Learning Contextual Retrieval for Robust Conversational Search

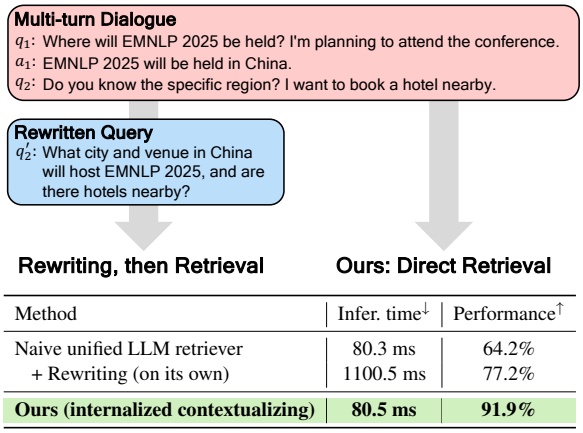
Anonymous ACL submission

Abstract

Effective conversational search demands a deep understanding of user intent across multiple dialogue turns. Users frequently use abbreviations and shift topics in the middle of conversations, posing challenges for conventional retrievers. While query rewriting techniques improve clarity, they often incur significant computational cost due to additional autoregressive steps. Moreover, although LLM-based retrievers demonstrate strong performance, they are not explicitly optimized to track user intent in multi-turn settings, often failing under topic drift or contextual ambiguity. To address these limitations, we propose `ContextualRetriever`, a novel LLM-based retriever that directly incorporates conversational context into the retrieval process. Our approach introduces: (1) a context-aware embedding mechanism that highlights the current query within the dialogue history; (2) intent-guided supervision based on high-quality rewritten queries; and (3) a training strategy that preserves the generative capabilities of the base LLM. Extensive evaluations across multiple conversational search benchmarks demonstrate that `ContextualRetriever` significantly outperforms existing methods while incurring no additional inference overhead.

1 Introduction

The rapid advancement of chatbots has significantly increased demand for conversational search engines (Gao et al., 2023; Mo et al., 2024). These systems must accurately retrieve information from large document collections to provide reliable, factual responses. Traditional search engines primarily handle single-turn queries and struggle in multi-turn conversational contexts, particularly when users heavily rely on abbreviated or context-dependent queries, as exemplified by q_2 in Figure 1. Thus, effective contextualization, which involves understanding user intent throughout the conversation, is crucial for accurate retrieval.



Method	Infer. time$^{\downarrow}$	Performance$^{\uparrow}$
Naive unified LLM retriever	80.3 ms	64.2%
+ Rewriting (on its own)	1100.5 ms	77.2%
Ours (internalized contextualizing)	**80.5 ms**	**91.9%**

Figure 1: **Potential of LLM-based retriever to contextualize the query in conversational search.** Even with a naive unified LLM retriever can be improved after query rewriting on its own (64.2% $\rightarrow$ 77.2%). Notable retrieval performance (91.9%) from the proposed `ContextualRetriever` by better leveraging the contextual understanding capability of LLMs, eliminating additional inference overhead.

A common strategy is query rewriting (Lin et al., 2020; Mo et al., 2023), which reformulates abbreviated or ambiguous user queries into fully specified ones by integrating conversational context (see Figure 1, blue box). Despite improved clarity, this approach necessitates additional rewriting models, significantly increasing inference time and computational overhead.

Recent approaches aim to build retrievers by directly fine-tuning Large Language Models (LLMs), leveraging their inherent language understanding capabilities (Jiang et al., 2023; Bai et al., 2023; Touvron et al., 2023). LLM-based retrievers such as SFR Embedding (Meng et al., 2024) and NV-Embed (Lee et al., 2024; Moreira et al., 2024) apply contrastive learning to pretrained LLMs, optimizing them specifically for retrieval tasks. Unified retrievers, such as GritLM (Muennighoff et al., 2024) and OneGen (Zhang et al., 2024), handle generation and retrieval tasks via multi-task learn-

ing within a single model. However, these methods primarily target single-turn queries and have not fully utilized LLMs' ability to model multi-turn conversational context. To incorporate conversational history, ChatRetriever (Mao et al., 2024) compresses prior turns into a limited number of special tokens. However, this compression strategy quickly saturates, yielding only marginal improvements as more tokens are added. This suggests a key limitation of current approaches: they fail to deeply embed rich conversational history into the retrieval representation itself.

Although prior work has made meaningful progress, few studies have fully leveraged the language understanding capabilities of LLMs to contextualize conversational queries. We hypothesize that LLM-based retrievers possess strong potential for modeling user intent across turns, but this ability remains underutilized in current designs. To validate this hypothesis, we examine GritLM, a unified LLM retriever trained for both generation and retrieval. We compare its retrieval performance when using embeddings derived from its own rewritten queries versus embeddings obtained directly from the original user queries. As shown in Figure 1, rewritten queries yield significantly better retrieval accuracy, suggesting that the model captures user intent and context well during the rewriting step. However, this contextual understanding is not effectively reflected in the retrieval embeddings generated from the original queries, indicating that the model's ability to embed conversational context remains underexploited. This insight motivates our work: we aim to directly encode user intent and dialogue context into the retrieval representation itself, enabling LLM-based retrievers to fully capitalize on their inherent contextual understanding ability.

To address this limitation, we introduce **`ContextualRetriever`**, a novel approach designed to better harness LLMs for retrieval in multi-turn conversations. `ContextualRetriever` comprises three core components: First, it employs a context-aware embedding mechanism that emphasizes the current query while encoding the full dialogue. Retrieval embeddings are computed solely from the current query segment, maintaining focus on the immediate information need while grounding it in broader conversational context. Second, it leverages intent-guided supervision by aligning model-generated embeddings with those derived from high-quality rewritten queries. These rewritten queries clarify user intent, allowing the model to learn intent-aware representations without requiring an explicit rewriting step at inference time. Third, it incorporates generation loss during training to preserve the LLM's intrinsic language understanding capabilities. This allows the model to retain its general linguistic competence, which is essential for interpreting ambiguous or context-dependent queries.

We evaluate `ContextualRetriever` on four standard conversational search benchmarks: TopiOCQA (Adlakha et al., 2022), QReCC (Anantha et al., 2021), TREC-CAsT (Dalton et al., 2020, 2021), and ORConvQA (Qu et al., 2020). Our method consistently outperforms strong baselines, demonstrating that embedding user intent directly into the retrieval space substantially improves multi-turn conversational search without introducing additional inference overhead.

2 Related Works

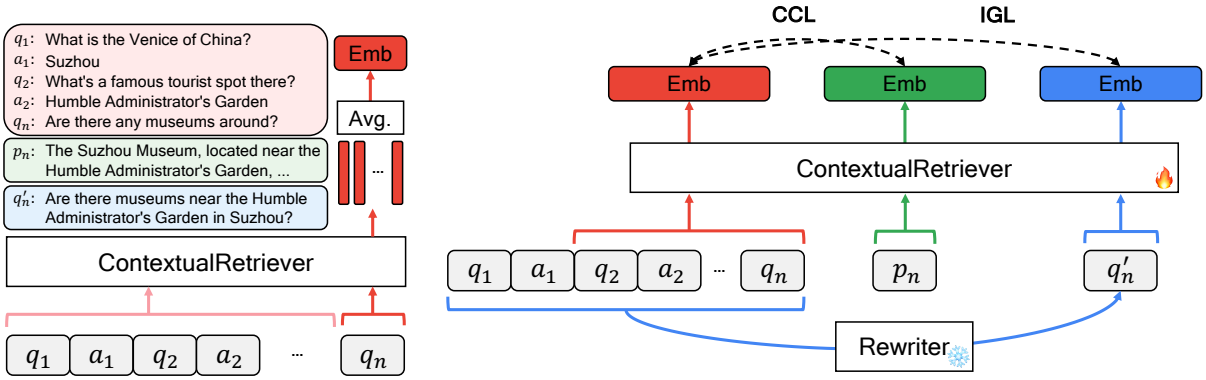
2.1 Dense Retrieval

Information retrieval has evolved from traditional lexical matching methods such as BM25 and TF-IDF (Robertson et al., 2009; Ramos et al., 2003) to dense retrieval approaches (Karpukhin et al., 2020; Khattab and Zaharia, 2020). Dense retrievers encode queries and passages into vector embeddings and perform retrieval based on their similarity.

Early dense retrievers built on BERT (Kenton and Toutanova, 2019) leverage its contextual representation power (Xiao et al., 2023; Wang et al., 2020). More recent approaches (Meng et al., 2024; Lee et al., 2024; Moreira et al., 2024; Li et al., 2024) utilize larger pre-trained LLMs to take advantage of superior language understanding. However, these models are typically trained on isolated query-passage pairs, which limits their ability to understand conversational context. Unified LLM retrievers such as GritLM (Muennighoff et al., 2024) and OneGen (Zhang et al., 2024) attempt to combine generation and retrieval in a single model for efficiency, but they fall short in embedding rich conversational context during retrieval.

2.2 Conversational Search

Most dense retrievers are trained on single-turn settings with clearly stated information needs. In contrast, conversational search introduces challenges such as ambiguity and context dependence. Query rewriting approaches (Lin et al., 2020; Mo et al., 2023) reformulate conversational queries into self-



(a) (b)

Figure 2: **Overview of our `ContextualRetriever`.** (a) `ContextualRetriever` processes the entire conversation history including the current query, but extracts retrieval embeddings specifically focused on the current query. (b) `ContextualRetriever` with the frozen rewriter is trained to align the query embeddings with both relevant passage embeddings and rewritten query embeddings for effective context understanding through Conversational Contrastive Learning (CCL) and User Intent-Guided Learning (IGL).

contained forms, but they incur significant computational overhead due to their reliance on a separate rewriting model.

Recent efforts have aimed to integrate conversational context directly into retrievers. CQE (Lin et al., 2021) and CONVAUG (Chen et al., 2024) generate context-aware embeddings via contrastive learning and data augmentation, respectively, but they do not explicitly model contextual ability. ConvDR (Yu et al., 2021) and DiSCo (Lupart et al., 2024) leverage knowledge distillation from rewritten queries to embed context, though their effectiveness heavily depends on the quality of the rewriting model. Shortcut Dependency (Kim and Kim, 2022) improves retrieval robustness by mitigating shortcut learning from topical cues, but it lacks deeper semantic modeling of dialogue context. ChatRetriever (Mao et al., 2024) leverages LLMs by encoding dialogue history into special tokens, but its performance quickly saturates, yielding marginal gains as context length increases. In contrast, our approach leverages the contextual capabilities of LLM by combining query rewriting–based supervision with generation-based training. This allows the model to encode both explicit intent signals and implicit contextual information without relying on external rewriting modules.

3 Method

3.1 Task Definition

Conversational search (Mo et al., 2024) aims to retrieve relevant passages from a collection $P = \{p_1, \ldots, p_m\}$ for each query in multi-turn dialogues. At the n-th conversation turn, the goal is to retrieve top-k passages for the current query q_n by leveraging the conversation history $\{q_i, a_i\}_{i=1}^{n-1}$, where q_i and a_i denote the query and response at the i-th turn, respectively. The retriever $R(\cdot)$ encodes both passages and queries into a shared embedding space. Each passage is pre-encoded offline, while the current query, together with its conversation history, is encoded during inference. Retrieval is performed by computing the cosine similarity between the query and passage embeddings.

3.2 Construction of Training Set

We introduce a *dynamic dialogue history sampling strategy*, creating varied training instances from conversation histories given a target query-passage pair (q_n, p_n). Specifically, we randomly select a starting point i $(i < n)$ and include all subsequent queries and responses $[q_i, a_i, \ldots, q_n]$ to form training pairs with the relevant passage p_n. This approach (1) augments the training data, (2) exposes the model to diverse context lengths, and (3) improves robustness to varying conversational histories. Our experiments confirm that this strategy significantly boosts the model's capacity to incorporate conversational context and generate high-quality retrieval embeddings (Table 6).

3.3 Retrieval Embedding Extraction

We design our retriever based on a decoder-only LLM architecture, following the previous LLM-based retrievers. As illustrated in Fig. 2(a), while our model takes the entire conversation history as input, it selectively extracts embeddings only from the tokens corresponding to the current query.

This selective extraction enables the model to effectively utilize the conversational history as contextual cues while ensuring the retrieval remains strongly aligned with the intent of the current query. This approach mitigates the risk of excessively prioritizing prior conversation context, which could otherwise hinder retrieval accuracy by overshadowing the immediate query intent. The retrieval embedding for the current query is computed by average pooling the sequence embeddings of the last m elements:

$$e_{q_n} = \text{AvgPool}(\{R([q_i, a_i, \ldots, q_n])_j\}_{j=N-m+1}^{N}), \quad (1)$$

where $R([q_i, a_i, \ldots, q_n])_j$ represents the embedding for the j-th token within the sequence. Here, m and N denote the token length of the current query and input conversation, respectively.

3.4 Training for Conversational Search

3.4.1 Conversational Contrastive Learning

We optimize the retriever to distinguish relevant from irrelevant passages using a contrastive learning objective applied to our constructed conversational dataset:

$$L_{CCL} = -\log \frac{f(q_n, p_n)}{f(q_n, p_n) + \sum_{p_k \in P_n^-} f(q_n, p_k)}, \quad (2)$$

where $f(q_n, p_n) = \exp((e_{q_n} \cdot e_{p_n})/\tau)$ is a similarity function with temperature τ, and P_n^- denotes a set of negative passages for query q_n. This contrastive framework serves two key purposes. First, it shapes the embedding space by pulling relevant query-passage pairs closer and pushing negatives apart. Second, because query embeddings are computed with the full dialogue context, L_{CCL} implicitly encourages the model to encode contextual information that improves retrieval performance.

3.4.2 User Intent-Guided Learning

To further enhance our retriever's ability to capture user intent, we propose an intent-guided learning approach that leverages signals from query rewriting. Our method employs LLMs as a query rewriter $QR(\cdot)$ to generate contextually explicit queries through carefully designed prompts (See Appendix A). The rewriter transforms abbreviated queries into self-contained formats by incorporating relevant context from previous interactions. We introduce an embedding alignment loss that bridges the gap between the embeddings of the original and rewritten queries:

$$L_{IGL} = \left\| e_{q_n} - e_{q'_n} \right\|_2^2, \quad (3)$$

where $q'_n = QR([q_1, a_1, \ldots, q_n])$ represents the rewritten query. While conversational contrastive learning optimizes query-passage relationships, intent-guided learning focuses on aligning query representations with their explicit, context-aware counterparts. As shown in Fig. 2(b), these learning objectives work together to ensure our model leverages comprehensive intent understanding to achieve effective retrieval performance.

3.4.3 Preserving LLM Capabilities

To maintain the rich language understanding capabilities of the base LLM while optimizing for retrieval performance, we introduce a generation-based regularization technique that shares the same computational path with retrieval. Specifically, we employ a next-token prediction loss that encourages the model to preserve its inherent ability to generate contextually appropriate responses:

$$L_G = -\log P(R(a_n) | R([q_i, a_i, ..., q_n, p_n])), \quad (4)$$

where the model predicts the next response a_n given the conversation and relevant passage.

3.4.4 Final Training Objective

Our complete training objective is:

$$L = (1 - \lambda_G)(L_{CCL} + \lambda_{IGL} L_{IGL}) + \lambda_G L_G, \quad (5)$$

where λ_{IGL} and λ_G control the balance among the loss components. We refer to the final retriever trained with this complete objective as **`ContextualRetriever`**.

4 Experiment

4.1 Experimental Setup

Datasets. We evaluate our approach on four widely-used conversational search datasets: TopiOCQA (Adlakha et al., 2022), QReCC (Anantha et al., 2021), TREC-CAsT (Dalton et al., 2020, 2021), and ORConvQA (Qu et al., 2020). All datasets feature multi-turn conversational queries, containing both current queries and conversation history. TopiOCQA contains frequent topic shifts within a conversation, requiring systems to determine whether to maintain or discard prior context. QReCC and ORConvQA are relatively topic-consistent, where the primary challenge is resolv-

Table 1: Retrieval performance comparison: Baseline models with and without our approach.

Method	TopiOCQA		CAsT-19	CAsT-20
	MRR↑	Hit@100↑	nDCG@3↑	
BGE-large	16.1	46.7	36.5	20.1
+ ours	+ 7.8	+ 16.1	+ 13.2	+ 14.8
SFR Embedding	17.8	57.3	32.6	24.5
+ ours	+ 12.4	+ 17.4	+ 8.2	+ 9.6
GritLM	24.3	68.5	30.7	18.2
+ ours	+ 17.9	+ 23.4	+ 31.6	+ 28.6

ing context-dependent expressions such as pronouns and ellipses by referencing previous conversation turns. TREC-CAsT 2019 and 2020 feature evolving user information needs within a controlled experimental setup. Conversations average around 9 to 10 turns and are manually curated to ensure coherence and diversity. We train our `ContextualRetriever` on TopiOCQA's training split and evaluate it on all four datasets: in-domain (TopiOCQA) and out-of-domain (QReCC, TREC-CAsT, ORConvQA). Statistics of each dataset are reported in Appendix B.

Evaluation. We employ standard information retrieval metrics to evaluate retrieval effectiveness, including Mean Reciprocal Rank (MRR), normalized Discounted Cumulative Gain at rank 3 (nDCG@3), and Hit Rate at rank k (Hit@k). MRR captures how early the first relevant document appears in the ranking. nDCG@3 evaluates both the presence and ranking quality of relevant documents within the top-3 results. Hit@k denotes the proportion of queries where at least one relevant document is retrieved within the top-k candidates.

Implementation details. We apply LoRA-based fine-tuning of our method to three different retrievers: BGE-large (Xiao et al., 2023), SFR Embedding (Meng et al., 2024), and GritLM (Muennighoff et al., 2024). As shown in Table 1, our method consistently improves performance across all models, with particularly strong gains when applied to GritLM. We attribute this compatibility to GritLM's joint training objective for generation and retrieval, which aligns naturally with our generation-preserving learning objective. Given this synergy, we select GritLM as the base retriever for our main experiments. We use LoRA with the following hyperparameters: 1 training epoch, batch size of 24, learning rate of 1e-4, LoRA rank of 16, and Adam optimizer. The weights for intent-guided learning (λ_{IGL}) and generation loss (λ_{G}) are set to 1.0 and 0.2, respectively.

4.2 Baselines

Query rewriter. We consider three query rewriting approaches: T5QR (Lin et al., 2020), GritLM (Muennighoff et al., 2024), and GPT-4-Turbo (Achiam et al., 2023). T5QR is a dedicated query rewriting model fine-tuned from the T5-base architecture (Raffel et al., 2020) using the TopiOCQA training set. In contrast, GritLM and GPT-4-Turbo are general-purpose language models that we leverage for prompt-based query rewriting, following recent trends in LLM-driven conversational rewriting (Ye et al., 2023) (see Appendix A).

Retriever. We consider two types of dense retrievers. First, we evaluate BERT-based models, including MiniLM (Wang et al., 2020) and BGE-large (Xiao et al., 2023), which are efficient and strong general-purpose retrievers (Muennighoff et al., 2022). Second, we evaluate LLM-based retrievers, including SFR Embedding (Meng et al., 2024), a retrieval-specialized Mistral-7B model; GritLM (Muennighoff et al., 2024), a unified retriever-generator also based on Mistral-7B; and ChatRetriever (Mao et al., 2024), a conversationally-tuned retriever built on Qwen-7B (Bai et al., 2023). For fair comparison, we re-evaluate all baselines under our evaluation setup.

Baseline configurations and query input types. We compare different query input strategies. In the rewriting setup, a query rewriter takes the conversation history and current query to produce a rewritten query, which is passed to the retriever. Without rewriting, we consider three variants: (1) *Current*: using only the current query; (2) *Window*: the current query with the last three query-response turns; and (3) *Full*: the entire conversation history concatenated with the current query.

4.3 Main Results

Table 2 presents the comparative evaluation of our approach against existing methods. On TopiOCQA, a benchmark known for its challenging topic shifts within conversations, `ContextualRetriever` achieves state-of-the-art performance. On QReCC, our method outperforms most baselines and shows competitive results even against GPT-4-Turbo rewrites. Without requiring a rewriting process, `ContextualRetriever` consistently outperforms GritLM applied to rewritten queries. This indicates that our method effectively leverages rewritten training queries and the inherent generative capability of pre-trained

Table 2: **Retrieval performance (%) of different retrievers and query rewriting approaches on TopiOCQA and QReCC datasets.** Best and second-best results are indicated in bold and underlined, respectively. Human rewrites are not included for TopiOCQA as they are not provided in the original dataset.

Retriever	Query Rewriter	Query Type	TopiOCQA			QReCC		
			MRR$^{\uparrow}$	Hit@20$^{\uparrow}$	Hit@100$^{\uparrow}$	MRR$^{\uparrow}$	Hit@20$^{\uparrow}$	Hit@100$^{\uparrow}$
MiniLM	-	Current	3.7	9.1	13.2	4.2	10.2	14.6
		Window	11.1	25.8	37.5	19.7	62.1	78.1
		Full	10.0	24.7	36.5	19.3	61.5	79.6
	GritLM	Rewritten	19.8	43.6	56.5	21.5	59.9	75.7
	Human		-	-	-	21.1	60.8	77.1
Bge-large	-	Current	4.9	10.7	14.4	4.6	11.1	13.9
		Window	16.1	33.4	46.7	22.3	69.1	83.9
		Full	13.5	29.3	42.0	21.3	67.4	84.0
	T5QR	Rewritten	18.9	42.2	55.3	16.3	49.7	64.2
	GritLM		28.3	53.0	64.1	26.0	73.7	86.2
	GPT-4-Turbo		37.2	71.2	82.5	28.6	80.2	**92.0**
	Human		-	-	-	26.3	73.9	86.8
SFR Embedding	-	Current	6.1	11.3	15.0	5.3	12.8	16.1
		Window	17.8	41.2	57.3	22.8	71.4	87.2
		Full	14.1	31.8	46.9	21.7	68.3	86.8
	T5QR	Rewritten	20.6	44.8	56.6	17.7	52.7	68.4
	GritLM		31.6	57.5	67.7	26.6	74.9	87.9
	GPT-4-Turbo		<u>40.5</u>	<u>76.3</u>	<u>86.6</u>	28.3	79.0	<u>91.6</u>
	Human		-	-	-	27.5	76.4	89.7
GritLM	-	Current	2.2	10.9	11.3	3.7	9.2	12.5
		Window	24.3	53.4	68.5	24.9	72.5	88.2
		Full	20.7	47.7	64.2	24.0	72.8	86.8
	T5QR	Rewritten	23.6	49.3	66.2	14.7	44.4	60.6
	GritLM		31.7	66.4	77.2	26.1	74.1	88.8
	GPT-4-Turbo		35.9	69.5	83.6	26.5	74.5	88.4
	Human		-	-	-	23.2	65.6	80.7
ChatRetriever	-	Full	38.1	71.1	84.2	<u>36.5</u>	<u>82.4</u>	91.4
ContextualRetriever (ours)	-	Full	**42.2**	**81.7**	**91.9**	**36.8**	**82.7**	91.5

LLMs to contextualize retrieval without needing explicit rewriting at inference time. Furthermore, our method clearly surpasses ChatRetriever demonstrating that our objective-driven approach to modeling contextual understanding yields more robust performance.

Impact of query input types. Experimental results reveal substantial performance differences across query input configurations. The *Current* setting performs poorly due to its inability to resolve abbreviations and lack of contextual cues. Comparisons between *Window* and *Full* configurations highlight key trade-offs: *Window* efficiently captures recent context but may overlook long-range dependencies, while *Full* offers broader coverage at the risk of introducing noise from irrelevant turns. Although dataset-specific input tuning can yield marginal improvements, it lacks generality and relies on heuristic decisions. In contrast, `ContextualRetriever` processes the full dialogue holistically and learns to attend to relevant context.

Effectiveness of query rewriting. The impact of query rewriting varies across datasets. In TopiOCQA, rewriting consistently improves retrieval performance by resolving context-dependent references and handling topic shifts. LLM-based rewriters such as GritLM and GPT-4-Turbo perform well, effectively capturing nuanced contextual signals. In contrast, T5QR exhibits limited capability, primarily resolving surface-level references such as pronouns (e.g., replacing "it" with its referent). In QReCC, however, rewriting can degrade performance. This degradation is often caused by over-summarization or loss of critical information during rewriting, which removes details necessary for accurate retrieval. In such cases, preserving the original conversational structure proves more effective than rewriting.

Analysis of retriever performance. Our analysis indicates that preserving the generative capacity of LLMs plays a crucial role in conversational retrieval. Compared to conventional BERT-based

Table 3: **nDCG@3 performance on the TREC-CAsT benchmark.** * indicates the result reported in the original ChatRetriever paper; other results are reproduced. **+ Response** denotes the use of retrieved responses as additional conversational context.

Method	CAsT-19	CAsT-20
Conversational Query Rewriting		
*LLM4CS	51.5	45.5
Dense Retrieval		
Bge-large	36.5	20.1
SFR Embedding	32.6	24.5
GritLM	30.7	18.2
Conversational Dense Retrieval		
*ConvDR	43.9	32.4
*LeCoRE	42.2	29.0
*ChatRetriever	52.1	40.0
ChatRetriever	54.1	38.7
ContextualRetriever (ours)	**62.3**	**46.8**
+ Response	**63.4**	**50.6**

models and LLM-based retrievers trained solely with retrieval objectives (i.e., SFR Embedding), both our model and GritLM incorporate generation loss during training. While all retrievers perform similarly on single-turn queries, regardless of whether the query is original or rewritten, performance gaps widen significantly in multi-turn settings (*Window* and *Full*). This difference is particularly pronounced on TopiOCQA, which involve complex topic shifts.

Performance on TREC-CAsT. On the TREC-CAsT benchmark, `ContextualRetriever` also delivers strong performance. Notably, our method achieves substantial gains over prior approaches (Table 3). It outperforms recent conversational search methods, including LLM-based query rewriting (e.g., LLM4CS (Mao et al., 2023a)) and dense retrievers such as ConvDR (Yu et al., 2021), LeCoRE (Mao et al., 2023b), and ChatRetriever (Mao et al., 2024). This result demonstrates that even in well-structured evaluation settings, our contextual embedding contributes significantly to improved retrieval quality.

In Appendix C and E, further analysis reveals that both our embedding extraction strategy and proposed loss substantially contribute to the performance gains. We also include evaluation results on ORConvQA to confirm the generalizability of our method. Additionally, we report generation performance to validate that our model preserves the language capabilities of the underlying LLM while optimizing for retrieval.

Table 4: Number of parameters (Params.) and inference time for query rewriters and retrievers.

Model	Query Type	Params.	Inference Time
Rewriter			
T5QR	Full	223M	156.5ms
GritLM	Full	7241M	1064.7ms
GPT-4-Turbo	Full	–	1293.6ms
Retriever			
BGE-large	Window	335M	18.4ms
GritLM (Current)	Current	7241M	35.8ms
GritLM (Window)	Window	7241M	44.8ms
GritLM (Full)	Full	7241M	80.3ms
ChatRetriever	Full	7721M	101.4ms
ContextualRetriever	Full	7241M	80.5ms

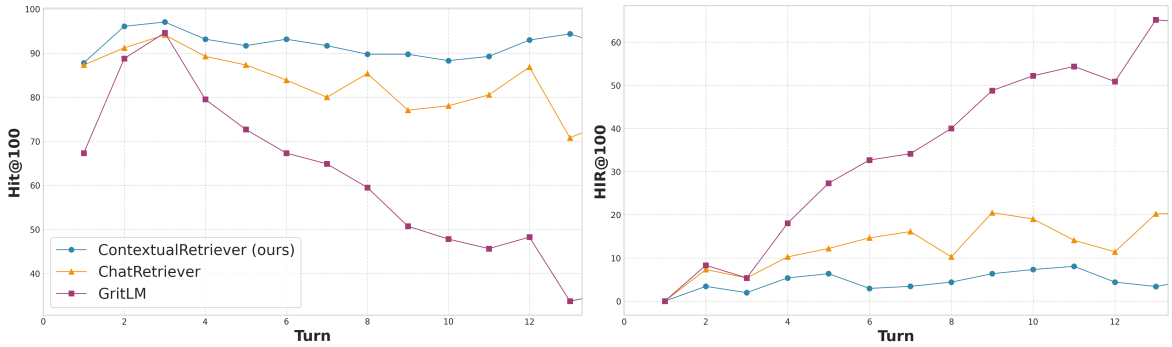
4.4 Analysis

Computational cost. Table 4 summarizes the computational requirements of various query rewriters and retrievers, in terms of model parameters and average inference time. Inference time was measured across 160 samples (10 conversations with 1–16 turns) using an Intel Xeon Gold 6342 CPU and a single NVIDIA RTX A5000 GPU.

Among query rewriters, we observe substantial differences in both model size and efficiency. All rewriters rely on autoregressive decoding for query generation, which introduces significant latency at inference time. This includes T5QR as well as LLM-based models such as GritLM and GPT-4 Turbo (accessed through OpenAI's API services). Larger models generally produce higher-quality rewrites, with GPT-4 Turbo delivering the best performance but also incurring the highest cost due to its increased model complexity.

For retrieval, BGE-large achieves the lowest inference time among all evaluated retrievers, reflecting its compact architecture but also its relatively lower retrieval performance compared to larger models. Both `ContextualRetriever` and ChatRetriever eliminate the need for separate rewriters and provide end-to-end solutions for conversational search. Notably, `ContextualRetriever` further outperforms ChatRetriever in performance, achieving a better balance between effectiveness and efficiency. This supports the advantage of our retriever design in real-world, latency-sensitive applications.

Ability to capture user intent. To evaluate how well models track evolving user intent throughout a dialogue, we conduct a turn-by-turn analysis comparing our method with GritLM and ChatRetriever. This analysis considers not only Hit@k but also whether the retrieved passages reflect the model's ability to isolate the current information need from



(a) (b)

Figure 3: (a) Hit@100 and (b) Historical Interference Rate (HIR@100) of our `ContextualRetriever`, ChatRetriever, and GritLM across conversation turns. HIR@100 measures how often a model retrieves passages related to previous queries rather than the current one.

earlier conversational turns.

Figure 3(a) reports Hit@100 across dialogue turns. Retrieval performance initially improves, as early turns include information that directly supports subsequent queries without significant topic shifts. However, as the conversation progresses, ambiguity and context dependencies accumulate, making retrieval more difficult. GritLM's performance declines sharply, indicating difficulty in maintaining contextual alignment. ChatRetriever is relatively more stable, while our model consistently achieves higher hit rates, especially in later turns where accurate disambiguation becomes critical.

To further analyze this behavior, we compute Historical Interference Rate (HIR@100), shown in Figure 3(b), which measures how often retrieved passages align with ground-truth passages from previous turns rather than the current one. A higher HIR@100 indicates that a model is overly influenced by earlier queries, retrieving outdated or irrelevant content. GritLM exhibits the highest HIR@100, often retrieving passages aligned with dominant earlier topics regardless of their current relevance. This suggests that the model relies on lexical or shallow semantic cues rather than modeling evolving user intent. ChatRetriever performs better, aided by conversational finetuning, but still suffers from interference. In contrast, our model consistently achieves lower HIR@100 across all turns, demonstrating greater robustness in distinguishing the current query from prior context.

This behavioral distinction is critical. While prior methods may appear context-aware, they often depend on memorization or anchoring to previously relevant contexts. Our model more faithfully tracks shifting user intent, enabling adaptive retrieval even in semantically entangled conversations. These findings reinforce our core design intuition: optimizing for intent-aware representations yields models that are not only accurate but also resilient to context interference and shortcut behaviors. These properties are particularly important in multi-turn settings where user goals evolve continuously.

5 Conclusions

We introduced `ContextualRetriever`, a unified retriever that generates context-aware embeddings without relying on external query rewriting. Our method integrates user intent understanding directly into the retrieval process by leveraging both conversational context and generation loss during training. Through extensive evaluations on four benchmark datasets, we demonstrate that `ContextualRetriever` not only improves retrieval accuracy but also generalizes well across diverse conversational styles and structures. These results suggest that integrating intent modeling within the retriever itself provides a scalable and robust solution for multi-turn conversational search.

6 Limitations

Our current implementation adopts LoRA-based parameter-efficient tuning due to resource constraints. While this setup yields strong performance, full-model fine-tuning at scale may further improve generalization and stability. Extending `ContextualRetriever` to support full-parameter optimization, along with training on more diverse topic shift scenarios, could lead to a more versatile conversational retriever capable of handling single-turn, multi-turn, and generative tasks. We leave this as a promising direction for future work.

References

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

Vaibhav Adlakha, Shehzaad Dhuliawala, Kaheer Suleman, Harm de Vries, and Siva Reddy. 2022. TopiOCQA: Open-domain conversational question answering with topic switching. *Transactions of the Association for Computational Linguistics*, 10:468–483.

Raviteja Anantha, Svitlana Vakulenko, Zhucheng Tu, Shayne Longpre, Stephen Pulman, and Srinivas Chappidi. 2021. Open-domain question answering goes conversational via question rewriting. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Haonan Chen, Zhicheng Dou, Kelong Mao, Jiongnan Liu, and Ziliang Zhao. 2024. Generalizing conversational dense retrieval via llm-cognition data augmentation. *arXiv preprint arXiv:2402.07092*.

Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2020. Trec cast 2019: The conversational assistance track overview. In *In Proceedings of TREC*.

Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2021. Cast 2020: The conversational assistance track overview. In *In Proceedings of TREC*.

Jianfeng Gao, Chenyan Xiong, Paul Bennett, and Nick Craswell. 2023. *Neural approaches to conversational information retrieval*. Springer.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880. Association for Computational Linguistics.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2. Minneapolis, Minnesota.

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

Sungdong Kim and Gangwoo Kim. 2022. Saving dense retriever from shortcut dependency in conversational search. *arXiv preprint arXiv:2202.07280*.

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, MingHao Qin, Shitao Xiao, Jianlyu Chen, Kun Luo, Yingxia Shao, Defu Lian, and Zheng Liu. 2024. Making text embedders few-shot learners.

Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2021. Contextualized query embeddings for conversational search. In *Findings of the Association for Computational Linguistics: EMNLP 2023*.

Sheng-Chieh Lin, Jheng-Hong Yang, Rodrigo Nogueira, Ming-Feng Tsai, Chuan-Ju Wang, and Jimmy Lin. 2020. Conversational question reformulation via sequence-to-sequence architectures and pretrained language models. *arXiv preprint arXiv:2004.01909*.

Simon Lupart, Mohammad Aliannejadi, and Evangelos Kanoulas. 2024. Disco meets llms: A unified approach for sparse retrieval and contextual distillation in conversational search. *arXiv preprint arXiv:2410.14609*.

Kelong Mao, Chenlong Deng, Haonan Chen, Fengran Mo, Zheng Liu, Tetsuya Sakai, and Zhicheng Dou. 2024. Chatretriever: Adapting large language models for generalized and robust conversational dense retrieval. *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.

Kelong Mao, Zhicheng Dou, Fengran Mo, Jiewen Hou, Haonan Chen, and Hongjin Qian. 2023a. Large language models know your contextual search intent: A prompting framework for conversational search. *arXiv preprint arXiv:2303.06573*.

Kelong Mao, Hongjin Qian, Fengran Mo, Zhicheng Dou, Bang Liu, Xiaohua Cheng, and Zhao Cao. 2023b. Learning denoised and interpretable session representation for conversational search. In *Proceedings of the ACM Web Conference 2023*.

Rui Meng, Ye Liu, Shafiq Rayhan Joty, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. 2024. Sfrembedding-mistral: enhance text retrieval with transfer learning. *Salesforce AI Research Blog*, 3.

Fengran Mo, Kelong Mao, Ziliang Zhao, Hongjin Qian, Haonan Chen, Yiruo Cheng, Xiaoxi Li, Yutao Zhu, Zhicheng Dou, and Jian-Yun Nie. 2024. A survey of conversational search. *arXiv preprint arXiv:2410.15576*.

Fengran Mo, Kelong Mao, Yutao Zhu, Yihong Wu, Kaiyu Huang, and Jian-Yun Nie. 2023. Convgqr: generative query reformulation for conversational search. *Association for Computational Linguistics*.

Gabriel de Souza P Moreira, Radek Osmulski, Mengyao Xu, Ronay Ak, Benedikt Schifferer, and Even Oldridge. 2024. Nv-retriever: Improving text embedding models with effective hard-negative mining. *arXiv preprint arXiv:2407.15831*.

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

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

Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W. Bruce Croft, and Mohit Iyyer. 2020. Open-Retrieval Conversational Question Answering. In *SIGIR*.

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

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

Cunxiang Wang, Sirui Cheng, Qipeng Guo, Yuanhao Yue, Bowen Ding, Zhikun Xu, Yidong Wang, Xiangkun Hu, Zheng Zhang, and Yue Zhang. 2023. Evaluating open-QA evaluation. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.

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

Fanghua Ye, Meng Fang, Shenghui Li, and Emine Yilmaz. 2023. Enhancing conversational search: Large language model-aided informative query rewriting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5985–6006, Singapore. Association for Computational Linguistics.

Shi Yu, Zhenghao Liu, Chenyan Xiong, Tao Feng, and Zhiyuan Liu. 2021. Few-shot conversational dense retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on research and development in information retrieval*, pages 829–838.

Jintian Zhang, Cheng Peng, Mengshu Sun, Xiang Chen, Lei Liang, Zhiqiang Zhang, Jun Zhou, Huajun Chen, and Ningyu Zhang. 2024. Onegen: Efficient one-pass unified generation and retrieval for llms. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4088–4119.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19724–19731.

Appendix

Learning Contextual Retrieval for Robust Conversational QA

A Prompt Template for Query Rewriting

We utilize rewritten queries for intent-guided learning, as described in Eq. 3. To generate accurate rewritings during training, we leverage both the gold context and the gold response. For each training instance, we employ the GPT-4-Turbo model with the following prompt template. Note that the prompt includes three few-shot examples, which are manually selected. The rewritten queries in these examples were generated by human annotators.

Prompt template for query rewriting for training set

```
Given a previous conversation, a current question,
related context regarding the current question,
and a ground truth response, your task is to
rewrite the current question to make it clearer
and more explicit. In your rewrite, please
avoid using pronouns or any abbreviated terms.
The aim is to ensure that the current question
stands alone, so that LLM can get the related
context and arrive at the ground truth response
without needing additional information.
Do not use any additional comments such as
"Here is a rewritten version of the current
question:". Only generate a rewritten question.

Examples: {Example1} {Example2} {Example3}

Previous Conversation: {Previous conversation}
Current Question: {Question}
Context: {Context}
Ground Truth Response: {Gt_response}
Output:
```

During evaluation, we adopt two prompt-based query rewriters: GritLM and GPT-4-Turbo. Unlike in training, gold context and gold responses are not available at test time. Therefore, rewriting must be performed solely based on the prior conversation history and the current user question. The prompt template used for GritLM and GPT-4-Turbo during evaluation is provided below:

Prompt template for query rewriting in evaluation

```
Given a previous conversation and a current
question, your task is to rewrite the current
question to make it clearer and more explicit.
In your rewrite, please avoid using pronouns
or any abbreviated terms. The aim is to ensure
that the current question stands alone, so that
the retriever can get the related context.
If the original question is already clear,
you can use the original question.

Example: {Example1} {Example2} {Example3}

Previous Conversation: {Previous conversation}
Current Question: {Question}
Rewrite Output:
```

For GritLM, we enclose the prompt with <|user|> token and add <|assistant|> token before the output.

B Evaluation Datasets

We summarize the statistics of the evaluation datasets in Table 5. Our model is trained on the TopiOCQA training split and evaluated on the development sets of all datasets.

Table 5: Statistics of the datasets, including the number of conversations (C), queries (Q), and passages (P).

Statistics	TopiOCQA		QReCC	CAsT-19	CAsT-20	ORConvQA
	Train	Test	Test	Test	Test	Test
C	3,509	205	2,775	50	25	490
Q	45,450	2,514	16,451	479	208	3,430
P	25M		54M	38M		11M

C Ablation Studies

All ablation results are reported on a sampled small passage set (5% of the full dataset), which differs from the main evaluation setup. This allows efficient comparison while preserving relative performance trends.

C.1 Effect of dialogue history sampling

As shown in Table 6, our dialogue history sampling strategy (Sec. 3.2) significantly outperforms the baseline that uses the original training set without augmentation, under the same number of training epochs. These results highlight the effectiveness of our method in exposing the model to diverse conversational contexts and better capturing user intent.

Table 6: Performance comparison (%) between baseline and our sampling strategy.

Sampling strategy	TopiOCQA	
	nDCG@3	Hit@5
Baseline	39.3	80.9
Dialogue history sampling (ours)	45.5	88.3

C.2 Impact of embedding extraction methods

We analyze our embedding extraction approach described in Sec. 3.3, which encodes the full conversation but uses only the current query's embeddings for retrieval. Table 7 shows that using all output embeddings leads to performance degradation, despite

access to full context. This validates our choice to maintain query-focused representations while still leveraging broader dialogue context during encoding.

Table 7: Performance comparison (%) of different retrieval embedding extraction methods.

Retrieval Embeddings	Retriever	TopiOCQA	
		nDCG@3	Hit@5
Full conversation	GritLM	15.9	41.5
Current query-focused	GritLM	29.2	64.0
	ContextualRetriever	45.5	88.3

C.3 Contribution of learning objectives

Table 8 reports the incremental impact of each learning objective. Adding Conversational Contrastive Learning (CCL) to the base configuration substantially improves performance in multi-turn settings. Further gains are observed by incorporating Intent-Guided Learning (IGL) and the Generation Loss (G). These findings confirm that each component contributes to more effective retrieval representations for conversational queries.

Table 8: Impact of different learning components.

Method	TopiOCQA	
	nDCG@3	Hit@5
GritLM	23.3	58.0
$+ L_{CCL}$	40.8	79.0
$+ L_{CCL} + L_{IGL}$	42.0	83.5
$+ L_{CCL} + L_{IGL} + L_G$	45.5	88.3

D Generation Performance

Our approach adopts a unified LLM architecture trained with an integrated generation loss, which preserves the model's ability to generate fluent and contextually relevant responses. To evaluate this capability, we compare our ContextualRetriever with GritLM using the following generation prompt:

Prompt template for generation

```
<|embed|>\n{Query}\n<|user|>\n {Context}
Optionally using the prior conversation
and context, answer the last query:
{Current Query}\n<|assistant|>\n
```

D.1 Evaluation Methodology

We assess generation performance using gold contexts with two metrics: (1) **Lexical Matching** (Wang et al., 2023; Izacard and Grave, 2021), which measures whether the generated answer contains any reference answer span; and (2) **Correctness** (Zhong et al., 2024), which uses GPT-4-Turbo to score semantic correctness with the following evaluation prompt:

Prompt template for measuring correctness

```
Evaluates if the response contains the correct
answer to the probing question
(labels: (0 : wrong, 0.5 : partial, 1 : correct)
You should return only digit 0, 0.5, or 1.
response: {response} answer: {answer}
```

D.2 Results and Analysis

As shown in Table 9, our model achieves superior generation accuracy compared to GritLM, validating the effectiveness of our approach in maintaining robust generation capabilities.

Table 9: Generation performance of Unified LLM.

Model	Gen. performance (%)	
	Lexical Matching	Correctness
GritLM	27.6	60.5
ContextualRetriever	31.7	70.3

Notably, our unified architecture enables efficient cache sharing between retrieval and generation. During inference, the query representations computed for retrieval can be reused for generation without incurring additional computational cost.

E Results on ORConvQA

We evaluate our model against various baselines on the ORConvQA dataset, which is characterized by consistent topic maintenance without shifts. Our experiments reveal that *Full* setting achieves the best performance due to the dataset's preference for comprehensive information preservation, while query rewriting approaches showed lower performance due to information loss. Notably, the BERT-based retriever (Bge-large) demonstrates strong performance on this dataset, primarily due to the strong correlation between previous conversations and current queries, and limited requirement for complex user intent understanding. Despite GritLM's relatively lower baseline performance, our approach achieves state-of-the-art results, outperforming both existing baselines and ChatRetriever across different query configurations.

Table 10: Retrieval performance (%) of different retrievers and query rewriting approaches on ORConvQA.

Retriever	Query Rewriter	Query Type	ORConvQA	
			nDCG@3	Hit@5
MiniLM	-	Current	8.2	16.2
		Window	46.5	74.1
		Full	62.5	90.0
	GritLM	Rewritten	40.6	69.0
Bge-large	-	Current	12.3	20.6
		Window	60.2	88.5
		Full	71.4	<u>97.0</u>
	T5QR	Rewritten	44.0	68.8
	GritLM		56.0	84.3
	GPT-4		65.4	94.2
	Human		54.3	82.9
SFR Embedding	-	Current	11.7	20.1
		Window	62.7	88.9
		Full	72.9	94.4
	T5QR	Rewritten	40.0	65.5
	GritLM		48.9	77.8
	GPT-4		59.0	89.3
	Human		50.5	79.7
GritLM	-	Current	10.8	17.4
		Window	59.1	84.4
		Full	**73.1**	95.2
	T5QR	Rewritten	36.9	59.2
	GritLM		46.1	71.7
	GPT-4		55.5	82.8
	Human		45.5	71.1
ChatRetriever	-	Full	70.6	95.0
ContextualRetriever (ours)	-	Full	<u>73.0</u>	**97.8**