information

Article

HiPC-QR: Hierarchical Prompt Chaining for Query Reformulation

Hua Yang [1,2,*], **Hanyang Li** [3] and **Teresa Gonçalves** [2,4,*]

[1] School of Artificial Intelligence, Zhongyuan University of Technology, Zhengzhou 450007, China
[2] VISTA Lab, Algoritmi Center, University of Évora, 7000-671 Évora, Portugal
[3] School of Computer Science, Zhongyuan University of Technology, Zhengzhou 450007, China; 2023107324@zut.edu.cn
[4] Department of Computer Science, University of Évora, 7000-671 Évora, Portugal
* Correspondence: huayang@zut.edu.cn (H.Y.); tcg@uevora.pt (T.G.)

Abstract

Query reformulation techniques optimize user queries to better align with documents, thus improving the performance of Information Retrieval (IR) systems. Previous methods have primarily focused on query expansion using techniques such as synonym replacement to improve recall. With the rapid advancement of Large Language Models (LLMs), the knowledge embedded within these models has grown. Research in prompt engineering has introduced various methods, with prompt chaining proving particularly effective for complex tasks. Directly prompting LLMs to reformulate queries has become a viable approach. However, existing LLM-based prompt methods for query reformulation often introduce irrelevant content into reformulated queries, resulting in decreased retrieval precision and misalignment with user intent. We propose a novel approach called Hierarchical Prompt Chaining for Query Reformulation (HiPC-QR). HiPC-QR employs a two-step prompt chaining technique to extract keywords from the original query and refine its structure by filtering out non-essential keywords based on the user's query intent. This process reduces the query's restrictiveness while simultaneously expanding essential keywords to enhance retrieval effectiveness. We evaluated the effectiveness of HiPC-QR on two benchmark retrieval datasets, namely MS MARCO and TREC Deep Learning.The experimental results show that HiPC-QR outperforms existing query reformulation methods on large-scale datasets in terms of both recall@10 and MRR@10.

Keywords: query reformulation; keyword expansion; keyword filtering; prompt chain

check for updates

Academic Editors: Shadi Banitaan and Mina Maleki

Received: 27 July 2025
Revised: 3 September 2025
Accepted: 9 September 2025
Published: 11 September 2025

Citation: Yang, H.; Li, H.; Gonçalves, T. HiPC-QR: Hierarchical Prompt Chaining for Query Reformulation. *Information* **2025**, *16*, 790. https://doi.org/10.3390/info16090790

1. Introduction

In information retrieval (IR) systems, users' queries often suffer from issues such as lexical mismatches (e.g., "mobile phone" vs. "mobile device"), semantic ambiguity (e.g., "Java" referring to either a programming language or coffee beans), or overly restrictive constraints (e.g., "ultra-thin laptops under $500 released after 20 January 2025"). These factors can lead to low relevance in retrieval results or even zero-result scenarios. To address this challenge, query reformulation techniques have been developed to enhance the effectiveness of original queries through methods such as query expansion, structural simplification, and constraint relaxation, with the goal of bridging the gap between user intent and document content.

To address lexical mismatches between queries and documents, traditional pseudo-relevance feedback (PRF) methods, such as Relevance Model 3 (RM3), enhance queries by selecting terms from initially retrieved relevant documents [1,2]. For semantic ambiguity, researchers utilize pretrained word embeddings trained on large corpora to produce context-aware representations, enabling precise semantic term expansion [3–5]. Additionally, deep neural ranking models based on BERT effectively capture semantic relationships between queries and documents, further reducing semantic mismatches [6–9].

Recent advancements in LLMs [10,11] have opened up new opportunities for developing innovative query reformulation strategies [12]. These methods employ LLMs' generative capabilities through prompting to directly produce alternative query variants [13] or suggest additional keywords to augment original queries [14] or to generate pseudo-relevant documents that are concatenated with the original query—either by repeating the query multiple times in the case of sparse retrieval or by inserting an [SEP] token in dense retrieval—and increase relevance [15]. Despite these advancements, most research continues to focus on lexical and semantic challenges, with relatively little attention given to addressing overly restrictive constraints. Moreover, current approaches often fail to address the overly restrictive constraints commonly found in complex queries. For instance, while Google's search engine successfully answers 94.3% of simple queries (Level 1), it manages only 9.2% of highly complex queries (Level 3) [16]. Research on everyday search behaviors also highlights the presence of structurally complex queries, making their effective handling a critical research direction [17].

To address the aforementioned challenges, we propose HiPC-QR, a prompt chain-based query reformulation framework designed to enhance the quality of rewritten queries through fine-grained and multi-step prompt strategies. We validate the effectiveness of our approach on two benchmark datasets, namely MS MARCO [18] and TREC Deep Learning [19].

The framework consists of two core modules: a keyword extraction module and a keyword filtering and expansion module. The keyword extraction module leverages the semantic understanding capabilities of LLMs to identify salient information from the original query, thereby reducing noise and preventing the inclusion of irrelevant content; building upon this, the keyword filtering and expansion module selectively removes overly restrictive keywords and introduces semantically relevant expansion terms, allowing for a broader query scope while maintaining alignment with the user's original intent. By integrating these two modules, HiPC-QR is capable of generating rewritten queries that are both semantically faithful and contextually comprehensive.

The main contributions of this paper are as follows:

- A novel strategy for controlled query reformulation: We introduce HiPC-QR, a conceptual approach that addresses the limitations of complex queries by distinguishing between constraining and non-constraining keywords. This strategy enables precise simplification (via removal of overly restrictive constraints) and targeted enhancement (via expansion of flexible keywords).
- A practical two-step prompt chaining framework to realize HiPC-QR: Building on the above strategy, we design a concrete two-stage LLM-driven framework. In the first step, the model extracts query keywords and their semantic roles; in the second step, it selectively filters or expands keywords based on these roles. This framework ensures that the conceptual strategy can be effectively implemented in practice without causing semantic drift.

The structure of the paper is as follows: Section 2 reviews related studies on traditional query reformulation and the use of LLMs for query reformulation, Section 3 presents the proposed method in detail, Section 4 covers the experimental setup and implementation details, Section 5 analyzes the experimental results, and, finally, Section 6 concludes the paper and outlines future work.

2. Related Work

This section reviews related work on query reformulation. The first subsection presents traditional query reformulation approaches, while the second focuses on reformulation techniques based on LLMs.

2.1. Traditional Query Reformulation Methods

Traditional query reformulation methods primarily modify user queries by expanding the vocabulary of the original query. Numerous experiments have validated the effectiveness of such an approach [20]. Following the advancement of IR, early query reformulation techniques relied on classical retrieval models, such as BM25 [21], which assess relevance based on exact match statistical features, including term frequency (TF) and document length. These methods typically utilize relevance-based language models, including RM3 [22], Rocchio's algorithm [23], and DFR Bo1 query expansion [24], selecting terms from top-ranked documents retrieved for the original query as feedback. Beyond query expansion, document expansion has also been explored. This technique predicts pseudo-queries for documents; for instance, Doc2Query [25] trains a sequence-to-sequence model to generate questions that the input document may answer, appending these queries to the original document for indexing.

Additionally, learned sparse retrieval models, such as SPLADE [26] and uniCOIL [27], focus on learning highly sparse representations while maintaining nearest-neighbor search efficiency; by incorporating explicit sparse regularization and applying log saturation effects to term weights, these models significantly improve retrieval effectiveness. The advancement of dense neural networks [6,28] has significantly contributed to the field of query reformulation, making pretrained embeddings a widely adopted approach for capturing complex semantics in query rewriting. For instance, BERT-QE [29] is a BERT-based query expansion model that leverages contextual understanding and fine-grained selection mechanisms to improve query expansion effectiveness while mitigating mismatch issues, thereby enhancing retrieval performance.

Query reformulation methods combining PRF with dense neural networks have been widely studied. These methods enhance query representations by leveraging feedback from initial retrieval results. For example, ColBERT-PRF [30,31] incorporates feedback embeddings into queries, ANCE-PRF [32] concatenates queries with feedback passages, and Vector-PRF [33] combines text-based and vector-based PRF approaches. All these methods aim to refine query representations, improving re-ranking and retrieval performance.

2.2. Query Reformulation Using Large Language Models

In recent years, LLMs based on deep learning have achieved breakthrough advancements in natural language processing (NLP). Some studies have explored applying LLMs to query reformulation in IR system, aiming to generate diverse query variants that better capture users' underlying intent. For instance, Query2Doc [15] employs few-shot prompting to guide LLMs in generating relevant documents, which are then concatenated with the original query. INTER [34] refines IR by synergizing retrieval models (RMs) and LLMs; specifically, the RM component expands query information using LLM-generated knowledge, while the LLM component enhances prompts with documents retrieved by the RMs. HyDE [35] is an LLM-based query expansion method that generates hypothetical document embeddings to enable zero-shot dense retrieval, eliminating the need for relevance labels; however, it is susceptible to LLM hallucinations, where generated hypothetical documents may contain factual inaccuracies. To mitigate this hallucinations, GRM [36] introduces a relevance assessment model, such as a scoring mechanism, to filter out LLM-generated documents with no relevance to the query. Unlike previous models that enhance queries with LLM-generated documents, LameR [37] employs PRF. It first retrieves documents using the original query and then uses LLMs to generate answers from the top-n results. These answers replace the original query for subsequent retrieval, mitigating hallucinations but depending on initial retrieval effectiveness. Mackie et al. [38] proposed Generative and Pseudo-Relevant Feedback (GRF), which uses LLMs to generate feedback documents for query reformulation independently of initial retrieval quality; GRF integrates traditional PRF by expanding queries with retrieved documents and optimizes retrieval performance through Weighted Reciprocal Rank Fusion (WRRF).

In addition to directly prompting LLMs to generate query-related documents, several researchers have explored the use of LLMs to produce diverse query variants. For instance, GenQREnsemble [39] is a simple yet effective approach for query expansion. It uses ten diverse but semantically equivalent prompts to instruct an LLM to generate keywords relevant to the original query. The generated keywords from all prompts are aggregated and concatenated with the original query to form an expanded version for retrieval. This ensemble strategy can effectively unlock the latent knowledge of LLMs, leading to more robust query reformulation results. Jagerman [12] utilized eight distinct prompt types to exploit the capabilities of general-purpose LLMs without the need for training or fine-tuning. These templates are applied in various scenarios, including zero-shot, few-shot, and chained reasoning tasks. The results indicate that chained reasoning prompts hold significant potential for query expansion. Inspired by chain-of-thought (CoT) prompting [40], prompt chaining [41] extends this idea by facilitating the decomposition of complex tasks into multiple subtasks through a series of sequential prompts, where each step depends on the output generated by its predecessor. While chain-of-thought prompting aims to elicit internal reasoning within a single prompt (e.g., “Let’s think step by step”), prompt chaining extends this idea by decomposing a complex task into a sequence of functionally distinct prompts, where each step produces a structured output for the next.

A common theme among existing query reformulation approaches is their focus on query expansion—either by appending keywords or phrases to the original query or by generating pseudo-relevant documents to enrich context. While these methods effectively increase lexical coverage and improve recall, they often risk diluting the original query intent, especially when irrelevant or redundant terms are introduced.

Our approach follows the prompt chaining paradigm, implementing a two-step framework for query reformulation by first extracting keywords and evaluating their constraint level and then adaptively filtering overly restrictive terms to prevent null retrieval and expanding less constraining ones to improve recall, thereby balancing query focus and retrieval coverage.

3. Methodology

This chapter outlines the methodology of our proposed approach. We begin by presenting an overview of the query reformulation framework based on prompt chaining. Following this, we delve into the details of keyword extraction and filtering and expansion. Each subsection provides a clear description of the respective processes and their roles in the overall system.

3.1. Query Reformulation Framework Based on Prompt Chaining

To improve the retrieval efficiency and relevance of natural language queries, we propose a query reformulation method based on a two-step prompt chain. The overall framework of this method is shown in Figure 1, which consists of two key steps: keyword extraction and keyword filtering and expansion.

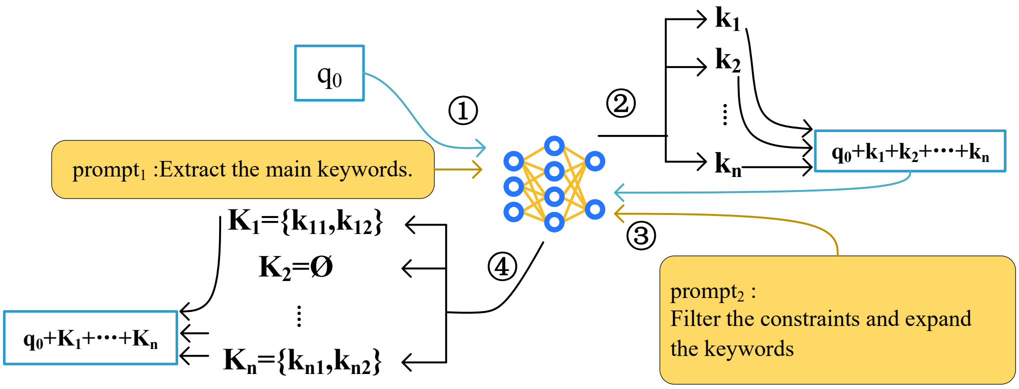

q₀
①
prompt₁ :Extract the main keywords.
②
k₁
k₂
kₙ
q₀+k₁+k₂+⋯+kₙ
③
prompt₂ :
Filter the constraints and expand the keywords
④
K₁={k₁₁,k₁₂}
K₂=Ø
Kₙ={kₙ₁,kₙ₂}
q₀+K₁+⋯+Kₙ

Figure 1. Hierarchical Prompt Chaining for Query Reformulation (HiPC-QR) architecture. Numbers ①–④ indicate the procedural steps in the query reformulation process: ① input original query and Prompt 1 to the LLM, ② extract keywords, ③ input keywords, query, and Prompt 2 to the LLM, and ④ perform filtering and expansion to generate a reformulated query.

Specifically, for the original query, denoted as q_0, we first leverage the powerful semantic understanding capability of the LLM. By designing specific prompt templates, we instruct the LLM to identify and extract keywords that are semantically most relevant to the original query. This results in a set of representative keywords $K : \{k_1, k_2, \ldots, k_n\}$, where the number of extracted keywords may vary across different queries. The keyword extraction process can be represented by Equation (1):

$$K : \{k_1, \ldots, k_n\} = LLM(prompt_1, q_0) \tag{1}$$

Then, we combine the keyword set K extracted in the previous step with the original query q_0 to form a new query q_k. This concatenation process is defined as follows in Equation (2):

$$q_k = concat(q_0, K, sep = '|') \tag{2}$$

The resulting q_k serves as the reformulated query string, and subsequent filtering/expansion operations are effectively applied to its components (q_0 and K).

In this way, we aim to preserve the core semantics of the original query during reformulation by focusing on extracted key terms, thereby minimizing interference from irrelevant information and providing semantically focused input for the next step.

In the query reformulation module, specifically in the constraint filtering and keyword expansion step, we utilize the LLM to assess the necessity of each keyword in relation to the original query. For each keyword k_i, if the LLM determines that expansion is needed, we create an expansion set $K_i : \{k_{i1}, k_{i2}, \ldots, k_{im}\}$, where m is the number of expanded terms and differs for each k_i. Moreover, if the LLM identifies a keyword as having restrictive constraints, it is considered for removal, and its corresponding expansion set becomes the empty set. This entire process is encapsulated by Equation (3), which formalizes both the expansion and filtering mechanisms.

$$K_i = LLM(prompt_2, q_0, k_i) = \begin{cases} \{k_{i1}, k_{i2}, \ldots, k_{im}\}, & \text{if } k_i \in \{\text{expansion}\} \\ \varnothing, & \text{if } k_i \in \{\text{filtering}\} \end{cases} \tag{3}$$

Finally, we linearly combine the original query with the processed keyword set to generate the reformulated query q_r. This process can be represented by Equation (4):

$$q_r = q_0 + \sum_{i=1}^{n} K_i \tag{4}$$

Through this approach, we not only expand the scope of the query but also prevent excessive expansion, thereby ensuring the relevance and accuracy of the retrieval results.

3.2. Keyword Extraction

Unlike traditional query expansion methods, which typically rely on initial query results or external knowledge bases to replace or expand keywords, our approach focuses on extracting the most accurate core keywords from the original query to better represent the user's query intent. By leveraging the powerful semantic understanding capabilities of LLMs, we can precisely identify and extract the core concepts most relevant to the query intent.

To achieve this, we began with an initial prompt template instructing the LLM to identify "focus on core concepts". However, through iterative testing and refinement, we found that explicitly guiding the LLM to "the main key terms from the query" improved the relevance and conciseness of the extracted keywords. Our final prompt template is shown in Table 1.

Although we use a single, manually refined prompt formulation based on iterative trials during development, it is carefully designed to elicit consistent and meaningful keyword outputs from the LLM. No post-processing is applied to the LLM output; instead, we rely on the model's strong semantic understanding capabilities to return high-quality keywords. To validate the effectiveness of our prompt design, we conducted a qualitative assessment by manually inspecting the keywords extracted for ten randomly selected queries. This evaluation confirmed that the prompt successfully guided the LLM to extract concise and representative terms that were closely aligned with the original query intent.

Table 1. The final version of the prompt template for Prompt 1 (keyword extraction).

	Content	**Purpose**
Instruction	"Extract the main key terms"	Direct the model to identify semantically central elements in the query.
Output Format	"Keywords: <keywords>"	Ensure structured output, facilitating further processing and analysis.

3.3. Keyword Filtering and Expansion

The second stage of our framework focuses on refining the extracted keyword set through a structured process of filtering overly restrictive terms and expanding semantically relevant ones. This step aims to enhance the flexibility of the query while preserving its core intent, thereby improving retrieval effectiveness.

To achieve this, the second prompt template is designed to explicitly instruct the LLM to perform two key tasks:

- Constraint Detection: Identify and relax overly specific spatiotemporal or numerical constraints (e.g., precise timestamps, narrow location ranges) that may unnecessarily limit the retrieval scope.
- Semantic Expansion: Recognize terms with high semantic relevance to the original query and suggest meaningful synonyms or related expressions that preserve the intent while increasing coverage.

Our final prompt template is shown in Table 2.

Table 2. The final version of the prompt template for Prompt 2 (keyword filtering and expansion).

	Content	Purpose
Instruction 1	"Perform rigorous constraint detection on the query to identify and optimize overly specific spatiotemporal/numerical constraints (e.g., excessively precise temporal or spatial limitations) while preserving essential core conditions."	Constraint detection.
Instruction 2	"Identify any key terms that can be replaced with synonyms or related terms, considering the original intent of the query."	Semantic expansion.
Output Format	"Reformulated query: <reformulated query>"	Ensure structured output, facilitating further processing and analysis.

The hierarchical nature of HiPC-QR lies in the ordered dependency between its two stages: extraction of keywords (Stage 1) informs the subsequent filtering and expansion decisions (Stage 2), thereby creating a structured pipeline that avoids the uncontrolled transformations often observed in end-to-end approaches. This decoupled design ensures that keyword semantics are explicitly modeled before rewriting occurs, which enhances both the interpretability and controllability of the reformulation process.

To further illustrate the query reformulation process and the effectiveness of our prompt chain structure, we present an example in Table 3; the table provides a step-by-step breakdown of how an original query is transformed into a more generalized and optimized version through keyword extraction and constraint optimization.

As shown in Table 3, the original query is first broken down into its key components through keyword extraction. This step ensures that only the most relevant information is retained for further processing. In the second step, the extracted keywords are subjected to constraint optimization and synonym identification. This involves relaxing overly specific constraints, such as the exact time range, and replacing key terms with more general or related concepts. The final reformulated query, "US stock price (performance) fluctuations (upward/downward trends) yesterday (last trading day)", demonstrates the effectiveness of this approach. By removing unnecessary specificity and incorporating synonyms, the query becomes more flexible and better suited for retrieval purposes while still preserving the original intent.

Although the system currently relies on a single, carefully refined prompt formulation—developed through extensive experimentation and validation during the design phase—we believe that this approach offers a clear and interpretable mechanism for query reformulation. It is particularly effective in mitigating issues such as low retrieval relevance or even zero-result scenarios caused by overly restrictive constraints.

Table 3. Query reformulation process and prompt chain structure.

Original_Query	US stock price fluctuations between 3:15 PM and 3:30 PM yesterday
Step 1 prompt	Given the original query: {original_query}, extract the main key terms. Return a list of the key terms or important concepts from the query. Keywords: <keywords>
Keywords	[US, stock price, fluctuations, yesterday, 3:15 PM–3:30 PM]
Step 2 prompt	Given the original query: {original_query} and the extracted key terms: {keywords}, perform the following tasks: 1. Perform rigorous constraint detection on the query to identify and optimize overly specific spatiotemporal/numerical constraints (e.g., excessively precise temporal or spatial limitations) while preserving essential core conditions. 2. Identify any key terms that can be replaced with synonyms or related terms, considering the original intent of the query. Reformulated query: <reformulated query>
Reformulated query	US stock price (performance) fluctuations (upward/downward trends) yesterday (last trading day).

4. Experimental Setup

This section provides a detailed introduction to the datasets, evaluation methods, system implementation details, and baseline models used for comparison. Through a rigorous experimental setup, we aim to validate the effectiveness of HiPC-QR in IR tasks.

4.1. Datasets and Evaluation Metrics

To comprehensively assess the performance of our proposed framework, we employ two widely used benchmark datasets along with a set of standard evaluation metrics.

MS MARCO, the Microsoft Machine Reading Comprehension dataset [18], is a large-scale retrieval corpus designed to advance deep learning applications in IR. It contains over 8.8 million passages and more than 500,000 query–passage pairs, providing a vast amount of training data for retrieval models. Additionally, the dataset reserves 6980 queries as a development set (Dev set) for evaluating retrieval performance. Due to its scale and diversity, MS MARCO serves as a key benchmark for measuring system recall and ranking accuracy in retrieval tasks. Our study focuses on this development set to assess the effectiveness of our proposed method.

To further evaluate retrieval systems under multi-level relevance scenarios, we also adopt the DL, TREC Deep Learning dataset [19]. This dataset is derived from MS MARCO web queries and relevant documents and features multi-level relevance judgments provided by NIST assessors, where each query–document pair is labeled on a scale from 0 (non-relevant) to 3 (highly relevant). Such annotations are richer than binary relevance schemes and are commonly referred to as hierarchical annotations in information retrieval. Specifically, the DL19 dataset contains 43 topics, while the DL20 dataset expands this to 52 topics. Both datasets primarily consist of fact-based queries, aiming to test a model's ability to accurately and comprehensively retrieve relevant information from large-scale document collections. In our evaluation, these graded annotations are leveraged differently depending on the metric: for nDCG, all relevance levels are used directly, while for metrics such as recall, we follow common practice by applying a threshold and treating documents with relevance levels of 2 or higher as relevant.

We adopt the following four evaluation metrics to measure the retrieval effectiveness:

- R@1K: This metric measures the proportion of relevant documents retrieved among the top 1000 results, as defined in Equation (5). A higher R@1K indicates better coverage of the retrieval system.

$$\text{R@1K} = \frac{|\{\text{relevant documents} \cap \text{top-1000 retrieved documents}\}|}{|\text{relevant documents}|} \tag{5}$$

- MRR@10: This metric evaluates the ranking quality by measuring the reciprocal rank of the first relevant document within the top 10 retrieved results, as defined in Equation (6), where rank_q denotes the rank position of the first relevant document for query q within the top 10 results and $|Q|$ is the total number of queries. If no relevant document is found within the top 10, $\frac{1}{\text{rank}_q}$ is taken as 0. A higher MRR@10 score indicates better ranking performance.

$$\text{MRR@10} = \frac{1}{|Q|} \sum_{q=1}^{|Q|} \frac{1}{\text{rank}_q} \tag{6}$$

- MAP (Mean Average Precision): MAP is defined as the mean of the average precision (AP) values across all queries:

$$\text{MAP} = \frac{1}{|Q|} \sum_{q=1}^{|Q|} \text{AP}_q \tag{7}$$

  where $|Q|$ is the total number of queries and AP_q is the average precision for query q:

$$\text{AP}_q = \frac{\sum_{i=1}^{n} \text{Precision@}i \cdot \text{Rel}(i)}{|\text{relevant documents in result}|}. \tag{8}$$

  In Equation (8), Precision@i denotes the precision at rank i and $\text{Rel}(i)$ is an indicator function equal to 1 if the document at position i is relevant (otherwise, it is equal to 0).
- NDCG@10 (Normalized Discounted Cumulative Gain at 10): This metric evaluates ranking quality by considering both the relevance and position and is calculated using two equations: Equation (9) defines DCG@10, while Equation (10) provides the specific form of NDCG@10 as the ratio of DCG@10 to the ideal IDCG@10. In Equation (9), rel_i is the relevance score of the document at rank i, and in Equation (10), IDCG@10 is the ideal DCG value obtained from a perfect ranking.

$$\text{DCG@10} = \sum_{i=1}^{10} \frac{2^{\text{rel}_i} - 1}{\log_2(i+1)} \tag{9}$$

$$\text{NDCG@10} = \frac{\text{DCG@10}}{\text{IDCG@10}} \tag{10}$$

  The NDCG@10 metric is particularly effective for datasets with multilevel relevance annotations (e.g., DL19 and DL20), where documents are assigned graded relevance scores (e.g., 0: not relevant; 1: marginally relevant; 2: relevant; 3: highly relevant).

The four metrics complement each other by evaluating different dimensions of retrieval performance. R@1K focuses on coverage of relevant documents among the top 1000 results, MRR@10 evaluates the ranking quality by rewarding early retrieval of the first relevant document, MAP considers both precision and recall across the entire ranked list, and NDCG@10 incorporates graded relevance levels to account for multi-level annotations.

4.2. LLM Deployment and Retrieval Configuration

Due to regional restrictions on commercial APIs and limited computational resources, we conduct our experiments using locally deployed open-source LLMs to ensure reproducibility and practical applicability. We implemented the proposed prompt chaining strategy using two large language models: Meta-LLaMA 3.1-8B-Instruct [42] and DeepSeek-R1-8B [43]. All models were deployed on a single GPU A5000 server to ensure consistent hardware conditions.

To enhance output stability and improve experimental reproducibility, we applied uniform inference settings across both models. Specifically, temperature controls the randomness of token sampling by scaling the logits before softmax (lower values make outputs more deterministic, while higher values increase diversity), and top_p (nucleus sampling) restricts generation to the smallest set of tokens whose cumulative probability exceeds p, balancing coherence and diversity. In addition, $frequency\_penalty$ decreases the likelihood of repeating tokens that have already appeared frequently, while $presence\_penalty$ discourages reusing tokens that have appeared at least once. In our experiments, we set $temperature = 0.1$ to reduce randomness while maintaining fluency, $top\_p = 0.9$ to allow sampling from the most probable subset, and $frequency\_penalty = 0$ and $presence\_penalty = 0$ since repetition was already minimized through structured prompting.

The prompt chaining strategy was applied in the same manner to both models, guiding them through a multi-step reasoning process with a fixed output format (e.g., “Keywords: <keywords>” and “Reformulated query: <reformulated query>”), ensuring consistent and structured responses. This implementation emphasizes our focus on evaluating the effectiveness of the designed prompting strategy, rather than comparing the intrinsic capabilities of the individual LLMs.

In the BM25-based retrieval system (used for both our method and all comparison methods), we utilize Pyserini’s BM25 parameter tuning method [44] to optimize the R@1K performance. A grid search approach is applied to fine-tune the BM25 model parameters on five different 10k-sample subsets, which were randomly selected using the Linux shuffle command. The final selected parameters for the BM25 model were $k_1 = 0.82$, $b = 0.68$. The BM25 ranking function computes the relevance score between a query and a document based on term frequency and document length normalization and is defined by Equation (11), where $f(q, D)$ represents the term frequency of word q in the document D, $|D|$ is the length of the document, $avgdl$ denotes the average length of the document in the corpus, and k_1 and b are hyperparameters controlling term saturation and length normalization, respectively.

$$BM25(D, Q) = \sum_{i=1}^{n} IDF(q_i) \times \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b\frac{|D|}{\text{avgdl}}\right)} \tag{11}$$

4.3. Comparison Methods

To comprehensively verify the effectiveness of HiPC-QR, we selected several commonly used query reformulation and expansion methods as baseline comparisons. All methods were evaluated using the same BM25 base retrieval system with fixed parameters ($k_1 = 0.82$, $b = 0.68$), ensuring a fair comparison.

RM3 [22] is a feedback-based query expansion method. Its core principle involves analyzing high-frequency relevant terms in the initial retrieval results and using these terms to expand the original query, thereby improving its coverage. In our experiments, we fine-tune the following RM3 parameters:

- fb_terms, the number of candidate expansion terms (search range: 5–95, step size: 5);
- fb_docs, the number of feedback documents (search range: 5–50, step size: 5);
- $original\_query\_weight$, the weight of the original query in the expanded version (search range: 0.2–0.8, step size: 0.1).

Although RM3 can improve recall, the introduction of expansion terms may increase noise, sometimes leading to lower precision than expected.

Rocchio [23] is a vector space model (VSM) query reformulation technique based on relevance feedback. It refines the query vector by adjusting it with weighted information from both relevant and non-relevant documents, thus better reflecting users' true intent. In this study, we adopt the default Rocchio weighting parameters in Pyserini, using it as an extended version of BM25. Although Rocchio enhances query representation, its performance heavily depends on the quality of the initial retrieval results.

GRF, Generative Relevance Feedback [38], is an LLM-based query reformulation method that does not rely on initial retrieval results. Instead, it expands and optimizes the original query by constructing diverse generative subtasks, such as Keywords, Entities, Queries, Summary, and Essay. To ensure a direct comparison with the HiPC-QR method, we selected two GRF subtasks: GRF_Keywords and GRF_Queries. We used the queries generated by these subtasks as input for the BM25 retrieval model.

GenQREnsemble [39] employs ten semantically equivalent but syntactically diverse prompts to elicit query-relevant keywords from an LLM. The generated keywords from all prompts are aggregated via union and concatenated with the original query to form an expanded version for retrieval. We apply GenQREnsemble using the same locally deployed LLMs (DeepSeek and Llama) and the same BM25 retriever with identical configurations as used in our proposed HiPC-QR method. This ensures a fair comparison, isolating the impact of the query reformulation strategy.

5. Experimental Results and Analysis

As mentioned, we compared the performance of our proposed two-step prompt chaining method HiPC-QR against both traditional query reformulation techniques and recent LLM-based baselines across three datasets: DL19, DL20, and Dev (MS MARCO Dev). RM3 and Rocchio represent classical query expansion methods based on pseudo-relevance feedback and vector space modeling, respectively, while GRF_Queries and GRF_Keywords serve as prompt-based query variant generation methods using LLMs. We also evaluate GenQREnsemble, which uses an ensemble of diverse prompts to generate and aggregate keywords for query reformulation, as a representative LLM-based baseline. As detailed in Section 3, the HiPC-QR framework adopts a two-step prompt chaining strategy: in step 1, the LLM extracts semantically relevant keywords from the original query and, in step 2, these keywords serve as intermediate reasoning cues to guide the generation of a reformulated query. The framework was implemented using both the LLaMA-3.1-8B-Instruction and DeepSeek-R1-8B models.

5.1. Analysis of Experimental Results

Table 4 presents the obtained results, where (L) and (D) refer to the LLaMA and DeepSeek models, respectively, and "−1" and "−2" indicate the two stages of HiPC-QR: "−1" for keyword extraction and "−2" for query reformulation using those keywords as reasoning cues. It can be observed that HiPC-QR-1 (D) achieves the best overall performance across most metrics and datasets. On DL20, it achieves MAP = 0.312, an improvement of +0.081 over GRF_Queries (0.231); on Dev, it achieves R@1K = 0.880, surpassing Rocchio (0.873) by +0.007. GenQREnsemble (D) achieves the highest recall on DL19 and DL20 with R@1K scores of 0.807 and 0.833, respectively, which we attribute to its ensemble-based keyword expansion strategy that generates a large and diverse set of keyword variants. This extensive lexical expansion aligns well with the BM25 retrieval mechanism, which heavily relies on term matching and term frequency. By introducing more relevant terms into the query, GenQREnsemble increases the likelihood of overlapping with matching documents, thereby enhancing retrieval coverage. However, its performance on ranking-sensitive metrics such as MAP and ndcg@10 remains suboptimal, suggesting that the unfiltered aggregation of keywords may introduce noise, which dilutes query focus and harms ranking quality. Notably, the Step1 strategy of HiPC-QR also yields higher recall, suggesting that using focused keywords can effectively bridge lexical gaps in BM25 retrieval. Meanwhile, HiPC-QR-2 (L) attains the highest MRR@10 of 0.2178 on Dev, indicating stronger early precision and better ranked results, which are valuable for real-world applications.

In contrast, GRF-based LLM variants underperform across most settings (e.g., GRF_Keywords attains R@1K = 0.6482 on DL19), likely due to lack of structured reasoning in prompt design.

Traditional methods such as Rocchio and RM3 still provide notable gains over BM25, but fall short compared to structured LLM prompting. These results confirm the effectiveness of our two-step HiPC-QR framework, which uses a prompt chain approach. It is especially effective when paired with high-reasoning-capacity LLMs like DeepSeek-R1.

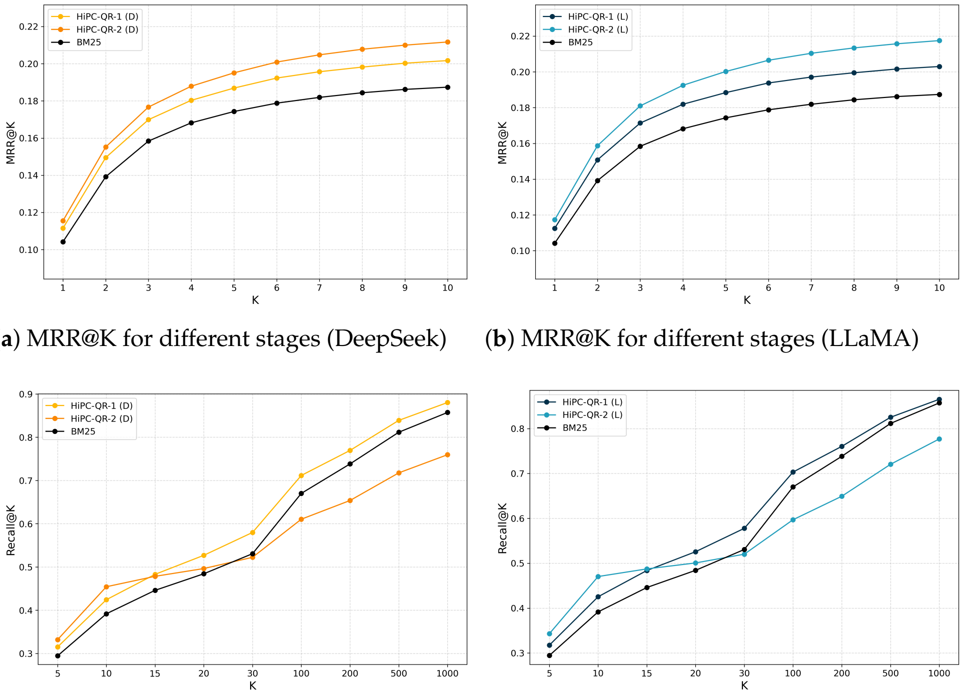
Table 4. Evaluation of the proposed HiPC-QR framework against classical and LLM-based query reformulation methods.

	DL19			**DL20**			**Dev**	
Model	**MAP**	**NDGC@10**	**R@1K**	**MAP**	**NDGC@10**	**R@1K**	**MRR@10**	**R@1K**
Retrieval Baseline								
BM25	0.290	0.497	0.745	0.288	0.488	0.803	0.187	0.857
Query Reformulation Methods								
RM3 [22]	0.334	0.515	0.795	0.302	0.492	0.829	0.165	0.870
Rocchio [23]	0.340	0.528	0.795	0.312	0.491	0.833	0.168	0.873
GRF_Queries [38]	0.272	0.455	0.729	0.231	0.373	0.637	– –	– –
GRF_Keywords [38]	0.190	0.349	0.648	0.201	0.314	0.572	– –	– –
GenQREnsemble (L)	0.222	0.375	0.667	0.203	0.349	0.646	– –	– –
GenQREnsemble (D)	0.327	0.513	0.807	0.279	0.465	0.833	– –	– –
HiPC-QR-1 (L)	0.302	0.509	0.750	0.307	**0.513**	0.821	0.203	0.865
HiPC-QR-2 (L)	0.289	0.485	0.756	0.282	0.477	0.793	**0.218**	0.777
HiPC-QR-1 (D)	0.324	**0.542**	**0.801**	0.312	0.511	0.816	0.202	**0.880**
HiPC-QR-2 (D)	0.311	0.500	0.783	0.299	0.477	0.819	0.212	0.760

Note: L = LLaMA, D = DeepSeek. "-1" and "-2" refer to the two steps of our proposed HiPC-QR framework. In step 1, the LLM extracts semantically relevant keywords from the original query; in step 2, these keywords are used as intermediate reasoning cues to generate the reformulated query. **Bold values** indicate the best results per column.

To further analyze the retrieval performance differences between the two strategies within the HiPC-QR framework, we plot the Recall@K and MRR@k curves at various K values on the MS MARCO Dev dataset, as shown in Figure 2. The figure presents the performance of four variants, including HiPC-QR-1 (L) and HiPC-QR-2 (L) under the LLaMA model and HiPC-QR-1 (D) and HiPC-QR-2 (D) under the DeepSeek model. Here, "−1" and "−2" refer to the keyword extraction and query reformulation stages, respectively, and (L)/(D) indicate the underlying LLM backbone.

It can be observed that in the lower K range (K = 5 to 15), HiPC-QR-2 (L) and HiPC-QR-2 (D) outperform both variants of HiPC-QR-1 (L) and HiPC-QR-1 (D) across all datasets in terms of recall. For example, on the Dev dataset, when K = 10, HiPC-QR-2 (L) achieves a recall score of 0.4704, which is significantly higher than that of HiPC-QR-1 (L), with a value of 0.4253. This indicates that the second-stage generation strategy is more effective in improving early retrieval performance, enabling BM25 to identify highly relevant documents at top ranks.



(**a**) MRR@K for different stages (DeepSeek) (**b**) MRR@K for different stages (LLaMA)

(**c**) Recall@K for different stages (DeepSeek) (**d**) Recall@K for different stages (LLaMa)

Figure 2. Recall@K and MRR@K on MS MARCO Dev using LLaMA and DeepSeek: HiPC-QR-1 and HiPC-QR-2. (**a**,**b**) MRR@K comparison across stages; (**c**,**d**) Recall@K comparison across stages.

However, as K increases, the recall trend reverses. When K = 20, HiPC-QR-1 (L) and HiPC-QR-1 (D) begin to outperform the two-step methods. For instance, HiPC-QR-1 (D) reaches a Recall@1K of 0.8803, the highest among all methods, while HiPC-QR-2 (D) achieves only 0.7596. Similar trends are observed on DL19 and DL20. This indicates that the keyword concatenation strategy has better coverage at large K values, as it enhances the prominence of key terms and improves overall recall.

This results may be due to the structural differences between the two query reformulation strategies. The second-step generation method rewrites the original query using a language model, removing restrictive terms and increasing semantic focus, which makes it easier to retrieve highly relevant documents at early ranks. However, this process may also compress or generalize the query, possibly leading to some semantically related but peripheral content being missed and reducing recall at deeper ranks. In contrast, the first-step method simply appends extracted keywords to the original query without altering its structure, preserving all context and enriching it with salient terms. This helps trigger more lexical matches in traditional retrieval systems (e.g., BM25), especially when K is large.

5.2. Ablation Study

To evaluate the role and synergy of the two key stages in HiPC-QR and to validate the effectiveness of the prompt chain, we performed ablation studies using two LLMs: LLaMA and DeepSeek. As shown in Table 5, the experimental results indicate that when only the keywords extracted in the first stage were appended to the original query (HiPC-QR-1 (L) and HiPC-QR-1 (D)), MRR@10 improved by 8.3% and 7.6%, respectively, compared to the baseline of BM25. This demonstrates the effectiveness of the keyword extraction stage and suggests that the extracted keywords provide valuable semantic signals for retrieval.

Table 5. Ablation study.

Method	MRR@10	R@1K
Baseline BM25	0.1874	0.8573
HiPC-QR-1 (L)	0.2030	0.8654
HiPC-QR-2 (L) $_{w/o\ HiPC-QR-1(L)}$	0.1551	0.7981
HiPC-QR-2 (L) $_{w/\ HiPC-QR-1(L)}$	**0.2175**	0.7770
HiPC-QR-1 (D)	0.2017	0.8803
HiPC-QR-2 (D) $_{w/o\ HiPC-QR-1(D)}$	0.1106	0.6491
HiPC-QR-2 (D) $_{w/\ HiPC-QR-1(D)}$	**0.2117**	0.7596

In contrast, when the keywords extracted in the first stage are used as input prompts, the LLM can more effectively perform the filtering and expansion strategy in the second stage (HiPC-QR-2$_{w/HiPC-QR-1}$). Specifically, the model is prompted to classify the keywords as follows: identify constraint-bearing terms (e.g., time, location, domain) and determine whether they should be retained or relaxed based on context while expanding non-constraint keywords to improve recall.

Compared to directly feeding the raw query into the LLM for rewriting, extracting keywords first provides the model with a more structured and semantically clear input. This step-by-step prompting approach improves the controllability and consistency of the LLMs' output. Experimental results show that this strategy achieves the best performance in terms of MRR@10, indicating that the proposed prompt chain (HiPC-QR) strikes a better balance between the retrieval comprehensiveness and accuracy by structuring the query rewriting process.

In addition, both LLaMA and DeepSeek demonstrated consistent performance improvements, further validating the generality and robustness of our method across different LLMs.

5.3. Comparison with Generative Methods

To further compare HiPC-QR with the GRF_Queries and GRF_Keywords, we conducted a statistical analysis on the DL19 dataset, focusing on the NDCG@10 metric. We chose NDCG@10 as the primary evaluation metric because it effectively captures both the relevance and ranking quality of the retrieved documents. Moreover, DL19 and DL20 are multi-grade relevance datasets, making NDCG a more appropriate measure compared to binary metrics such as Recall@10 or MRR@10. As shown in Table 6, the performance of HiPC-QR is always superior to that of GRF_queries and GRF_keywords using both LLMs.

Table 6. Comparison of HiPC-QR and GRF-based methods on the DL19 dataset.

Method	NDCG@10	Improvement over GRF_Keywords	Outperformed Queries (Count/%)	Std Dev
GRF_Keywords	0.348714	—	—	0.273745
HiPC-QR-1 (L)	0.509351	46.1%	28/65.1%	0.26014
HiPC-QR-1 (D)	0.542188	55.5%	31/72.1%	0.230846
Method	**NDCG@10**	**Improvement over GRF_Queries**	**Outperformed Queries (Count / %)**	**Std Dev**
GRF_Queries	0.455084	—	—	0.306530
HiPC-QR-2 (L)	0.484772	6.5%	24/55.8%	0.266856
HiPC-QR-2 (D)	0.499916	9.9%	22/51.2%	0.266682

In addition to the statistical comparison, we present one representative example in Table 7 to illustrate the reformulation behavior of HiPC-QR compared to GRF. This query was randomly sampled from the DL19 test set and serves as a typical case demonstrating how HiPC-QR preserves user intent while relaxing overly restrictive constraints to improve retrieval performance. The original query is "*what is the most popular food in Switzerland*", and we compare the query variants generated by GRF and HiPC-QR with both LLaMA and DeepSeek.

The GRF_Keywords method produced irrelevant or overly generic tokens such as "*List*", "*Relevant*", and "*Documents*", resulting in a poor NDCG@10 score of 0.2900. Although the GRF_Queries generated multiple alternative phrasings (e.g., "*famous Swiss food*", "*Swiss cuisine*"), it achieved an NDCG@10 of 0.7744.

In contrast, our proposed method HiPC-QR-1, using both LLaMA and DeepSeek, directly extracted relevant keyword phrases (e.g., "*popular food, Switzerland*"), resulting in improved focus and clarity, each achieving 0.6170 in NDCG@10. Building upon these intermediate keywords, the HiPC-QR-2 models further refined the query. The LLaMA version introduced semantically rich expansions such as "*dish*" and "*Swiss cuisine*", yielding a significantly higher NDCG@10 of 0.9202. Notably, the DeepSeek-based HiPC-QR-2 achieved the highest score of 0.9355, demonstrating its superior capability in capturing nuanced user intent through more precise reformulation (e.g., "*the most famous foods in Switzerland*"). This demonstrates that our method offers distinct advantages in keyword precision and effective query reformulation.

Table 7. Comparison of query samples between GRF and HiPC-QR from the DL19 dataset.

Method	Query (ID:833860)	Ndcg@10
Original Query	what is the most popular food in switzerland	0.4827
GRF_Keywords	Switzerland, Food, Popular, List, Relevant, Documents	0.2900
GRF_Query	Swiss food, Swiss cuisine, Traditional Swiss food, Popular Swiss food, Most popular Swiss food, famous Swiss food, best Swiss food	0.7744
HiPC-QR-1 (L)	what is the most popular food in switzerland|popular food, Switzerland	0.6170
HiPC-QR-2 (L)	what is the most popular (food, dish) in (Switzerland, Swiss cuisine)?	0.9202
HiPC-QR-1 (D)	what is the most popular food in switzerland|Switzerland, popular food	0.6170
HiPC-QR-2 (D)	What are the most famous foods (popular food) in Switzerland?	0.9355

5.4. Handling Constraints in Query Reformulation

User queries often contain constraints such as domain, time, or spatial limitations, which can restrict the range of relevant results. As show in Table 8, the original query, "*what is a nonconformity? earth science*", is limited by the domain constraint of "*earth science*", potentially excluding relevant documents from related fields like geology. In step 2, by removing the "*earth science*" constraint, the query scope is expanded, and the relevant term "*geologic nonconformity*" is added, enhancing the comprehensiveness of the retrieval. The value of MAP@1000 in Table 8 increases from 0.0043 for the original query to 0.5283, indicating that our query reformulation method not only improves the effectiveness of IR but also captures more cross-domain relevant information. While the recall rate is improved, it may introduce some noise, affecting the precision of the retrieval results. Therefore, balancing comprehensive retrieval while minimizing irrelevant results remains a significant challenge for information retrieval systems.

Table 8. Case study of handling constraints in query reformulation from the DL20 dataset.

Method	Query (ID:1116380)	Map@1000
Original Query	what is a nonconformity? earth science	0.0043
HiPC-QR-1 (L)	what is a nonconformity? earth science | nonconformity, earth science	0.0051
HiPC-QR-2 (L)	what is a nonconformity (geologic nonconformity)?	0.5283

6. Conclusions and Future Work

In this study, we propose HiPC-QR, a progressive two-step query reformulation method using prompt chain and LLMs. Our approach consists of keyword extraction followed by semantic constraint-based query reformulation, aiming to balance ranking accuracy and recall performance. Through fine-grained query reformulation, our approach enhances the accuracy and relevance of retrieval systems, demonstrating strong performance particularly in metrics such as MRR@10 and R@10. Notably, in the Dev dataset experiments, HiPC-QR demonstrated strong adaptability in large-scale retrieval tasks, providing a solid foundation for its practical application in real-world scenarios.

It is worth noting that this two-stage prompting mechanism introduces additional processing latency. Specifically, the end-to-end latency of the complete HiPC-QR pipeline is approximately 7.46 s (LLaMA) and 10.03 s (DeepSeek), corresponding to two sequential stages, namely step 1 (keyword extraction) and step 2 (keyword filtering and expansion), both executed via the LLM. This latency is overwhelmingly dominated by the cost of LLM inference.

In future research, we will focus on refining the methods for assessing keyword constraint strength, aiming to improve the precision and robustness of the filtering and expansion process. By developing more sophisticated criteria for determining whether a keyword is overly restrictive or semantically beneficial, we can further enhance the effectiveness of constraint-aware query reformulation.

To address the current latency bottleneck caused by LLM inference, we plan to investigate model distillation, quantization, and lightweight architectures to accelerate inference speed while preserving the effectiveness of the proposed method. These efficiency-oriented optimizations are critical for real-world deployment in low-latency retrieval scenarios.

The proposed HiPC-QR method will be integrated with state-of-the-art retrieval models, such as deep learning-based ranking systems, to enhance overall retrieval performance. We also aim to validate HiPC-QR on larger and more diverse benchmarks, such as BEIR [45], to assess its generalization across domains.

Finally, we will develop refined reformulation strategies tailored to different query types, aiming to minimize potential negative effects and achieve consistent performance improvements across diverse user intents.

Author Contributions: H.Y., conceptualization, methodology, data curation, validation, writing—review and editing, funding acquisition, supervision; H.L., methodology, formal analysis, validation, soft ware, writing—original draft; T.G., resources, funding acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: This research work was funded by: (1) the Key Scientific Research Project of Higher Education Institutions in Henan Province, grant no. 24A520060. (2) Graduate Education and Teaching Reform Research Project of Zhongyuan University of Technology, grant no. JG202434. (3) Guidance Project of the China National Textile and Apparel Council, grant no. 2025054.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The datasets used in this study are openly available to the public.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Robertson, S.E. On term selection for query expansion. *J. Doc.* **1990**, *46*, 359–364. [CrossRef]
2. Lavrenko, V.; Croft, W.B. Relevance-based language models. In *Proceedings of the ACM SIGIR Forum*; ACM: New York, NY, USA, 2017; Volume 51, pp. 260–267.
3. Kuzi, S.; Shtok, A.; Kurland, O. Query expansion using word embeddings. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, Indianapolis, IN, USA, 24–28 October 2016; pp. 1929–1932.

4. Roy, D.; Paul, D.; Mitra, M.; Garain, U. Using word embeddings for automatic query expansion. *arXiv* **2016**, arXiv:1606.07608. [CrossRef]
5. Zamani, H.; Croft, W.B. Embedding-based query language models. In Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval, Newark, DE, USA, 12–16 September 2016; pp. 147–156.
6. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
7. Khattab, O.; Zaharia, M. 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, Virtual, 25–30 July 2020; pp. 39–48.
8. MacAvaney, S.; Yates, A.; Cohan, A.; Goharian, N. CEDR: Contextualized embeddings for document ranking. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019; pp. 1101–1104.
9. Nogueira, R.; Yang, W.; Cho, K.; Lin, J. Multi-stage document ranking with BERT. *arXiv* **2019**, arXiv:1910.14424. [CrossRef]
10. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
11. Peng, B.; Li, C.; He, P.; Galley, M.; Gao, J. Instruction tuning with gpt-4. *arXiv* **2023**, arXiv:2304.03277. [CrossRef]
12. Jagerman, R.; Zhuang, H.; Qin, Z.; Wang, X.; Bendersky, M. Query expansion by prompting large language models. *arXiv* **2023**, arXiv:2305.03653. [CrossRef]
13. Nogueira, R.; Lin, J.; Epistemic, A. From doc2query to docTTTTTquery. *Online Preprint* **2019**, *6*.
14. Wang, X.; MacAvaney, S.; Macdonald, C.; Ounis, I. Generative query reformulation for effective adhoc search. *arXiv* **2023**, arXiv:2308.00415. [CrossRef]
15. Wang, L.; Yang, N.; Wei, F. Query2doc: Query expansion with large language models. *arXiv* **2023**, arXiv:2303.07678. [CrossRef]
16. Vemuru, S.; John, E.; Rao, S. Handling Complex Queries Using Query Trees. *Authorea Preprints* **2023**. [CrossRef]
17. Azad, H.K.; Deepak, A. Query expansion techniques for information retrieval: A survey. *Inf. Process. Manag.* **2019**, *56*, 1698–1735. [CrossRef]
18. Bajaj, P.; Campos, D.; Craswell, N.; Deng, L.; Gao, J.; Liu, X.; Majumder, R.; McNamara, A.; Mitra, B.; Nguyen, T.; et al. Ms marco: A human generated machine reading comprehension dataset. *arXiv* **2016**, arXiv:1611.09268.
19. Craswell, N.; Mitra, B.; Yilmaz, E.; Campos, D.; Voorhees, E.M. Overview of the TREC 2019 deep learning track. *arXiv* **2020**, arXiv:2003.07820. [CrossRef]
20. Carpineto, C.; Romano, G. A survey of automatic query expansion in information retrieval. *ACM Comput. Surv. (CSUR)* **2012**, *44*, 1–50. [CrossRef]
21. Robertson Stephen, E.; Steve, W.; Susan, J.; Micheline, H.B.; Mike, G. Okapi at TREC-3. In Proceedings of the Third Text REtrieval Conference (TREC 1994), Gaithersburg, MA, USA, 2–4 November 1994.
22. Abdul-Jaleel, N.; Allan, J.; Croft, W.B.; Diaz, F.; Larkey, L.; Li, X.; Smucker, M.D.; Wade, C. UMass at TREC 2004: Novelty and HARD. In Proceedings of the TREC-13, Gaithersburg, MA, USA, 16–19 November 2004; pp. 715–725.
23. Croft, W.B.; Metzler, D.; Strohman, T. *Search Engines: Information Retrieval in Practice*; Addison-Wesley Reading: Boston, MA, USA, 2010; Volume 520.
24. Amati, G.; Van Rijsbergen, C.J. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst. (TOIS)* **2002**, *20*, 357–389. [CrossRef]
25. Nogueira, R.; Yang, W.; Lin, J.; Cho, K. Document expansion by query prediction. *arXiv* **2019**, arXiv:1904.08375. [CrossRef]
26. Formal, T.; Piwowarski, B.; Clinchant, S. SPLADE: Sparse lexical and expansion model for first stage ranking. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, MontrEal, QC, Canada, 11–15 July 2021; pp. 2288–2292.
27. Lin, J.; Ma, X. A few brief notes on deepimpact, coil, and a conceptual framework for information retrieval techniques. *arXiv* **2021**, arXiv:2106.14807. [CrossRef]
28. Xiong, L.; Xiong, C.; Li, Y.; Tang, K.F.; Liu, J.; Bennett, P.; Ahmed, J.; Overwijk, A. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv* **2020**, arXiv:2007.00808. [CrossRef]
29. Zheng, Z.; Hui, K.; He, B.; Han, X.; Sun, L.; Yates, A. BERT-QE: Contextualized query expansion for document re-ranking. *arXiv* **2020**, arXiv:2009.07258. [CrossRef]
30. Wang, X.; Macdonald, C.; Tonellotto, N.; Ounis, I. ColBERT-PRF: Semantic pseudo-relevance feedback for dense passage and document retrieval. *ACM Trans. Web* **2023**, *17*, 1–39. [CrossRef]
31. Wang, X.; Macdonald, C.; Tonellotto, N.; Ounis, I. Pseudo-relevance feedback for multiple representation dense retrieval. In Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval, New York, NY, USA, 11–15 July 2021; pp. 297–306.

32. Yu, H.; Xiong, C.; Callan, J. Improving query representations for dense retrieval with pseudo relevance feedback. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, Virtual, 1–5 November 2021; pp. 3592–3596.
33. Li, H.; Mourad, A.; Zhuang, S.; Koopman, B.; Zuccon, G. Pseudo relevance feedback with deep language models and dense retrievers: Successes and pitfalls. *ACM Trans. Inf. Syst.* **2023**, *41*, 1–40. [CrossRef]
34. Feng, J.; Tao, C.; Geng, X.; Shen, T.; Xu, C.; Long, G.; Zhao, D.; Jiang, D. Synergistic Interplay between Search and Large Language Models for Information Retrieval. *arXiv* **2023**, arXiv:2305.07402. [CrossRef]
35. Gao, L.; Ma, X.; Lin, J.; Callan, J. Precise zero-shot dense retrieval without relevance labels. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Toronto, ON, Canada, 9–14 July 2023; pp. 1762–1777.
36. Mackie, I.; Sekulic, I.; Chatterjee, S.; Dalton, J.; Crestani, F. GRM: Generative relevance modeling using relevance-aware sample estimation for document retrieval. *arXiv* **2023**, arXiv:2306.09938. [CrossRef]
37. Shen, T.; Long, G.; Geng, X.; Tao, C.; Zhou, T.; Jiang, D. Large language models are strong zero-shot retriever. *arXiv* **2023**, arXiv:2304.14233. [CrossRef]
38. Mackie, I.; Chatterjee, S.; Dalton, J. Generative relevance feedback with large language models. In Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, Taipei, Taiwan, 23–27 July 2023; pp. 2026–2031.
39. Dhole, K.D.; Agichtein, E. Genqrensemble: Zero-shot llm ensemble prompting for generative query reformulation. In *Proceedings of the European Conference on Information Retrieval*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 326–335.
40. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.H.; Le, Q.V.; Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24824–24837.
41. Saravia, E. Prompt Engineering Guide. GitHub. 2022. Available online: https://github.com/dair-ai/Prompt-Engineering-Guide (accessed on 1 June 2023).
42. Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; et al. The llama 3 herd of models. *arXiv* **2024**, arXiv:2407.21783. [CrossRef]
43. Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv* **2025**, arXiv:2501.12948.
44. Lin, J.; Ma, X.; Lin, S.C.; Yang, J.H.; Pradeep, R.; Nogueira, R. Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, 11–15 July 2021; pp. 2356–2362.
45. Thakur, N.; Reimers, N.; Rücklé, A.; Srivastava, A.; Gurevych, I. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv* **2021**, arXiv:2104.08663. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.