用初步检索结果中的潜在相关文档，同时结合大语言模型生成更具针对性和语义丰富度的扩展内容，能够有效增强扩展文本的准确性和相关性，从而提升检索性能。然而，尽管伪相关文档在相关性上与原始查询较为匹配，但由于其通常由多个不同文本片段拼接而成，语义连贯性较差，且往往包含冗余信息和噪声，这些问题在很大程度上限制了大语言模型的生成质量。因此，本文通过引入知识图谱，与大语言模型协同完成查询扩展任务。

2.2 大语言模型

大语言模型的发展历程可以追溯到2017年Transformer (Vaswani et al., 2017)架构的提出，为大规模预训练模型的诞生奠定了基础。OpenAI于2018年推出的GPT系列标志着这一领域的突破，特别是GPT-3 (Brown et al., 2020)通过1750亿参数实现了前所未有的语言生成能力。随着ChatGPT的发布，通过引入强化学习与人类反馈（RLHF）(Ouyang et al., 2022)，大语言模型在对话系统中展现了更强的交互性与推理能力。2023年以来，GPT-4 (Achiam et al., 2023)的发布进一步扩展了在多模态任务中的能力，使得模型在更复杂的任务中展现出卓越的性能，如图文问答、视觉-语言推理等任务，近期发布的GPT-4.1 (Taniguchi and Lindsey, 2025)更是在编程和指令遵循方面，取得了重大进展，进一步提升了其应用场景的广度和深度。此外，Meta发布的LLaMA (Touvron et al., 2023)系列开源模型也展示了较小参数规模模型在多个NLP任务中的优越性。

2024年，随着技术的不断进步，大模型在复杂问题解决和深度推理能力上取得了显著进展，OpenAI-o1 (Jaech et al., 2024)，OpenAI-o3 (Pfister and Jud, 2025)等推理模型的推出，标志着人工智能在模拟人类思维模式上迈出了重要一步。2025年初，中国推出了具有开创性且高性价比的大型语言模型DeepSeek-R1(Guo et al., 2025)。DeepSeek-R1在数学计算、代码生成、自然语言推理等关键领域表现出色，性能已比肩OpenAI的GPT-o1正式版。此外，DeepSeek-R1还以其“超成本效益”和“开源”设计挑战了AI领域的传统规范，推动了先进大模型的普及。

2.3 融合大语言模型与知识图谱的方法

融合大语言模型与知识图谱的方法是近年来自然语言处理一个重要的发展方向，大语言模型与知识图谱的融合可以实现两者各自的优势互补，推动更高效、更智能的知识处理和推理。

一方面，知识图谱作为一种结构化、可验证的知识表示形式，能够为大语言模型提供稳定、权威的事实支撑，显著提升其生成内容的准确性和可解释性，缓解“幻觉”问题。例如，在语义推理方面，近期提出的R3方法(Toroghi et al., 2024)利用知识图谱增强问答任务中的推理过程。该方法将常识性知识图谱问答问题转化为树状结构的搜索任务，借助大语言模型的语言能力配合图谱中的常识公理，实现了对推理路径的结构化建模。这种方法不仅提升了模型的推理准确性，还使得推理过程更具可验证性，从而有效实现了将来自知识图谱的语义理解能力注入到大语言模型中。

另一方面，大语言模型凭借着在大规模语料上训练所获得的丰富语言知识与语义理解能力，也能为知识图谱的构建与完善提供了全新手段。例如，BertNet (Hao et al., 2023)通过从大语言模型中抽取任意关系的实体对，进而构建通用知识图谱。该方法通过多轮改写初始提示，利用大语言模型对改写后提示的回答生成候选实体对，并进行排序，最终保留排名靠前的实体对作为图谱的组成部分。Kommineni (2024)等人提出了一种半自动化的知识图谱构建流程，借助ChatGPT-3.5强大的生成能力，首先生成涵盖特定主题的高层次问题，然后进一步引导模型抽取问题中的实体与关系，构建本体，并将文档中提取的信息映射至该本体以构建知识图谱。

3 方法

本文提出的查询扩展方法主要包括5大核心模块。检索模块：检索出与查询相关的前k个文档；关键句提取模块：从前k个文档中提取出与查询直接相关的句子并聚合；三元组抽取模块：从聚合的信息中抽取实体关系三元组；知识图谱扩展模块：识别原始查询与三元组语义联系，扩展三元组，完善知识图谱；查询扩展模块：结合原始查询和知识图谱生成的高质量扩展文本。LKQE的算法如表1所示。我们在附录A中提供了各子模块的提示词。

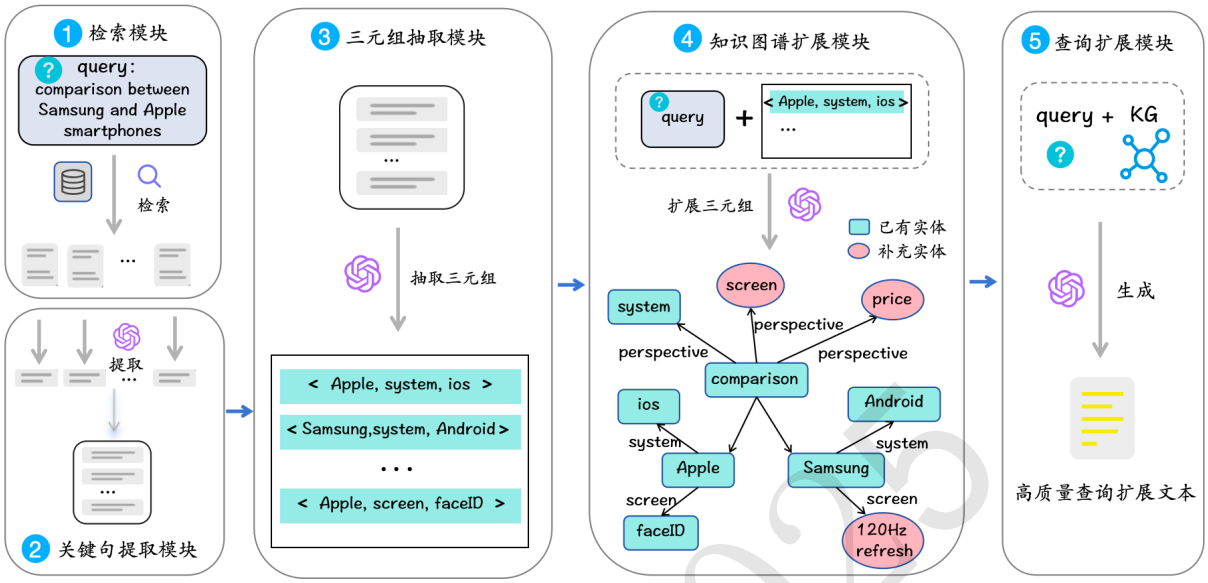


图 1. LKQE 框架图

3.1 检索模块

检索模块采用BM25检索模型，根据用户查询从文档集合中检索出前k个最相关的文档$D = \{d1, d2, ..., dk\}$。BM25模型的公式核心是词频（TF）和逆文档频率（IDF），仅依赖文本索引，根据相关性对文档进行排序。因此具有高效性和良好的检索性能，可以直接用于大规模数据集检索。该模块可以快速从大规模语料库中检索出相关文档，确保为后续处理提供高质量候选文档，保证语义扩展的基本方向，如式（1）所示。

$$D = BM25(query) = \{d_1, d_2, \ldots, d_k\} \tag{1}$$

3.2 关键句提取模块

从检索模块接收到前k个文档$D = \{d_1, d_2, ..., d_k\}$后，关键句提取模块使用大语言模型来理解并提取与原始查询最相关的句子，同时过滤掉不相关的内容。具体而言，对于每个文档d_i，我们提取出关键句子$\{s_1, s_2, ..., s_k\}$，并将它们聚合成一个集合S，如式（2）所示。此过程可确保只有每个文档中最相关的信息才会传递给后续模块，从而确保在后续构建知识图谱的过程中能够提取出关键信息，提高查询扩展的精度。

$$S = LLM(prompt_1, D) = \{s_1, s_2, \ldots, s_n\} \quad (n \geq k) \tag{2}$$

3.3 三元组抽取模块

关键句提取模块所提取的句子虽然与原始查询直接相关，但句子之间彼此独立，缺乏连贯的上下文语义，难以直接支持后续的查询扩展。因此，三元组抽取模块的核心目标，是对提取模块中的句子集合S进行重新整理和结构化表达。在获得相关句子集合S后，三元组抽取模块从中抽取结构化的实体关系信息，整理成三元组的形式，具体步骤如下：首先，利用大语言模型，从句子集合S中提取实体集$Ei = \{e_1, e_2, ..., e_m\}$和关系集$Ri = \{r_1, r_2, ..., r_k\}$，然后将其组织为结构化的三元组$\{e_i, r_i, e_i^{'}\}$，如式（3）所示。

$$T = LLM(prompt_2, S) = \{\langle e_1, r_1, e_1^{'}\rangle, \langle e_2, r_2, e_2^{'}\rangle, \dots \langle e_m, r_m, e_m^{'}\rangle\} \tag{3}$$

LKQE 算法:

输入：用户查询q
输出：扩展后的查询q'
对于每一个查询q:
检索出前k个相关文档:
$D = BM25(q) = \{d_1, d_2, \dots, d_k\}$
提取出相关文档中的关键句:
$S = LLM(prompt_1, D)$
从关键句中抽取出三元组:
$T = LLM(prompt_2, S)$
扩展三元组完善知识图谱:
$KG = LLM(prompt_3, q, T)$
生成查询扩展文本:
$expansion = LLM(prompt_4, q, KG)$
新查询：$q' = concat(3 * q, expansion)$

表 1. LKQE 算法

3.4 知识图谱扩展模块

知识图谱扩展模块结合原始查询$query$和已有三元组T之间的语义关联，分析知识图谱中仍缺失的实体与关系，从而补充三元组，提高知识图谱的完整性。具体来说，模块首先对$query$和已有三元组$(h, r, t) \in T$进行语义对齐，识别$query$所隐含但尚未在T中显式体现的信息片段。随后，模块基于语言模型的推理能力，生成包含新实体与新关系的补充三元组(h', r', t')，这些三元组与原始查询语义紧密相关。最终，扩展后的知识图谱KG包含了更全面的实体与关系，能够为查询扩展模块提供更加充分的语义支持，如式（4）所示。

我们的策略主要基于“横向”与“纵向”两个维度对图谱进行扩展，具体说明如下：横向扩展（实体扩展）：我们尝试为当前图谱中已有的核心实体引入更多“兄弟节点”或“语义相关实体”。纵向扩展（关系深化）：对于已有实体对，我们引导LLM挖掘更多潜在的关系。表2展示了DL19中的知识图谱扩展示例。最终，构建的知识图谱KG将作为查询扩展模块的输入，为大语言模型生成更精准条理的查询扩展文本提供结构化知识支撑。

$$KG = LLM(prompt_3, query, T) = \{\langle e_1, r_1, e_1^{'}\rangle, \langle e_2, r_2, e_2^{'}\rangle, \dots \langle e_n, r_n, e_n^{'}\rangle\} \tag{4}$$

3.5 查询扩展模块

在知识图谱KG构建完成后，查询扩展模块利用该结构化信息，结合原始查询$query$，生成高质量的查询扩展文本。具体而言，该模块结合原始查询、知识图谱中的实体关系三元组，利用大语言模型的理解和推理能力，充分理解查询的意图，生成最终的查询扩展文本。该扩展文本不仅涵盖原始查询的核心信息，还融入了知识图谱提供的额外语义信息，既够从知识图谱的全局结构考虑，保证生成文本的完整性和条理性，又能充分利用三元组表达的关键实体和联系，从而确保生成查询扩展的质量，如式（5）所示。

经过5个模块的信息整合与处理后，我们将生成的查询扩展文本$expansion$与原始查询$query$结合，构建新的查询输入。值得注意的是，考虑到扩展后的查询长度会覆盖掉一部分原始查询的语义空间，我们将原始查询$query$重复三次，以强化其在最终扩展文本中的语义权重，如式（6）所示。

$$expansion = LLM(prompt_4, query, KG) \tag{5}$$

$$query' = \text{concat}(3 * query, expansion) \tag{6}$$

第二十四届中国计算语言学大会论文集，第422页-第433页，济南，中国，2025年8月11日至14日。

总体而言，LKQE是一种结合知识图谱优化伪相关文档注入，协同大语言模型共同工作的查询扩展方法。并且，LKQE的整个过程不需要对任何模型进行额外的训练和微调，仅依靠大语言模型的固有能力在少样本环境下生成高质量的查询扩展文本。这种基于现成的大语言模型和检索模型的方法在各种搜索系统中具有巨大的应用潜力。

Input	**Output**
Query:	**Output KG Triples:**
what are the three distinct layers of the earth?	⟨ Earth; has layers; crust, mantle, core ⟩
	⟨ Crust; type; continental and oceanic ⟩
	⟨ Crust; thickness; thinner than mantle ⟩
Existing Knowledge Triples:	⟨ Crust; function; where all life on Earth exists ⟩
⟨ Earth; layer composition; four distinct layers ⟩	⟨ Mantle; depth; about miles ⟩
⟨ Inner core; location; centre ⟩	⟨ Mantle; function; movement of tectonic plates ⟩
⟨ Mantle; description;widest section of the Earth⟩	⟨ Mantle; function; generation of volcanic activity ⟩
⟨ Crust, Mantle, Core; type; chemical layers ⟩	⟨ Core; division; liquid outer core and solid inner core ⟩
⟨ Crust; location; surface of the Earth ⟩	⟨ Outer core; composition; molten iron and nickel ⟩
⟨ Crust; composition; chemically distinct layer ⟩	⟨ Inner core; composition; solid metal due to pressure ⟩
	⟨ Earth; has layers; working together ⟩

表 2. 知识图谱扩展模块在DL19上的扩展示例。当多个尾实体共享一个头实体时，我们用逗号隔开。例如，⟨ Earth; has layers; crust, mantle, core ⟩。

4 实验设计

4.1 数据集和评价指标

本文的方法在TREC-DL-2019和TREC-DL-2020这两个广泛使用的公共数据集上进行了系统性实验，以评估其在实际信息检索任务中的有效性和稳定性。TREC-DL-2019和TREC-DL-2020数据集基于MS MARCO语料库构建，涵盖了新闻文章、博客和网页内容等多种类型。数据集由专业评审员进行高质量相关性标注，广泛用于评估深度学习检索模型的性能。表3统计了这两个数据集的查询(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。

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

表 3. 数据集统计信息

关于评价指标，本文实验中，我们主要关注MAP、nDCG@10、Recall@1k，从多维度来衡量模型的排序性能。MAP：计算所有查询的平均精度，反映整体检索系统的精度表现。nDCG@10：衡量检索结果的排序质量，考虑相关性和位置权重，@10表示关注前10个结果。Recall@1k：衡量前1000个返回文档中包含的相关文档比例，侧重于召回能力。

4.2 基线模型

为了全面评估我们所提出方法的有效性，我们选择了多种具有代表性的基线模型进行对比，以确保对比的全面性和权威性。具体而言，我们采用以下几类基线方法：

（1）传统的查询扩展方法：Bo1、KL、RM3。

（2）基于大语言的查询扩展方法：Query2Term、Query2Term-FS（Query2Term的少样本版本）、Query2Term-PRF（PRF文档增强Query2Term）、Query2Doc、Query2Doc-FS、Query2Doc-PRF、CoT(Jagerman et al., 2023)、CoT-PRF、MILL。

	TREC-DL19			TREC-DL20		
	MAP	nDCG@10	Recall@1k	MAP	nDCG@10	Recall@1k
BM25	37.00	49.75	73.62	35.87	49.36	75.12
传统的查询扩展方法						
Bo1	39.99	50.86	75.11	39.67	49.47	79.48
KL	39.77	50.57	74.66	39.53	49.27	79.39
RM3	40.45	51.56	75.43	40.22	50.43	79.94
基于大语言模型的查询扩展方法						
Query2Term	29.91	44.17	68.81	38.49	50.12	79.07
Query2Term-FS	37.51	50.38	76.13	35.59	47.80	78.76
Query2Term-PRF	37.18	48.56	70.20	33.70	47.76	76.68
Query2Doc	49.04	62.77	84.21	47.03	61.22	83.38
Query2Doc-FS	49.65	63.83	83.75	45.27	61.45	82.57
Query2Doc-PRF	44.56	59.00	82.12	43.49	55.28	82.57
CoT	45.67	63.44	83.43	42.34	58.39	80.11
CoT-PRF	44.73	61.63	80.37	44.04	60.81	80.49
MILL	**53.11**	63.80	85.92	48.17	61.79	**85.27**
LKQE(ours)	50.88	**66.23**	**86.27**	**49.15**	**64.27**	84.61

表 4. LKQE 在 TREC 数据集上的表现

4.3 超参数设置

本文的实验均使用 PyTerrier 框架完成。对于检索模块，我们使用 BM25 检索模型并默认参数，词项频率饱和度参数 $k_1 = 1.3$，文档长度归一化参数 $b = 0.75$。对于大语言模型，我们使用 OpenAI 的 API 接口实现对 GPT-3.5-turbo 模型的调用，随机度参数 $temperature = 0.7$，核采样参数 $top\_p = 0.9$，最大生成长度参数 $max\_takens = Infinite$，以确保模型输出的完整性。对于实验过程中部分超参数，我们设置如下：检索模块中检索 top_k 个文档，k 设置为 8；查询扩展模块生成的扩展文本长度 len 限制为原始查询的 15 倍。

5 实验结果及分析

5.1 整体性能分析

表 4 展示了多种查询扩展方法在 TREC-DL19 与 TREC-DL20 两个数据集上的性能表现，涵盖传统方法、基于大语言模型的方法以及我们提出的 LKQE 方法。从整体结果来看，我们的方法在各项指标上均表现优异。

在TREC-DL19 上，LKQE 方法在 nDCG@10 和 Recall@1k上分别达到了66.23%和86.27%，在所有方法中排名第一，仅在 MAP 上低于 MILL。值得注意的是，尽管 MILL 在 MAP 上领先，但 LKQE 在 nDCG@10 和 Recall@1k 上具有明显优势，表明其在实际检索中能够更有效地提升前排结果的相关性与整体覆盖率。在 TREC-DL20上，LKQE 在 MAP 和 nDCG@10 上取得了 49.15% 和 64.27% 的最高成绩，同时 Recall@1k 达到 84.61%，仅次于MILL的 85.27%，显示出强大的通用性与稳定性。

相较于传统查询扩展方法（如Bo1、KL、RM3），LKQE 在所有指标上均有显著提升，这表明大语言模型和知识图谱协同能够有效地补充原始查询的语义信息。同时，相比于其他基于大语言模型的扩展方法（如 Query2Doc、CoT、MILL 等），LKQE 在保持良好召回率的同时进一步提升了排序质量和 MAP，展现出更强的综合检索能力。

综上所述，LKQE 方法在两个评测数据集上均取得了最优或次优的性能，验证了我们的方法在查询扩展任务中的有效性与稳定性。

	TREC-DL19			TREC-DL20		
	MAP	nDCG@10	Recall@1k	MAP	nDCG@10	Recall@1k
LKQE	**50.88**	**66.23**	**86.27**	**49.15**	**64.27**	**84.61**
w/o M_4	48.36	63.19	84.58	47.68	62.71	82.50
w/o M_4&M_3	47.12	62.41	82.74	46.13	61.09	81.58

表 5. 消融实验结果

5.2 消融实验

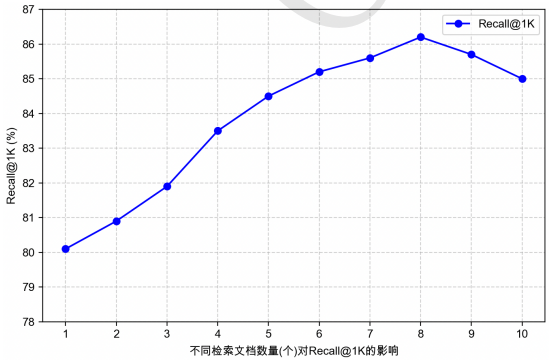
在本实验中，我们分析方法中各个模块对最终查询扩展效果的贡献，验证大语言模型协同知识图谱是否提升了查询扩展的质量。为了评估各模块的作用，我们设计了以下消融实验：（1）完整模型（LKQE）：包括检索模块 M_1、关键句提取模块 M_2、三元组抽取模块 M_3、知识图谱扩展模块 M_4、查询扩展模块 M_5。（2）去除知识图谱扩展模块（*w/o* M_4）：直接用抽取的三元组作为上下文进行查询扩展，不进行知识图谱扩展。（3）去除知识图谱扩展模块和三元组抽取模块（*w/o* M_4&M_3）：直接用关键句子作为上下文进行扩展。

根据实验结果表 5 我们可以得到以下结果：（1）LKQE 的核心模块对的查询扩展质量的提升都有贡献。（2）去除知识图谱扩展模块，三个指标都明显下降。说明知识图谱扩展模块能够为原始查询扩展了大量关键实体信息与关系信息，知识图谱提供的结构化信息有助于大语言模型生成更流畅、信息更丰富的扩展查询。（3）去除知识图谱扩展模块和三元组抽取模块，三个指标进一步下降。说明三元组抽取模块抽取的三元组与关键句子相比，能够更好的表达信息之间的语义关系。（4）*w/o* $M4$ 要比 *w/o* M_4&M_3 下降幅度大，说明知识图谱扩展模块补充了丰富的三元组信息，为查询扩展模块提供了详细的指导。

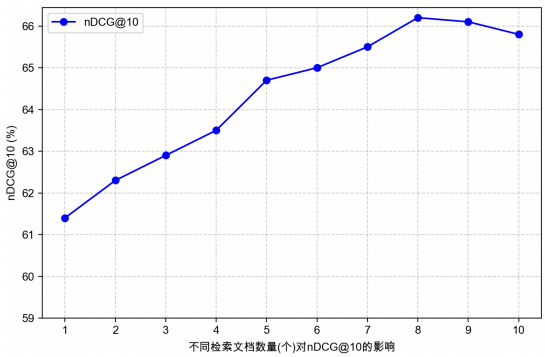
5.3 不同检索文档数量实验对比分析

我们设定初次检索文档数量为从 1 到 10 个文档，并分析其对检索效果的影响。我们使用 nDCG@10 和 Recall@1k 作为评测指标来评估不同检索文档数量对检索效果的影响，下图展示了在 DL19 上不同检索文档数量对检索效果的影响。（1）召回率（Recall@1k）：随着初次检索文档数量的增加，召回率逐渐提高。例如，N=8 时，召回率达到了 86.27%，比 N=1 时的 80.11% 提高了 6.16%。这表明，增加初始检索的文档数量能够提供大量的与原始查询相关的上下文信息，进而提升召回能力。（2）排序质量（nDCG@10）：nDCG@10 也随着初次检索文档数量的增加而提高。例如，N=8 时，nDCG@10为 66.23%，远高于 N=1 时的 61.46%。增加初次检索的文档数量帮助系统在扩展查询时更好地排序相关文档，提高了检索的质量。

因此，我们可以得出，初次检索文档数量对召回率和排序质量有显著影响。随着初始检索文档数量的增加，召回率和排序质量逐渐提高，尤其在 N=8 时效果最为显著。



(a) 不同检索文档数量 N 对 Recall@1k 的影响



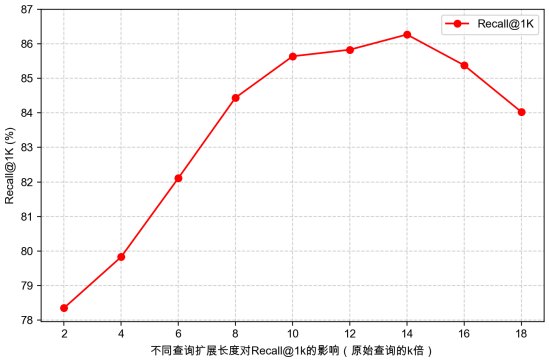
(b) 不同检索文档数量 N 对 nDCG@10 的影响

5.4 不同查询扩展长度实验对比分析

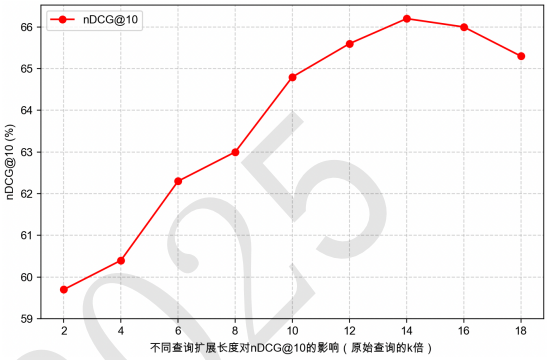
我们通过将查询扩展的文本长度设置为原始查询的 K 倍，分析不同扩展长度对检索效果的影响。具体而言，我们设定以下实验设置：设置 K = 2, 4, 6, 8 ,10, 12, 14 ,16, 18。同样使

用nDCG@10和Recall@1k作为评测指标进行评估。下图展示了不同查询扩展长度在DL19上的检索效果差异，通过分析实验数据，我们可以得到以下结果：（1）召回率（Recall@1k）：随着扩展长度的增加（K倍从2到14），召回率逐步提高。这表明，增加扩展查询的长度可以覆盖更多可能相关的信息，尤其在查询本身信息有限的情况下，扩展能够帮助系统召回更多相关结果。具体来说，K=14的扩展查询召回了最多的相关文档（86.27%），相比于K=2时的78.31%，提升了7.96%。（2）排序质量（nDCG@10）：排序质量也随着扩展长度增加而提高，表明扩展查询不仅召回更多文档，而且排序结果更符合用户需求。特别是K=14时，nDCG@10达到66.23%，远高于K=2时的59.75%，这意味着更长的查询扩展能有效提升排序质量，帮助检索系统更准确地排列相关文档。（3）扩展查询长度（K倍）与检索效果呈正相关，增加查询长度可以显著提高召回率和排序质量，尤其在查询信息不足时，查询扩展能显著提升检索系统的性能。过长的查询扩展（K倍较大）可能带来冗余信息，虽然性能提升但边际效益逐渐减小，在某些情况下，扩展查询的长度并不是越长越好。

从实验结果来看，K=14的扩展长度是召回率、排序质量、扩展质量等方面的最佳平衡点。此时，查询扩展能够有效提升检索系统的性能，同时避免过长查询带来的冗余信息。



(a) 不同查询扩展长度K对Recall@1k的影响



(b) 不同查询扩展长度K对nDCG@10的影响

5.5 不同模型效果对比

	TREC-DL19			TREC-DL20		
	MAP	nDCG@10	Recall@1k	MAP	nDCG@10	Recall@1k
GPT-3.5-turbo	50.88	**66.23**	**86.27**	49.15	64.27	**84.61**
DeepSeek-R1-8B	**51.42**	65.96	86.19	**49.64**	**65.28**	84.17
Llama-3-8B-Instruct-fp16	48.81	63.73	83.45	47.21	62.82	82.33

表 6. 不同模型效果对比

在表6中，我们比较了三种主流语言模型在DL19与DL20数据集上的检索性能。整体来看，GPT-3.5-turbo在Recall@1k上表现最优，分别达到86.27%（DL19）和84.61%（DL20），显示出其在提升召回方面的优势。DeepSeek-R1-8B在MAP和nDCG@10上表现稳定，分别在两个数据集中取得了最高或次高分，说明其对高质量文档排序具有较强能力。相比之下，Llama-3-8B-Instruct-fp16在各项指标上略低，但仍维持较为合理的性能水平，体现出开源模型在特定场景下的实用潜力。

6 结语

本文针对伪相关反馈文档在大语言模型查询扩展中的语义不连贯与噪声干扰问题，提出了一种大语言模型和知识图谱协同的查询扩展方法（LKQE）。该方法通过引入结构化的实体关系信息，有效提升了扩展文本的语义准确性和相关性。实验结果表明，LKQE在公开检索数据集DL19和DL20上均取得了优于现有主流方法的效果，验证了其在复杂信息检索场景下的有效性和通用性。未来工作中，我们将进一步探索将知识图谱与大语言模型的推理能力相结合，提升模型在复杂语义理解和深层关系推断场景下的泛化能力。

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.

Hasan Abu-Rasheed, Christian Weber, and Madjid Fathi. 2024. Knowledge graphs as context sources for LLM-based explanations of learning recommendations. In *2024 IEEE Global Engineering Education Conference (EDUCON)*, pages 1–5. IEEE.

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.

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. In *Advances in Neural Information Processing Systems*, volume 33, pages 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.

Xiaojun Chen, Shengbin Jia, and Yang Xiang. 2020. A review: Knowledge reasoning over knowledge graph. *Expert Systems with Applications*, 141:112948. Elsevier.

Zhe Chen, Yuehan Wang, Bin Zhao, Jing Cheng, Xin Zhao, and Zongtao Duan. 2020. Knowledge graph completion: A review. *IEEE Access*, 8:192435–192456. IEEE.

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.

Jinyuan Fang, Zaiqiao Meng, and Craig Macdonald. 2025. KiRAG: Knowledge-Driven Iterative Retriever for Enhancing Retrieval-Augmented Generation. *arXiv preprint arXiv:2502.18397*.

Jinyuan Fang, Zaiqiao Meng, and Craig Macdonald. 2024. TRACE the evidence: Constructing knowledge-grounded reasoning chains for retrieval-augmented generation. *arXiv preprint arXiv:2406.11460*.

Dieter Fensel, Umutcan Şimşek, Kevin Angele, Elwin Huaman, Elias Kärle, Oleksandra Panasiuk, Ioan Toma, Jürgen Umbrich, Alexander Wahler, Dieter Fensel, et al. 2020. Introduction: what is a knowledge graph? In *Knowledge Graphs: Methodology, Tools and Selected Use Cases*, pages 1–10. Springer.

Dieter Fensel, Umutcan Şimşek, Kevin Angele, Elwin Huaman, Elias Kärle, Oleksandra Panasiuk, Ioan Toma, Jürgen Umbrich, Alexander Wahler, Dieter Fensel, et al. 2020. Introduction: What is a Knowledge Graph? In *Knowledge Graphs: Methodology, Tools and Selected Use Cases*, pages 1–10. Springer.

Jiazhan Feng, Chongyang Tao, Xiubo Geng, Tao Shen, Can Xu, Guodong Long, Dongyan Zhao, and Daxin Jiang. 2023. Synergistic Interplay between Search and Large Language Models for Information Retrieval. *arXiv preprint arXiv:2305.07402*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2501.12948*.

Shibo Hao, Bowen Tan, Kaiwen Tang, Bin Ni, Xiyan Shao, Hengzhe Zhang, Eric P. Xing, and Zhiting Hu. 2023. BERTNet: Harvesting Knowledge Graphs with Arbitrary Relations from Pretrained Language Models. *arXiv preprint arXiv:2206.14268*.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. 2024. OpenAI O1 System Card. *arXiv preprint arXiv:2412.16720*.

Rolf Jagerman, Honglei Zhuang, Zhen Qin, Xuanhui Wang, and Michael Bendersky. 2023. Query Expansion by Prompting Large Language Models. *arXiv preprint arXiv:2305.03653*.

Pengyue Jia, Yiding Liu, Xiangyu Zhao, Xiaopeng Li, Changying Hao, Shuaiqiang Wang, and Dawei Yin. 2023. Mill: Mutual verification with large language models for zero-shot query expansion. *arXiv preprint arXiv:2310.19056*.

Vamsi Krishna Kommineni, Birgitta König-Ries, and Sheeba Samuel. 2024. From Human Experts to Machines: An LLM Supported Approach to Ontology and Knowledge Graph Construction. *arXiv preprint arXiv:2403.08345*.

Amanda Kau, Xuzeng He, Aishwarya Nambissan, Aland Astudillo, Hui Yin, and Amir Aryani. 2024. Combining knowledge graphs and large language models. *arXiv preprint arXiv:2407.06564*.

Yibin Lei, Yu Cao, Tianyi Zhou, Tao Shen, and Andrew Yates. 2024. Corpus-steered query expansion with large language models. *arXiv preprint arXiv:2402.18031*.

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. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744.

Lars-Peter Meyer, Claus Stadler, Johannes Frey, Norman Radtke, Kurt Junghanns, Roy Meissner, Gordian Dziwis, Kirill Bulert, and Michael Martin. 2023. LLM-assisted knowledge graph engineering: Experiments with ChatGPT. In *Working Conference on Artificial Intelligence Development for a Resilient and Sustainable Tomorrow*, pages 103–115. Springer Fachmedien Wiesbaden.

Rolf Pfister and Hansueli Jud. 2025. Understanding and Benchmarking Artificial Intelligence: OpenAI's o3 Is Not AGI. *arXiv preprint arXiv:2501.07458*.

Masahiko Taniguchi and Jonathan S. Lindsey. 2025. Acquisition of absorption and fluorescence spectral data using chatbots. *Digital Discovery*, 4(1):21–34. Royal Society of Chemistry.

Armin Toroghi, Willis Guo, Mohammad Mahdi Abdollah Pour, and Scott Sanner. 2024. Right for Right Reasons: Large Language Models for Verifiable Commonsense Knowledge Graph Question Answering. *arXiv preprint arXiv:2403.01390*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.

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

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*.

附录A　提示词设计

子模块	Prompt
关键句提取模块	Given a query and the retrieved documents, extract only the sentences that directly answer, address, or provide relevant information for the query. Remove any unrelated or less relevant content. Maintain the original meaning while ensuring the extracted sentences are concise and informative. Query: {query} Documents: { documents} Sentences: { }
三元组抽取模块	You are a knowledge graph constructor specializing in extracting structured information from text. Given a document, identify and extract knowledge triples in the format <head entity; relation; tail entity>: 1.Each triple represents a specific relation or event between entities. 2.The head entity and tail entity must be relevant phrases from the text. 3.If multiple tail entities share a relation with a head entity, aggregate them with commas. Document: { document} Knowledge Triples: { }
知识图谱扩展模块	You are an expert in knowledge graph construction. Your task is to analyze the semantic relationship between a given query and existing knowledge triples, expand the knowledge triples with relevant entities and relationships to improve the knowledge graph. Query: {query} Existing Knowledge Triples: {existing knowledge triples} Output KG Triples: { } **Example:** **Query:** which numbers are the account number on the check? **Existing Knowledge Triples:** ⟨check; location; middle of the bottom⟩ ⟨account number; representation; long string of numbers⟩ ⟨account number; description; bank account number⟩ ⟨bank account number; determination; longer number⟩ ⟨bank account number; extraction; from check⟩ **Output KG Triples:** ⟨check; contains; account number⟩ ⟨account number; location; bottom of the check⟩ ⟨routing number; followed by; account number⟩ ⟨account number; uniqueness; unique identifier for specific bank account⟩ ⟨account number; purpose; setting up direct deposits, making electronic payments, verifying account information⟩ ⟨check number; difference from; account number⟩ ⟨check number; location; top right corner of the check⟩ ⟨account number; sensitivity; sensitive information that should be kept confidential⟩
查询扩展模块	Given a query and some knowledge triples, please write a passage based on them to answer the query. Query: {query} Knowledge Triples: {kg triples} Passage: { }

表 7: 子模块提示词设计