# When Should Queries Be Decomposed? A Stage-Aware Study of Query Decomposition for Multi-Condition Retrieval

**Bochao Yin**[1]*, **Xuan Lu**[1,2]*, **Zhengyu Qi**[1], **Xiaoyu Shen**[1,3]†

[1]Eastern Institute of Technology, Ningbo [2]Shanghai Jiao Tong University
[3]Zhejiang Key Laboratory of Industrial Intelligence and Digital Twin
lux1997@sjtu.edu.cn xyshen@eitech.edu.cn

## Abstract

Multi-condition retrieval requires systems to identify documents that satisfy multiple distinct constraints, moving beyond mere topical relevance. While query decomposition is widely adopted as an intuitive remedy, its effectiveness across different retrieval pipeline stages remains underexplored. In this paper, we conduct a stage-aware empirical study and uncover a stark, stage-dependent effect: decomposition during initial retrieval frequently harms retrieval performance due to semantic dilution, yet substantially improves reranking by enabling more fine-grained constraint verification. Motivated by these insights, we propose a principled **Stage-Aware Decomposition** framework that retains the monolithic query during initial retrieval to preserve global semantic context, while employing sub-queries exclusively during reranking for fine-grained constraint matching. Extensive evaluations on the MultiConIR and SSRB benchmarks demonstrate that our framework consistently improves ranking performance for compositional queries across multiple retrieval and reranking models. We release our code at https://github.com/EIT-NLP/Query-Decompose.

## 1 Introduction

Information Retrieval (IR) underpins modern information access by enabling systems to identify and rank relevant documents. Retrieval has evolved from lexical matching and structured filtering (Robertson and Zaragoza, 2009; Yee et al., 2003) to neural semantic retrieval (Karpukhin et al., 2020; Khattab and Zaharia, 2020) and, more recently, LLM-based query reformulation and reranking (Sun et al., 2023; Gao et al., 2023). Despite these advances, many real-world queries remain challenging because they contain multiple conditions that must be considered jointly. For example, a query such as "*a lightweight laptop with 16GB RAM under $1500 for gaming, and at least 512GB SSD storage*" specifies several independent requirements. We refer to this setting as *multi-condition retrieval*, where the goal is to rank documents according to how well they satisfy the query's multiple conditions collectively.

Recent benchmarks such as MultiConIR (Lu et al., 2025) and SSRB (Zhang et al., 2025a) show that modern retrievers struggle with such queries, particularly in distinguishing highly relevant documents from hard negatives that match the general topic but satisfy fewer specific conditions. A natural approach is *query decomposition*, which breaks a complex query into simpler sub-queries corresponding to its constituent conditions. This idea has been explored in complex question answering (Min et al., 2019), retrieval-augmented frameworks (Ammann et al., 2025), and agentic search systems (Press et al., 2023; Jin et al., 2025; Zhao et al., 2025).

However, despite its growing adoption, the effectiveness of query decomposition for multi-condition retrieval remains poorly understood. In particular, it is unclear whether decomposition is equally useful at different stages of a multi-stage retrieval pipeline, where initial retrieval aims to preserve recall while reranking focuses on fine-grained discrimination. This leads to a critical but underexplored research question: *Does the effectiveness of query decomposition depend on where it is applied within the multi-stage retrieval pipeline?*

In this work, we conduct a systematic empirical study of query decomposition and uncover a stark, stage-dependent effect: *decomposition frequently impairs initial retrieval, but improves reranking on the same queries*. Quantitatively, on complex queries, decomposition decreases retrieval NDCG@10 by 1.6–4.3 points across different retrievers, while improving reranking NDCG@10 by 3.9–7.6 points over standard reranking across dif-

*Equal contribution
†Corresponding author



*Findings of the Association for Computational Linguistics: EMNLP 2026*, pages 34990–35016
October 24-29, 2026 ©2026 Association for Computational Linguistics

ferent reranker models. We trace this asymmetry to a phenomenon we term *semantic dilution*. A complex query combines multiple conditions that collectively provide a strong signal for identifying the most relevant documents. Decomposition strips away this conjunctive structure, leaving each sub-query with only a fragment of the original intent. Each sub-query therefore matches a much larger candidate pool, including documents that satisfy one condition but fail to satisfy the remaining ones. During initial retrieval, where the system must search a large corpus and preserve recall, this loss of global context weakens the discriminative signal of the original query. Partial matches can consequently be promoted over documents that satisfy substantially more of the original conditions.

In contrast, decomposition becomes beneficial during reranking. Rerankers operate over a small, pre-filtered candidate pool and can model fine-grained query–document interactions at the token level (Lu et al., 2026a; Fan et al., 2026a). At this stage, the goal is not broad candidate discovery but precise discrimination among already plausible candidates. Decomposed sub-queries make individual conditions explicit, allowing the reranker to assess how well each candidate matches different parts of the original intent. Thus, the same decomposition that weakens global retrieval signals can provide useful condition-level information when applied to reranking.

Motivated by these insights, we propose a principled **Stage-Aware Decomposition** framework that aligns query representations with the objectives of different retrieval stages. We retain the original monolithic query for initial retrieval, preserving the complete query intent and safeguarding global recall. We then apply query decomposition exclusively during reranking, where fine-grained interactions between candidate documents and individual sub-queries can be exploited for more precise ranking. This design combines the complementary strengths of the two representations: the monolithic query preserves global relevance during candidate generation, while decomposed sub-queries expose individual conditions for fine-grained reranking.

To summarize, our primary contributions are:

- **Stage-Dependent Insights**: We systematically evaluate query decomposition across different stages of multi-condition retrieval and identify *semantic dilution* as a key cause of performance degradation during the initial retrieval stage.
- **Stage-Aware Framework**: We introduce a framework that strategically aligns query with pipeline stages, preserving the monolithic query for initial retrieval while exploiting decomposed sub-queries for reranking.
- **Extensive Evaluation**: Through comprehensive evaluations on the MultiConIR and SSRB benchmarks across various embedding and reranking models, we demonstrate that our approach substantially and consistently enhances ranking precision for complex multi-condition queries.

## 2 Related Work

**Multi-condition retrieval** Traditional retrieval benchmarks such as MS MARCO (Nguyen et al., 2016) and BEIR (Thakur et al., 2021) have driven the development of modern retrieval models, but they mainly evaluate retrieval for individual natural language queries based on lexical or semantic similarity. Such settings often fail to capture user intent involving multiple independent constraints. Recent benchmarks like BRIGHT (Su et al., 2025) introduce reasoning-intensive queries beyond surface matching. Nevertheless, retrieval is still framed as satisfying a single unified query objective. More recent work explicitly studies multi-condition retrieval. BIRCO (Wang et al., 2024) targets retrieval tasks with multi-faceted objectives, MultiConIR (Lu et al., 2025) evaluates retrieval robustness as query conditions increase, MCMR (Lu et al., 2026d) evaluates multi-condition multimodal retrieval, focusing on whether retrieval models can satisfy multiple fine-grained semantic constraints across modalities, and SSRB (Zhang et al., 2025a) constructs a large-scale benchmark for natural language queries that combine exact matching constraints and fuzzy semantic requirements across multiple fields. These studies highlight the increasing difficulty of satisfying multiple constraints simultaneously, indicating that existing retrieval models still struggle with complex query requirements (Huang et al., 2026; Han and Huang, 2026). This gap motivates further research on robust multi-condition retrieval. In particular, it calls for more effective ways to handle complex queries with multiple constraints, which we discuss in the following subsection.

**Query Decomposition** Query decomposition has been widely studied as a strategy for handling complex information needs. Prior work on query reformulation and rewriting has been shown to improve retrieval effectiveness, particularly in conversational search and information retrieval settings (Lin et al., 2021; Nogueira and Cho, 2017). Building on this idea, query decomposition splits a complex query into simpler sub-queries, and has been applied in multi-hop reasoning and retrieval-augmented generation (RAG), where decomposed queries can improve evidence retrieval and answer accuracy (Trivedi et al., 2022; Lin et al., 2023; Ammann et al., 2025; Fan et al., 2025, 2026b). More recent work further optimizes decomposition to better align with downstream retrievers or dynamically select or prioritize sub-queries during retrieval (Liu et al., 2025; Petcu et al., 2025). Decomposition has also been explored in structured retrieval settings, where sub-queries are used to capture fine-grained attributes or sub-intents (Zhu et al., 2025; Zhong et al., 2025). Despite these advances, existing approaches often treat decomposed sub-queries independently and focus on improving recall. This limits their ability to model global consistency across multiple constraints, particularly in multi-condition retrieval. In contrast, we study query decomposition from a stage-aware perspective and show that its effectiveness critically depends on how it is integrated into retrieval and reranking pipelines.

## 3 Problem & Experimental Setup

### 3.1 Task Definition

**Multi-Condition Retrieval** We study information retrieval under multi-condition queries, where a query specifies multiple constraints that must be jointly satisfied. Formally, we represent a query as a set of atomic conditions $Q = \{c_1, c_2, \ldots, c_k\}$. Given a document $d$, we say $d$ is *fully relevant* to $Q$ if it satisfies all $k$ conditions simultaneously; otherwise it is partially relevant or irrelevant. The retrieval objective is to rank fully relevant documents above all others.

**Query Decomposition** To model compositional queries, we consider query decomposition, which transforms a query $Q$ into a set of sub-queries via a decomposition function $\mathcal{D}$:

$$\mathcal{D}(Q) = \{q_1, q_2, \ldots, q_m\}. \tag{1}$$

Each sub-query $q_i$ is associated with a set of conditions $C(q_i) \subseteq \{c_1, \ldots, c_j\}$. Collectively, these sub-queries provide partial views of the original query, each capturing a subset of the constraints.

**Two-Stage Retrieval Pipeline** We consider a standard two-stage retrieval pipeline. In the first stage, a retriever produces a candidate set $\mathcal{C}_K$ of size $K$ from the corpus. In the second stage, a reranker assigns fine-grained relevance scores to each candidate in $\mathcal{C}_K$ and returns a reordered list. At each stage, the input query can be either the original query $Q$ or a set of decomposed sub-queries $\mathcal{D}(Q)$. Our central question is whether the effectiveness of query decomposition depends on the stage at which it is applied.

**Dataset** We evaluate on the MultiConIR benchmark (Lu et al., 2025). which covers five domains—Books, Movies, People, Medical Cases, and Legal Documents—with queries containing $k \in \{1, \ldots, 10\}$ conditions. Each query is paired with a positive document $d^+$ satisfying all $k$ conditions and graded hard negatives $\{d_j\}_{j=0}^{k-1}$, where $d_j$ satisfies exactly $j$ conditions.

### 3.2 Evaluation Metrics

We evaluate retrieval and ranking performance using **Recall@$K$**, **NDCG@$K$**, and **Win Rate**. Recall evaluates whether relevant documents are preserved during the initial retrieval stage; NDCG is highly suitable for tasks involving graded relevance judgment (Thakur et al., 2021); Win Rate measures the model's fine-grained discrimination capability (Lu et al., 2025).

The Win Rate is the probability that the true positive document $d^+$ scores higher than its hardest negative $d_*^-$, defined as the incorrect document that satisfies the largest number of query conditions. Formally, over a set of queries $Q$:

$$\text{Win Rate} = \frac{1}{|Q|} \sum_{q \in Q} \mathbf{1}\big(S(q, d^+) > S(q, d_*^-)\big) \tag{2}$$

where $S(q, d)$ denotes the relevance score predicted by the model for a query-document pair.

Detailed Win Rate results are provided in Appendix B.

### 3.3 Retrieval and Reranking Setup

**Retrieval Setup** We evaluate the effect of query decomposition on retrieval performance across five datasets. We use two representative embedding model architectures: (1) BGE-large-en-v1.5 (Xiao

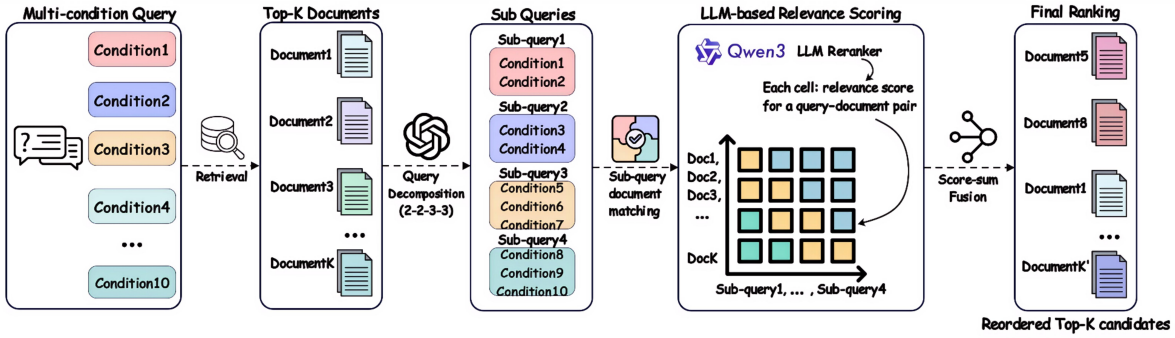



Figure 1: Overview of the Stage-Aware Decomposition Framework

et al., 2024), an encoder-only model based on BERT (Devlin et al., 2019); and (2) the Qwen3-Embedding series (Zhang et al., 2025b), a decoder-only(Vaswani et al., 2017) model family built on LLMs.

For the Qwen3-Embedding series, we select variants of different sizes (0.6B, 4B, and 8B). For decomposed retrieval, we retrieve the top-1000 candidates for each sub-query and assign a score of zero to all documents beyond this cutoff before fusion. For this retrieval stage, we retrieve top-50 candidates then passed to the reranking stage.

**Reranking Setup** We adopt a pointwise reranking paradigm (Lu et al., 2026b,c), where each candidate document is independently evaluated against each decomposed sub-query. For reranking models, we use bge-reranker-v2-m3 (Li et al., 2023; Chen et al., 2024) and the Qwen3-Reranker series (Zhang et al., 2025b) (0.6B, 4B, and 8B). We frame reranking as a binary relevance classification task and compute scores using the normalized logits of the "Yes" and "No" tokens. Discussions on listwise reranking and token-based scoring experiments are provided in Appendix C.

## 4 Methodology

Intuitively, query decomposition offers a natural way to improve multi-condition retrieval: decomposing a long, information-dense query into shorter sub-queries allows the model to be more precised about each condition. By aggregating the top candidates retrieved for each sub-query through a fusion mechanism, the final result better reflects documents that simultaneously satisfy all conditions.

### 4.1 Query decomposition

Within our experimental framework, we investigate the efficacy of query decomposition by introducing three strategies, which are formulated as follows:

**Naive Decomposition** In this approach, a complex query $Q$ is strictly partitioned into single-condition sub-queries. The decomposition function $\mathcal{D}_{\text{naive}}$ is formulated as:

$$\mathcal{D}_{\text{naive}}(Q) = \{q_i\}_{i=1}^{n} \quad \text{s.t.} \quad |C(q_i)| = 1, \forall i,$$

where $n$ is the total number of conditions in $Q$, and $C(q_i)$ denotes the set of semantic conditions contained within sub-query $q_i$.

**Fixed-size Decomposition** In contrast to the naive approach, which restricts each sub-query to a single condition, this strategy partitions a complex query $Q$ into sub-queries that each contain between $k$ and $j$ conditions, where $1 \leq k \leq j \leq n$ and $n$ denotes the total number of conditions in $Q$. The decomposition function $\mathcal{D}_{\text{fixed}}$ is formulated as:

$$\mathcal{D}_{\text{fixed}}(Q) = \{q_i\}_{i=1}^{m} \quad \text{s.t.} \quad k \leq |C(q_i)| \leq j, \forall i,$$

where $m$ is the total number of resulting sub-queries and $C(q_i)$ denotes the set of semantic conditions contained within sub-query $q_i$.

**Adaptive Decomposition** Unlike the previous rule-based strategies, this approach leverages an LLM to dynamically determine the optimal granularity of decomposition. The decomposition function $\mathcal{D}_{\text{adaptive}}$, parameterized by the LLM (denoted as $\mathcal{M}$), is formulated as:

$$\mathcal{D}_{\text{adaptive}}(Q; \mathcal{M}) = \{q_i\}_{i=1}^{k} \quad \text{s.t.} \quad |C(q_i)| = d_i,$$

where $d_i \geq 1$ denotes the number of semantic conditions dynamically assigned to the $i$-th sub-query, and $k$ is the total number of generated sub-queries. Both $d_i$ and $k$ are inferred by $\mathcal{M}$ based on the semantic context of the original query $Q$.

### 4.2 Fusion Strategy

After query decomposition, we perform independent retrieval/reranking for each sub-query. To aggregate the result, we perform two different fusion

strategies: Reciprocal Rank Fusion (RRF) (Cormack et al., 2009) and Score-Sum Fusion.

**Reciprocal Rank Fusion (RRF)** For a given document $d$, its RRF score is computed by summing the inverse of its rank across all decomposed sub-queries $q_i \in Q_{\text{sub}}$:

$$\text{RRF}(d) = \sum_{q_i \in Q_{\text{sub}}} \frac{1}{k + \text{rank}_i(d)} \tag{3}$$

where $\text{rank}_i(d)$ denotes the rank position of document $d$ retrieved by sub-query $q_i$, and $k$ is a smoothing constant. The final candidate list is generated by sorting the documents in descending order of their $\text{RRF}(d)$ scores. RRF fusion provides a highly robust, training-free mechanism by relying strictly on the rank permutations. We provide a sensitivity analysis of $k$ in Appendix F.

**Score-Sum Fusion** Rather than utilizing rank positions, this method computes the final relevance score of document $d$ by directly aggregating its raw retrieval or reranking scores $s_i(d)$ across all sub-queries:

$$\text{Score}(d) = \sum_{q_i \in Q_{\text{sub}}} s_i(d) \tag{4}$$

Similar to RRF, documents are subsequently sorted in descending order based on their aggregated $\text{Score}(d)$ to produce the final ranked list. Score-Sum Fusion retains the absolute magnitude of relevance scores, ensuring that strong confidence signals are not lost.

## 5 Experimental Results

**Decomposition and Fusion Strategies** To determine the optimal query decomposition and fusion pipeline, we conduct a study evaluating three proposed decomposition strategies alongside two fusion methods. Specifically, we randomly sample 30% of the data across all datasets in the Multi-ConIR benchmark and evaluate each configuration on both the retrieval and reranking stages. As a reference, we first measure the performance of the undecomposed baseline. We observe that queries containing 2 or 3 conditions achieve higher scores than queries with more conditions. Motivated by this, we set $k = 2$ and $j = 3$ for fixed-size decomposition. Accordingly, queries with three or fewer conditions (Q1–Q3) already fall within this range and are left undecomposed. For Adaptive Decomposition, we let Qwen3-8B dynamically determine the optimal granularity for each query; the detailed prompt is provided in Appendix A.

As shown in Tables 1 and 2, fixed-size decomposition consistently yields the best overall performance across both the retrieval and reranking stages. Furthermore, the Score-Sum fusion method is shown to consistently outperform RRF. We therefore adopt these optimal configurations—fixed-size decomposition with $k = 2$ and $j = 3$, coupled with Score-Sum fusion—for all subsequent experiments. Beyond this direct conclusion, this study also reveals a striking stage-dependent pattern: query decomposition is detrimental at the retrieval stage but substantially boosts performance at the reranking stage. This observation motivates our Stage-Aware Decomposition Framework, illustrated in Figure 1. We next verify whether the framework remains effective across the full benchmark and different retrieval and reranking models.

**Retrieval Result** As shown in Table 3, applying query decomposition to complex queries (Queries 4–10) consistently degrades performance across all evaluated models in terms of both NDCG@10 and Recall@50. Specifically, for the NDCG@10 metric, all models experience a consistent drop following decomposition. The most severe decline is observed with BGE-large-en-v1.5, which exhibits a 4.3-point degradation, while the Qwen series models show drops ranging from 1.6 to 2.2 points. Similarly, for Recall@50, both query decomposition strategies consistently underperform the standard baseline, registering an absolute reduction of approximately 1 to 3 points. These findings confirm that query decomposition fundamentally impairs performance at the initial retrieval stage.

**Reranking Result** As Table 4 indicates, standard reranking does not guarantee improvements and can even degrade retrieval quality. For instance, Qwen3-Reranker-0.6B suffers a 4.1-point drop on complex queries (Queries 4–10) compared to the unreranked baseline. Conversely, integrating query decomposition into the reranking stage yields substantial gains, outperforming standard reranking by 7.6 points and successfully surpassing the baseline. Notably, this performance gain over the baseline scales robustly with model size: the larger Qwen3-Reranker-8B achieves an 11.6-point improvement over the baseline and a 5.9-point gain over standard reranking. This result also aligns with our initial hypothesis, confirming that query decomposition is beneficial at the reranking stage.

| Model | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **NDCG@10** | | | | | | | | | | | |
| Base† | 67.7 | 84.3 | 86.6 | 85.4 | 82.7 | 78.4 | 73.6 | 66.7 | 58.2 | 43.1 | 72.7 |
| Naive-sum | 67.7 | 75.3 | 76.1 | 67.6 | 64.8 | 59.8 | 56.0 | 50.5 | 40.5 | 28.8 | 58.7 |
| Naive-rrf | 67.7 | 70.1 | 67.5 | 62.7 | 59.9 | 55.1 | 51.9 | 46.2 | 37.7 | 27.6 | 54.6 |
| Adaptive-sum | 67.7 | 78.9 | 81.8 | 73.6 | 81.2 | 74.6 | 67.2 | 63.0 | **55.6** | 41.5 | 68.5 |
| Adaptive-rrf | 67.7 | 76.2 | 79.9 | 68.1 | 79.5 | 71.0 | 61.9 | 59.6 | 53.1 | 39.5 | 65.7 |
| Fixed-size-sum | 67.7 | **84.3** | **86.6** | **82.5** | **81.5** | **78.0** | **72.0** | **65.2** | 55.5 | **42.3** | **71.6** |
| Fixed-size-rrf | 67.7 | **84.3** | **86.6** | 78.3 | 79.6 | 75.0 | 68.6 | 62.6 | 54.0 | 39.9 | 69.7 |

Table 1: NDCG@10 performance comparison in retrieval stage across different decomposition and fusion strategies and query lengths using Qwen3-Embedding-0.6B. †Base (retrieve without decomposition) is shown as a reference. Values are scaled by 100. The "X-Y" naming convention denotes the combination of decomposition strategy X (e.g., naive, adaptive, fixed-size) and fusion method Y (e.g., scoresum, rrf).

| Model | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **NDCG@10** | | | | | | | | | | | |
| Pure Rerank† | 80.1 | 87.1 | 86.2 | 81.9 | 76.9 | 72.4 | 67.4 | 57.0 | 50.9 | 40.4 | 70.0 |
| Naive-sum | 80.1 | 86.9 | 86.4 | 86.8 | 83.7 | 79.5 | 74.7 | 67.8 | 61.2 | 49.9 | 75.7 |
| Naive-rrf | 80.1 | 86.7 | 86.1 | 85.6 | 81.6 | 77.7 | 71.7 | 66.3 | 59.4 | 49.5 | 74.5 |
| Adaptive-sum | 80.1 | 87.0 | **86.6** | 85.3 | **83.7** | **80.6** | 72.8 | **69.3** | 60.0 | 51.2 | 75.7 |
| Adaptive-rrf | 80.1 | 86.8 | 86.2 | 85.5 | 83.3 | 79.0 | 73.2 | 67.9 | 59.0 | 51.4 | 75.2 |
| Fixed-size-sum | 80.1 | **87.1** | 86.2 | **87.3** | 83.5 | 79.3 | **76.3** | 68.8 | **61.6** | **52.1** | **76.2** |
| Fixed-size-rrf | 80.1 | **87.1** | 86.2 | 86.1 | 83.5 | 78.6 | 74.2 | 66.8 | 59.0 | 52.0 | 75.4 |

Table 2: NDCG@10 performance comparison in reranking stage across different decomposition and fusion strategies and query lengths using Qwen3-Reranker-0.6B. †Pure Rerank (rerank without decomposition) is shown as a reference. Values are scaled by 100. The "X-Y" naming convention same as Table 1.

| Method | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 | Avg.(Q4–Q10) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **BGE-large-en-v1.5** | | | | | | | | | | | |
| Base (N@10) | 69.8 | 80.7 | 84.7 | 84.2 | 81.5 | 77.1 | 72.0 | 64.8 | 55.8 | 42.3 | **68.2** |
| Decomp-Merge (N@10) | 69.8 | 80.7 | 84.7 | 78.0 | 77.5 | 74.9 | 65.8 | 60.7 | 52.6 | 38.0 | 63.9 |
| Base (R@50) | 85.8 | 94.5 | 97.7 | 98.7 | 99.5 | 99.5 | 99.7 | 99.5 | 99.6 | 99.8 | **99.5** |
| Decomp-Merge (R@50) | 85.8 | 94.5 | 97.7 | 95.3 | 97.3 | 98.4 | 95.5 | 96.9 | 97.7 | 94.1 | 96.5 |
| **Qwen3-Embedding-0.6B** | | | | | | | | | | | |
| Base (N@10) | 67.7 | 84.0 | 86.2 | 85.8 | 82.5 | 78.7 | 73.4 | 66.5 | 58.5 | 45.5 | **70.1** |
| Decomp-Merge (N@10) | 67.7 | 84.0 | 86.2 | 82.4 | 80.3 | 77.1 | 70.6 | 65.1 | 56.7 | 42.5 | 67.8 |
| Base (R@50) | 85.7 | 96.5 | 98.6 | 99.4 | 99.7 | 99.9 | 100.0 | 100.0 | 100.0 | 100.0 | **99.9** |
| Decomp-Merge (R@50) | 85.7 | 96.5 | 98.6 | 97.5 | 98.6 | 99.2 | 98.7 | 99.5 | 99.6 | 99.0 | 98.9 |
| **Qwen3-Embedding-4B** | | | | | | | | | | | |
| Base (N@10) | 69.8 | 83.7 | 86.9 | 86.0 | 82.5 | 78.2 | 72.7 | 65.9 | 56.7 | 42.9 | **69.3** |
| Decomp-Merge (N@10) | 69.8 | 83.7 | 86.9 | 83.3 | 80.7 | 76.9 | 70.1 | 64.2 | 55.8 | 43.1 | 67.7 |
| Base (R@50) | 86.1 | 95.7 | 98.7 | 99.3 | 99.9 | 99.8 | 100.0 | 100.0 | 100.0 | 100.0 | **99.9** |
| Decomp-Merge (R@50) | 86.1 | 95.7 | 98.7 | 97.3 | 98.5 | 99.3 | 98.6 | 99.4 | 99.5 | 99.3 | 98.8 |
| **Qwen3-Embedding-8B** | | | | | | | | | | | |
| Base (N@10) | 72.6 | 85.1 | 87.4 | 86.3 | 83.5 | 79.3 | 73.5 | 67.0 | 58.6 | 45.5 | **70.5** |
| Decomp-Merge (N@10) | 72.6 | 85.1 | 87.4 | 83.4 | 81.0 | 77.6 | 70.7 | 64.3 | 57.0 | 44.3 | 68.3 |
| Base (R@50) | 87.7 | 96.6 | 98.9 | 99.6 | 99.9 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | **99.9** |
| Decomp-Merge (R@50) | 87.7 | 96.6 | 98.9 | 97.6 | 98.9 | 99.5 | 98.7 | 99.4 | 99.5 | 99.4 | 99.0 |

Table 3: Impact of query decomposition on retrieval across different query complexities. Values are scaled by 100. Base represents retrieve without decomposition. We provide full result in Appendix B.

| Method | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 | Avg.(Q4–Q10) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Qwen3-Reranker-0.6B** | | | | | | | | | | | |
| Orig. Base | 69.8 | 80.7 | 84.7 | 84.2 | 81.5 | 77.1 | 72.0 | 64.8 | 55.8 | 42.3 | 68.2 |
| Pure Rerank | 79.6 | 86.9 | 86.4 | 82.3 | 77.8 | 72.0 | 67.7 | 58.1 | 50.8 | 40.2 | 64.1 |
| Decomp-Merge | 79.6 | 86.9 | 86.4 | 87.4 | 83.5 | 78.2 | 75.4 | 67.4 | 59.3 | 50.8 | 71.7 |
| ΔDecomp-Merge | 0.0 | 0.0 | 0.0 | ↑5.1 | ↑5.7 | ↑6.2 | ↑7.7 | ↑9.3 | ↑8.5 | ↑10.6 | ↑7.6 |
| **bge-reranker-v2-m3** | | | | | | | | | | | |
| Orig. Base | 69.8 | 80.7 | 84.7 | 84.2 | 81.5 | 77.1 | 72.0 | 64.8 | 55.8 | 42.3 | 68.2 |
| Pure Rerank | 79.4 | 89.0 | 90.1 | 88.2 | 84.8 | 80.6 | 74.4 | 67.3 | 57.9 | 43.9 | 71.0 |
| Decomp-Merge | 79.4 | 89.0 | 90.1 | 89.5 | 86.8 | 82.4 | 78.3 | 71.7 | 63.6 | 51.7 | 74.9 |
| ΔDecomp-Merge | 0.0 | 0.0 | 0.0 | ↑1.3 | ↑2.0 | ↑1.8 | ↑3.9 | ↑4.4 | ↑5.7 | ↑7.8 | ↑3.9 |
| **Qwen3-Reranker-4B** | | | | | | | | | | | |
| Orig. Base | 69.8 | 80.7 | 84.7 | 84.2 | 81.5 | 77.1 | 72.0 | 64.8 | 55.8 | 42.3 | 68.2 |
| Pure Rerank | 81.6 | 89.5 | 89.8 | 86.3 | 81.7 | 76.7 | 70.4 | 63.2 | 55.5 | 46.3 | 68.6 |
| Decomp-Merge | 81.6 | 89.5 | 89.8 | 89.6 | 86.2 | 81.5 | 77.7 | 71.6 | 64.2 | 57.1 | 75.4 |
| ΔDecomp-Merge | 0.0 | 0.0 | 0.0 | ↑3.3 | ↑4.5 | ↑4.8 | ↑7.3 | ↑8.4 | ↑8.7 | ↑10.8 | ↑6.8 |
| **Qwen3-Reranker-8B** | | | | | | | | | | | |
| Orig. Base | 69.8 | 80.7 | 84.7 | 84.2 | 81.5 | 77.1 | 72.0 | 64.8 | 55.8 | 42.3 | 68.2 |
| Pure Rerank | 81.3 | 90.8 | 92.1 | 90.3 | 86.9 | 82.1 | 76.5 | 68.8 | 60.8 | 51.9 | 73.9 |
| Decomp-Merge | 81.3 | 90.8 | 92.1 | 91.6 | 88.4 | 85.2 | 81.2 | 75.7 | 70.0 | 66.6 | 79.8 |
| ΔDecomp-Merge | 0.0 | 0.0 | 0.0 | ↑1.3 | ↑1.5 | ↑3.1 | ↑4.7 | ↑6.9 | ↑9.2 | ↑14.7 | ↑5.9 |

Table 4: Impact of query decomposition on reranking across different query complexities. Values are scaled by 100. Orig. Base: retrieval-only baseline; Pure Rerank: rerank without decomposition; Decomp-Merge: rerank each sub-query then merge; ΔDecomp-Merge: improvement over Pure Rerank. We provide full results in Appendix B.

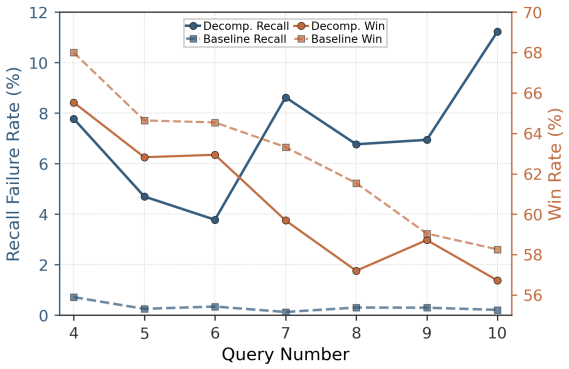



Figure 2: Recall failure rate (%) and average win rate of baseline queries versus decomposed sub-queries in retrieval. Undecomposed queries consistently exhibit a lower recall failure than sub-queries, while baseline retrieval achieves a higher win rate than retrieval with query decomposition.

## 6 Analysis

### 6.1 Why Retrieval Fails

**High Incidence of Recall Failure in Sub-queries** Query decomposition degrades initial retrieval primarily by missing target documents entirely. We define this as *recall failure*: failing to retrieve any positive document within the top 50 candidates. As Figure 2 illustrates, baseline queries maintain a recall failure rate strictly below 1%, whereas decomposed sub-queries experience a dramatic spike to 3.5%–12%. This drop in foundational recall establishes a hard bottleneck, explaining why early decomposition is detrimental to the overall pipeline.

**Increased Susceptibility to Easy Negatives** Beyond recall failure, applying decomposition during the initial retrieval phase also makes the system more susceptible to easy negative (EN) documents. As shown in Table 5, decomposed queries consistently retrieve more ENs within their top 50 candidates compared to the baseline. We provide an example in Appendix D illustrating how semantic dilution in a decomposed sub-query leads to the retrieval of easy negative documents.

| Dataset | Baseline | Decompose | Δ |
|---|---|---|---|
| Books | 78.88 | 80.81 | +1.93 |
| People | 78.18 | 78.39 | +0.21 |
| Movies | 79.14 | 80.46 | +1.32 |
| Medical Cases | 78.59 | 80.17 | +1.58 |
| Legal Documents | 78.33 | 80.04 | +1.71 |
| Average | 78.62 | 79.97 | +1.35 |

Table 5: Probability (%) of retrieving easy negative documents within the top 50 candidates for baseline retrieval and retrieval with query decomposition.

**Diminished Fine-Grained Capacity** We measure fine-grained discriminative capacity using the win rate metric to evaluate whether decomposition improves the relative ranking among highly similar items. As shown in Figure 2, baseline retrieval consistently outperforms the decomposed

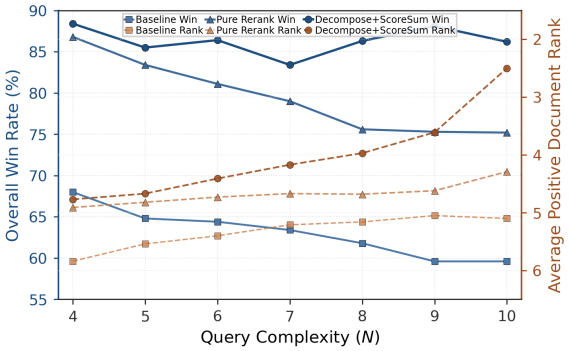



Figure 3: Average win rate (%) and average rank position of positive documents for baseline reranking and reranking with query decomposition across different query complexity levels. Decomposition in reranking achieves a higher win rate and a better (lower) average rank of positive documents than both pure reranking and baseline retrieval.

approach across all query lengths. While capacity declines for both methods as queries lengthen, decomposition provides no fine-grained advantage, maintaining a strictly lower win rate throughout.

The concurrent rise in recall failures and easy negative (EN) retrieval highlights a fundamental flaw in early query decomposition. Fragmenting a cohesive, information-rich query into isolated subqueries dilutes its overall semantic constraints. In a large corpus, these weakened constraints not only cause the model to miss target documents but also allow it to be misled by ENs. Furthermore, the win rate analysis confirms that decomposition offers no compensatory benefits; it fails to enhance fine-grained discriminative capacity, leaving the model unable to distinguish between similar candidates.

### 6.2 Why Reranking Works

**Enhanced Fine-Grained Capacity** We further evaluate fine-grained discriminative capacity using the win rate metric. As Figure 3 indicates, decomposition method significantly outperform standard reranking. Notably, this performance gap widens for highly complex, multi-condition queries. This demonstrates that query decomposition effectively isolates specific constraints, enabling the model to better capture and distinguish fine-grained details.

**Elevated Positive Document Ranking** Given that initial recall is already high for complex queries (Queries 4–10), the primary objective of the reranker is to elevate true positives above hard negatives. To evaluate this capability, we measure the average rank position of positive documents. As shown in Figure 3, applying query decomposition during reranking places positive documents significantly higher than standard reranking. This demonstrates that decomposition empowers the model to better differentiate true positives from hard negatives, thereby improving overall ranking quality.

The concurrent improvements in win rate and positive document ranking confirm the efficacy of query decomposition in reranking. By fragmenting complex queries, the model is forced to attend to individual semantic constraints, significantly enhancing its fine-grained discriminative capacity. Furthermore, because reranking operates on a constrained candidate pool, it avoids the context dilution inherent to large-scale retrieval. This focused environment makes decomposition a strictly beneficial strategy for final document scoring.

### 6.3 Generalization Analysis on SSRB

To demonstrate the generalizability of our end-to-end pipeline, we extend our evaluation to the Semi-Structured Retrieval Benchmark (SSRB) (Zhang et al., 2025a), a widely adopted dataset for multi-condition information retrieval. We randomly select three representative schemas for this analysis.

To ensure high-quality fixed-size query decomposition, we utilize GPT-5.3 (OpenAI, 2026) with tailored prompts (detailed in Appendix E). For the underlying models, we deploy Qwen3-Embedding-4B for retrieval and Qwen3-Reranker-8B for reranking. In our initial experiments, applying decomposition directly during the retrieval phase resulted in a sharp performance drop compared to the baseline. However, by deploying our proposed end-to-end pipeline, we observe substantial performance improvements. Detailed results are presented in Table 6.

| Stage | Setting | Schema | | |
|---|---|---|---|---|
| | | Camera | Employee | API |
| **NDCG@10** | | | | |
| Retrieval | Base | **11.1** | **10.0** | **19.3** |
| | Decomp | 8.9 | 6.8 | 12.6 |
| Reranking | Normal Rerank | 13.2 | 10.6 | 10.3 |
| | Stage-Aware | **14.9** | **12.7** | **11.4** |
| **Recall@100** | | | | |
| Retrieval | Base | **20.3** | **21.4** | **26.8** |
| | Decomp | 14.7 | 18.4 | 17.1 |

Table 6: Impact of query decomposition on retrieval and reranking performance across selected SSRB schemas. Values are scaled by 100.

### 6.4 Computational Efficiency Analysis

To quantify the computational overhead introduced by query decomposition, we evaluate the inference efficiency during the reranking stage. We compare the No-Decompose baseline with decomposition-based reranking across four reranker models: **bge-reranker-v2-m3**, **Qwen3-Reranker-0.6B**, **Qwen3-Reranker-4B**, and **Qwen3-Reranker-8B**, using the Books dataset.

For each query, we measure the total time required to rerank its 50 candidate documents. Under the decomposition setting, the reported runtime includes scoring all sub-query–candidate pairs and aggregating their scores. We use vLLM for inference acceleration with a batch size of 50. All experiments are conducted on a single NVIDIA A800 80GB GPU.

| Model | No-Decomp. | Decomp. | Ratio |
|---|---|---|---|
| Qwen3-0.6B | 0.2202 | 0.4693 | 2.13× |
| BGE-v2-m3 | 0.1706 | 0.3439 | 2.02× |
| Qwen3-4B | 0.6001 | 1.2969 | 2.16× |
| Qwen3-8B | 0.4536 | 0.9776 | 2.16× |

Table 7: Inference efficiency of decomposition-based reranking on the Books dataset. Runtime is reported in seconds per query (s/q), where each query contains 50 candidate documents.

## 7 Conclusion

In this work, we conducted a systematic empirical study on the role of query decomposition in multi-condition information retrieval. Our findings reveal a critical stage-dependency: applying decomposition during the initial retrieval phase degrades performance by inducing severe recall failure and introducing easy negatives. Conversely, deploying decomposition exclusively during the reranking phase significantly enhances the model's fine-grained discriminative capacity against hard negatives. Based on these insights, we proposed a novel **Stage-Aware Decomposition Framework**. By retaining the monolithic query for robust candidate generation and leveraging strategically partitioned sub-queries for cross-encoder reranking, our framework effectively balances global semantic context with strict logical verification. Ultimately, this approach offers a principled and highly effective pipeline for modern retrieval systems to handle complex, compositional user intents.

## Limitations

First, while our study explores three representative query decomposition strategies and two merging techniques across distinct multi-condition retrieval benchmarks, the combinatorial design space of multi-stage pipelines is vast. Future work could investigate alternative decomposition granularities along with various fusion methods to further explore the performance boundaries and trade-offs of stage-aware retrieval. Second, to establish a highly controlled and rigorous experimental baseline, our evaluation primarily focuses on the Qwen3 model family across both pipeline stages. While these consistent model configurations provide clean, unconfounded insights into stage-dependent behavior, exploring the cross-architecture generalizability of our findings to other emerging dense retrievers and rerankers represents a promising direction for broader validation. Third, this work is conceptually dedicated to isolating and diagnosing the intrinsic mechanics of query decomposition across pipeline stages, rather than focusing on engineering-driven performance competition with miscellaneous retrieval-augmentation frameworks. Extending this stage-aware principle to integrate with other advanced indexing or prompting paradigms remains an open avenue for future exploration.

## References


Paul J. L. Ammann, Jonas Golde, and Alan Akbik. 2025. Question decomposition for retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 497–507, Vienna, Austria. Association for Computational Linguistics.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *Preprint*, arXiv:2402.03216.

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston, MA, USA, July 19-23, 2009*, pages 758–759. ACM.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Yingqi Fan, Xuan Lu, Anhao Zhao, Junlong Tong, Ping Nie, Kai Zou, Yunpu Ma, Wei Zhang, and Xiaoyu Shen. 2026a. minireranker: Efficient multimodal reranking through visual cache reuse and interaction sparsity. *Preprint*, arXiv:2606.10759.

Yingqi Fan, Junlong Tong, Anhao Zhao, and Xiaoyu Shen. 2026b. What do visual tokens really encode? uncovering sparsity and redundancy in multimodal large language models. *arXiv preprint arXiv:2603.00510*.

Yingqi Fan, Anhao Zhao, Jinlan Fu, Junlong Tong, Hui Su, Yijie Pan, Wei Zhang, and Xiaoyu Shen. 2025. Visipruner: Decoding discontinuous cross-modal dynamics for efficient multimodal llms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 18896–18913.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. 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), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1762–1777. Association for Computational Linguistics.

Guangzeng Han and Xiaolei Huang. 2026. What makes good instruction-tuning data? an in-context learning perspective. *Preprint*, arXiv:2604.25132.

Haohang Huang, Xuan Lu, Mingyi Su, Xuan Zhang, Ziyan Jiang, Ping Nie, Kai Zou, Tomas Pfister, Wenhu Chen, Wei Zhang, Xiaoyu Shen, and Rui Meng. 2026. MMEB-v3: Measuring the performance gaps of omni-modality embedding models. In *Third Conference on Language Modeling*.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *CoRR*, abs/2503.09516.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics.

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, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 39–48. ACM.

Chaofan Li, Zheng Liu, Shitao Xiao, and Yingxia Shao. 2023. Making large language models a better foundation for dense retrieval. *Preprint*, arXiv:2312.15503.

Kevin Lin, Kyle Lo, Joseph Gonzalez, and Dan Klein. 2023. Decomposing complex queries for tip-of-the-tongue retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5521–5533, Singapore. Association for Computational Linguistics.

Sheng-Chieh Lin, Jheng-Hong Yang, Rodrigo Nogueira, Ming-Feng Tsai, Chuan-Ju Wang, and Jimmy Lin. 2021. Multi-stage conversational passage retrieval: An approach to fusing term importance estimation and neural query rewriting. *ACM Trans. Inf. Syst.*, 39(4):48:1–48:29.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Yaoyang Liu, Junlin Li, Yinjun Wu, and Zhen Chen. 2025. POQD: performance-oriented query decomposer for multi-vector retrieval. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net.

Xuan Lu, Haohang Huang, Yingqi Fan, Junlong Tong, Yuxuan Zhang, Ping Nie, Rui Meng, and Xiaoyu Shen. 2026a. Comprank: Efficient llm reranking via token-level compression and decoding-free scoring. *Preprint*, arXiv:2606.11700.

Xuan Lu, Haohang Huang, Rui Meng, Yaohui Jin, Wenjun Zeng, and Xiaoyu Shen. 2026b. Rethinking reasoning in document ranking: Why chain-of-thought falls short. In *The Fourteenth International Conference on Learning Representations*.

Xuan Lu, Haohang Huang, Rui Meng, Yaohui Jin, Wenjun Zeng, and Xiaoyu Shen. 2026c. Tools are underdocumented: Simple document expansion boosts tool retrieval. In *The Fourteenth International Conference on Learning Representations*.

Xuan Lu, Kangle Li, Haohang Huang, Rui Meng, Wenjun Zeng, and Xiaoyu Shen. 2026d. Beyond global similarity: Multi-conditional retrieval for fine-grained cross-modal understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9699–9709.

Xuan Lu, Sifan Liu, Bochao Yin, Yongqi Li, Xinghao Chen, Hui Su, Yaohui Jin, Wenjun Zeng, and Xiaoyu Shen. 2025. MultiConIR: Towards multi-condition information retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 13471–13494, Suzhou, China. Association for Computational Linguistics.

Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019. Multi-hop reading comprehension through question decomposition and rescoring. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 6097–6109. Association for Computational Linguistics.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Rodrigo Nogueira and Kyunghyun Cho. 2017. Task-oriented query reformulation with reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 574–583. Association for Computational Linguistics.

OpenAI. 2026. Openai GPT-5 system card. *CoRR*, abs/2601.03267.

Roxana Petcu, Kenton Murray, Daniel Khashabi, Evangelos Kanoulas, Maarten de Rijke, Dawn J. Lawrie, and Kevin Duh. 2025. Query decomposition for RAG: balancing exploration-exploitation. *CoRR*, abs/2510.18633.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, Findings of ACL, pages 5687–5711. Association for Computational Linguistics.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Hongjin Su, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han-yu Wang, Haisu Liu, Quan Shi, Zachary S. Siegel, Michael Tang, Ruoxi Sun, Jinsung Yoon, Sercan Ö. Arik, Danqi Chen, and Tao Yu. 2025. BRIGHT: A realistic and challenging benchmark for reasoning-intensive retrieval. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is chatgpt good at search? investigating large language models as re-ranking agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 14918–14937. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *CoRR*, abs/2104.08663.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multi-hop questions via single-hop question composition. *Trans. Assoc. Comput. Linguistics*, 10:539–554.

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 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Xiaoyue Wang, Jianyou Wang, Weili Cao, Kaicheng Wang, Ramamohan Paturi, and Leon Bergen. 2024. BIRCO: A benchmark of information retrieval tasks with complex objectives. *CoRR*, abs/2402.14151.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, pages 641–649. ACM.

Ka-Ping Yee, Kirsten Swearingen, Kevin Li, and Marti A. Hearst. 2003. Faceted metadata for image search and browsing. In *Proceedings of the 2003 Conference on Human Factors in Computing Systems, CHI 2003, Ft. Lauderdale, Florida, USA, April 5-10, 2003*, pages 401–408. ACM.

Xin Zhang, Mingxin Li, Yanzhao Zhang, Dingkun Long, Yongqi Li, Yinghui Li, Pengjun Xie, Meishan Zhang, Wenjie Li, Min Zhang, and Philip S. Yu. 2025a. SSRB: Direct natural language querying to massive heterogeneous semi-structured data. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025b. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *CoRR*, abs/2506.05176.

Shu Zhao, Tan Yu, Anbang Xu, Japinder Singh, Aaditya Shukla, and Rama Akkiraju. 2025. Parallelsearch: Train your llms to decompose query and search sub-queries in parallel with reinforcement learning. *CoRR*, abs/2508.09303.

Yunfei Zhong, Jun Yang, Yixing Fan, Jiafeng Guo, Lixin Su, Maarten de Rijke, Ruqing Zhang, Dawei Yin, and Xueqi Cheng. 2025. Reasoning-enhanced query

understanding through decomposition and interpretation. *CoRR*, abs/2509.06544.

Yilun Zhu, Nikhita Vedula, and Shervin Malmasi. 2025. Hint-augmented re-ranking: Efficient product search using LLM-based query decomposition. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 200–216, Mumbai, India. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics.

## A Prompt Template for Qwen3-8B to Conduct Adaptive Decomposition

In this appendix, we detail the exact prompt instructions provided to Qwen3-8B to perform adaptive query decomposition on the MultiConIR benchmark. Unlike fixed-size decomposition, the adaptive strategy allows the model to dynamically determine both the number of sub-queries and the number of conditions assigned to each sub-query, based on the semantic structure of the original query. The conversation follows the standard chat template with three roles: `system`, `user`, and `assistant`.

### System Message

You are an expert in complex, multi-condition information retrieval. Your task is to adaptively decompose a given complex search query into distinct, independent sub-queries to optimize downstream retrieval performance. You should dynamically determine both the number of sub-queries and how many conditions each sub-query contains, based on the semantic coherence of the original query.

**Rules:**

1. **Adaptive Granularity:** Decide the optimal number of sub-queries based on the semantic structure of the original query.
2. **Semantic Coherence:** Group semantically related conditions together within the same sub-query. Different sub-queries may contain different numbers of conditions, depending on how the conditions naturally cluster.
3. **No Information Overlap:** The conditions must be mutually exclusive across sub-queries. Do not duplicate specific constraints or information across the decomposed segments.
4. **Preserve Original Wording:** Extract and segment the text while strictly preserving the original phrasing, vocabulary, and sentence structure as much as possible. Avoid aggressively paraphrasing, rewriting, or hallucinating new information.

**Output Format:** Strictly output the decomposed sub-queries in the following format, with one sub-query per line and no additional conversational text, reasoning, or thinking process:
`[Original_ID]_subq_[Index], "[Sub-query text]"`

### User Message (Few-shot Example Input)

```
4_Query_8, "Find a movie matching these conditions: 1. Chaplin, dressed as tramp, sits with wife. 2. Charlie holding onions, investigates odd smell. 3. Origin: American. 4. Charlie drunkenly sees dummy as opponent. 5. Mabel reveals dummy to fighting Charlie. 6. Director: Charlie Chaplin. 7. Cast includes Charlie Chaplin, Mabel Normand. 8. Man with tennis racquet approaches wife in bar."
```

### Assistant Message (Few-shot Example Output)

```
4_Query_8_subq_1, "Find a movie matching these conditions: Chaplin, dressed as tramp, sits with wife. Man with tennis racquet approaches wife in bar."
4_Query_8_subq_2, "Find a movie matching these conditions: Charlie holding onions, investigates odd smell. Charlie drunkenly sees dummy as opponent. Mabel reveals dummy to fighting Charlie."
4_Query_8_subq_3, "Find a movie matching these conditions: Origin: American. Director: Charlie Chaplin. Cast includes Charlie Chaplin, Mabel Normand."
```

### User Message (Actual Query Input)

```
{query_id}, "{query_text}"
```

## B Detailed Results

### B.1 Detailed Retrieval Results

Table 3 reports the average retrieval performance across the five datasets. For completeness, we provide the detailed results for each dataset (Books, Movies, People, Medical Cases, and Legal Documents) in the Appendix.

Tables 8, 9, 10 and 11 present the detailed results for each dataset in MultiConIR.

| Dataset | Setting | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | **NDCG@10** | | | | | |
| | Baseline | 75.3 | 76.1 | 77.8 | 75.8 | 74.5 | 69.9 | 66.1 | 58.1 | 49.4 | 36.7 |
| **Books** | Decomp-RRF | 75.3 | 76.1 | 77.8 | 63.4 | 64.8 | 61.3 | 56.0 | 50.9 | 42.7 | 29.8 |
| | Decomp-Sum | 75.3 | 76.1 | 77.8 | 70.1 | 69.4 | 66.3 | 58.6 | 53.7 | 45.7 | 31.8 |
| | Baseline | 55.0 | 75.3 | 88.0 | 88.5 | 87.1 | 81.8 | 76.9 | 69.8 | 59.5 | 45.5 |
| **Movies** | Decomp-RRF | 55.0 | 75.3 | 88.0 | 74.2 | 77.9 | 77.9 | 64.8 | 59.2 | 52.7 | 37.9 |
| | Decomp-Sum | 55.0 | 75.3 | 88.0 | 79.6 | 80.7 | 80.3 | 67.9 | 62.6 | 55.6 | 39.4 |
| | Baseline | 80.7 | 87.8 | 86.9 | 85.8 | 81.5 | 77.2 | 71.5 | 64.2 | 55.1 | 43.9 |
| **People** | Decomp-RRF | 80.7 | 87.8 | 86.9 | 80.9 | 78.7 | 75.0 | 67.3 | 62.1 | 52.9 | 40.9 |
| | Decomp-Sum | 80.7 | 87.8 | 86.9 | 84.8 | 80.4 | 77.0 | 69.7 | 64.0 | 55.0 | 42.2 |
| | Baseline | 61.6 | 78.6 | 83.7 | 84.3 | 81.4 | 77.9 | 72.3 | 66.9 | 59.6 | 43.8 |
| **Medical Cases** | Decomp-RRF | 61.6 | 78.6 | 83.7 | 71.2 | 72.0 | 71.5 | 63.5 | 60.0 | 51.9 | 38.9 |
| | Decomp-Sum | 61.6 | 78.6 | 83.7 | 77.8 | 76.7 | 75.1 | 67.4 | 63.1 | 55.1 | 39.4 |
| | Baseline | 76.2 | 85.6 | 87.2 | 86.7 | 83.2 | 78.9 | 73.3 | 65.2 | 55.6 | 41.4 |
| **Legal Documents** | Decomp-RRF | 76.2 | 85.6 | 87.2 | 73.2 | 76.2 | 73.1 | 62.6 | 57.5 | 48.7 | 35.7 |
| | Decomp-Sum | 76.2 | 85.6 | 87.2 | 77.7 | 80.3 | 75.7 | 65.6 | 60.4 | 51.8 | 37.1 |
| | | | | | | **Recall@50** | | | | | |
| | Baseline | 89.6 | 93.0 | 95.6 | 95.4 | 97.8 | 97.9 | 98.6 | 97.9 | 98.1 | 98.9 |
| **Books** | Decomp-RRF | 89.6 | 93.0 | 95.6 | 92.8 | 95.4 | 96.1 | 94.6 | 94.5 | 94.9 | 92.1 |
| | Decomp-Sum | 89.6 | 93.0 | 95.6 | 92.8 | 95.7 | 96.3 | 95.1 | 94.6 | 95.1 | 91.7 |
| | Baseline | 69.2 | 87.9 | 96.8 | 99.0 | 99.8 | 99.8 | 99.9 | 99.8 | 100.0 | 100.0 |
| **Movies** | Decomp-RRF | 69.2 | 87.9 | 96.8 | 91.6 | 94.9 | 98.8 | 94.3 | 96.5 | 98.4 | 98.6 |
| | Decomp-Sum | 69.2 | 87.9 | 96.8 | 91.7 | 95.0 | 98.9 | 91.8 | 94.3 | 97.8 | 94.5 |
| | Baseline | 94.5 | 98.8 | 99.4 | 99.8 | 100.0 | 100.0 | 100.0 | 100.0 | 99.9 | 100.0 |
| **People** | Decomp-RRF | 94.5 | 98.8 | 99.4 | 99.8 | 100.0 | 99.9 | 99.8 | 100.0 | 100.0 | 99.5 |
| | Decomp-Sum | 94.5 | 98.8 | 99.4 | 100.0 | 100.0 | 100.0 | 99.8 | 100.0 | 100.0 | 99.8 |
| | Baseline | 82.7 | 95.0 | 99.0 | 99.7 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| **Medical Cases** | Decomp-RRF | 82.7 | 95.0 | 99.0 | 97.2 | 97.8 | 98.7 | 97.6 | 98.5 | 98.1 | 97.6 |
| | Decomp-Sum | 82.7 | 95.0 | 99.0 | 97.4 | 97.9 | 98.7 | 97.1 | 98.6 | 97.6 | 93.7 |
| | Baseline | 92.9 | 97.7 | 97.6 | 99.5 | 99.8 | 99.6 | 100.0 | 99.8 | 100.0 | 100.0 |
| **Legal Documents** | Decomp-RRF | 92.9 | 97.7 | 97.6 | 94.4 | 98.4 | 98.0 | 93.5 | 96.9 | 97.6 | 92.7 |
| | Decomp-Sum | 92.9 | 97.7 | 97.6 | 94.6 | 98.1 | 98.2 | 93.8 | 96.8 | 97.8 | 90.7 |

Table 8: Detailed retrieval results for each dataset using **BGE-large-en-v1.5**. Values are scaled by 100.

| Dataset | Setting | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | **NDCG@10** | | | | | | |
| | Baseline | 67.9 | 82.7 | 82.4 | 82.4 | 79.0 | 77.1 | 70.8 | 64.0 | 55.6 | 43.5 |
| **Books** | Decomp-RRF | 67.9 | 82.7 | 82.4 | 73.6 | 74.1 | 71.7 | 65.4 | 58.3 | 52.1 | 37.8 |
| | Decomp-Sum | 67.9 | 82.7 | 82.4 | 78.4 | 76.3 | 75.0 | 68.5 | 62.3 | 54.7 | 39.2 |
| | Baseline | 62.1 | 84.1 | 89.3 | 88.4 | 85.3 | 81.0 | 75.8 | 68.4 | 59.0 | 45.1 |
| **Movies** | Decomp-RRF | 62.1 | 84.1 | 89.3 | 78.6 | 79.9 | 75.8 | 67.2 | 62.3 | 54.1 | 39.1 |
| | Decomp-Sum | 62.1 | 84.1 | 89.3 | 82.5 | 83.0 | 79.1 | 70.0 | 66.0 | 56.7 | 42.9 |
| | Baseline | 78.2 | 90.6 | 91.3 | 88.5 | 84.5 | 80.3 | 75.7 | 68.4 | 61.1 | 50.6 |
| **People** | Decomp-RRF | 78.2 | 90.6 | 91.3 | 83.7 | 80.5 | 77.6 | 71.2 | 64.8 | 58.1 | 45.5 |
| | Decomp-Sum | 78.2 | 90.6 | 91.3 | 86.8 | 81.4 | 78.5 | 73.5 | 66.5 | 59.0 | 47.0 |
| | Baseline | 57.9 | 77.8 | 82.1 | 83.2 | 80.5 | 76.9 | 71.7 | 66.6 | 59.3 | 46.0 |
| **Medical Cases** | Decomp-RRF | 57.9 | 77.8 | 82.1 | 77.8 | 76.3 | 73.3 | 67.4 | 62.5 | 54.1 | 39.8 |
| | Decomp-Sum | 57.9 | 77.8 | 82.1 | 81.2 | 79.2 | 75.6 | 70.1 | 65.2 | 57.0 | 41.4 |
| | Baseline | 72.5 | 84.9 | 85.8 | 86.5 | 83.0 | 78.1 | 73.2 | 65.1 | 57.6 | 42.3 |
| **Legal Documents** | Decomp-RRF | 72.5 | 84.9 | 85.8 | 77.5 | 78.2 | 75.7 | 65.9 | 62.6 | 54.8 | 39.1 |
| | Decomp-Sum | 72.5 | 84.9 | 85.8 | 83.3 | 81.6 | 77.5 | 70.8 | 65.7 | 56.5 | 42.0 |
| | | | | | **Recall@50** | | | | | | |
| | Baseline | 86.6 | 96.4 | 97.3 | 98.9 | 99.6 | 100.0 | 100.0 | 99.9 | 99.8 | 99.8 |
| **Books** | Decomp-RRF | 86.6 | 96.4 | 97.3 | 97.2 | 97.5 | 99.3 | 97.9 | 98.9 | 99.4 | 97.5 |
| | Decomp-Sum | 86.6 | 96.4 | 97.3 | 96.8 | 97.5 | 99.5 | 98.4 | 99.2 | 99.6 | 97.9 |
| | Baseline | 75.4 | 93.3 | 99.0 | 99.4 | 99.7 | 99.5 | 100.0 | 100.0 | 100.0 | 100.0 |
| **Movies** | Decomp-RRF | 75.4 | 93.3 | 99.0 | 94.6 | 98.1 | 97.8 | 96.5 | 98.4 | 99.3 | 99.0 |
| | Decomp-Sum | 75.4 | 93.3 | 99.0 | 94.5 | 98.1 | 97.7 | 96.5 | 98.7 | 99.4 | 98.2 |
| | Baseline | 93.6 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| **People** | Decomp-RRF | 93.6 | 100.0 | 100.0 | 99.9 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| | Decomp-Sum | 93.6 | 100.0 | 100.0 | 99.9 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| | Baseline | 83.5 | 96.4 | 98.8 | 99.6 | 99.8 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| **Medical Cases** | Decomp-RRF | 83.5 | 96.4 | 98.8 | 98.7 | 99.2 | 99.6 | 99.4 | 99.6 | 99.5 | 99.3 |
| | Decomp-Sum | 83.5 | 96.4 | 98.8 | 98.9 | 99.2 | 99.7 | 99.5 | 99.6 | 99.5 | 99.8 |
| | Baseline | 89.5 | 96.5 | 98.0 | 99.3 | 99.4 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| **Legal Documents** | Decomp-RRF | 89.5 | 96.5 | 98.0 | 97.0 | 98.5 | 99.2 | 98.1 | 99.8 | 98.8 | 98.2 |
| | Decomp-Sum | 89.5 | 96.5 | 98.0 | 97.2 | 98.5 | 99.3 | 99.3 | 99.8 | 99.5 | 99.0 |

Table 9: Detailed retrieval results for each dataset using **Qwen3-Embedding-0.6B**. Values are scaled by 100.

| Dataset | Setting | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | **NDCG@10** | | | | | | |
| | Baseline | 74.1 | 85.1 | 86.5 | 85.2 | 82.3 | 77.8 | 72.1 | 65.1 | 55.9 | 43.2 |
| **Books** | Decomp-RRF | 74.1 | 85.1 | 86.5 | 78.8 | 77.6 | 73.7 | 66.5 | 59.9 | 51.3 | 39.1 |
| | Decomp-Sum | 74.1 | 85.1 | 86.5 | 83.3 | 80.6 | 76.0 | 69.1 | 62.7 | 53.7 | 40.4 |
| | Baseline | 53.1 | 76.4 | 85.7 | 86.3 | 84.7 | 80.0 | 74.8 | 66.8 | 56.3 | 42.9 |
| **Movies** | Decomp-RRF | 53.1 | 76.4 | 85.7 | 71.8 | 74.6 | 75.5 | 63.3 | 60.4 | 54.0 | 39.4 |
| | Decomp-Sum | 53.1 | 76.4 | 85.7 | 78.1 | 79.5 | 78.6 | 68.9 | 64.3 | 58.3 | 43.8 |
| | Baseline | 82.8 | 91.6 | 91.1 | 89.0 | 84.3 | 80.2 | 73.7 | 66.9 | 58.8 | 44.2 |
| **People** | Decomp-RRF | 82.8 | 91.6 | 91.1 | 84.8 | 81.7 | 76.9 | 68.4 | 61.8 | 54.0 | 39.7 |
| | Decomp-Sum | 82.8 | 91.6 | 91.1 | 87.3 | 83.1 | 78.3 | 71.1 | 63.5 | 55.7 | 41.4 |
| | Baseline | 62.5 | 79.0 | 83.5 | 83.0 | 78.4 | 74.7 | 69.7 | 65.0 | 56.5 | 44.8 |
| **Medical Cases** | Decomp-RRF | 62.5 | 79.0 | 83.5 | 79.4 | 76.0 | 72.5 | 65.2 | 61.0 | 52.1 | 41.9 |
| | Decomp-Sum | 62.5 | 79.0 | 83.5 | 83.0 | 77.7 | 73.7 | 67.5 | 62.8 | 53.7 | 44.1 |
| | Baseline | 76.7 | 86.5 | 87.8 | 86.5 | 83.0 | 78.0 | 73.3 | 65.7 | 56.0 | 39.6 |
| **Legal Documents** | Decomp-RRF | 76.7 | 86.5 | 87.8 | 81.1 | 80.8 | 76.2 | 70.4 | 64.2 | 55.4 | 41.9 |
| | Decomp-Sum | 76.7 | 86.5 | 87.8 | 85.1 | 82.5 | 77.9 | 73.6 | 67.7 | 57.5 | 45.7 |
| | | | | | **Recall@50** | | | | | | |
| | Baseline | 89.9 | 97.4 | 99.0 | 99.5 | 99.9 | 99.9 | 100.0 | 100.0 | 99.8 | 100.0 |
| **Books** | Decomp-RRF | 89.9 | 97.4 | 99.0 | 98.7 | 99.3 | 99.7 | 99.3 | 99.8 | 99.3 | 99.6 |
| | Decomp-Sum | 89.9 | 97.4 | 99.0 | 98.7 | 99.3 | 99.7 | 99.6 | 99.8 | 99.5 | 99.6 |
| | Baseline | 64.5 | 86.4 | 96.2 | 98.1 | 99.7 | 99.5 | 100.0 | 99.9 | 100.0 | 100.0 |
| **Movies** | Decomp-RRF | 64.5 | 86.4 | 96.2 | 89.5 | 93.6 | 97.7 | 91.1 | 95.0 | 96.7 | 98.8 |
| | Decomp-Sum | 64.5 | 86.4 | 96.2 | 89.5 | 93.7 | 97.7 | 94.2 | 97.5 | 98.5 | 98.2 |
| | Baseline | 95.9 | 99.8 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| **People** | Decomp-RRF | 95.9 | 99.8 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| | Decomp-Sum | 95.9 | 99.8 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| | Baseline | 88.2 | 97.5 | 99.7 | 99.8 | 99.9 | 99.8 | 99.9 | 100.0 | 100.0 | 100.0 |
| **Medical Cases** | Decomp-RRF | 88.2 | 97.5 | 99.7 | 99.6 | 100.0 | 99.5 | 99.1 | 99.8 | 99.5 | 99.8 |
| | Decomp-Sum | 88.2 | 97.5 | 99.7 | 99.7 | 100.0 | 99.6 | 99.6 | 99.8 | 99.5 | 99.8 |
| | Baseline | 92.2 | 97.4 | 98.4 | 99.3 | 100.0 | 100.0 | 100.0 | 99.9 | 100.0 | 99.8 |
| **Legal Documents** | Decomp-RRF | 92.2 | 97.4 | 98.4 | 98.4 | 99.3 | 99.6 | 99.4 | 99.8 | 99.6 | 99.0 |
| | Decomp-Sum | 92.2 | 97.4 | 98.4 | 98.5 | 99.3 | 99.5 | 99.6 | 99.8 | 99.8 | 99.2 |

Table 10: Detailed retrieval results for each dataset using **Qwen3-Embedding-4B**. Values are scaled by 100.

| Dataset | Setting | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | **NDCG@10** | | | | | | |
| | Baseline | 76.0 | 85.3 | 85.9 | 84.1 | 82.0 | 77.7 | 71.5 | 65.2 | 56.0 | 47.1 |
| **Books** | Decomp-RRF | 76.0 | 85.3 | 85.9 | 79.4 | 78.4 | 74.8 | 67.8 | 62.0 | 53.7 | 45.1 |
| | Decomp-Sum | 76.0 | 85.3 | 85.9 | 82.8 | 80.3 | 76.5 | 69.7 | 63.1 | 55.0 | 46.9 |
| | Baseline | 57.8 | 79.8 | 88.8 | 89.3 | 88.1 | 83.9 | 79.3 | 72.5 | 62.9 | 45.8 |
| **Movies** | Decomp-RRF | 57.8 | 79.8 | 88.8 | 76.5 | 79.3 | 77.8 | 67.1 | 63.8 | 58.0 | 39.3 |
| | Decomp-Sum | 57.8 | 79.8 | 88.8 | 81.2 | 82.9 | 81.5 | 72.6 | 68.5 | 62.9 | 44.3 |
| | Baseline | 87.4 | 93.0 | 91.3 | 89.0 | 85.8 | 81.5 | 75.4 | 69.1 | 62.2 | 51.4 |
| **People** | Decomp-RRF | 87.4 | 93.0 | 91.3 | 84.9 | 82.2 | 77.8 | 71.5 | 64.8 | 57.6 | 43.6 |
| | Decomp-Sum | 87.4 | 93.0 | 91.3 | 87.5 | 83.2 | 79.2 | 72.7 | 65.7 | 57.9 | 44.9 |
| | Baseline | 66.2 | 82.7 | 85.0 | 84.1 | 80.0 | 75.7 | 69.6 | 64.2 | 58.0 | 45.6 |
| **Medical Cases** | Decomp-RRF | 66.2 | 82.7 | 85.0 | 78.4 | 76.1 | 72.2 | 65.6 | 60.5 | 53.7 | 44.0 |
| | Decomp-Sum | 66.2 | 82.7 | 85.0 | 82.4 | 78.3 | 74.2 | 67.8 | 63.0 | 55.9 | 46.1 |
| | Baseline | 75.4 | 84.5 | 86.1 | 85.0 | 81.6 | 77.6 | 71.9 | 63.9 | 54.2 | 37.3 |
| **Legal Documents** | Decomp-RRF | 75.4 | 84.5 | 86.1 | 79.2 | 77.7 | 74.6 | 68.3 | 59.7 | 50.6 | 36.8 |
| | Decomp-Sum | 75.4 | 84.5 | 86.1 | 83.3 | 80.2 | 76.6 | 70.9 | 61.4 | 53.4 | 39.5 |
| | | | | | **Recall@50** | | | | | | |
| | Baseline | 91.3 | 98.0 | 99.7 | 99.8 | 100.0 | 100.0 | 100.0 | 100.0 | 99.8 | 100.0 |
| **Books** | Decomp-RRF | 91.3 | 98.0 | 99.7 | 99.0 | 99.7 | 99.9 | 99.6 | 99.9 | 99.6 | 99.8 |
| | Decomp-Sum | 91.3 | 98.0 | 99.7 | 99.1 | 99.7 | 99.9 | 99.7 | 99.9 | 99.7 | 99.8 |
| | Baseline | 69.4 | 88.8 | 96.5 | 98.9 | 99.8 | 99.9 | 100.0 | 100.0 | 100.0 | 100.0 |
| **Movies** | Decomp-RRF | 69.4 | 88.8 | 96.5 | 90.1 | 95.0 | 98.3 | 94.0 | 97.2 | 98.5 | 99.2 |
| | Decomp-Sum | 69.4 | 88.8 | 96.5 | 90.4 | 95.0 | 98.1 | 94.3 | 98.0 | 99.2 | 97.8 |
| | Baseline | 97.2 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| **People** | Decomp-RRF | 97.2 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| | Decomp-Sum | 97.2 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| | Baseline | 89.3 | 98.7 | 99.8 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| **Medical Cases** | Decomp-RRF | 89.3 | 98.7 | 99.8 | 99.4 | 100.0 | 99.9 | 99.8 | 99.8 | 99.5 | 99.8 |
| | Decomp-Sum | 89.3 | 98.7 | 99.8 | 99.6 | 100.0 | 100.0 | 99.9 | 99.9 | 99.9 | 100.0 |
| | Baseline | 91.5 | 97.6 | 98.8 | 99.3 | 99.9 | 100.0 | 100.0 | 99.8 | 100.0 | 100.0 |
| **Legal Documents** | Decomp-RRF | 91.5 | 97.6 | 98.8 | 98.5 | 99.5 | 99.6 | 99.3 | 99.0 | 99.1 | 98.5 |
| | Decomp-Sum | 91.5 | 97.6 | 98.8 | 98.7 | 99.7 | 99.8 | 100.0 | 99.3 | 100.0 | 99.5 |

Table 11: Detailed retrieval results for each dataset using **Qwen3-Embedding-8B**. Values are scaled by 100.

### B.2 Detailed Reranking Results

Table 4 reports the average reranking performance across the five datasets. For completeness, we provide the detailed results for each dataset (Books, Movies, People, Medical Cases, and Legal Documents) in the Appendix.

Tables 12, 13 and 14 present the detailed results for each dataset in MultiConIR.

| Dataset | Setting | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | **NDCG@10** | | | | | |
| **Books** | Orig. Base | 75.3 | 76.1 | 77.8 | 75.8 | 74.5 | 69.9 | 66.1 | 58.1 | 49.4 | 36.7 |
| | Pure Rerank | 85.2 | 88.0 | 86.9 | 82.1 | 79.8 | 73.7 | 68.0 | 59.6 | 51.4 | 43.8 |
| | Decomp-Sum | 85.2 | 88.0 | 86.9 | 85.6 | 84.9 | 80.0 | 77.0 | 71.6 | 64.0 | 55.9 |
| | Decomp-RRF | 85.2 | 88.0 | 86.9 | 84.6 | 83.9 | 79.3 | 76.1 | 71.2 | 63.3 | 58.7 |
| **Movies** | Orig. Base | 55.0 | 75.3 | 88.0 | 88.5 | 87.1 | 81.8 | 76.9 | 69.8 | 59.5 | 45.5 |
| | Pure Rerank | 65.9 | 83.4 | 87.3 | 82.6 | 76.8 | 69.9 | 56.8 | 49.9 | 49.9 | 38.4 |
| | Decomp-Sum | 65.9 | 83.4 | 87.3 | 90.8 | 85.6 | 79.6 | 76.6 | 70.7 | 58.8 | 52.9 |
| | Decomp-RRF | 65.9 | 83.4 | 87.3 | 88.8 | 84.5 | 78.0 | 74.1 | 66.7 | 56.0 | 47.1 |
| **People** | Orig. Base | 80.7 | 87.8 | 86.9 | 85.8 | 81.5 | 77.2 | 71.5 | 64.2 | 55.1 | 43.9 |
| | Pure Rerank | 91.1 | 92.8 | 89.0 | 84.6 | 79.2 | 74.6 | 66.4 | 59.5 | 51.5 | 39.2 |
| | Decomp-Sum | 91.1 | 92.8 | 89.0 | 89.5 | 85.3 | 81.9 | 78.3 | 72.1 | 64.1 | 53.3 |
| | Decomp-RRF | 91.1 | 92.8 | 89.0 | 89.0 | 85.3 | 82.2 | 77.9 | 72.1 | 63.9 | 53.9 |
| **Medical Cases** | Orig. Base | 61.6 | 78.6 | 83.7 | 84.3 | 81.4 | 77.9 | 72.3 | 66.9 | 59.6 | 43.8 |
| | Pure Rerank | 72.9 | 87.5 | 88.1 | 83.3 | 77.7 | 71.8 | 64.4 | 56.3 | 50.1 | 40.1 |
| | Decomp-Sum | 72.9 | 87.5 | 88.1 | 90.0 | 85.2 | 80.8 | 76.2 | 70.5 | 62.1 | 53.4 |
| | Decomp-RRF | 72.9 | 87.5 | 88.1 | 87.8 | 84.4 | 80.5 | 74.2 | 68.0 | 60.8 | 51.7 |
| **Legal Documents** | Orig. Base | 76.2 | 85.6 | 87.2 | 86.7 | 83.2 | 78.9 | 73.3 | 65.2 | 55.6 | 41.4 |
| | Pure Rerank | 85.0 | 86.8 | 82.4 | 80.0 | 74.8 | 70.4 | 65.0 | 56.9 | 50.4 | 38.3 |
| | Decomp-Sum | 85.0 | 86.8 | 82.4 | 86.8 | 81.3 | 75.1 | 72.8 | 66.5 | 58.2 | 48.4 |
| | Decomp-RRF | 85.0 | 86.8 | 82.4 | 85.2 | 80.4 | 75.3 | 73.1 | 66.3 | 58.4 | 48.9 |

Table 12: Detailed reranking results for each dataset using **Qwen3-Reranker-0.6B**. Values are scaled by 100.

| Dataset | Setting | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | **NDCG@10** | | | | | | |
| **Books** | Orig. Base | 75.3 | 76.1 | 77.8 | 75.8 | 74.5 | 69.9 | 66.1 | 58.1 | 49.4 | 36.7 |
| | Pure Rerank | 86.4 | 89.2 | 88.4 | 84.3 | 81.6 | 76.4 | 70.6 | 64.1 | 56.6 | 52.7 |
| | Decomp-Sum | 86.4 | 89.2 | 88.4 | 86.6 | 84.9 | 81.1 | 75.4 | 68.5 | 62.3 | 59.0 |
| | Decomp-RRF | 86.4 | 89.2 | 88.4 | 84.9 | 83.5 | 79.5 | 73.2 | 67.6 | 61.0 | 58.3 |
| **Movies** | Orig. Base | 55.0 | 75.3 | 88.0 | 88.5 | 87.1 | 81.8 | 76.9 | 69.8 | 59.5 | 45.5 |
| | Pure Rerank | 66.3 | 85.3 | 91.0 | 88.7 | 83.2 | 77.4 | 71.2 | 63.3 | 53.9 | 42.2 |
| | Decomp-Sum | 66.3 | 85.3 | 91.0 | 91.4 | 89.0 | 84.9 | 81.4 | 74.0 | 66.7 | 56.0 |
| | Decomp-RRF | 66.3 | 85.3 | 91.0 | 89.7 | 87.2 | 83.3 | 78.3 | 71.3 | 62.5 | 52.8 |
| **People** | Orig. Base | 80.7 | 87.8 | 86.9 | 85.8 | 81.5 | 77.2 | 71.5 | 64.2 | 55.1 | 43.9 |
| | Pure Rerank | 93.0 | 93.9 | 90.7 | 85.5 | 80.4 | 76.0 | 69.8 | 62.8 | 56.9 | 45.6 |
| | Decomp-Sum | 93.0 | 93.9 | 90.7 | 90.5 | 84.8 | 80.5 | 77.7 | 71.6 | 65.5 | 58.9 |
| | Decomp-RRF | 93.0 | 93.9 | 90.7 | 88.8 | 83.2 | 79.5 | 75.9 | 68.7 | 63.5 | 55.1 |
| **Medical Cases** | Orig. Base | 61.6 | 78.6 | 83.7 | 84.3 | 81.4 | 77.9 | 72.3 | 66.9 | 59.6 | 43.8 |
| | Pure Rerank | 74.2 | 89.4 | 92.0 | 89.1 | 85.0 | 79.9 | 72.8 | 66.7 | 58.7 | 50.6 |
| | Decomp-Sum | 74.2 | 89.4 | 92.0 | 91.0 | 87.3 | 82.4 | 78.0 | 74.7 | 64.4 | 58.6 |
| | Decomp-RRF | 74.2 | 89.4 | 92.0 | 89.2 | 85.8 | 81.1 | 74.4 | 68.4 | 62.1 | 54.4 |
| **Legal Documents** | Orig. Base | 76.2 | 85.6 | 87.2 | 86.7 | 83.2 | 78.9 | 73.3 | 65.2 | 55.6 | 41.4 |
| | Pure Rerank | 88.0 | 89.7 | 86.9 | 83.8 | 78.3 | 73.7 | 67.5 | 59.1 | 51.6 | 40.4 |
| | Decomp-Sum | 88.0 | 89.7 | 86.9 | 88.6 | 84.8 | 78.7 | 75.7 | 69.4 | 62.2 | 53.2 |
| | Decomp-RRF | 88.0 | 89.7 | 86.9 | 87.4 | 83.5 | 78.3 | 74.4 | 66.6 | 59.1 | 51.1 |

Table 13: Detailed reranking results for each dataset using **Qwen3-Reranker-4B**. Values are scaled by 100.

| Dataset | Setting | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | **NDCG@10** | | | | | | |
| **Books** | Orig. Base | 75.3 | 76.1 | 77.8 | 75.8 | 74.5 | 69.9 | 66.1 | 58.1 | 49.4 | 36.7 |
| | Pure Rerank | 86.5 | 89.1 | 88.8 | 85.0 | 82.6 | 77.4 | 72.1 | 65.7 | 60.9 | 60.8 |
| | Decomp-Sum | 86.5 | 89.1 | 88.8 | 87.4 | 84.5 | 81.1 | 76.5 | 69.9 | 65.0 | 66.5 |
| | Decomp-RRF | 86.5 | 89.1 | 88.8 | 84.6 | 83.2 | 79.0 | 73.3 | 67.5 | 62.1 | 68.2 |
| **Movies** | Orig. Base | 55.0 | 75.3 | 88.0 | 88.5 | 87.1 | 81.8 | 76.9 | 69.8 | 59.5 | 45.5 |
| | Pure Rerank | 65.5 | 86.0 | 93.1 | 93.5 | 91.0 | 87.4 | 82.9 | 75.2 | 66.0 | 53.7 |
| | Decomp-Sum | 65.5 | 86.0 | 93.1 | 93.7 | 91.0 | 88.6 | 84.5 | 79.9 | 74.4 | 63.8 |
| | Decomp-RRF | 65.5 | 86.0 | 93.1 | 90.2 | 88.8 | 86.2 | 80.9 | 75.9 | 70.8 | 59.9 |
| **People** | Orig. Base | 80.7 | 87.8 | 86.9 | 85.8 | 81.5 | 77.2 | 71.5 | 64.2 | 55.1 | 43.9 |
| | Pure Rerank | 92.5 | 95.7 | 93.5 | 91.4 | 88.0 | 83.3 | 77.0 | 70.4 | 60.1 | 50.2 |
| | Decomp-Sum | 92.5 | 95.7 | 93.5 | 92.1 | 88.1 | 85.7 | 81.1 | 75.9 | 72.0 | 70.9 |
| | Decomp-RRF | 92.5 | 95.7 | 93.5 | 89.7 | 87.3 | 84.1 | 79.7 | 74.2 | 70.0 | 71.9 |
| **Medical Cases** | Orig. Base | 61.6 | 78.6 | 83.7 | 84.3 | 81.4 | 77.9 | 72.3 | 66.9 | 59.6 | 43.8 |
| | Pure Rerank | 74.3 | 91.0 | 94.8 | 92.6 | 88.0 | 82.9 | 76.3 | 67.1 | 58.3 | 46.6 |
| | Decomp-Sum | 74.3 | 91.0 | 94.8 | 94.3 | 91.0 | 88.9 | 83.9 | 79.7 | 73.5 | 69.7 |
| | Decomp-RRF | 74.3 | 91.0 | 94.8 | 91.8 | 89.1 | 86.4 | 80.1 | 75.6 | 68.4 | 67.7 |
| **Legal Documents** | Orig. Base | 76.2 | 85.6 | 87.2 | 86.7 | 83.2 | 78.9 | 73.3 | 65.2 | 55.6 | 41.4 |
| | Pure Rerank | 87.5 | 92.1 | 90.3 | 89.0 | 85.1 | 79.2 | 74.3 | 65.6 | 58.7 | 48.3 |
| | Decomp-Sum | 87.5 | 92.1 | 90.3 | 90.4 | 87.2 | 81.7 | 80.2 | 73.2 | 65.3 | 61.9 |
| | Decomp-RRF | 87.5 | 92.1 | 90.3 | 87.8 | 85.2 | 80.1 | 76.6 | 69.6 | 64.2 | 63.0 |

Table 14: Detailed reranking results for each dataset using **Qwen3-Reranker-8B**. Values are scaled by 100.

### B.3 Detailed Recall Failure Rate In Retrieval

Figure 2 reports the average recall failure rate of sub-queries in retrieval process across the five datasets. For completeness, we provide the detailed results for each dataset (Books, Movies, People, Medical Cases, and Legal Documents) in the Appendix.

Tables 15 and Table 16 present the detailed results for each dataset in MultiConIR.

| Dataset | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 |
|---|---|---|---|---|---|---|---|---|---|---|
| **Books** | 7.9 | 4.2 | 2.5 | 10.3 | 7.3 | 7.7 | 11.3 | 9.7 | 11.1 | 16.1 |
| **People** | 3.6 | 0.2 | 0.2 | 1.3 | 0.6 | 0.7 | 1.7 | 1.1 | 1.3 | 1.9 |
| **Movies** | 23.0 | 8.8 | 2.2 | 10.6 | 6.0 | 3.2 | 10.6 | 7.5 | 4.7 | 11.7 |
| **Medical Cases** | 12.5 | 3.3 | 0.4 | 7.7 | 6.0 | 4.0 | 10.6 | 7.9 | 8.2 | 12.1 |
| **Legal Documents** | 4.5 | 1.2 | 1.6 | 8.9 | 3.5 | 3.3 | 9.0 | 7.6 | 9.4 | 14.3 |

Table 15: Probability (%) that decomposed sub-queries in retrieval fail to retrieve any positive document in the top 50.

| Dataset | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 |
|---|---|---|---|---|---|---|---|---|---|---|
| **Books** | 7.9 | 4.2 | 2.5 | 2.5 | 1.1 | 1.3 | 0.6 | 1.1 | 1.5 | 1.1 |
| **People** | 3.6 | 0.2 | 0.2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| **Movies** | 23.0 | 8.8 | 2.2 | 0.8 | 0.2 | 0.2 | 0.0 | 0.2 | 0.0 | 0.0 |
| **Medical Cases** | 12.5 | 3.3 | 0.4 | 0.2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| **Legal Documents** | 4.5 | 1.2 | 1.6 | 0.0 | 0.0 | 0.3 | 0.0 | 0.3 | 0.0 | 0.0 |

Table 16: Probability (%) that baseline retrieval fails to retrieve any positive document in the top 50.

### B.4 Detailed Win Rate In Retrieval

Figure 2 reports the average win rate in retrieval process across the five datasets. For completeness, we provide the detailed results for each dataset (Books, Movies, People, Medical Cases, and Legal Documents) in the Appendix.

Tables 17 present the detailed results for each dataset in MultiConIR.

| Dataset | Setting | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Books** | Baseline | 74.7 | 68.3 | 64.9 | 59.2 | 57.3 | 56.4 | 55.2 | 51.2 | 53.2 | 51.1 |
| | Decomp | 74.7 | 68.3 | 64.9 | 57.5 | 56.7 | 53.2 | 54.8 | 51.6 | 53.0 | 48.1 |
| **Movies** | Baseline | 84.0 | 84.6 | 79.0 | 74.5 | 72.2 | 69.3 | 67.6 | 67.5 | 63.6 | 68.1 |
| | Decomp | 84.0 | 84.6 | 79.0 | 68.5 | 69.1 | 65.4 | 63.3 | 59.4 | 65.3 | 64.2 |
| **People** | Baseline | 74.1 | 71.4 | 69.3 | 68.3 | 65.4 | 67.0 | 63.3 | 64.6 | 59.0 | 66.3 |
| | Decomp | 74.1 | 71.4 | 69.3 | 66.8 | 65.4 | 66.5 | 62.0 | 60.4 | 59.8 | 58.8 |
| **Medical Cases** | Baseline | 84.6 | 77.2 | 74.3 | 70.2 | 65.2 | 67.3 | 67.1 | 66.5 | 66.4 | 56.8 |
| | Decomp | 84.6 | 77.2 | 74.3 | 70.9 | 65.6 | 66.6 | 61.6 | 63.7 | 62.3 | 61.5 |
| **Legal Documents** | Baseline | 86.2 | 75.1 | 73.9 | 67.9 | 63.2 | 62.8 | 63.4 | 58.0 | 53.0 | 49.0 |
| | Decomp | 86.2 | 75.1 | 73.9 | 64.0 | 57.3 | 63.0 | 56.7 | 51.0 | 53.3 | 51.0 |

Table 17: Win Rate (%) between Baseline and Decomposition in retrieval methods across different datasets.

### B.5 Detailed Win Rate In Reranking

Figure 3 reports the average win rate in reranking process across the five datasets. For completeness, we provide the detailed results for each dataset (Books, Movies, People, Medical Cases, and Legal Documents) in the Appendix.

Tables 18 present the detailed results for each dataset in MultiConIR.

| Dataset | Setting | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | **Win Rate** | | | | | |
| **Books** | Orig. Base | 73.6 | 68.3 | 65.9 | 58.8 | 57.1 | 55.8 | 55.1 | 51.3 | 52.8 | 51.4 |
| | Pure Rerank | 93.2 | 93.9 | 88.8 | 86.2 | 84.5 | 83.9 | 82.2 | 84.5 | 87.3 | 89.7 |
| | Decomp-Sum | 93.2 | 93.9 | 88.8 | 88.0 | 86.0 | 85.0 | 83.7 | 88.2 | 90.4 | 90.5 |
| | Decomp-RRF | 93.2 | 93.9 | 88.8 | 87.1 | 86.2 | 85.0 | 81.1 | 87.8 | 88.4 | 91.0 |
| **Movies** | Orig. Base | 82.0 | 83.9 | 78.6 | 74.7 | 72.3 | 69.3 | 67.7 | 67.4 | 63.6 | 68.1 |
| | Pure Rerank | 89.9 | 90.6 | 88.7 | 89.2 | 88.1 | 85.8 | 85.0 | 81.3 | 79.8 | 76.0 |
| | Decomp-Sum | 89.9 | 90.6 | 88.7 | 87.1 | 84.6 | 85.6 | 83.1 | 86.2 | 89.2 | 78.5 |
| | Decomp-RRF | 89.9 | 90.6 | 88.7 | 84.0 | 83.0 | 84.0 | 79.7 | 81.9 | 82.5 | 74.6 |
| **People** | Orig. Base | 74.4 | 71.1 | 69.2 | 68.5 | 65.4 | 67.0 | 63.3 | 64.6 | 59.0 | 66.3 |
| | Pure Rerank | 92.4 | 93.9 | 87.7 | 88.5 | 85.2 | 81.3 | 76.2 | 71.1 | 69.7 | 70.6 |
| | Decomp-Sum | 92.4 | 93.9 | 87.7 | 91.0 | 85.4 | 89.4 | 82.4 | 85.5 | 88.0 | 89.5 |
| | Decomp-RRF | 92.4 | 93.9 | 87.7 | 90.5 | 88.6 | 89.4 | 87.6 | 84.0 | 86.0 | 87.4 |
| **Medical Cases** | Orig. Base | 84.0 | 77.5 | 74.3 | 70.2 | 65.2 | 67.3 | 67.1 | 66.5 | 66.4 | 56.8 |
| | Pure Rerank | 87.6 | 92.3 | 91.3 | 88.8 | 82.3 | 79.0 | 75.9 | 68.9 | 69.5 | 68.5 |
| | Decomp-Sum | 87.6 | 92.3 | 91.3 | 92.5 | 89.6 | 90.5 | 85.3 | 88.4 | 90.2 | 88.3 |
| | Decomp-RRF | 87.6 | 92.3 | 91.3 | 92.8 | 89.1 | 89.4 | 84.6 | 86.3 | 86.6 | 87.1 |
| **Legal Documents** | Orig. Base | 84.6 | 75.0 | 73.6 | 67.7 | 63.3 | 62.7 | 63.7 | 59.4 | 56.3 | 55.1 |
| | Pure Rerank | 88.4 | 83.4 | 81.4 | 80.6 | 76.0 | 74.6 | 74.4 | 70.3 | 67.8 | 68.9 |
| | Decomp-Sum | 88.4 | 83.4 | 81.4 | 83.3 | 81.4 | 81.7 | 82.1 | 82.5 | 82.0 | 85.4 |
| | Decomp-RRF | 88.4 | 83.4 | 81.4 | 81.4 | 78.5 | 77.8 | 77.6 | 78.6 | 79.8 | 81.3 |

Table 18: Detailed win rate (%) results after reranking for each dataset.

### B.6 Detailed Positive Document Rank In Reranking

Figure 3 reports the average positive document rank in reranking process across the five datasets. For completeness, we provide the detailed results for each dataset (Books, Movies, People, Medical Cases, and Legal Documents) in the Appendix.

Tables 19 present the detailed results for each dataset in MultiConIR.

| Dataset | Setting | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Positive Document Average Position** | | | | | | | | | | | |
| **Books** | Orig. Base | 8.46 | 8.19 | 7.48 | 7.03 | 6.70 | 6.62 | 6.24 | 6.37 | 6.20 | 6.58 |
| | Pure Rerank | 5.72 | 5.31 | 5.20 | 5.13 | 5.12 | 5.13 | 5.06 | 5.06 | 4.67 | 3.10 |
| | Decomp-Sum | 5.72 | 5.31 | 5.20 | 4.89 | 4.91 | 4.75 | 4.60 | 4.54 | 4.15 | 2.50 |
| | Decomp-RRF | 5.72 | 5.31 | 5.20 | 5.22 | 5.14 | 5.08 | 5.13 | 5.06 | 4.68 | 2.48 |
| **Movies** | Orig. Base | 11.36 | 8.02 | 5.96 | 5.29 | 4.85 | 4.76 | 4.66 | 4.48 | 4.49 | 4.28 |
| | Pure Rerank | 6.13 | 5.19 | 4.88 | 4.57 | 4.37 | 4.15 | 3.95 | 3.90 | 3.93 | 3.98 |
| | Decomp-Sum | 6.13 | 5.19 | 4.88 | 4.57 | 4.39 | 4.03 | 3.78 | 3.51 | 3.05 | 2.61 |
| | Decomp-RRF | 6.13 | 5.19 | 4.88 | 5.05 | 4.78 | 4.52 | 4.49 | 4.22 | 3.82 | 3.18 |
| **People** | Orig. Base | 8.01 | 6.62 | 6.07 | 5.49 | 5.37 | 5.29 | 5.23 | 5.13 | 5.11 | 4.81 |
| | Pure Rerank | 5.51 | 5.26 | 5.12 | 4.95 | 4.82 | 4.75 | 4.75 | 4.70 | 4.80 | 4.64 |
| | Decomp-Sum | 5.51 | 5.26 | 5.12 | 4.85 | 4.78 | 4.49 | 4.31 | 4.13 | 3.66 | 2.22 |
| | Decomp-RRF | 5.51 | 5.26 | 5.12 | 5.14 | 4.96 | 4.77 | 4.70 | 4.52 | 4.08 | 2.31 |
| **Medical Cases** | Orig. Base | 11.23 | 8.00 | 6.71 | 5.66 | 5.39 | 5.13 | 4.96 | 4.78 | 4.42 | 4.68 |
| | Pure Rerank | 6.58 | 5.36 | 4.96 | 4.78 | 4.74 | 4.65 | 4.69 | 4.83 | 4.81 | 5.01 |
| | Decomp-Sum | 6.58 | 5.36 | 4.96 | 4.60 | 4.41 | 4.07 | 3.84 | 3.53 | 3.19 | 2.29 |
| | Decomp-RRF | 6.58 | 5.36 | 4.96 | 4.99 | 4.78 | 4.52 | 4.57 | 4.30 | 4.08 | 2.52 |
| **Legal Documents** | Orig. Base | 9.05 | 7.02 | 6.17 | 5.69 | 5.38 | 5.19 | 4.95 | 5.03 | 4.97 | 5.12 |
| | Pure Rerank | 5.99 | 5.64 | 5.31 | 5.17 | 5.13 | 5.06 | 4.98 | 5.01 | 5.01 | 4.98 |
| | Decomp-Sum | 5.99 | 5.64 | 5.31 | 4.99 | 4.94 | 4.76 | 4.37 | 4.19 | 4.04 | 2.90 |
| | Decomp-RRF | 5.99 | 5.64 | 5.31 | 5.34 | 5.23 | 5.05 | 4.89 | 4.80 | 4.50 | 3.23 |

Table 19: Detailed positive document position results after reranking for each dataset.

## C Additional Details on Reranking Setup

### C.1 Why We Use Pointwise Reranking Instead of Listwise Reranking

Pointwise and listwise reranking are two standard paradigms for reranking. While listwise reranking has shown strong performance in many ranking benchmarks, applying it to large candidate pools is computationally expensive due to the long-context interaction among multiple documents. In addition, long-context listwise reranking is more susceptible to positional biases such as the "lost in the middle" effect (Liu et al., 2024), where documents located in the middle of the context receive less effective attention.

In our setting, query decomposition introduces multiple sub-queries for each original query. Combining listwise reranking with decomposition would substantially increase context length and computational overhead, since all candidate documents and multiple sub-queries would need to be jointly processed. To avoid these issues, we adopt a pointwise reranking strategy, where each candidate document is independently scored against each decomposed sub-query.

### C.2 Preliminary Token-Based Reranking Exploration

In our preliminary experiments, we explored a zero-shot **token-based reranking** approach using Qwen3-0.6B. In this setup, the model was prompted to directly generate a discrete numerical relevance score (e.g., 0–10) for each query-document pair.

However, we found that this generative scoring strategy was unstable and consistently underperformed the original retrieval baseline. We attribute this degradation to two main factors:

- **Generation instability:** autoregressive generation introduces decoding variance and formatting inconsistency for numerical outputs.
- **Weak score calibration:** discrete generated scores are not well calibrated across different query-document pairs, making score aggregation less reliable.

To address these limitations, we transitioned to a **logit-based reranking** formulation using the Qwen3-Reranker series (0.6B, 4B, and 8B). Specifically, we formulate reranking as a binary classification task. Instead of relying on generated text outputs, we directly extract the logits corresponding to the "Yes" and "No" tokens from the final prediction layer. These logits are normalized using a softmax function to produce a continuous relevance score between 0 and 1:

$$P(\text{Yes}|q,d) = \frac{\exp(z_{\text{Yes}})}{\exp(z_{\text{Yes}}) + \exp(z_{\text{No}})} \tag{5}$$

where $z_{\text{Yes}}$ and $z_{\text{No}}$ denote the logits of the corresponding tokens.

Table 20 compares the performance of the token-based and logit-based scoring strategies on the Books dataset. The results show that the logit-based formulation consistently outperforms both the token-based approach and the original retrieval baseline.

| Query | Baseline | Token-Based | Logit-Based |
|---|---|---|---|
| Query 1 | 0.753 | 0.696 | 0.852 |
| Query 2 | 0.761 | 0.722 | 0.880 |
| Query 3 | 0.778 | 0.752 | 0.867 |
| Query 4 | 0.758 | 0.750 | 0.821 |
| Query 5 | 0.745 | 0.735 | 0.798 |
| Query 6 | 0.699 | 0.682 | 0.737 |
| Query 7 | 0.661 | 0.645 | 0.680 |
| Query 8 | 0.581 | 0.564 | 0.596 |
| Query 9 | 0.494 | 0.463 | 0.514 |
| Query 10 | 0.367 | 0.345 | 0.438 |

Table 20: Performance comparison (NDCG@10) of different scoring mechanisms on the Books dataset. The logit-based approach consistently resolves the degradation caused by token-based scoring and significantly surpasses the initial retrieval baseline.

### C.3 Implementation Details of Logit-Based Reranking

In our logit-based reranking phase (using the Qwen3-Reranker series), the model evaluates the relevance of a document to a given query using a strictly structured instructional prompt. The exact input is concatenated as follows:

```
<Instruct>: {instruction}
<Query>: {query}
<Document>: {doc}
```

Instead of decoding the full generated sequence, we directly extract the logits corresponding to the positive and negative label tokens at the final prediction layer. In our implementation, these are represented by `token_true_id` (for "Yes") and `token_false_id` (for "No"). The final relevance score is computed by applying a log-softmax formulation over these two specific logits:

$$\text{score} = \frac{\exp(L_{\text{true}})}{\exp(L_{\text{true}}) + \exp(L_{\text{false}})} \tag{6}$$

where $L$ represents the raw logit value for the respective token. This methodology ensures the model leverages its internal probability distribution, providing a well-calibrated, continuous probability score that serves as a highly deterministic signal for our End-to-End Pipeline.

## D Semantic Dilution in Decomposed Sub-queries

We present an example from the Books dataset to illustrate how decomposition weakens semantic constraints and leads to easy negative retrieval. The original query specifies 8 conditions and is partitioned into sub-queries. We examine one such sub-query, which retains 3 constraints. Under this sub-query, the dense retriever assigns higher similarity score to an easy negative (0.650) than to the true positive (0.576).

**Case Study: Semantic Dilution Example**

**# Original Query (8 conditions)**
Find a book that fits: 1. Authored by Steve Chandler, 2. Helps with personal change, 3. Contains humor and reassurance, 4. Recommended by peers, 5. Offers an action plan, 6. Includes feedback-based methods, 7. Uses coaching techniques, 8. Banishes negative thoughts.

**# Decomposed Sub-query (3 conditions retained)**
Find a book that satisfies these conditions: Contains humor and reassurance. Recommended by peers. Offers an action plan.

**# Positive Document (similarity: 0.576)**
" 'If you take the best of Wayne Dyer and add it to the best of Anthony Robbins, what you would have would only be half as good as Steve Chandler.'—Dale Dauten, Chicago Tribune 'Some books that can help you awaken and begin to change are ones by Steve Chandler, who, I am reading lately. Great stuff. I'm becoming a fan of Steve Chandler.'—Joe Vitale, best-selling author of The Attraction Factor and contributor to The Secret 'Steve Chandler lights you up with the glow of his internal neon. **[He is] one of the funniest men I've ever known—what he proposes is so rock solid and reassuring.**'—Lisa Schnebly, The Arizona Republic '100 Ways to Motivate Yourself is wonderful, inspirational, honest, and courageous. It speaks from every page. **It is definitely a book I will recommend to my clients and friends.**'—Devers Branden, coauthor of What Love Asks of Us With the third refreshed edition of 100 Ways to Motivate Yourself, **Steve Chandler helps you create an action plan for living your vision**, in business and in life. It features 100 proven methods to positively change the way you think and act—methods based on feedback from the hundreds of thousands of corporate and public seminar attendees Chandler speaks to each year. The book now also includes techniques and breakthroughs he has created for individual coaching clients. 100 Ways to Motivate Yourself will help you break through the negative barriers and banish the pessimistic thoughts that are preventing you from fulfilling your lifelong goals and dreams. This edition also contains new mental and spiritual techniques that give readers more immediate access to action and results in their lives. Steve Chandler invites you to maintain your current mindset and stick to conventional methods, leading to minimal transformation and delaying your progress toward your objectives."

**# Easy Negative Document (similarity: 0.650)**
"You can wait for the right moment before setting out on your dreams. You can learn to successfully deal with any difficult times or negative people you may encounter. You can unlock enthusiasm, determination, and energy using the keys of an optimistic point of view: releasing the past, keeping the future in mind, and taking everything one day at a time. Day by day, this book addresses a different topic relating to issues you may face as you work to improve your mental outlook. Beginning with a 'game plan' and mapping out where you want to go, you'll gain insight about working through obstacles and measuring your progress. You'll also be reminded about the value of appreciating your present blessings and looking forward to a hopeful future. You'll find practical advice as well as writings specifically chosen to inspire and motivate you on your journey. The empowering message of this book is also a simple one: You have the power in your hands today to be happier, more creative, and more at peace with yourself and everyone around you. Take this week out of your life and begin to build a positive attitude that will reward you in wonderful ways and make a lasting difference in this world. Category: Self-help, Personal Growth, General"

## E Prompt Template for GPT-5.3 in SSRB Generalization Analysis

In this appendix, we detail the exact prompt instructions provided to GPT-5.3 to perform fixed-size query decomposition on the SSRB dataset. The prompt is designed to strictly enforce structural boundaries while preserving the semantic integrity of the original multi-condition queries.

**System Prompt for Query Decomposition**

**# Role & Task**
You are an expert in complex, multi-condition information retrieval. Your task is to decompose a given complex search query into distinct, independent sub-queries to optimize downstream retrieval performance.

**# Strict Rules for Decomposition**

1. **Decomposition Limit:** Split the original query into sub-queries. If the original query only contains 2-3 conditions, output the original query as a single sub-query without splitting.
2. **Condition Threshold:** Each generated sub-query MUST contain 2-3 distinct search conditions or constraints.
3. **No Information Overlap:** The semantic conditions must be mutually exclusive across sub-queries. Do not duplicate specific constraints or information across the decomposed segments.
4. **Preserve Original Wording:** Extract and segment the text while strictly preserving the original phrasing, vocabulary, and sentence structure as much as possible. Avoid aggressively paraphrasing, rewriting, or hallucinating new information. Ensure each segment remains a coherent sentence.

**# Output Format**
You must strictly output the decomposed sub-queries in the following format without any additional conversational text:
`[Original_ID]_subq_[Index], "[Sub-query text]"`

**# Example**
**Input:**
`product_search-camera_search-q-1347, "I'm planning a hiking trip and need a durable, high-quality camera that can handle outdoor conditions. I'm looking for a camera with a resolution of at least 20 megapixels and one that's known for its excellent image quality, something with a really good reputation amongst photography enthusiasts."`

**Output:**
`product_search-camera_search-q-1347_subq_1, "I'm planning a hiking trip and need a durable, high-quality camera that can handle outdoor conditions. I'm looking for a camera with a resolution of at least 20 megapixels"`
`product_search-camera_search-q-1347_subq_2, "I'm looking for a high quality camera and one that's known for its excellent image quality, something with a really good reputation amongst photography enthusiasts."`

## F Sensitivity to the RRF Hyperparameter

RRF introduces a smoothing constant $k$ that controls the contribution of rank positions during score aggregation. To examine whether RRF is sensitive to this hyperparameter, we vary $k$ from 10 to 120 and report the resulting performance in Table 21.

| $k$ | **NDCG@10** | **Recall@10** | **Recall@50** |
|---|---|---|---|
| 10 | 0.5935 | 0.7706 | 0.9407 |
| 20 | 0.5995 | 0.7753 | 0.9396 |
| 40 | 0.6007 | 0.7757 | 0.9390 |
| 60 | 0.5994 | 0.7737 | 0.9387 |
| 80 | 0.5983 | 0.7718 | 0.9376 |
| 100 | 0.5974 | 0.7703 | 0.9361 |
| 120 | 0.5968 | 0.7692 | 0.9348 |

Table 21: Sensitivity analysis of the RRF smoothing parameter $k$.

As shown in Table 21, performance remains relatively stable across a broad range of $k$ values. NDCG@10 varies only from 0.5935 to 0.6007, while Recall@10 and Recall@50 also show only minor fluctuations. These results indicate that RRF is not highly sensitive to the choice of $k$ in our setting.