MultiConIR: Towards Multi-Conditional Information Retrieval

Anonymous ACL submission

Abstract

In this paper, we introduce MultiConIR, the first benchmark specifically designed to evaluate retrieval models in multi-condition scenarios. Unlike existing datasets that primarily focus on single-condition queries from search engines, MultiConIR captures real-world complexity by incorporating five diverse domains: books, movies, people, medical cases, and legal documents. We propose three tasks to systematically assess retrieval and reranking models on multi-condition robustness, monotonic relevance ranking, and query format sensitivity. Our findings reveal that existing retrieval and reranking models struggle with multi-condition retrieval, with rerankers suffering severe performance degradation as query complexity increases. We further investigate the performance gap between retrieval and reranking models, exploring potential reasons for these discrepancies. Finally, we highlight the strengths of GritLM and Nv-Embed, which demonstrate enhanced adaptability to multi-condition queries, offering insights for future retrieval models.

1 Introduction

Information retrieval (IR) systems are critical for helping users find relevant information across various domains. Traditionally, these systems match queries to documents using lexical similarity (Carpineto and Romano, 2012; Ponte and Croft, 2017). Recent advances in Dense Retrieval have significantly improved retrieval effectiveness by encoding queries and documents into embeddings, capturing deeper semantic relationships (Karpukhin et al., 2020; Zhan et al., 2021). These retrieval systems are highly effective when the relationship between the query and document is simple and direct (Su et al., 2024). However, as user needs become more complex—especially in real-world scenarios—retrieval systems increasingly struggle to capture the full range of user intent (Zhu et al., 2023; Su et al., 2024).

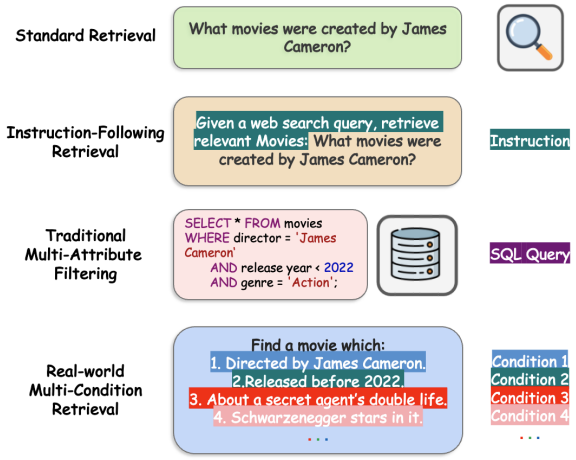


Figure 1: From single-condition to multi-conditional retrieval. Standard retrieval relies on single-condition queries, while instruction-following retrieval enhances queries with explicit instructions. Traditional SQL-based filtering handles predefined attributes, whereas real-world multi-condition retrieval involves queries specifying multiple conditions, which can be attribute-level or semantic-level.

A significant challenge in information retrieval arises with multi-conditional queries, where users specify multiple requirements simultaneously, as illustrated in Fig 1. Whether searching for a movie with specific attributes or selecting a product that meets various criteria, multi-conditional search has become an integral part of modern information-seeking behavior. Traditional IR systems handle such scenarios using structured filtering mechanisms, such as SQL-based queries that retrieve information from backend databases based on predefined conditions (e.g., release year, genre). However, this approach is inherently rigid and limited, as it relies on explicitly defined attributes and lacks the flexibility to accommodate evolving or diverse user preferences. As a result, it struggles to support nuanced or semantic-level queries that go beyond structured data filtering.

The advent of Large Language Models (LLMs) has enhanced IR by introducing instruction-

following capabilities (Asai et al., 2023; Weller et al., 2024a; Oh et al., 2024). This approach augments standard queries with explicit instructions, which serve as additional constraints to refine search results, as shown in Fig.1. Despite these advancements, existing evaluation benchmarks remain predominantly focused on single-condition queries and binary relevance assessments—classifying documents as either relevant or irrelevant (Nguyen et al., 2016; Kwiatkowski et al., 2019; Muennighoff et al., 2022)—thus overlooking the nuanced challenges of multi-conditional queries, where relevance depends on the degree to which multiple conditions are satisfied.

An ideal multi-conditional retrieval system should exhibit the following properties: (1) **Complexity Robustness:** The system should maintain high performance regardless of query complexity (i.e., the number of conditions specified); (2) **Relevance Monotonicity:** The relevance scores should scale monotonically with the number of matched conditions; for example, a document matching all n conditions should be ranked higher than one matching $n-1$; (3) **Format Invariance:** The system should yield consistent results regardless of the query format, whether presented as a structured list or as free-form natural language.

Existing benchmarks do not offer a structured framework for evaluating multi-conditional retrieval along these dimensions. To address this gap, we introduce **MultiCondIR**—the first benchmark designed to comprehensively evaluate multi-conditional retrieval systems. Through systematic experiments on 12 state-of-the-art models (including dense retrievers, cross-encoders, and LLM-based agents), we uncover several critical insights:

- **Multi-Condition Struggle:** Retrieval and reranking models degrade as the number of query conditions increases, struggling to distinguish positives from hard negatives.
- **Monotonicity Failure:** Models fail to maintain a consistent ranking hierarchy, with performance declining when differentiating HN_n from HN_{n-1}.
- **Reranker Breakdown:** While effective in single-condition tasks, rerankers struggle significantly in multi-condition retrieval and are more sensitive to query format variations.

By quantifying these gaps, our work provides actionable guidelines for developing IR systems that truly understand multi-conditional intent, laying the groundwork for advancing IR toward human-like reasoning in complex search scenarios.

2 Related Works

Retriever: From Sparse To Dense Traditional sparse retrieval methods are based on BM25 (Robertson and Zaragoza, 2009), TF-IDF (Ramos et al., 2003), etc., rely on keyword matching and statistical weighting to evaluate relevance, which suffers from the well-known issue of lexical gap (Berger et al., 2000), restricting their ability to effectively capture semantic relationships (Luan et al., 2021; Nian et al., 2024).

Dense retrieval addresses this limitation by encoding both queries and documents as embeddings within a joint latent space, where the semantic relationship is captured through the similarity scores between their embeddings (Li et al., 2023a). Pre-trained language models like BERT (Devlin et al., 2019) and T5 (Raffel et al., 2020), are widely used as backbone encoders for dense retrieval (Li et al., 2023b; Sturua et al., 2024; Xiao et al., 2023).

Recent advancements have shown that LLMs offer significant potential as backbone encoders for dense retrieval (Wang et al., 2024a; Weller et al., 2024c; BehnamGhader et al., 2024). For instance, Repllama (Ma et al., 2023) enhanced retrieval performance by fine-tuning Llama-2 to serve as dense retrievers. GritLM (Muennighoff et al., 2024) unified text embedding and generation within a single LLM. LLM2Vec (BehnamGhader et al., 2024) introduced an unsupervised approach for transforming decoder-only LLMs into dense retrievers.

Benchmarks In Complex Retrieval Tasks Existing datasets for information retrieval, such as MS MARCO (Nguyen et al., 2016), Natural Questions (Kwiatkowski et al., 2019), and MTEB (Muennighoff et al., 2022), primarily focus on queries sourced from search engines. The relationships between queries and documents are typically simple and direct (Su et al., 2024).

Recent studies have expanded retrieval benchmarks to address more complex scenarios. Instruction-based datasets (Weller et al., 2024a; Qin et al., 2024; Oh et al., 2024), for instance, evaluate the instruction-following capabilities of retrieval models by embedding explicit instructions within queries to better represent users' retrieval intents. Furthermore, some works have assessed retrieval models' abilities to handle logical rea-

soning tasks, including Boolean logic (Mai et al., 2024; Zhang et al., 2024c), negation (Zhang et al., 2024a; Weller et al., 2024b), and multi-hop reasoning (Su et al., 2024). These efforts mark significant progress in increasing query complexity. However, while research in the generative modeling domain has explored the ability of LLMs to handle multi-constraint instructions (He et al., 2024; Ferraz et al., 2024; Zhang et al., 2024b), studies on retrieval models in multi-condition scenarios remain sparse.

3 MultiConIR

We introduce **MultiConIR**, a benchmark designed to evaluate the capacity of retrieval models to process multi-conditional queries. Formally, given a query q_k composed of k conditions $C = \{c_1, c_2, \ldots, c_k\}$ with $k \in \{1, \ldots, 10\}$, we construct a structured retrieval setup consisting of:

(1) **Two query formulations**, denoted as q_k^{inst} and q_k^{desc}, where q_k^{inst} corresponds to a structured instruction-style query, formally expressed as a tuple $q_k^{\text{inst}} = \langle f, C \rangle$ where f is an explicit function describing retrieval constraints, and q_k^{desc} is a natural language descriptive-style query, represented as an unstructured sequence about the same set C,

(2) A **positive document** d^+ that satisfies all k conditions, i.e., $d^+ \models C$,

(3) A sequence of **hard negative documents** $\{d_0, d_1, \ldots, d_{k-1}\}$, where each d_j satisfies exactly j out of k conditions, formally expressed as $d_j \models \{c_1, \ldots, c_j\}$ and $d_j \not\models \{c_{j+1}, \ldots, c_k\}$.

This controlled design enables a principled evaluation of multi-conditional retrieval along three fundamental axes: (1) **Complexity Robustness:** The model's retrieval effectiveness as k increases, measured by its ability to distinguish d^+ from d_{k-1}; (2) **Relevance Monotonicity:** The extent to which the retrieval model enforces a strict ordering such that $S(q_k, d_j) > S(q_k, d_{j+1})$ for all j, ensuring that documents satisfying more conditions are ranked higher; and (3) **Format Invariance:** The stability of retrieval performance under transformations of query representation, quantified by discrepancies in ranking outcomes across query formats.

3.1 Domain Selection

MultiConIR was constructed from real-world datasets from five diverse domains: books, movies, people, medical cases, and legal documents. These domains were carefully chosen for their practical significance and their intrinsic requirement for multi-condition retrieval. **Books & Movies:** Queries in these domains often blend attribute-based (e.g., *a sci-fi movie*) with narrative-based (e.g., *a story about time travel*), requiring semantic understanding beyond keyword matching. **People:** Queries are often based on vague recollections or specific but incomplete information about a person (e.g., *a Nobel laureate in Physics who studied black holes*) **Medical Case & Legal Document:** These domains require fine-grained condition matching, where case-based reasoning is essential (e.g., *a patient with breathing issues and antibiotic allergies*), demanding models understand condition dependencies, not just match isolated terms.

3.2 Dataset Construction Pipeline

To construct MultiConIR, we design a multi-step data generation framework as shown in Fig. 2, this pipeline is highly adaptable across multiple domains, enabling the generation of queries and hard negative (HN) documents that progressively satisfy 1 to 10 conditions. To preserve dataset integrity and mitigate the generalization issues associated with fully synthetic datasets (Li et al., 2023c; Wang et al., 2024b), we employ LLM-based generation (GPT-4o) exclusively for modifying sentences within hard negatives, rather than altering entire documents.[1] The data creation process consists of the following steps, with detailed prompt templates provided in Appendix B:

Step 1: Condition Sentence Extraction. To capture the fine-grained constraints for multi-condition retrieval, we prompt GPT-4o to extract ten key sentences from each document, where each sentence represents a distinct and semantically complete condition. To ensure robustness, extracted sentences must be contextually relevant, non-redundant, and representative of the document's core information.

Step 2: Query Generation. Given the extracted condition sentences, we prompt GPT-4o to generate ten hierarchical queries (Query1 to Query10), each incorporating an increasing number of conditions (from 1 to 10). To enhance linguistic diversity, queries are formulated in two distinct styles: (1) Instruction-style (explicitly listing conditions in a structured format). (2) Descriptive-style (embedding conditions naturally within a coherent sentence).

[1] Discussion about the challenges associated with fully LLM-generated datasets can be found in Appendix E.1.

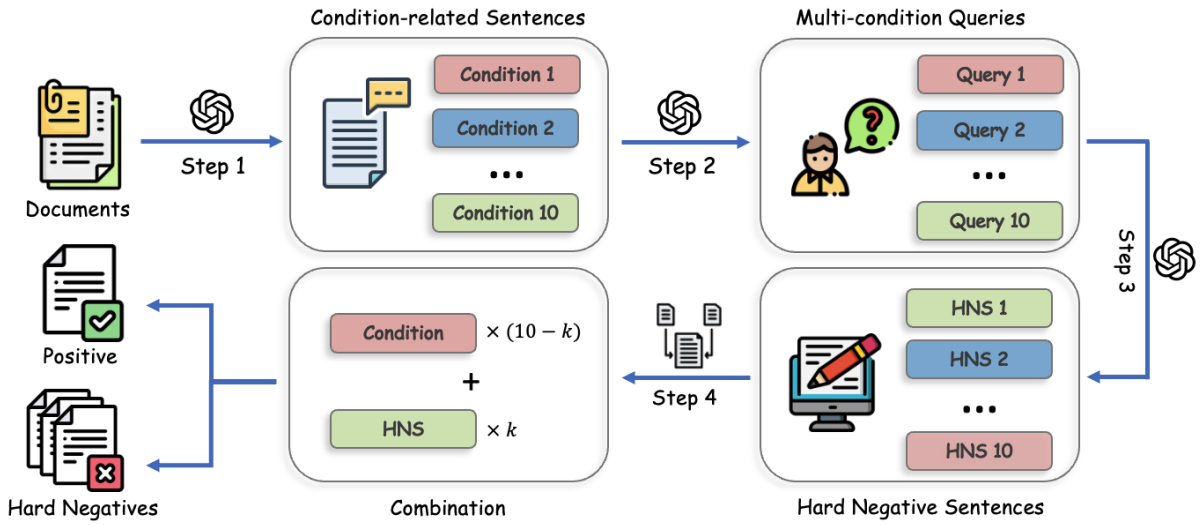


Figure 2: MultiCon Dataset Construction Pipeline.

Step 3: Hard Negative Sentence Construction. To create hard negatives, we prompt GPT-4o to modify extracted condition sentences into Hard Negative Sentences (HNS) that subtly deviate from the original meaning while preserving lexical plausibility. Unlike naive negation, these modifications introduce semantic shifts that retain fluency but fail to satisfy the original retrieval conditions, thereby making retrieval harder. We experimented with two approaches for HNS construction: one modifies key information (applied to books, movies, medical cases, and legal documents), while the other retains keywords but adds dummy information (used for the people dataset). [2]

Step 4: HN Document Generation. To construct multi-condition retrieval datasets, we systematically assemble documents: (1) Positive documents: Composed of all ten original condition sentences, ensuring full query relevance. (2) Hard Negative documents d_1 to d_{10}: Generated by progressively replacing 1 to 10 original condition sentences with their corresponding hard negative versions, creating a continuum of relevance degradation. This structured data generation ensures that each query condition incrementally refines document ranking, allowing for fine-grained retrieval performance assessment.

3.3 Evaluation Metrics

Complexity Robustness Queries range from `query1` to `query10`, each progressively incorporating 1 to 10 conditions. The candidate set comprises a `Positive` document that fully satisfies all conditions and a `HN1` document, which is derived from the positive by modifying a single condition.

Complexity robustness is measured using **Win Rate (WR)** under various k, defined as:

$$\mathrm{WR}_k = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}(S(q_k, d^+) > S(q_k, d_{k-1})),$$

where $S(q_i, d^+)$ and $S(q_i, d^-)$ denote similarity scores for the positive document and hard negative.

Relevance Monotonicity The query is fixed as `query10` (containing all 10 conditions), while the candidate set includes one `positive` and ten hard negatives ($d_0 - d_9$), each containing 0–9 conditions.

We evaluate performance using Win Rate between adjacent hard negatives:

$$\mathrm{WR}_{k,k-1} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\big(S(q_{10}, d_k) > S(q_{10}, d_{k-1})\big), \quad (1)$$

Format invariance. We compare two query formats: (1) Instruction-style, which explicitly lists conditions (e.g., *Find a movie that meets the following conditions: 1. Action genre, 2. Directed by James Cameron*). (2) Descriptive-style, which integrates conditions into a natural query (e.g., *Find an action movie directed by James Cameron*).

To quantify ranking variability between query styles, we define the **Flip Rate (FR)**:

$$\mathrm{FR} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\big(\mathbf{1}(S_{\text{inst}}(q_{10}, d_k) > S_{\text{inst}}(q_{10}, d_{k-1})) \neq \mathbf{1}(S_{\text{desc}}(q_{10}, d_k) > S_{\text{desc}}(q_{10}, d_{k-1}))\big),$$

[2] The differences between these two approaches are discussed in Appendix E.2.

where S_{inst} and S_{desc} denote similarity scores under instruction-style and descriptive-style queries. The indicator function returns 1 if the ranking order of positive and hard negative documents changes between query styles and 0 otherwise. A higher FR indicates greater sensitivity to query formulation.

4 Experiments

We evaluate 13 representative retrieval models from diverse architectures and varying model sizes, including one spare retrieval model: BM25 (Robertson and Zaragoza, 2009); Two BERT-based retrieval models: gte-large-en-v1.5 (Li et al., 2023b) and jina-embeddings-v3 (Sturua et al., 2024); Seven LLM-based retrieval models: NV-Embed-v2 (Lee et al., 2024), bge-en-icl (Li et al., 2024), gte-Qwen2-7B-instruct (Li et al., 2023b), gte-Qwen2-1.5B-instruct (Li et al., 2023b), e5-mistral-7b-instruct (Wang et al., 2024a), GritLM-7B (Muennighoff et al., 2024), LLM2Vec (BehnamGhader et al., 2024); Three reranking models: bge-reranker-v2-m3 (Chen et al., 2024), bge-reranker-v2-gemma (Chen et al., 2024), FollowIR-7B (Weller et al., 2024a). Details of each model are provided in Appendix A.

4.1 Results for Complexity Robustness

Table 1 presents the average Win Rate scores for evaluating complexity robustness across five datasets, with individual dataset performances detailed in the Appendix D.1. To visualize how model performance evolves as the number of query conditions increases, we plot a line graph with Gaussian smoothing (window size = 1), as shown in Fig. 3. The results reveal several notable trends:

Performance decline with increasing query conditions As the number of conditions in the query increases, the performance of both retrieval and reranking models declines. This suggests that with more conditions, models struggle to accurately distinguish between positives and HNs. Among all models, *GritLM-7B* exhibits the lowest performance degradation, declining by only 6.13% from Query1 to Query10. *NV-Embed-v2* follows with a relatively low decline of 12.51%.

Reranking models exhibit steeper performance drop Initially, reranking models outperform retrieval models when queries contain fewer conditions. However, as the number of conditions increases, their performance declines more sharply. As shown in Table 1, at Query1, rerankers almost outperformed all retrieval models. bge-reranker-v2-gemma achieves a Win Rate of 91.07% at Query1 but drops to 56.09% at Query10, a drastic 34.98% decrease. Similarly, bge-reranker-v2-m3 declines from 87.14% to 44.87%, a drop of 42.27%, and followIR shows a drop from 83.41% to 43.52%, with a drop of 39.89%. According to Table 1, the average Win Rate decline for rerankers reaches **39.05%**, while for retrievers, it is **14.06%**.

Sparse retrieval models show performance improvement Unlike dense retrievers and rerankers, BM25's performance improves with query complexity, increasing from 28.59% (Query1) to 39.87% (Query10). This is likely due to its lexical matching mechanism, where more query conditions lead to greater lexical overlap with candidate documents, enhancing retrieval effectiveness. However, BM25's overall win rate remains lower than dense retrievers, and positives rank below HNs (win rate < 0.5), suggesting that hard negatives consistently outrank positives. This may stem from dataset construction, where queries and HNs were generated or paraphrased by GPT-4o, potentially introducing lexical bias. Appendix E further explores challenges in using generated datasets.

4.2 Results for Relevance Monotonicity

Fig. 4 shows the trend of average Win Rate in the multi-condition retrieval setting of Task 2. The complete results, provided in Appendix D.2, evaluate the model's ability to differentiate the relevance hierarchy among documents with varying conditions. The findings reveal key insights:

Relevance monotonicity failure As documents become increasingly hard (i.e., satisfying more conditions in the query), retrieval and reranking models struggle to distinguish between d_k and d_{k-1}, leading to a decline in Win Rate performance. This failure emphasizes the challenge of preserving relevance monotonicity in multi-conditional retrieval settings and highlights a gap in current model capabilities when handling complex queries.

Sensitive to " exact matches" and " complete mismatches" We observe a slight upward trend at the end of win rate curves for most dense retrievers, likely due to their contrastive learning-based training. Dense retrievers aim to pull query-positive pairs closer while pushing negatives further apart, which may weaken their ability to dis-

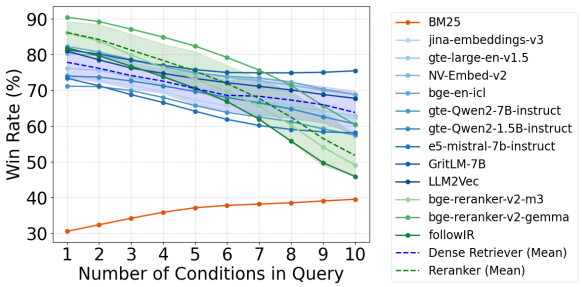


Figure 3: Effect of Condition Quantity on Average Win Rate (reflecting complexity robustness). Win Rate measures the success rate of Positive over Hard Negatives as query conditions increase. Blue lines represent Dense Retrievers; Green lines represent Rerankers; Red line represents BM25.

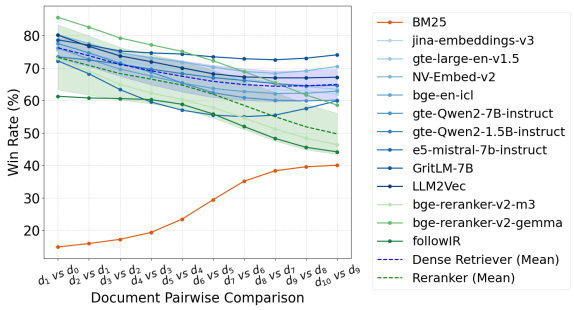


Figure 4: Relevance Monotonicity Distinction. Win Rate reflects the success rate between documents satisfying different numbers of conditions under a multi-condition query (query10), i.e., d_k vs. d_{k-1}.

Model	Query1	Query2	Query3	Query4	Query5	Query6	Query7	Query8	Query9	Query10	**Decline**
Spare Retriever											
BM25	28.59	34	33.14	36.53	37.38	37.95	38.08	38.41	38.86	39.87	-11.28
Dense Retriever											
jina-embeddings-v3	76.09	71.26	71.84	71.04	65.59	65.65	64.75	64.24	64.62	60.71	15.38
gte-large-en-v1.5	75.87	77.26	73.79	70.22	70.71	67.40	67.53	64.97	65.36	61.36	14.51
NV-Embed-v2	80.53	80.32	78.81	75.70	75.68	72.61	73.28	71.54	70.00	68.02	12.51
bge-en-icl	83.42	80.65	78.44	76.77	74.54	73.00	74.23	73.25	69.70	68.00	15.42
gte-Qwen2-7B-instruct	70.75	72.22	69.99	68.51	65.20	63.53	62.22	62.20	59.15	56.17	14.58
gte-Qwen2-1.5B-instruct	73.64	74.97	72.23	71.37	69.94	67.46	66.92	64.64	63.65	58.68	14.96
e5-mistral-7b-instruct	75.05	70.85	68.18	67.45	63.70	61.60	59.70	59.07	57.85	58.12	16.93
GritLM-7B	82.08	80.32	78.38	76.40	76.40	73.50	75.69	**74.62**	**74.53**	**75.95**	**6.13**
LLM2Vec	83.13	77.42	75.49	75.48	72.49	72.56	70.56	70.73	68.71	67.00	16.13
Reranker											
bge-reranker-v2-m3	87.14	85.56	78.62	76.05	74.29	68.41	67.86	59.48	55.59	44.87	42.27
bge-reranker-v2-gemma	**91.07**	**90.02**	**86.70**	**84.99**	**83.17**	**79.00**	**75.89**	72.29	67.11	56.09	34.98
followIR	83.41	79.72	76.25	74.60	70.12	67.94	62.62	55.93	48.59	43.52	39.89

Table 1: Impact of increasing condition quantity in queries on average Win Rate (Task 1). The Decline reflects the degree of Win Rate reduction from query1 to query10.

tinguish hard negatives, especially when HN_n and HN_{n-1} are highly similar. In contrast, they perform more reliably in clear-cut "exact match" or "complete mismatch" cases.

4.3 Results for Format Invariance

Table 2 presents the Flip Rate induced by query format variations. The results reveal that most models exhibit a Flip Rate exceeding 10%, indicating *a substantial impact of query formulation on retrieval performance*. BM25 achieves a Flip Rate of 16.19%, which serves as a baseline for the lexical differences caused by query format variations.

Dense retrieval models show relatively lower sensitivity to changes in query format, with Flip Rates ranging from 8% to 16%. Among them, GritLM-7B (8.21%), NV-Embed-v2 (9.12%), and LLM2Vec (9.78%) exhibit less variation, suggesting higher robustness to query styles.

Model	People	Books	Movies	Medical	Legal	Avg.
Spare Retriever						
BM25	14.86	17.88	16.84	19.25	12.14	16.19
Dense Retriever						
jina-embeddings-v3	10.55	8.65	10.24	14.72	13.10	11.45
gte-large-en-v1.5	11.74	8.84	12.96	15.70	15.16	12.88
NV-Embed-v2	10.17	8.80	7.52	10.17	8.94	9.12
bge-en-icl	13.48	12.74	15.18	19.81	14.44	15.13
gte-Qwen2-7B-instruct	12.71	15.56	13.62	16.37	17.56	15.16
gte-Qwen2-1.5B-instruct	12.38	13.26	10.48	16.81	12.51	13.09
e5-mistral-7b-instruct	9.17	9.92	8.20	10.75	12.25	10.06
GritLM-7B	8.52	5.35	8.32	8.98	9.86	**8.21**
LLM2Vec	12.81	7.93	9.56	8.12	10.49	9.78
Reranker						
bge-reranker-v2-m3	42.40	32.22	34.82	28.35	31.24	33.81
bge-reranker-v2-gemma	27.50	18.94	16.52	13.42	24.41	20.16
followIR	35.81	31.60	23.70	25.07	28.43	28.92

Table 2: Flip Rate for query format shift (Task 3). The Flip Rate reflects the win rate reversal when switching the query format from instruction-style to descriptive-style.

In contrast, *reranking models show significantly higher sensitivity to query format changes*, with Flip Rates exceeding 20%. The highest Flip Rate

observed is 33.81% for bge-reranker-v2-m3, indicating that reranking models are more susceptible to changes in query formulation.

5 Analysis

5.1 Retrievers vs. Rankers

In our experiments, we observed significant differences in the performance of retrieval models and reranking models across the three tasks—retrieval models demonstrate greater robustness to query complexity, better preservation of relevance monotonicity, and stronger query format invariance compared to reranking models. We attempt to explain this from the perspectives of relevance computation mechanisms and attention mechanisms.

Retrieval models: robustness induced by bidirectional attention and dual-encoder. Retrieval models typically employ a dual-encoder architecture, where queries and documents are independently encoded before computing their similarity using dot-product or cosine similarity. This independent computation ensures that the generation of query and document embeddings remains unaffected by each other. At the same time, bidirectional attention enables the model to capture the overall semantic meaning of the query better.

Reranking models: sensitivity to query complexity. Unlike retrieval models, rerankers compute relevance by jointly processing the query and document as input. There are two main reranker architectures: (1) cross-encoders, which perform token-level relevance comparison through cross-attention (e.g., bge-reranker-v2-m3), and (2) generative relevance estimation using LLM agents (e.g., bge-reranker-v2-gemma and FollowIR). Both require deep query-document interaction, making them more susceptible to input length variations.

Our experiments reveal that in traditional single-condition retrieval tasks, rerankers outperform retrieval models in ranking effectiveness. This suggests that rerankers are better at capturing query-document relevance when dealing with short and simple queries. However, as query complexity increases, the performance of rerankers deteriorates more rapidly than retrieval models, eventually falling behind them. According to Table 1, the average decline in Win Rate score for rerankers is **39.05%**, more than twice the average decline of retrieval models, which is **14.06%.**

These findings highlight a key limitation in existing reranking models: **their reliance on fine-grained query-document interactions becomes a bottleneck in multi-condition scenarios**. While reranking models outperform retrieval models for queries with fewer conditions, their effectiveness diminishes as query complexity grows.

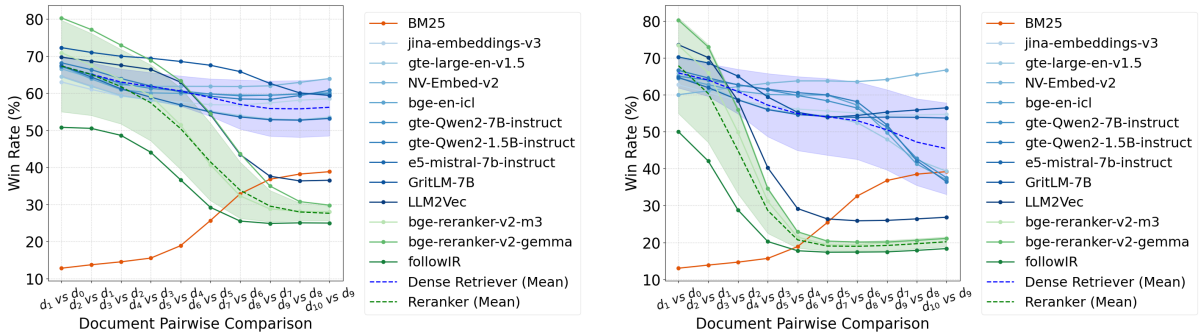
5.2 Effect Of Document Length

Figure 5 shows the impact of document length (512 vs. 1024 words) on relevance monotonicity in multi-condition retrieval, with full results in Appendix D.3. Figure 4 presents win rate results for original documents (without length modification). Comparing Figures 4 and 5, we find that increasing document length does not change the overall trend in Task 2: as hard negatives (HNs) become more challenging, models struggle more to distinguish relevance between documents satisfying k vs. $k-1$ conditions. This decline is more pronounced when document length increases to 1024 words.

Rerankers exhibit greater sensitivity to document length, as evidenced by a sharper decline in win rate compared to retrieval models. This phenomenon also explains the performance degradation of rerankers—since they process both the query and document together as input, their architectures, whether cross-encoder-based or LLM-based generative relevance estimation, struggle with longer inputs, leading to a decline.

5.3 Robustness of GritLM and NV-Embed

GritLM and hybrid attention. GritLM exhibits strong robustness in multi-condition retrieval, likely due to its hybrid attention mechanism, which integrates causal attention for generative tasks and bidirectional attention for embedding learning. Prior research has shown that unidirectional attention constrains a model's ability to generate effective embeddings (Wang et al., 2020; Lee et al., 2024). Recent studies have explored introducing bidirectional attention to improve embedding representations in LLMs. For instance, LLM2Vec (BehnamGhader et al., 2024) introduces a masked prediction task to warm-up the bidirectional attention, while Nv-Embed (Lee et al., 2024) removes causal attention masks during contrastive learning. GritLM (Muennighoff et al., 2024) utilizes a hybrid objective with both bidirectional representation learning and causal generative training. However, both LLM2Vec and NV-Embed discard the causal attention mechanism after transforming



(a) Impact of document length (512 words) on retrieval

(b) Impact of document length (1024 words) on retrieval

Figure 5: Effect of document length on retrieval performance. Figures (a) and (b) show the retrieval performance when padding the document set to 512 words and 1024 words. We use repeated filler text, e.g., *"The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again."*, following the setting in Wang et al. (2023).

the model into an embedding model, which may limit their ability to handle complex, constraint-heavy queries. In contrast, GritLM retains both causal and bidirectional attention, enabling fine-grained semantic modeling while preserving the structural understanding of multi-condition queries. This likely contributes to its superior adaptability in multi-condition retrieval.

Nv-Embed and latent attention layer pooling. Nv-Embed exhibits suboptimal performance across most multi-condition retrieval tasks, which we attribute to its Latent Attention Layer pooling strategy. Common embedding methods include: 1) mean pooling, and 2) the last <EOS> token embedding. Mean pooling averages all token representations to generate a global representation, but this approach can dilute critical information. In contrast, last <EOS> token embedding relies solely on the final token, which introduces recency bias.

Despite its overall suboptimal retrieval performance, Nv-Embed demonstrates notable robustness to increasing document length. Results in Appendix D.3 indicate that while most models suffer significant performance drops when document length expands from 512 to 1024 words, Nv-Embed not only maintains its performance but also achieves the highest average win rate at 1024 words. We hypothesize that this resilience stems from its latent attention layer pooling mechanism, which mitigates information loss more effectively than conventional pooling methods.

Both methods offer distinct advantages: GritLM preserves bidirectional and causal attention, enhancing reasoning capabilities in embeddings, while Nv-Embed's pooling strategy prevents information degradation, particularly in longer documents. These complementary strategies contribute to multi-condition retrieval, balancing semantic richness, reasoning capacity, and information retention, offering insights for future model design.

6 Conclusion

In this paper, we introduce MultiConIR, the first benchmark specifically designed to evaluate retrieval models in multi-condition scenarios. Unlike existing datasets that primarily focus on single-condition queries from search engines, MultiConIR captures real-world complexity by incorporating five diverse domains: books, movies, people, medical cases, and legal documents.

We propose three tasks to systematically assess retrieval and reranking models on multi-condition robustness, monotonic relevance ranking, and query format sensitivity. Through extensive experiments, we analyze model performance across architectures and sizes, examining their ability to distinguish fine-grained conditions, preserve semantic information, and adapt to contextual variations.

Our findings reveal that existing retrieval and reranking models struggle with multi-condition retrieval, with rerankers suffering severe performance degradation as query complexity increases. We further investigate the performance gap between retrieval and reranking models, exploring potential reasons for these discrepancies. Finally, we highlight the strengths of GritLM and Nv-Embed, which demonstrate enhanced adaptability to multi-condition queries, offering insights for future retrieval models.

Limitations

While MultiConIR provides a novel benchmark for evaluating retrieval models in multi-condition scenarios, several limitations should be acknowledged. First, our dataset relies on automated query generation and hard negative creation, which may introduce biases in condition representation despite efforts to ensure accuracy. These biases could affect retrieval models' ability to distinguish fine-grained differences. Second, our evaluation focuses on retrieval tasks and does not cover reasoning-based retrieval or interactive search scenarios. Real-world systems often incorporate reranking, user feedback, and hybrid retrieval, which are not explicitly modeled. Lastly, our dataset does not fully consider query reformulation strategies or multi-turn retrieval, limiting its applicability to dynamic search environments. These limitations highlight the need for further research into multi-condition retrieval, particularly in addressing dataset biases, expanding evaluation scopes, and integrating retrieval with realistic user interactions.

Ethics Statement

This study adheres to ethical standards in AI research, ensuring transparency and reproducibility in dataset construction and model evaluation while exclusively using publicly available pre-trained models for experiments. Dataset Considerations: MultiConIR is built from publicly available sources and does not contain sensitive or personally identifiable information. Given its inclusion of medical and legal documents, we apply strict data filtering and safety measures to respect model safety constraints and prevent the generation of harmful or misleading content. Additionally, we recognize that automatically generated queries and hard negatives may introduce biases. Therefore, during dataset construction, we take measures to minimize the impact of inherent language model biases on retrieval tasks. MultiConIR aims to advance multi-condition retrieval research while ensuring data fairness and ethical compliance. We encourage future research to further explore bias detection strategies in retrieval dataset, enhancing model fairness and reliability in diverse corpus environments.

References

Akari Asai, Timo Schick, Patrick Lewis, Xilun Chen, Gautier Izacard, Sebastian Riedel, Hannaneh Hajishirzi, and Wen-tau Yih. 2023. Task-aware retrieval with instructions. *ACL Findings*.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders. *Preprint*, arXiv:2404.05961.

Adam Berger, Rich Caruana, David Cohn, Dayne Freitag, and Vibhu Mittal. 2000. Bridging the lexical chasm: statistical approaches to answer-finding. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 192–199.

Claudio Carpineto and Giovanni Romano. 2012. A survey of automatic query expansion in information retrieval. *Acm Computing Surveys (CSUR)*, 44(1):1–50.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2022. LexGLUE: A benchmark dataset for legal language understanding in english. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4310–4330. 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.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Thomas Palmeira Ferraz, Kartik Mehta, Yu-Hsiang Lin, Haw-Shiuan Chang, Shereen Oraby, Sijia Liu, Vivek Subramanian, Tagyoung Chung, Mohit Bansal, and Nanyun Peng. 2024. Llm self-correction with decrim: Decompose, critique, and refine for enhanced following of instructions with multiple constraints. *arXiv preprint arXiv:2410.06458*.

Qianyu He, Jie Zeng, Qianxi He, Jiaqing Liang, and Yanghua Xiao. 2024. From complex to simple: Enhancing multi-constraint complex instruction following ability of large language models. *arXiv preprint arXiv:2404.15846*.

HPE AI Solutions. 2023. Medical cases classification tutorial dataset.

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)*, pages 6769–6781, Online. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Nv-embed: Improved techniques for training llms as generalist embedding models. *arXiv preprint arXiv:2405.17428*.

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

Chaofan Li, MingHao Qin, Shitao Xiao, Jianlyu Chen, Kun Luo, Yingxia Shao, Defu Lian, and Zheng Liu. 2024. Making text embedders few-shot learners. *Preprint*, arXiv:2409.15700.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023b. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. 2023c. Synthetic data generation with large language models for text classification: Potential and limitations. *arXiv preprint arXiv:2310.07849*.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345.

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2023. Fine-tuning llama for multi-stage text retrieval. *Preprint*, arXiv:2310.08319.

Sameer S. Mahajan. 2017. People wikipedia data.

Quan Mai, Susan Gauch, and Douglas Adams. 2024. Setbert: Enhancing retrieval performance for boolean logic and set operation queries. *arXiv preprint arXiv:2406.17282*.

Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. 2024. Generative representational instruction tuning. *Preprint*, arXiv:2402.09906.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

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.

Jinming Nian, Zhiyuan Peng, Qifan Wang, and Yi Fang. 2024. W-RAG: Weakly Supervised Dense Retrieval in RAG for Open-domain Question Answering. *arXiv preprint arXiv:2408.08444*.

Hanseok Oh, Hyunji Lee, Seonghyeon Ye, Haebin Shin, Hansol Jang, Changwook Jun, and Minjoon Seo. 2024. Instructir: A benchmark for instruction following of information retrieval models. *arXiv preprint arXiv:2402.14334*.

Jay M Ponte and W Bruce Croft. 2017. A language modeling approach to information retrieval. In *ACM SIGIR Forum*, volume 51, pages 202–208. ACM New York, NY, USA.

Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. 2024. Infobench: Evaluating instruction following ability in large language models. *arXiv preprint arXiv:2401.03601*.

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

Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. Citeseer.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Jonathan Robischon. 2018. Wikipedia movie plots.

Elvin Rustamov. 2021. Books dataset.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. 2024. Ai models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759.

Saba Sturua, Isabelle Mohr, Mohammad Kalim Akram, Michael Günther, Bo Wang, Markus Krimmel, Feng Wang, Georgios Mastrapas, Andreas Koukounas, Nan Wang, et al. 2024. jina-embeddings-v3: Multilingual embeddings with task lora. *arXiv preprint arXiv:2409.10173*.

Hongjin Su, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han-yu Wang, Haisu Liu, Quan Shi, Zachary S Siegel, Michael Tang, et al. 2024. Bright: A realistic and challenging benchmark for reasoning-intensive retrieval. *arXiv preprint arXiv:2407.12883*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2020. Superglue: A stickier benchmark for general-purpose language understanding systems. *Preprint*, arXiv:1905.00537.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024a. Improving text embeddings with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11897–11916, Bangkok, Thailand. Association for Computational Linguistics.

Yifei Wang, Jizhe Zhang, and Yisen Wang. 2024b. Do generated data always help contrastive learning? *Preprint*, arXiv:2403.12448.

Orion Weller, Benjamin Chang, Sean MacAvaney, Kyle Lo, Arman Cohan, Benjamin Van Durme, Dawn Lawrie, and Luca Soldaini. 2024a. Followir: Evaluating and teaching information retrieval models to follow instructions. *arXiv preprint arXiv:2403.15246*.

Orion Weller, Dawn Lawrie, and Benjamin Van Durme. 2024b. Nevir: Negation in neural information retrieval. *Preprint*, arXiv:2305.07614.

Orion Weller, Benjamin Van Durme, Dawn Lawrie, Ashwin Paranjape, Yuhao Zhang, and Jack Hessel. 2024c. Promptriever: Instruction-trained retrievers can be prompted like language models. *arXiv preprint arXiv:2409.11136*.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1503–1512.

Wenhao Zhang, Mengqi Zhang, Shiguang Wu, Jiahuan Pei, Zhaochun Ren, Maarten de Rijke, Zhumin Chen, and Pengjie Ren. 2024a. Excluir: Exclusionary neural information retrieval. *arXiv preprint arXiv:2404.17288*.

Xinghua Zhang, Haiyang Yu, Cheng Fu, Fei Huang, and Yongbin Li. 2024b. Iopo: Empowering llms with complex instruction following via input-output preference optimization. *arXiv preprint arXiv:2411.06208*.

Zongmeng Zhang, Jinhua Zhu, Wengang Zhou, Xiang Qi, Peng Zhang, and Houqiang Li. 2024c. Boolquestions: Does dense retrieval understand boolean logic in language? *Preprint*, arXiv:2411.12235.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zheng Liu, Zhicheng Dou, and Ji-Rong Wen. 2023. Large language models for information retrieval: A survey. *arXiv preprint arXiv:2308.07107*.

A Details of Models

For each model used in this paper, Table 3 provides details on model size, architecture, maximum input context length, and whether instructions is included.

Model	Size	Architecture	Instruction	Max length
Sparse Retriever				
BM25 (Robertson and Zaragoza, 2009)	N/A	Sparse	No	N/A
Dense Retriever				
jina-embeddings-v3 (Sturua et al., 2024)	572M	Encoder	No	4K
gte-large-en-v1.5 (Li et al., 2023b)	434M	Encoder	No	8K
NV-Embed-v2 (Lee et al., 2024)	7.8B	Decoder	Yes	32K
bge-en-icl (Li et al., 2024)	7.1B	Decoder	Yes	32K
gte-Qwen2-7B-instruct (Li et al., 2023b)	7.6B	Decoder	Yes	131K
gte-Qwen2-1.5B-instruct (Li et al., 2023b)	1.5B	Decoder	Yes	131K
e5-mistral-7b-instruct (Wang et al., 2024a)	7.1B	Decoder	Yes	32K
GritLM-7B (Muennighoff et al., 2024)	7.2B	Decoder	Yes	4K
LLM2Vec (BehnamGhader et al., 2024)	7.5B	Decoder	Yes	8K
Reranker				
bge-reranker-v2-m3 (Chen et al., 2024)	568M	Cross-Encoder	No	8k
bge-reranker-v2-gemma (Chen et al., 2024)	2.5B	Decoder	Yes	1k
followIR (Weller et al., 2024a)	7.2B	Decoder	Yes	4k

Table 3: **Details of models used in experiments.** We list the number of parameters of each model except the sparse model (BM25). Regarding the model architecture, we distinguish between sparse models, dense models, and rerankers. When known, dense models are further classified as Encoders or Decoders. Rerankers are categorized into Cross Encoders and Decoders (LLM-based generative relevance scoring). Max length denotes the maximum input length used for each model in the experiments. The Instruction column indicates whether instructions are included in the retrieval process.

For Dense Retrieval models that require instructions (NV-Embed-v2, bge-en-icl, gte-Qwen2-7B-instruct, gte-Qwen2-1.5B-instruct, e5-mistral-7b-instruct, GritLM-7B, and LLM2Vec), we use the following instruction:

"Given a domain retrieval query, retrieve documents that meet the specified conditions."

For LLM-based rerankers (bge-reranker-v2-gemma and followIR), we adopt the model's default prompt. For example, bge-reranker-v2-gemma uses the following prompt:

"Given a query A and a passage B, determine whether the passage contains an answer to the query by providing a prediction of either 'Yes' or 'No'."

For models that do not require instructions, we directly input the query and document, such as jina-embeddings-v3, gte-large-en-v1.5, and bge-reranker-v2-m3.

B Prompt Templates For Constructing *MultiCon Dataset*

Table 4, 5, and 6 present the prompts used in Steps 1 to 3 for constructing our ***MultiCon*** dataset.

For placeholders, {*domain*} ∈ {*People, Books, Movies, Medical Case, Legal Document*}. {*domain_features*} specifies key attributes within a particular domain. In the medical case domain, {*domain_features*} ∈ { *patient symptoms, clinical diagnosis, drug allergies, family medical history, surgical details, postoperative outcomes, hospitalization duration, recovery status.*} In the legal document domain, {*domain_features*} ∈ { *case type, involved parties, court ruling, legal provisions, evidence summary, defense strategy.* } In the movies domain, {*domain_features*} ∈ { *summary, lead actors, release date, release area, genre, detailed plots.* } In the books domain, {*domain_features*} ∈ { *author, publication*

year, genre, main content, detailed plots. } In the people domain, {*domain_features*} ∈ { *profession, nationality, notable achievements, social impact, related events.* }

Task	Prompt
Step 1: Condition Sentence Extraction	I will provide you a document of {domain}, you should extract ten detailed sentences that represent the key conditions the document satisfies. Please adhere to the following guidelines: - Extract fine-grained condition-related sentences relevant to {domain}, such as {domain_features}. - Do not paraphrase; use the original sentences from the document. - Ensure each sentence is semantically intact and not conflict with the context. - Format the output as an array, e.g., ["sentence1", "sentence2", ..., "sentence10"]. Here is the document: {domain_document}. Return array only.

Table 4: Prompt for GPT-4o to extract condition sentence (Step 1).

C Details And Examples Of The *MultiCon* Dataset

Table 7 presents the five domains and their source data used for constructing our ***MultiCon*** dataset. Tables 9, 10, 8, 11, and 12 illustrate examples from their respective domains.

D Complete Results

D.1 Complete Results Of Task1

Table 13, 14, 15, 16, and 17 present the complete results of Task 1 for the five datasets: People, Books, Movies, Medical Case, and Legal Document.

D.2 Complete Results Of Task 2

Table 18 presents the experimental results of Task 2, where Win Rate reflects the success rate between documents that satisfy different numbers of conditions under a multi-condition query (query10, which contains ten conditions), i.e., d_k vs. d_{k-1}.

D.3 Complete Results Of Document Length

Table 19 and Table 20 present the effect of document length on retrieval performance, with documents padded to 512 and 1024 words, respectively. We use repeated filler text, such as *"The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again."*, following the setting in Wang et al. (2023). The filler text is inserted between the original document sentences until the total text length reaches 512 or 1024 words.

E Findings In Constructing *MultiCon* Dataset

E.1 The Use Of LLM-generated Data in Retrieval

In recent years, artificial datasets generated by LLMs have become a common practice for training and evaluating retrieval models (Su et al., 2024; Lee et al., 2024; Weller et al., 2024a). For instance, E5-Mistral (Wang et al., 2024a) rely entirely on LLM-generated datasets for fine-tuning. While this approach can significantly expand training corpora, prior studies have highlighted its potential drawbacks, including introducing inherit linguistic biases of the underlying LLMs (Shumailov et al., 2024), potentially constraining the retrieval model's performance and generalizability. Furthermore, purely artificial data

Task	Prompt
Step 2: Query Generation (Instruction-style)	I will provide you {num} condition-related sentences; formulate a retrieval query for me. Here are a few examples for reference: - {Instruction-style example 1} - {Instruction-style example 2} Please adhere to the following guidelines: - Each sentence represents a condition; with {num} sentences, the number of conditions is {num}. - The query should be instruction-style, explicitly listing all conditions. - Each condition should be around 10 words. - Make conditions concise, summarizing each sentence. - You can paraphrase and modify keywords while maintaining meaning. Here are the sentences: {info}. Return one query only. Do not include extra information.
Step 2: Query Generation (Descriptive-style)	I will provide you {num} condition-related sentences; formulate a retrieval query for me. Here are a few examples for reference: - {Descriptive-style example 1} - {Descriptive-style example 2} Please adhere to the following guidelines: - Each sentence represents a condition; with {num} sentences, the number of conditions is {num}. - The query should be descriptive-style, integrating and describing all conditions in natural language. - Each condition should be around 10 words. - Make conditions concise, summarizing each sentence. - You can paraphrase and modify keywords while maintaining meaning. Here are the sentences: {info}. Return one query only. Do not include extra information.

Table 5: Prompt for GPT-4o to generate queries with varying conditions (Step 2).

often lacks the contextual richness and complexity found in real-world retrieval scenarios (Li et al., 2023c; Wang et al., 2024b), making it difficult to capture the actual needs of users' queries accurately.

During our dataset construction, we observed similar issues. When using LLM-generated transformations to modify positive documents into hard negatives, the model often restructured expressions to fit its learned patterns, even when explicitly instructed to modify only a few condition-related words while keeping the rest unchanged. For example, in the legal documents dataset, a positive sentence like: *"The defendant was convicted of fraud under Section 420 of the Penal Code and sentenced to five years in prison."* was frequently modified by the LLM into a generic pattern, such as: *"The defendant was found guilty of fraud and received a prison sentence."*

Similarly, in medical case documents, a sentence like: *"The patient reported experiencing persistent chest pain and shortness of breath, leading to a diagnosis of angina."* was often transformed into a standardized version: *"The patient was diagnosed with a heart condition after reporting chest pain."*

Task	Prompt
Step 3: Hard Negative Sentence Making (For Books, Movies, Medical Case, and Legal Document Datasets)	I will provide you one query and one sentence, generate a modified sentence for me. Here are a few examples for reference: Query: - {query} Sentence: - {condition sentence} Modified: - {hard negative sentence} Please adhere to the following guidelines: - Modify the sentence so that its meaning no longer aligns with the query. - Keep key terms unchanged. - Ensure the new sentence is semantically different from the original. Here is the query: {query}. Here is the Sentence: {information}. Return only the modified sentence.
Step 3: Hard Negative Sentence Making (For People Dataset)	I will provide you one query and one sentence, generate a modified sentence for me. Here are a few examples for reference: Query: - Who is the American artist that went to RISD? Sentence: - He went to RISD for graduate school. Modified: - He went to ACCA for graduate school, but his sister went to RISD. Please adhere to the following guidelines: - Modify the sentence so that its meaning no longer aligns with the query. - Keep key terms unchanged, but introduce dummy information to mislead the retrieval model. For example, if the original sentence states, "He went to RISD for graduate school," you can modify it to, "He went to ACCA for graduate school, but his sister went to RISD," where the key term (RISD) remains but is assigned to an irrelevant entity (his sister). - Ensure the new sentence is semantically different from the original by using different wording and synonymous substitution. - The changed sentence should prevent the query from retrieving it as relevant information. Here is the query: {query}. Here is the sentence: {information}. Return only the modified sentence.

Table 6: Prompt for GPT-4o to modify the condition sentence to hard negative sentence (Step 3).

These modifications eroded the diversity and long-tail characteristics of real-world data, reducing the fine-grained variability necessary for retrieval tasks. Instead of preserving rich domain-specific details, LLM-generated transformations tended to normalize distinct cases into overly generic patterns, which could misrepresent real-world retrieval challenges.

Empirical results further confirm the limitations of fully LLM-generated training data. The E5-Mistral model, which relies entirely on synthetic data, performs the worst on MultiConIR. In Task 1, as shown in Table 1, it exhibits the highest performance decline (16.93%) among retrieval models, and in Task 2, as shown in Table 18, its average win rate (60.36%) is the lowest among retrieval models, trailing the

Domain	Records	Source Data
People	420	People Wikipedia Dat(Mahajan, 2017)
Books	482	Books Dataset (Rustamov, 2021)
Movies	500	Wikipedia Movie Plots (Robischon, 2018)
Medical Case	479	Medical Cases (HPE AI Solutions, 2023)
Legal Document	426	LexGLUE (Chalkidis et al., 2022)

Table 7: Domain-specific Source Data References. The Records column indicates the number of data entries for each domain in the MultiCon dataset.

second-worst model (Jina-Embeddings-V2) by 5%. These results reinforce the generalization challenges posed by fully synthetic datasets in retrieval tasks, highlighting the importance of incorporating real-world document structures and constraints in training data.

To mitigate this, our pipeline minimizes document-wide modifications, instead restricting LLM interventions to condition sentences only. This targeted approach preserves real-world data authenticity while introducing controlled semantic perturbations, ensuring that retrieval models are trained on meaningful and realistic hard negatives rather than fully synthetic documents.

E.2 Impact of Different Hard Negative Construction Strategies

To systematically examine the impact of hard negative sentence (HNS) construction on retrieval models, we experimented with two distinct approaches: (1) Key Information Modification – altering critical details while maintaining overall sentence structure (applied to books, movies, medical cases, and legal documents). (2) Keyword Retention with Dummy Information – keeping all original keywords intact while injecting irrelevant dummy information (used for the people dataset).

A key objective of this study was to investigate how different HNS construction strategies affect retrieval difficulty. Our initial hypothesis was that the second approach (retaining keywords but adding dummy information) would pose a greater challenge for retrieval models, particularly Dense Retrievers, since hard negatives in this setting contain all the key terms present in positive documents.

However, our experimental results contradicted this expectation. As shown in Fig.6 on the people dataset, Dense Retrieval models remained highly stable, demonstrating a strong ability to differentiate semantic nuances even when all keywords were retained. This suggests that Dense Retrieval primarily relies on contextual embeddings rather than simple keyword matching, allowing it to distinguish between truly relevant documents and distractors with superficial lexical overlap.

In contrast, Reranker models exhibited a significant performance drop when dealing with dummy-information-based HNS. This suggests that Rerankers are more sensitive to this type of negative construction, likely due to their cross-encoder or generative architectures, which process both the query and document jointly. Since Rerankers typically assign scores based on fine-grained textual relevance, the presence of keyword overlap without genuine semantic alignment may mislead them more than Dense Retrieval models.

These findings highlight important considerations for hard negative sampling in multi-condition retrieval. While Dense Retrievers appear robust to surface-level keyword retention, Rerankers are more vulnerable to semantically misleading negatives, suggesting that future retrieval pipelines should adapt negative sampling strategies based on the target retrieval model architecture.

Query 10	Positive	HN 1
Find a notable individual who meets these criteria: 1. Studies plasma melatonin to assess biological rhythm disorders. 2. Identified 25-hour circadian rhythms in totally blind individuals. 3. Worked at NIMH in Bethesda, Maryland before 1981. 4. Collaborated closely with Robert Sack in Oregon. 5. Researches chronobiologic sleep and mood disorders. 6. Graduated from University of Chicago in nineteen seventies with MD and PhD. 7. Had around 90 publications on PubMed by 2005. 8. Is a full professor and vice-chair of Psychiatry at OHSU. 9. Focuses on bright light exposure and melatonin treatments. 10. **Studies disorders like winter depression, jet lag, and shift work maladaptation.**	Relying on a very precise assay for plasma melatonin, a hormone that has a clearly defined 24-hour pattern of secretion, biological rhythm disorders can be assessed and their treatment can be monitored. Totally blind individuals have 25-hour circadian rhythms, drifting an hour later each day unless they take a melatonin capsule at a certain time every day. Prior to moving to Oregon in 1981, Lewy was at the National Institute of Mental Health (NIMH) in Bethesda, Maryland, working with senior colleague Thomas Wehr. In Oregon, he has worked closely with Robert L. Sack. He describes his research as follows: 'My laboratory studies chronobiologic sleep and mood disorders.' Alfred J. Lewy, aka Sandy Lewy, graduated from University of Chicago in 1973 after studying psychiatry, pharmacology, and ophthalmology. As of December 2005, he had 94 publications available on PubMed. He is a full professor and vice-chair of the Department of Psychiatry at OHSU (Oregon Health Science University) and holds an MD and PhD. Current research is focused on developing bright light exposure and melatonin administration as treatment modalities for these disorders. **These disorders include winter depression, jet lag, maladaptation to shift work, and certain types of sleep disturbances.**	Relying on a very precise assay for plasma melatonin, a hormone that has a clearly defined 24-hour pattern of secretion, biological rhythm disorders can be assessed and their treatment can be monitored. Totally blind individuals have 25-hour circadian rhythms, drifting an hour later each day unless they take a melatonin capsule at a certain time every day. Prior to moving to Oregon in 1981, Lewy was at the National Institute of Mental Health (NIMH) in Bethesda, Maryland, working with senior colleague Thomas Wehr. In Oregon, he has worked closely with Robert L. Sack. He describes his research as follows: 'My laboratory studies chronobiologic sleep and mood disorders.' Alfred J. Lewy, aka Sandy Lewy, graduated from University of Chicago in 1973 after studying psychiatry, pharmacology, and ophthalmology. As of December 2005, he had 94 publications available on PubMed. He is a full professor and vice-chair of the Department of Psychiatry at OHSU (Oregon Health Science University) and holds an MD and PhD. Current research is focused on developing bright light exposure and melatonin administration as treatment modalities for these disorders. **These lifestyle changes address issues like winter depression, jet lag, shift work maladaptation, and certain types of sleep disturbances.**

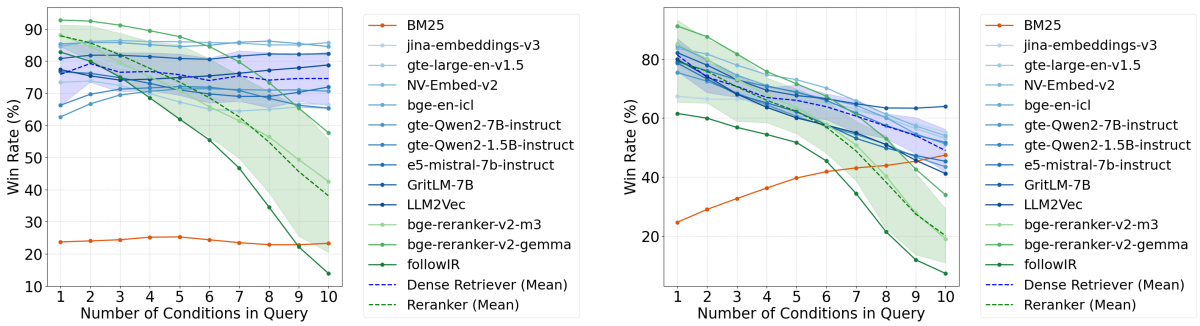
Table 8: An example in domain of Famous People

Query 10	Positive	HN 1
Find a notable individual who meets these criteria: 1. Details Bernie Madoff's $65B Ponzi collapse. 2. Covers the impact on national media. 3. Investigates Madoff's history of fraud. 4. Offers deep insight into Madoff's family. 5. Contains exclusive news and material. 6. Includes Bernie's Little Black Book. 7. Features Madoff's past calendars. 8. Provides details on the origins of his scam. 9. Includes cooperation from Eleanor Squillari. 10. **Reveals SEC connections to Madoff.**	The collapse of Bernie Madoff's Ponzi scheme led to the instant evaporation of $65 billion of wealth. The effects of Madoff's brazen fraud were felt most closely in New York and Palm Beach but the story was, and continues to be, front page news across the country. Brian Ross and his team of investigators shed an unyielding light onto Madoff's scheme–how he got started, how he succeed for so long, who helped him, and who shielded him from early investigations. This is an incisive and voyeuristic look into this first family of financial crime. The Madoff Chronicles includes a vast array of news and material that readers won't find anywhere else. Contains a reproduction of Bernie's Little Black Book. Ross has also secured Madoff's calendar for the past three years and other never-before-seen documents from inside the Madoff empire, straight from his desk. Read key details of how Madoff carried out his scam and the revelation that he began the fraud from almost the first day, in the 1960s. Extensive cooperation by Madoff's personal assistant, Eleanor Squillari. **Contains incriminating connections between Madoff and certain members of the SEC.**	The collapse of Bernie Madoff's Ponzi scheme led to the instant evaporation of $65 billion of wealth. The effects of Madoff's brazen fraud were felt most closely in New York and Palm Beach but the story was, and continues to be, front page news across the country. Brian Ross and his team of investigators shed an unyielding light onto Madoff's scheme–how he got started, how he succeed for so long, who helped him, and who shielded him from early investigations. This is an incisive and voyeuristic look into this first family of financial crime. The Madoff Chronicles includes a vast array of news and material that readers won't find anywhere else. Contains a reproduction of Bernie's Little Black Book. Ross has also secured Madoff's calendar for the past three years and other never-before-seen documents from inside the Madoff empire, straight from his desk. Read key details of how Madoff carried out his scam and the revelation that he began the fraud from almost the first day, in the 1960s. Extensive cooperation by Madoff's personal assistant, Eleanor Squillari. **Contains no documented communications between Madoff and SEC members.**

Table 9: An example in domain of Book

Query 10	Positive	HN 1
Find a movie that matches all conditions: 1. Originated from American. 2. Plot: Mrs. Lowe and Black were lovers. 3. Plot: Terry carelessly spends money. 4. Plot: Terry promoted to ship foreman. 5. Cast: Mary Miles Minter, Allan Forrest. 6. Plot: Julia Deep works behind exchange desk. 7. Director: Lloyd Ingraham. 8. **Plot: Terry's spending takes a toll.** 9. Plot: Lottie sees Terry and Julia at park. 10. Release Year: 1918.	Origin/Ethnicity: American Meanwhile, it is revealed Mrs. Lowe and Black were once lovers. He is spending his money carelessly and doesn't put any time in paying the bills, much to the dislike of the department store owner Timothy Black. Soon, Terry is promoted to a foreman on a ship. Cast: Mary Miles Minter, Allan Forrest Julia Deep is a young woman working behind the exchange desk at a department store. Director: Lloyd Ingraham **After a while, Terry's money spending takes its toll.** Lottie gets distracted and does not notice Terry and Julia at the park. Release Year: 2022	Origin/Ethnicity: American Meanwhile, it is revealed Mrs. Lowe and Black were once lovers. He is spending his money carelessly and doesn't put any time in paying the bills, much to the dislike of the department store owner Timothy Black. Soon, Terry is promoted to a foreman on a ship. Cast: Mary Miles Minter, Allan Forrest Julia Deep is a young woman working behind the exchange desk at a department store. Director: Lloyd Ingraham **Eventually, Terry's frugality leads to financial growth.** Lottie gets distracted and does not notice Terry and Julia at the park. Release Year: 2022

Table 10: An example in domain of Movie



(a) Task1 on Peole dataset

(b) Task1 on Legal dataset

Figure 6: Impact of different HNS construction strategies.

Query 10	Positive	HN 1
Find a case where the patient: 1. Underwent ascending aortic arch angiogram. 2. Had left common carotid artery angiogram. 3. Received right common carotid artery angiogram. 4. Undergone left subclavian artery angiogram. 5. Had right iliac angiogram with runoff. 6. Performed bilateral cerebral angiograms. 7. Experienced TIA and moderate carotid stenosis. 8. Had 400 ml blood loss. 9. Provided informed consent for the procedure. 10. **Received a 6-French sheath in the right femoral artery.**	PROCEDURE PERFORMED: 1. Selective ascending aortic arch angiogram. 2. Selective left common carotid artery angiogram. 3. Selective right common carotid artery angiogram. 4. Selective left subclavian artery angiogram. 5. Right iliac angio with runoff. 6. Bilateral cerebral angiograms were performed as well via right and left common carotid artery injections. INDICATIONS FOR PROCEDURE: TIA, aortic stenosis, postoperative procedure. Moderate carotid artery stenosis. ESTIMATED BLOOD LOSS: 400 ml. After obtaining informed consent, the patient was brought to the cardiac catheterization suite in postabsorptive and nonsedated state. **Using modified Seldinger technique, a 6-French sheath was placed into the right common femoral artery and vein without complication.**	PROCEDURE PERFORMED: 1. Selective ascending aortic arch angiogram. 2. Selective left common carotid artery angiogram. 3. Selective right common carotid artery angiogram. 4. Selective left subclavian artery angiogram. 5. Right iliac angio with runoff. 6. Bilateral cerebral angiograms were performed as well via right and left common carotid artery injections. INDICATIONS FOR PROCEDURE: TIA, aortic stenosis, postoperative procedure. Moderate carotid artery stenosis. ESTIMATED BLOOD LOSS: 400 ml. After obtaining informed consent, the patient was brought to the cardiac catheterization suite in postabsorptive and nonsedated state. **A 6-French sheath was used in the left femoral artery and vein with minor complications, employing the modified Seldinger technique.**

Table 11: An example in domain of Medical Case

Query 10	Positive	HN 1
Find a case where: 1. Michigan Legislature enacted a statute in 1987. 2. Petitioners challenged the statute under Contract Clause and Due Process Clause. 3. The statute affected workers injured before March 31, 1982. 4. Petitioners argued a 1981 law allowed reduction of workers' compensation benefits. 5. The Michigan Supreme Court accepted petitioners' interpretation in 1985. 6. Legislature introduced a bill to overturn the court's decision. 7. House Bill 5084 was introduced in October 1985. 8. The bill became law on May 14, 1987. 9. Petitioners were ordered to refund nearly $25 million. 10. **Michigan Supreme Court upheld the statute for lacking vested rights and rational purpose.**	In 1987, the Michigan Legislature enacted a statute that had the effect of requiring petitioners General Motors Corporation (GM) and Ford Motor Company (Ford) to repay workers' compensation benefits GM and Ford had withheld in reliance on a 1981 workers' compensation statute. Petitioners challenge the provision of the statute mandating these retroactive payments on the ground that it violates the Contract Clause and the Due Process Clause of the Federal Constitution. The benefit coordination provision did not specify whether it was to be applied to workers injured before its effective date, March 31, 1982. Petitioners took the position that the 1981 law allowed them to reduce workers' compensation benefits to workers injured before March 31, 1982, who were receiving benefits from other sources. In 1985, petitioners' interpretation was accepted by the Michigan Supreme Court. Chambers v. General Motors Corp., decided together with Franks v. White Pine Copper Div., Copper Range Co., 422 Mich. 636, 375 N.W.2d 715. The Michigan Legislature responded almost immediately by introducing legislation to overturn the court's decision. On October 16, 1985, before the Michigan Supreme Court had ruled on the motion for rehearing in Chambers, House Bill 5084 was introduced. The amended Senate bill passed into law on May 14, 1987. 1987 Mich.Pub.Acts No. 28. As a result of the 1987 statute, petitioners were ordered to refund nearly $25 million to disabled employees. **The Michigan Supreme Court upheld the statute against these challenges, on the ground that the employers had no vested rights in coordination for Contract Clause purposes, and that the retroactive provisions furthered a rational legislative purpose.** 436 Mich. 515, 462 N.W.2d 555 (1990).	In 1987, the Michigan Legislature enacted a statute that had the effect of requiring petitioners General Motors Corporation (GM) and Ford Motor Company (Ford) to repay workers' compensation benefits GM and Ford had withheld in reliance on a 1981 workers' compensation statute. Petitioners challenge the provision of the statute mandating these retroactive payments on the ground that it violates the Contract Clause and the Due Process Clause of the Federal Constitution. The benefit coordination provision did not specify whether it was to be applied to workers injured before its effective date, March 31, 1982. Petitioners took the position that the 1981 law allowed them to reduce workers' compensation benefits to workers injured before March 31, 1982, who were receiving benefits from other sources. In 1985, petitioners' interpretation was accepted by the Michigan Supreme Court. Chambers v. General Motors Corp., decided together with Franks v. White Pine Copper Div., Copper Range Co., 422 Mich. 636, 375 N.W.2d 715. The Michigan Legislature responded almost immediately by introducing legislation to overturn the court's decision. On October 16, 1985, before the Michigan Supreme Court had ruled on the motion for rehearing in Chambers, House Bill 5084 was introduced. The amended Senate bill passed into law on May 14, 1987. 1987 Mich.Pub.Acts No. 28. As a result of the 1987 statute, petitioners were ordered to refund nearly $25 million to disabled employees. **The Michigan Supreme Court found the statute invalid on the grounds that the retroactive provisions did not further a rational legislative purpose and that the employers had vested rights in coordination for Contract Clause purposes.** 436 Mich. 515, 462 N.W.2d 555 (1990).

Table 12: An example in domain of Legal Document

Model	Query1	Query2	Query3	Query4	Query5	Query6	Query7	Query8	Query9	Query10	Decline
Spare Retriever											
BM25	23.10	25.24	22.62	26.43	26.19	23.57	24.05	22.14	22.38	23.81	-0.71
Dense Retriever											
jina-embeddings-v3	88.81	78.33	78.33	76.90	72.14	68.57	67.38	66.19	67.62	65.95	22.86
gte-large-en-v1.5	72.14	76.19	73.10	68.57	69.29	61.43	66.19	62.38	69.05	63.81	8.33
NV-Embed-v2	82.14	89.52	85.48	85.48	87.38	84.05	87.38	84.05	84.05	86.67	-4.52
bge-en-icl	85.00	85.95	86.43	85.00	83.81	84.29	86.43	87.14	85.48	83.81	1.19
gte-Qwen2-7B-instruct	58.10	70.95	69.05	70.71	71.90	71.43	71.43	69.76	72.62	69.76	-11.67
gte-Qwen2-1.5B-instruct	61.90	74.76	70.24	71.19	73.57	70.48	73.81	66.90	65.71	65.00	-3.10
e5-mistral-7b-instruct	76.43	77.86	73.33	75.48	68.10	71.43	67.62	68.81	69.05	73.57	2.86
GritLM-7B	79.05	84.52	80.71	81.67	81.67	77.86	83.10	82.62	81.43	82.62	-3.57
LLM2Vec	79.05	74.76	71.90	75.71	74.05	75.95	75.00	78.81	76.19	80.00	-0.95
Reranker											
bge-reranker-v2-m3	90.00	85.95	78.57	73.81	74.29	62.14	63.57	56.67	52.14	36.67	53.33
bge-reranker-v2-gemma	92.62	93.33	91.67	88.33	89.29	84.76	81.43	72.38	69.52	50.71	41.90
followIR	84.52	80.48	76.43	68.57	60.71	56.90	50.00	34.05	20.95	8.57	75.95

Table 13: Win Rate Score as the Number of Conditions in the Query Increases on the People Dataset (Task 1)

Model	Query1	Query2	Query3	Query4	Query5	Query6	Query7	Query8	Query9	Query10	decline
Spare Retriever											
BM25	45.85	47.93	47.93	49.17	47.72	51.24	50.21	48.34	53.94	46.06	-0.21
Dense Retriever											
jina-embeddings-v3	76.14	72.41	71.99	71.37	67.01	64.73	67.43	63.07	63.28	58.51	17.63
gte-large-en-v1.5	73.86	75.93	71.99	70.33	68.67	69.71	65.77	66.60	64.32	70.95	2.90
NV-Embed-v2	78.01	73.65	75.93	68.88	69.71	69.50	68.46	68.05	67.84	71.58	6.43
bge-en-icl	78.42	76.76	75.31	72.20	72.41	69.29	70.54	70.12	70.95	72.41	6.02
gte-Qwen2-7B-instruct	71.99	63.90	64.52	61.83	54.77	50.00	50.00	49.17	48.55	50.83	21.16
gte-Qwen2-1.5B-instruct	70.54	69.09	70.33	69.92	65.98	60.58	64.11	62.45	64.11	56.85	13.69
e5-mistral-7b-instruct	73.03	65.77	68.88	65.98	60.58	59.96	59.34	60.58	60.37	68.67	4.36
GritLM-7B	80.91	74.69	81.33	73.86	75.73	73.86	79.25	77.80	78.01	80.50	0.41
LLM2Vec	80.08	70.12	72.41	70.12	68.05	66.80	64.32	65.35	65.15	64.52	15.56
Reranker											
bge-reranker-v2-m3	83.20	79.05	76.35	71.99	72.20	72.82	74.48	68.67	60.58	51.66	31.54
bge-reranker-v2-gemma	87.76	84.44	82.78	81.74	80.08	75.31	74.48	72.82	70.75	70.54	17.22
followIR	85.89	78.22	77.59	72.41	68.67	68.46	61.41	60.37	53.11	47.30	38.59

Table 14: Win Rate Score as the Number of Conditions in the Query Increases on the Books Dataset (Task 1)

Model	Query1	Query2	Query3	Query4	Query5	Query6	Query7	Query8	Query9	Query10	decline
Spare Retriever											
BM25	28.60	34.60	33.60	37.20	34.80	36.80	37.00	38.40	37.60	39.00	-10.40
Dense Retriever											
jina-embeddings-v3	77.60	77.00	75.20	75.40	70.80	72.00	69.40	68.40	71.80	67.20	10.40
gte-large-en-v1.5	78.40	81.00	76.60	75.60	73.80	71.00	71.00	69.00	67.20	59.40	19.00
NV-Embed-v2	81.00	81.80	80.40	78.60	80.00	76.00	77.40	76.60	72.20	67.00	14.00
bge-en-icl	84.40	83.20	80.60	79.20	78.80	77.00	76.00	75.80	70.40	66.60	17.80
gte-Qwen2-7B-instruct	74.60	80.00	74.60	72.80	72.40	72.00	68.80	75.00	67.60	67.40	7.20
gte-Qwen2-1.5B-instruct	80.40	83.00	76.00	74.80	73.00	73.80	71.00	74.60	72.20	68.40	12.00
e5-mistral-7b-instruct	73.40	71.80	66.00	64.80	70.00	61.20	65.60	62.80	63.00	55.60	17.80
GritLM-7B	85.00	83.60	80.00	80.80	79.80	72.80	76.40	72.60	71.40	73.80	11.20
LLM2Vec	87.20	89.40	86.40	84.40	81.80	81.60	79.60	77.80	75.40	74.20	13.00
Reranker											
bge-reranker-v2-m3	92.80	90.80	80.20	81.60	79.20	75.60	71.00	63.40	67.00	60.40	32.40
bge-reranker-v2-gemma	92.20	91.60	86.00	89.20	83.40	79.40	77.60	78.20	70.80	61.60	30.60
followIR	92.60	86.40	83.40	88.00	81.00	81.00	83.40	81.40	82.60	85.00	7.60

Table 15: Win Rate Score as the Number of Conditions in the Query Increases on the Movies Dataset (Task 1)

Model	Query1	Query2	Query3	Query4	Query5	Query6	Query7	Query8	Query9	Query10	Decline
Spare Retriever											
BM25	25.68	28.18	31.52	32.99	37.37	36.33	35.49	39.25	37.16	40.71	-15.03
Dense Retriever											
jina-embeddings-v3	68.89	64.93	65.34	65.55	55.32	59.08	56.16	63.67	60.54	59.29	9.60
gte-large-en-v1.5	77.24	81.84	76.83	75.57	71.82	70.56	70.35	67.01	68.48	62.42	14.82
NV-Embed-v2	77.24	73.07	74.53	73.70	65.97	62.63	67.43	68.68	67.22	63.67	13.57
bge-en-icl	80.79	78.91	76.41	77.04	68.89	68.68	72.23	71.19	69.31	67.64	13.15
gte-Qwen2-7B-instruct	71.61	73.70	73.49	72.65	66.60	67.64	66.18	64.72	60.75	51.57	20.04
gte-Qwen2-1.5B-instruct	73.90	72.86	72.03	70.77	67.43	65.55	63.47	64.09	60.13	53.86	20.04
e5-mistral-7b-instruct	69.31	67.01	67.64	64.09	57.83	55.53	56.16	51.98	50.10	48.64	20.67
GritLM-7B	79.75	82.25	76.62	78.08	77.66	75.16	75.16	77.87	78.91	78.29	1.46
LLM2Vec	84.34	81.21	79.12	83.09	80.58	79.75	78.71	79.33	80.79	78.50	5.85
Reranker											
bge-reranker-v2-m3	81.21	91.02	86.64	86.43	79.33	75.37	75.78	66.39	72.65	62.00	19.21
bge-reranker-v2-gemma	89.35	92.48	90.19	92.90	91.02	85.80	85.39	82.67	80.79	70.35	19.00
followIR	92.28	90.81	89.35	89.56	86.22	84.97	81.42	86.22	76.41	71.82	20.46

Table 16: Win Rate Score as the Number of Conditions in the Query Increases on the Medical Case Dataset (Task 1)

Model	Query1	Query2	Query3	Query4	Query5	Query6	Query7	Query8	Query9	Query10	decline
Spare Retriever											
BM25	19.72	34.04	30.05	36.85	40.85	41.78	43.66	43.90	43.19	49.77	-30.05
Dense Retriever											
jina-embeddings-v3	69.01	63.62	68.31	65.96	62.68	63.85	63.38	59.86	59.86	52.58	16.43
gte-large-en-v1.5	77.70	71.36	70.42	61.03	69.95	64.32	64.32	59.86	57.75	50.23	27.46
NV-Embed-v2	84.27	83.57	77.70	71.83	75.35	70.89	65.73	60.33	58.69	51.17	33.10
bge-en-icl	88.50	78.40	73.47	70.42	68.78	65.73	65.96	61.97	52.35	49.53	38.97
gte-Qwen2-7B-instruct	77.46	72.54	68.31	64.55	60.33	56.57	54.69	52.35	46.24	41.31	36.15
gte-Qwen2-1.5B-instruct	81.46	75.12	72.54	70.19	69.72	66.90	62.21	55.16	56.10	49.30	32.16
e5-mistral-7b-instruct	83.10	71.83	65.02	66.90	61.97	59.86	49.77	51.17	46.71	44.13	38.97
GritLM-7B	85.68	76.53	73.24	67.61	67.14	67.84	64.55	62.21	62.91	64.55	21.13
LLM2Vec	84.98	71.60	67.61	64.08	57.98	58.69	55.16	52.35	46.01	37.79	47.18
Reranker											
bge-reranker-v2-m3	88.50	80.99	71.36	66.43	66.43	56.10	54.46	42.25	25.59	13.62	74.88
bge-reranker-v2-gemma	93.43	88.26	82.86	72.77	72.07	69.72	60.56	55.40	43.66	27.23	66.20
followIR	61.74	62.68	54.46	54.46	53.99	48.36	36.85	17.61	9.86	4.93	56.81

Table 17: Win Rate Score as the Number of Conditions in the Query Increases on the Legal Documents Dataset (Task 1)

Model	d_1_vs_d_0	d_2_vs_d_1	d_3_vs_d_2	d_4_vs_d_3	d_5_vs_d_4	d_6_vs_d_5	d_7_vs_d_6	d_8_vs_d_7	d_9_vs_d_8	d_{10}_vs_d_9	Avg.
Spare Retriever											
BM25	13.91	16.50	16.81	18.14	22.10	29.04	37.78	38.87	39.93	40.19	25.90
Dense Retriever											
jina-embeddings-v3	73.43	70.52	67.45	66.66	65.32	63.40	63.15	62.82	65.13	60.35	65.82
gte-large-en-v1.5	76.74	73.85	72.70	69.91	70.32	68.05	67.39	64.09	65.14	62.58	69.08
NV-Embed-v2	82.57	76.39	74.45	72.10	73.27	69.15	69.48	66.74	68.75	71.57	72.45
bge-en-icl	79.40	70.58	69.36	68.13	64.80	63.12	63.01	61.72	61.31	63.69	66.51
gte-Qwen2-7B-instruct	79.84	74.02	70.57	69.97	65.44	60.54	61.35	59.42	59.55	60.40	66.11
gte-Qwen2-1.5B-instruct	74.30	71.80	72.28	68.49	69.32	65.69	67.08	64.97	63.46	65.09	68.25
e5-mistral-7b-instruct	75.11	67.88	62.73	58.61	56.87	54.52	55.26	54.03	56.68	61.94	60.36
GritLM-7B	79.59	77.73	73.40	74.71	75.56	72.15	73.52	71.87	72.01	75.21	74.58
LLM2Vec	83.50	74.25	73.43	72.24	70.36	67.21	66.99	67.07	66.49	67.48	70.90
Reranker											
bge-reranker-v2-m3	76.08	68.06	63.83	62.06	60.65	58.35	54.79	50.60	48.57	44.96	58.80
bge-reranker-v2-gemma	87.98	82.08	78.07	77.10	76.21	72.13	68.84	65.63	62.32	56.13	72.65
followIR	61.99	59.82	60.87	60.76	59.91	56.41	51.63	47.93	44.71	43.52	54.76

Table 18: Average Win Rate Comparison Between Documents in Task 2

Model	d_1_vs_d_0	d_2_vs_d_1	d_3_vs_d_2	d_4_vs_d_3	d_5_vs_d_4	d_6_vs_d_5	d_7_vs_d_6	d_8_vs_d_7	d_9_vs_d_8	d^+_vs_d_9	Avg.
Spare Retriever											
BM25	11.91	14.34	14.75	14.45	15.99	24.75	36.58	37.68	38.02	39.34	24.78
Dense Retriever											
jina-embeddings-v3	64.8	60.11	58.61	58.31	57.83	56.36	56.42	57.20	58.56	58.59	58.68
gte-large-en-v1.5	66.63	61.15	57.66	59.44	56.26	55.10	53.64	53.14	51.34	54.51	56.89
NV-Embed-v2	68.83	62.87	60.97	62.16	61.78	61.80	62.04	61.10	62.75	64.78	62.91
bge-en-icl	70.55	62.18	59.35	60.33	59.70	60.19	59.28	58.95	60.16	60.23	61.09
gte-Qwen2-7B-instruct	68.63	64.87	61.91	60.28	61.98	58.70	59.28	59.06	60.28	59.69	61.47
gte-Qwen2-1.5B-instruct	69.51	66.38	63.47	61.89	60.13	58.76	58.57	57.36	58.62	62.11	61.68
e5-mistral-7b-instruct	69.98	63.50	60.30	59.3	56.77	54.52	53.05	53.14	51.47	53.99	57.60
GritLM-7B	73.46	70.37	69.19	70.36	68.36	67.05	68.34	61.55	58.38	59.57	66.66
LLM2Vec	70.57	68.37	67.10	67.10	66.35	58.73	36.04	36.81	35.37	37.03	54.35
Reranker											
bge-reranker-v2-m3	73.65	66.55	63.28	61.65	54.60	38.54	28.42	27.92	28.14	28.02	47.08
bge-reranker-v2-gemma	82.39	77.63	71.80	70.00	65.10	56.91	41.75	32.01	28.84	29.96	55.64
followIR	50.48	51.78	49.64	46.27	37.60	25.22	24.41	24.51	25.67	24.71	36.03

Table 19: Effect of document length on retrieval performance (padded to 512 words).

Model	d_1_vs_d_0	d_2_vs_d_1	d_3_vs_d_2	d_4_vs_d_3	d_5_vs_d_4	d_6_vs_d_5	d_7_vs_d_6	d_8_vs_d_7	d_9_vs_d_8	d^+_vs_d_9	Avg.
Spare Retriever											
BM25	12.25	14.32	14.86	14.81	15.97	24.48	36.00	37.53	38.85	39.53	24.86
Dense Retriever											
jina-embeddings-v3	64.56	59.74	58.13	57.20	55.47	55.90	55.33	53.70	54.75	54.81	56.96
gte-large-en-v1.5	68.62	61.61	58.47	54.77	54.97	54.52	54.44	49.88	39.09	38.34	53.47
NV-Embed-v2	59.23	61.26	62.58	62.81	64.57	63.95	62.98	63.49	65.65	67.55	63.41
bge-en-icl	66.08	61.93	60.83	59.63	59.34	61.04	61.08	51.67	36.00	35.95	55.36
gte-Qwen2-7B-instruct	66.06	63.23	63.36	61.35	59.63	58.56	56.62	57.97	36.61	35.96	55.94
gte-Qwen2-1.5B-instruct	68.46	63.66	62.02	62.21	59.47	60.78	59.38	58.71	35.01	34.91	56.46
e5-mistral-7b-instruct	66.97	61.11	59.05	54.53	54.40	54.47	53.02	54.55	53.88	53.57	56.56
GritLM-7B	71.47	67.97	69.41	56.02	54.49	52.58	54.21	56.35	54.87	57.26	59.46
LLM2Vec	74.81	72.05	71.55	26.85	26.68	26.35	25.24	26.16	26.04	27.27	40.30
Reranker											
bge-reranker-v2-m3	76.56	70.75	56.83	18.33	19.86	19.09	18.55	20.75	19.88	21.78	34.24
bge-reranker-v2-gemma	83.48	77.02	66.34	19.76	20.54	19.81	20.64	19.34	20.65	21.46	36.90
followIR	52.36	51.43	19.70	18.35	17.15	17.11	17.78	16.94	17.77	18.75	24.73

Table 20: Effect of document length on retrieval performance (padded to 1024 words).