Dense Query Rewriting with Retrieval Loss for Conversational Search

Anonymous ACL submission

Abstract

Conversational search aims to construct proper query in conversational settings to retrieve relevant knowledge as a user question is often inadequate for retrieval purpose. Existing approaches typically employ generative language models for reformulating user questions into search queries relying on human rewrite queries as an ideal solution. However, these approaches may yield suboptimal retrieval results due to the independent learning of reformulation models and retrievers, leading to discrepancy between the training and target objectives. Building on recent advancements in dense retrieval, we propose a novel query rewriting model that represents a query as dense vectors, considered as an reformulated input for a dense retriever. This approach allows the model to be optimized directly from the retrieval objective. Moreover, our model does not require golden queries for the training, ensuring it adaptable to environments where human-rewritten queries are unavailable. Experimental results on widely used benchmark datasets demonstrate the superiority of the proposed model, achieving outstanding performances. The comprehensive analysis offers valuable insights into the development of dense query rewriting models.

1 Introduction

With the evergrowing abilities of recent large language models (LLMs), conversational questions answering has gained increasing attention, emphasizing the understanding of users' question intent within a conversational history. Unlike traditional information retrieval (IR), conversational search (Gao et al., 2022) specifically addresses reformulating a user question into a search query based on multi-turn conversational histories (Anantha et al., 2021). A user question in a conversation is often not suitable for retrieval purpose because of pronouns or ellipsis appearing where the actual meanings are in the previous turns. Therefore they have to be well resolved to retrieve knowledge that helps providing a proper answer.

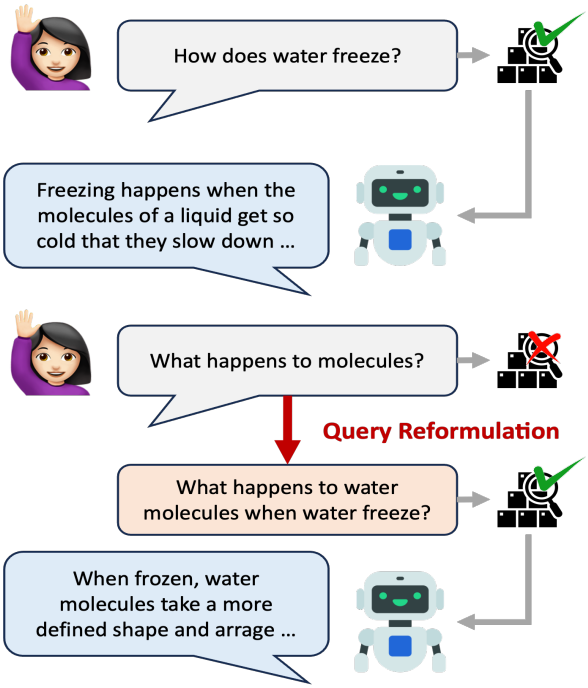


Figure 1: A user question often contain pronouns or ellipsis that degrade retrieval performances. The question has to be reformulated as a stand-alone query that fully contains users' search intent.

Existing methods for conversational search can be divided into two main approaches; training specific retriever models for conversational search or training query reformulation models. The former approach trains retrievers to handle whole conversational contexts (Qu et al., 2020; Hashemi et al., 2020; Yu et al., 2021; Kim and Kim, 2022). While this method effectively represents a question within its conversational context, training a retriever from the scratch requires substantial cost. Additionally, some users may only have access to off-the-shelf retrievers designed for stand-alone queries. To alleviate the issue, the latter approach focuses on reformulating context-dependent questions into de-

contextualized queries. For example, in Figure 1, the user's second question *"Did he sing well"* contains the pronoun *"he"* which has to be resolved by referring to *"Gary Fransis Caine Cherone"* from the previous contexts. The query reformulation model enriches the semantic meaning of the user's question, enabling the reformulated query to be independently expressive and compatible with off-the-shelf retrievers.

The most widely studied query reformulation strategies are query rewriting and query expansion. Query rewriting usually involves training generative models using human rewrite queries as gold-standard labels. However, collecting human rewrite queries is both expansive and time-consuming. Furthermore, prior studies have demonstrated that even human rewrite queries do not always guarantee the optimal retrieval results (Lin et al., 2021; Wu et al., 2022). This results in the inherent limitation of the generative query rewriting models. On the other hand, query expansion attempts to resolve the limitations of human rewrite query by expanding a query with relevant terms, such as potential answer, considering entities previously mentioned. Nevertheless, both strategies still suffer from suboptimality because of the mismatch between the objectives of query reformulation models and retrievers. The query reformulation models are trained with language modeling loss to generate terms based on given human rewrite queries as labels, whereas the retrievers are usually optimized by ranking loss to place relevant passages, best matching the query, at the top through either sparse or dense representations: terms or vectors. Moreover, the generative models cannot be trained when there are no available human rewrite queries.

In this work, we introduce a **Dense Q**uery **R**ewriting model (DenseQR) that constructs vector representations of an input query which are subsequently fed into a dense retriever, replacing traditional token embeddings of the query. Unlike decoder-based generative models, an encoder-based pretrained language model (PLM) is employed for the query rewriting to meet the vocabulary distributions of the rewriter and retriever. Since the encoder is specialized to reconstruct input sequences through the masked language modeling (MLM) objective, the final representations of the model effectively capture the semantic meaning of the input as well as maintain original meanings of tokens. The model is then optimized using contrastive loss between queries and passages, aligned with the objective of training the dense retriever. Experimental results prove that the proposed model substantially outperforms traditional generative query rewriting models in terms of retrieval. Notably, our model achieves state-of-the-art performance with the dense retriever, even on the dataset where a term-based sparse retriever have steadily resulted in better performances. Our contributions are summarized as follows:

- We propose a novel model that rewrites queries as dense vectors, allowing the model to be trained by the retrieval objective. This is the first study that directly trains the query rewriting model through the retrieval objective as far as we know.
- Experimental results on the widely used benchmark datasets demonstrate the superiority of the proposed model compared to the prior studies. The performances of our model achieve the state-of-the-art results.
- Our in-depth analyses provide an insight about dense query rewriting through the empirical experiments.

2 Related Work

2.1 Conversational Question Answering

Conversational question answering (CQA) is a task that aims to answer questions within a conversational setting. Due to the linguistic phenomenon such as pronouns and ellipsis, CQA is more challenging compared to traditional machine reading comprehension (MRC) task. For CQA, QReCC (Anantha et al., 2021) and TopiCQA (Adlakha et al., 2022) datasets are widely used. Both datasets provide question and answer pairs for CQA. QReCC is dataset inspired by query reformulation and hence provides human rewrite queries for retrieval purpose. TopioCQA highlights topic shifts within a conversation. About 28% of turns require contextually different documents with respect to their previous turns because of the topic shifts.

2.2 Conversational Query Reformulation

The goal of the conversational query reformulation is to convert a user question dependent on conversational history into a stand-alone query so that the reformulated query can be directly applied to off-the-shelf retrievers without additional training. In

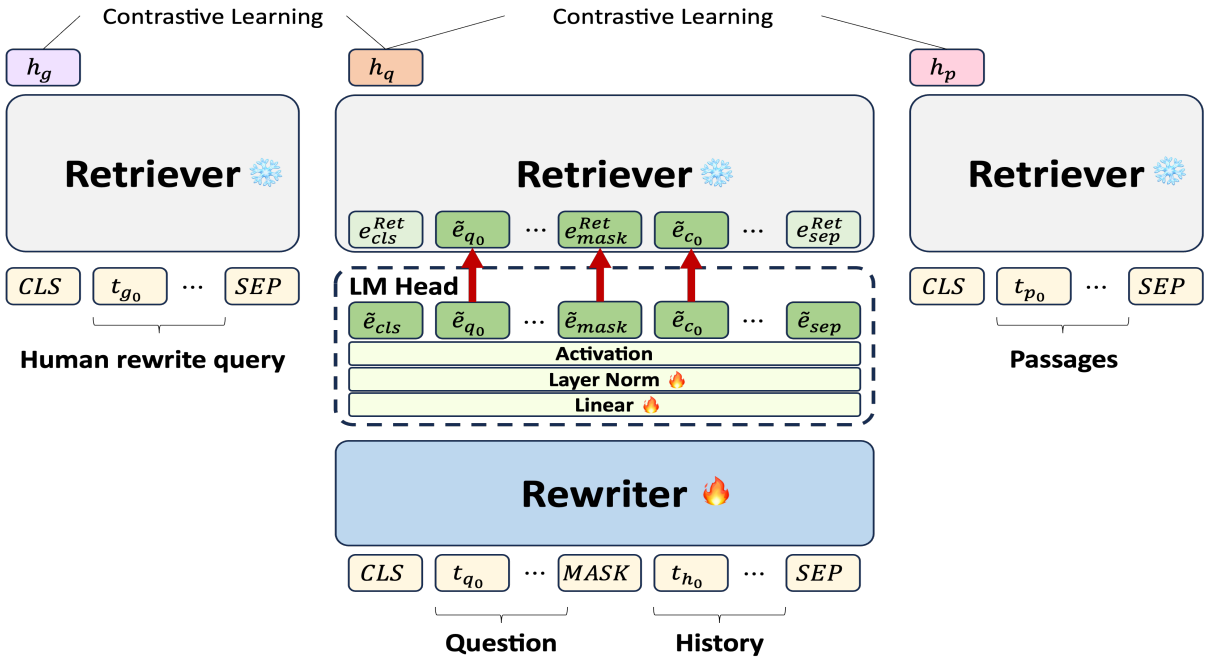


Figure 2: The overview of the proposed model. The reconstructed token embeddings from the rewriter goes through the retriever except for special tokens. It is important to note that the last layer in the LM-head, i.e., transposed token embeddings, is not used for neither the training nor the inference.

early-stage studies, query rewriting method training generative language models to mimic human rewrite queries was proposed. However, it has been shown that given human rewrite queries might not be the optimal queries (Lin et al., 2021; Wu et al., 2022). To address this problem, query expansion method emerged that selects useful terms from conversational context (Voskarides et al., 2020; Kumar and Callan, 2020). Even the query expansion method proved its effectiveness in performance, it requires an additional term selection model, which are not directly optimized for retrieval signals. Wu et al. (2022) leverage reinforcement learning with rewards derived by judgement on generated candidate queries whether they have positive impact on retrieval or not. Jang et al. (2024) enhances the retriever signal by employing a real-valued reward instead of the binary reward used by Wu et al. (2022). Mo et al. (2023) integrate both query writing and query expansion for better conversational query reformulation. Although these studies incorporate retrieval signals for generating proper terms but they are also restricted by sub-optimality.

3 Methodology

In conversational search, questions in a conversation have to be accurately rewritten into standalone queries to effectively leverage off-the-shelf retrievers. However, such rewritten queries in text do not always guarantee optimal retrieval results, as they may contain noisy terms or phrases that are misaligned with the retriever's interpretation. Consequently, it is crucial to construct standalone queries that actually leverage retrieval performances.

Given that input queries for dense retrievers are represented as high-dimensional vectors (a.k.a dense representations or embeddings), we design a model that also reformulates queries as dense vectors, allowing the model to be directly optimized by retrieval loss through the dense vectors. The specifics of the model are described in the following subsections.

3.1 Dense Query Rerwriting

Dense retrievers primarily aim to construct contextually well represented vectors that ensure a high vector similarity between a query and its relevant passages. Typically, a single query (or passage) is represented by one high-dimensional dense vector, which is the result of combining several input token embeddings through deep layers of self-attention. In this manner, dense retrievers do not require their input to be in natural language form.

Meanwhile, the learning objective of encoder-based pretrained language models (PLMs) is to

reconstruct masked tokens into their original form. Input tokens are represented as embeddings, which are progressively transformed into hidden states during the encoding process to capture the semantics of the input. The language model's LM-head then reconstructs these hidden states back into tokens. Since the the last layer of LM-head shares parameters with the embedding layer, the hidden states passing through the dense layer—just before the last layer in LM-head—closely resemble their original token embeddings. These vectors are expected to represent the semantic meaning as well as the meaning of each token, making them suitable for use as inputs to dense retrievers.

While previous studies have demonstrated the positive effects of query expansion, we cannot predict which specific terms will enhance retrieval results across diverse queries without training additional expansion models. To leverage the MLM objective, we concatenate a fixed number of *mask* tokens with an input query, expecting these tokens to function as expanded terms once processed through the encoding layers. The expanded query is then concatenated with its corresponding conversational history and tokenized, allowing it to be mapped to the embeddings in the rewriter:

$$x_q = \texttt{[CLS]}\ t_{q_0}, ..., \texttt{[MASK]},\ t_{c_0} ...,\ \texttt{[SEP]}, \quad (1)$$

$$E_q^{Rwt} = \texttt{Embedding}^{Rwt}(x^q). \quad (2)$$

The embeddings, $E_q^{Rwt} \in \mathbb{R}^{|q| \times D}$ where $|q|$ is the length of the input and D is the dimension of each embedding, are fed into the rewriter, outputting hidden states, $H_q^{Rwt} \in \mathbb{R}^{|q| \times D}$, that contain semantic information of the input. Finally, the hidden states go through in order of dense, normalization, and activation layer, resulting in reconstructed token embeddings $\tilde{E}_q^{Rwt} \in \mathbb{R}^{|q| \times D}$:

$$H_q^{Rwt} = \texttt{Rewriter}(E_q^{Rwt}), \quad (3)$$

$$\tilde{E}_q^{Rwt} = \sigma(\texttt{LayerNorm}(H^q \cdot W_D^\top)), \quad (4)$$

where $W_D^\top \in \mathbb{R}^{D \times D}$ and σ denote the trainable parameters of the dense layer and gelu non-linear activation function. The reconstructed embeddings are then used as an input of a dense retriever except for start and separation tokens, i.e., *cls* and *sep*, as shown in Figure 2. Since the special tokens always appear when the dense retriever is fine-tuned (especially the hidden state of *cls* token is used for calculating vector similarity), it may hurt the stability of the model if these token embeddings are dynamically changed across the inputs.

3.2 Advanced Contrastive Learning

The input dense embeddings of the query for the retriever are constructed by concatenating the special tokens' embeddings of the retriever and the reconstructed embeddings from the rewriter:

$$E_q^{Ret} = [e_{cls}^{Ret}; \tilde{E}_{1:-1}^{Rwt}; e_{sep}^{Ret}]. \quad (5)$$

The retriever encodes the input and outputs a dense representation of the query, corresponding to the last hidden state of *cls* token. Similarly, a passage representation, used for contrastive learning, is obtained by the retriever but without using the rewriter:

$$h_q = \texttt{Retriever}(E_q^{Ret})_{cls}, \quad (6)$$

$$h_p = \texttt{Retriever}(x_p)_{cls}. \quad (7)$$

Finally, a similarity between the query and the passage is calculated by the inner product, denoted as f, of the two dense vectors and is used for calculating retrieval loss. The rewriter is then optimized by the contrastive loss which is directly backpropagated from the retriever through the reconstructed token embeddings:

$$\mathcal{L}_p = -\log \frac{e^{f(h_q \cdot h_{p^+})}}{e^{f(h_q \cdot h_{p^+})} + \sum_{p_i \in \mathcal{N}} e^{f(h_q \cdot h_{p_i})}}, \quad (8)$$

where p^+ and $\mathcal{N}$ indicate a positive passage and a set of negative passages, respectively. In addition to the typical contrastive loss based on query-passage representation pairs, various contrastive losses are adopted to elaborately rewrite queries. The specifics of the losses are described in the following paragraphs.

Guidance contrastive loss Ideally, the goal of the rewriter is to reconstruct token embeddings of an input question that closely resemble those of an optimal query. While human rewrite queries are not optimal, they can provide valuable guidance for the training process. Accordingly, a guidance loss, $\mathcal{L}g$, is incorporated using the same contrastive approach that leverages input queries and their corresponding human rewrite queries if available. Given the lack of hard negative samples for human rewrite queries, we utilize natural in-batch negative samples instead. The formula follows a structure similar to Eq.8, with the key distinction being the substitution of h_p with h_g which denotes a representation of a human rewrite query.

Identifying contrastive loss Motivated by the quantization-based techniques (Zeng et al., 2024; Babenko and Lempitsky, 2014; Chen et al., 2010), we utilize identified dense representations to compute an additional contrastive loss, aiming to propagate the ability of the dense retriever as much as possible to the rewriter. Since retrieval outputs of dense retrievers can vary due to minor differences in inputs, such as a single word or an element of a dense representation, it is crucial for the dense retrievers to map the query and its relevant passages into closely aligned vector spaces. Although dense representations are challenging to interpret, it is expected that representations encoded from inputs sharing common words or similar contextual meanings will exhibit similarity. In other words, the representations can be identified by subtracting a common representation from these similar ones. Using both positive and negative samples, we compute an average representation of passages, denoted as m_p, and subtract it from all representations within a mini-batch (including the query representation) as follows:

$$u_q = h_q - m_p, \tag{9}$$

$$u_p = h_p - m_p. \tag{10}$$

Then, the new representations u replace the h in Eq.8, resulting in an identifying loss $\mathcal{L}_u$. Finally, the overall loss for training the rewriter is joint sum of the diverse contrastive losses:

$$\mathcal{L} = \mathcal{L}_p + \mathcal{L}_g + \mathcal{L}_u. \tag{11}$$

It is noteworthy that only the rewriter is trained and the retriever is fixed because the retriever is already optimized to well capture the semantics of input queries and passages by its own training algorithm using a typical IR dataset. Therefore, continually training the retriever with different algorithm on new datasets, relatively small compared to the IR dataset, may cause unintended performance decreases.

4 Experiments

Dataset For the fair comparison, we conducted experiments on QReCC (Anantha et al., 2021) and TopiOCQA (Adlakha et al., 2022) datasets following the prior studies. Questions in both datasets are paired with passages which are necessary for answers at each turn. QReCC dataset consists of 14K open-domain English conversations with 80K question-answer pairs and about 54M passage collections. Note that human rewrite queries are exclusively available for the QReCC dataset. TopiOCQA dataset includes 3.9K conversations with 51k question-answer pairs and about 25.7M passages. Compared to QReCC dataset, conversations of TopiOCQA are longer and introduce more complexities in query reformulation.

Evaluation Metrics We evaluated the retrieval performance of our model with four evaluation metrics as widely used in earlier studies: mean reciprocal rank (MRR), NDCG@3 Recall@10, Recall@100. The higher value indicates the better performance for all the metrics.

Baselines We compared the proposed model with following baselines: (1) **Raw** used a question of current turn as an input query. (2) **GPT2QR** (Anantha et al., 2021) trained GPT-2 based Transformer++ model with QReCC train dataset that generates rewritten queries. (3) **QuReTec** (Voskarides et al., 2020) expanded a query of current turn with relevant terms from corresponding conversation history. (4) **T5QR** (Lin et al., 2020) trained seq-to-seq T5 model to generate human rewrite queries using a conversational history and a current question as an input. (5) **ConQRR** (Wu et al., 2022) introduced reinforcement learning, enabling the T5 model to be trained using a retriever-based rewards. (6) **ConvGQR** (Mo et al., 2023) combined query rewriting and query expansion through knowledge infusion mechanism. (7) **IterCQR** (Jang et al., 2024) iteratively trained a query reformulation model with refined retriever-based rewards.

Implementation Details The retriever was initialized from ANCE (Xiong et al., 2021) of which backbone model is RoBERTa (Liu et al., 2019). Consequently, the rewriter also used RoBERTa initialization to meet the vocabulary space of the retriever. We implemented both models in PyTorch (Paszke et al., 2019) and Huggingface `Transformer` library (Wolf et al., 2020). More details about the training parameters are described in Appendix A.

4.1 Main Results

The retrieval performance of the proposed model demonstrates a significant improvement over existing methods across most evaluation metrics, as detailed in Table 1. Specifically, DenseQR outperforms previous results obtained from the dense

Type	Method	QReCC				TopiOCQA			
		MRR	NDCG@3	R@10	R@100	MRR	NDCG@3	R@10	R@100
Sparse (BM25)	Raw	6.5	5.5	11.1	21.5	2.1	1.8	4.0	9.2
	Human rewrite	39.4	37.2	64.2	98.0	-	-	-	-
	GPT2QR	30.4	27.9	50.5	82.3	6.2	5.3	12.4	26.4
	QuReTec	34.0	30.5	55.5	86.0	8.5	7.3	16.0	31.3
	T5QR	33.4	30.2	53.8	86.1	11.3	9.8	22.1	44.7
	ConQRR	38.3	-	60.1	**88.9**	-	-	-	-
	ConvGQR	44.1	41.0	64.4	85.5	12.4	10.7	23.8	45.6
	IterCQR	46.7	44.1	64.4	85.5	16.5	14.9	29.3	54.1
Dense (ANCE)	Raw	7.3	6.0	11.5	17.2	3.8	3.8	7.4	13.4
	Human rewrite	38.5	36.6	60.1	79.3	-	-	-	-
	GPT2QR	33.9	30.9	53.1	72.9	12.6	12.0	22.0	33.1
	QuReTec	35.0	32.6	55.0	72.9	11.2	10.2	20.2	34.4
	T5QR	34.5	31.8	53.1	72.8	23.0	22.2	37.6	54.4
	ConQRR	41.8	-	65.1	84.7	-	-	-	-
	ConvGQR	42.0	39.1	63.5	81.8	25.6	24.3	41.8	58.8
	IterCQR	42.9	40.2	65.5	84.1	26.3	25.1	42.6	62.0
	DenseQR (Ours)	**47.0**	**45.8**	**68.4**	82.4	**32.3**	**32.1**	**53.7**	**74.0**

Table 1: Retrieval results both on sparse (BM25) and dense (ANCE) retriever. Note that our model is not available to sparse retriever but the results are reported for the comparison. The higher value indicates the better performance in all metrics. The best results are **bolded**, and the second-best results are underlined.

retriever by $4.1\%p$ on the QReCC dataset and $6.0\%p$ on the TopiOCQA dataset in terms of MRR. Although sparse retrievers have consistently surpassed dense retrievers on the QReCC dataset, it is noteworthy that our model exceeded the previous best results with the sparse retriever except for Recall@100. The lower performance in Recall@100 may be attributed to the presence of specific words in rewritten queries. Some existing models, trained through reinforcement or iterative learning, may rely on silver bullets such as peculiar words—neither present in the conversational history nor contextually similar—that help leverage lower-ranked relevant passages. However, these words appear insufficient to significantly improve overall ranking scores. In contrast, our model rewrites queries as dense vectors by directly learning errors of the retriever that ensures the representations are aligned with the retriever. Consequently, this approach results in higher rankings for relevant passages rather than simply boosting low-ranked ones, which can be considered more effective for practical applications.

4.2 Ablation Study

Diverse contrastive losses Our model is optimized using the three contrastive losses derived from the retriever. Since TopiOCQA dataset does not provide human rewrite queries, the model is trained with $\mathcal{L}_p$ and $\mathcal{L}_u$. Table 2 presents the retrieval performances of the model in relation to the losses employed during training.

Ablation	QReCC		TopiOCQA	
	MRR	NDCG@3	MRR	NDCG@3
DenseQR	**47.0**	**45.8**	**32.3**	**32.1**
(1) w/o $\pounds_p$	46.3	44.9	32.3	32.0
(2) w/o $\pounds_g$	46.3	44.9	-	-
(3) w/o $\pounds_u$	42.9	41.6	25.7	25.3

Table 2: Performances for the ablation study according to the use of loss functions.

A comparison between the results (1) and (3) reveals that removing the conventional contrastive loss yielded competitive performance, while omitting $\mathcal{L}_u$ led to substantial performance declines in both datasets. This underscores the importance of learning well-identified representations. Similar to prior generative models, our model encodes both the current question and the conversational history. As a result, previous conversations may surface multiple times, even for questions posed in different turns. Since passages relevant to previous turns contain contents somewhat related to the current question, the retriever may incorrectly rank passages related to previous conversations higher than those relevant to the current one. Our identifying contrastive loss was designed to address this issue by more sophisticatedly distinguishing relevant passages from those hard negative passages, such

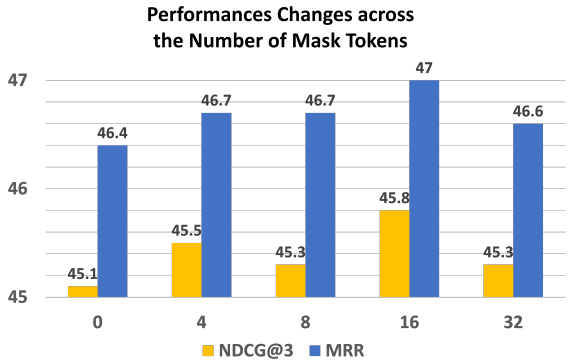


Figure 3: Retrieval results on QReCC dataset across the number of mask tokens.

Retriever	Query Type	QReCC	
		MRR	NDCG@3
SimLM	Raw	1.2	1.1
	Human rewrite	7.3	6.5
	(1) Rewriter (BERT)	28.3	26.9
	(2) Rewriter (PT-BERT)	41.4	39.8
	(3) Rewriter (RoBERTa)	6.2	5.5
ANCE	Raw	7.3	6.0
	Human rewrite	38.5	36.6
	(4) Rewriter (BERT)	10.4	9.6
	(5) Rewriter (RoBERTa)	**47.0**	**45.8**

Table 3: Retrieval performances depending on the backbone model of the dense retriever.

as previous turns' relevant ones having common words or contextual meanings related to conversational history.

For the QReCC dataset, the guidance loss appears to have minimal impact on training, as indicated by result (2). A key advantage of DenseQR is its capacity to be trained without the requirement for human rewrite queries. Nevertheless, if available, incorporating them into the training process can further enhance performance, particularly when combined with the overall loss functions.

Effect of mask tokens A fixed number of *mask* tokens were included as expanded terms in the input queries. To assess the impact of the number of *mask* tokens on performance, we performed an experiment using the QReCC dataset. The results, depicted in Figure 3, show performance tendency across different numbers of *mask* tokens. Compared to the case without *mask* tokens, the inclusion of *mask* tokens consistently enhanced performance, demonstrating the effectiveness of dense query rewriting. While performance generally improves as the number of *mask* tokens increases, a decline was observed when 32 *mask* tokens were used. This performance drop is likely due to noise in representations caused by the excessive number of *mask* tokens, as the average query length in the QReCC dataset is about only 8 tokens.

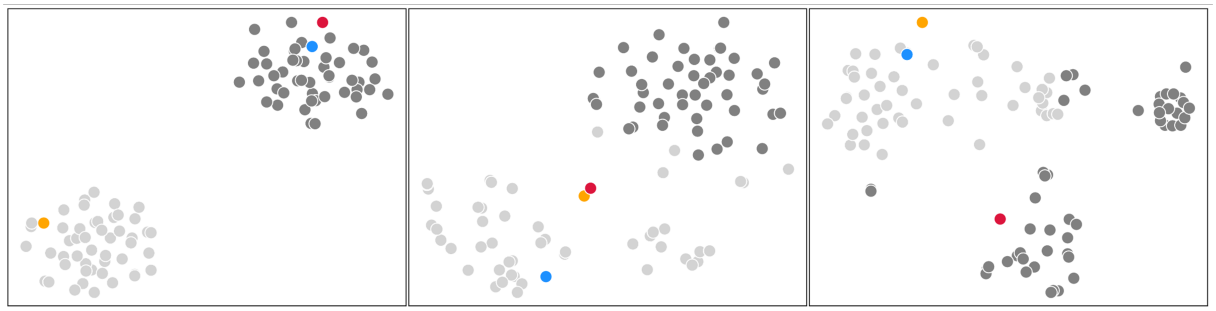
4.3 In-depth Analysis

Alignment of rewriter and retriever We conducted further experiments by applying various combinations of backbone models for both the rewriter and retriever. Since most dense retrievers are based on BERT (Devlin et al., 2019) or RoBERTa, we employed these models for the rewriter, and SimLM (Wang et al., 2023) for the BERT-based dense retriever. The findings, reported in Table 3, yielded two key insights.

First, it is clear that aligning the backbone model of the rewriter and retriever always showed the better performances. When comparing the results obtained by human rewrite queries, their performance on the QReCC dataset differed significantly, with MRR scores of 38.5 and 7.3, respectively, although both SimLM and ANCE were fine-tuned on the MSMARCO dataset. This disparity may stem from differences in the pretraining text corpora used for the backbone models. RoBERTa, for instance, was pretrained on the CommonCrawl dataset, which overlaps with the passage collection of the QReCC dataset, whereas BERT was not. As a result, the difference in basic retrieval performances between the retrievers appear to have similar effect when the different backbone models are used for the rewriter and retriever; comparison between (3) and (4). Notably, despite poor performance with raw and human rewrite queries, the rewriter substantially improved the performance of the SimLM retriever.

Second, when comparing the results of (1) and (2), where the rewriters were initialized with BERT and PT-BERT[1], respectively, performance of (2) achieved a remarkable improvement. We observed that retrieval performance improved when the rewriter and retriever had more closely aligned embedding parameters. Specifically, the embedding similarities between the rewriter and the retriever were 0.69, 0.99, and 1.0 for (1), (2), and (5), respectively. These findings suggest that rewritten queries are not well aligned with dense retrievers

[1]PT-BERT is a particularly pretrained model for SimLM on MSMARCO passages with a shallow decoder using a similar objective of MLM.



(a) when was the first one established (b) what is parnassius smintheus (c) how many states in india ruling by bjp

dense query raw query relevant passage
passages retrieved by dense query passages retrieved by raw query

Figure 4: t-SNE visualization of dense representations for queries and retrieved passages. The left plot shows a case where the rewritten query retrieves the relevant passage at a higher rank, while the center and right plots depict the opposite cases.

when the initial embeddings place in substantially different vector spaces.

Case study for dense query rewriting To ensure that DenseQR effectively addresses the challenges in conversational search, we visualized the dense representations of some queries and their corresponding retrieved passages in TopiOCQA dataset, not available for human rewrite queries as labels. In Figure 4 (a), the raw query, *"when was the first one established,"* contains high ambiguity due to the use of the pronoun *"one,"* which necessitates precise resolution in able to refer to *"record plant"* in the previous turns. As shown in the visualization, despite of the absence of a *human rewrite* query, the rewriter successfully captures the semantic meaning of the query in context with its conversational history, placing the representation of the dense query in close proximity to that of the relevant passage.

In contrast, the queries depicted in (b) and (c) are stand-alone ones, yet the ranks of the relevant passages were lower when the dense queries were entered. Specifically, although both the raw and dense queries in (b) are encoded into a similar vector space, the retrieved passages are clearly distinct. The passages retrieved by the dense query focus on the names of the *"Roman Empire"* and philosophers whose names frequently end with *"us."* The rewriter appears to persistently focus on the suffix *"us,"* also presented in *"smintheus,"* after the tokenization. Meanwhile, for (c), the query includes the abbreviation *"bjp,"* which serves as a crucial term for retrieving the relevant passage. While the raw query well retrieved the relevant passage, the rewriter fails to accurately rewrite the query resulting in retrieved passages widely spread. These two cases exhibit the representative future direction of the dense query rewriting; controlling attention to core tokens for elaborate dense representation and capturing the semantic meanings of named entities or abbreviations. We expect the both challenges can be dealt with taking advantage of the strengths of sparse retrievers but we left it as future work. Detailed descriptions of these three cases can be found in the Appendix B.

5 Conclusion

In this paper, we introduced a novel model that enables training with immediate retrieval loss by rewriting queries into dense representations. The proposed model achieves state-of-the-art performance, surpassing even the results of sparse retrievers on the QReCC dataset. Additionally, empirical experiments demonstrate the effectiveness of the training methods employed. Our in-depth analysis further explains the alignment between the rewriter and dense retrievers in terms of backbone models and how dense rewriter operates on several input queries, providing valuable insights for subsequent research of dense query rewriting.

Limitations

The remaining limitation of the proposed model is unavailability for sparse retrievers. As explained

in Section 4.3, there are cases that dense rewriter fails to select core token to attend, and abbreviation or name entities play a role of silver bullet to retrieve relevant passages. Therefore, a method to align sparse retrievers with DenseQR is needed. A simple way is to learn MLM objective so that the input *mask* tokens predict terms presenting in conversational history or having similar meanings that do not appear in a raw query. These predicted terms can be used for expansion terms with the raw query. We left the aligning dense rewriters with both sparse and dense retrievers for the future work.

References

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. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 520–534, Online. Association for Computational Linguistics.

Artem Babenko and Victor Lempitsky. 2014. Additive quantization for extreme vector compression. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 931–938.

Yongjian Chen, Tao Guan, and Cheng Wang. 2010. Approximate nearest neighbor search by residual vector quantization. *Sensors (Basel, Switzerland)*, 10:11259 – 11273.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Jianfeng Gao, Chenyan Xiong, Paul Bennett, and Nick Craswell. 2022. Neural approaches to conversational information retrieval. *Preprint*, arXiv:2201.05176.

Helia Hashemi, Hamed Zamani, and W. Bruce Croft. 2020. Guided transformer: Leveraging multiple external sources for representation learning in conversational search. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 1131–1140, New York, NY, USA. Association for Computing Machinery.

Yunah Jang, Kang-il Lee, Hyunkyung Bae, Hwanhee Lee, and Kyomin Jung. 2024. IterCQR: Iterative conversational query reformulation with retrieval guidance. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8121–8138, Mexico City, Mexico. Association for Computational Linguistics.

Sungdong Kim and Gangwoo Kim. 2022. Saving dense retriever from shortcut dependency in conversational search. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10278–10287, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Vaibhav Kumar and Jamie Callan. 2020. Making information seeking easier: An improved pipeline for conversational search. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3971–3980.

Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2021. Contextualized query embeddings for conversational search. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1004–1015.

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. *Preprint*, arXiv:2004.01909.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Fengran Mo, Kelong Mao, Yutao Zhu, Yihong Wu, Kaiyu Huang, and Jian-Yun Nie. 2023. Convgqr: Generative query reformulation for conversational search. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4998–5012.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. *PyTorch: an imperative style, high-performance deep learning library*. Curran Associates Inc., Red Hook, NY, USA.

Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W. Bruce Croft, and Mohit Iyyer. 2020. Open-retrieval conversational question answering. In *Proceedings of the 43rd International ACM SIGIR Conference on*

Research and Development in Information Retrieval, SIGIR '20, page 539–548, New York, NY, USA. Association for Computing Machinery.

Nikos Voskarides, Dan Li, Pengjie Ren, Evangelos Kanoulas, and Maarten de Rijke. 2020. Query resolution for conversational search with limited supervision. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 921–930.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2023. SimLM: Pre-training with representation bottleneck for dense passage retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2244–2258, Toronto, Canada. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zeqiu Wu, Yi Luan, Hannah Rashkin, David Reitter, Hannaneh Hajishirzi, Mari Ostendorf, and Gaurav Singh Tomar. 2022. Conqrr: Conversational query rewriting for retrieval with reinforcement learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10000–10014.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

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*, SIGIR '21, page 829–838, New York, NY, USA. Association for Computing Machinery.

Hansi Zeng, Chen Luo, Bowen Jin, Sheikh Muhammad Sarwar, Tianxin Wei, and Hamed Zamani. 2024. Scalable and effective generative information retrieval. In *Proceedings of the ACM Web Conference 2024*, WWW '24, page 1441–1452, New York, NY, USA. Association for Computing Machinery.

A Implementation

DenseQR was trained for 3 epochs with learning rate of 2e-5 on 2x RTX A6000 48GB GPUs. For contrastive learning, we set batch size per GPU to 136; 8 for query and 128 for passages. With in-batch negative sampling, each query wass mapped to 1 positive and 127 negative samples. The same hyper-parameters were used for both QReCC and TopiOCQA datasets.

B Case Study

The top-ranked passages using raw or dense queries are reported in Table 4, 5, and 6. Specifically, the ranks of relevant passages are annotated separately. When the relevant passages are not in top-100 results, the ranks are marked as N/A.

Key	**Value**
Raw query	when was the first one established
Top retrieved passage with raw query	On August 6, actress Lucille Ball was born in Jamestown. The first plane to ever fly over Jamestown occurred on September 28, 1911. The Norden Clubhouse was completed in 1914. On April 8, 1917, Company E left for guard duty. The Emerson Glass Company started the same year. In 1918, Jamestown Corp. formed to make airplane propellers. A steamship, the "City of Pittsburgh" sank at the Boatlanding, also in 1918. In 1921, the Zonta Club was organized. In 1925, the Hotel Samuels was sold. That same year, the Scottish Rite Temple was formally opened and taxpayers voted a $350,000 bond issue for the Third Street Bridge.
Rank of relevant passage with raw query	N/A
Top retrieved passage with rewritten query	The Record Plant is a recording studio established in New York City in 1968 and currently operating in Los Angeles, California. Known for innovations in the recording artists' workspace, it has produced highly influential albums, including Blondie's "Parallel Lines", Metallica's Load and Reload, The Eagles' "Hotel California", Fleetwood Mac's "Rumours", Eminem's "The Marshall Mathers LP", Guns N' Roses' "Appetite for Destruction," and Kanye West's "The College Dropout". More recent albums with songs recorded at Record Plant include Lady Gaga's "ARTPOP", Justin Bieber's "Purpose", Beyoncé's "Lemonade", and Ariana Grande's "Thank U, Next". The studio was founded in 1968 in New York City by Gary Kellgren and Chris Stone, who opened a Los Angeles branch the following year and a Sausalito, California, location in 1972.
Rank of relevant passage with dense query	1

Table 4: Specifics of the case (a) in Figure 4.

Key	**Value**
Raw query	what is parnassius smintheus
Top retrieved passage with raw query	Parnassius smintheus, which is a closely related species found in North America, is often misidentified as Parnassius phoebus. Some researchers also tend to split the North American population of Parnassius smintheus into two or three species, the northernmost of which are usually regarded as being part of P. phoebus, while the rest is considered to be P. smintheus.
Rank of relevant passage with raw query	3
Top retrieved passage with rewritten query	Priscus of Epirus (c. 305 – c. 395 AD) was a Neoplatonist philosopher and theurgist, a colleague of Maximus of Ephesus, and a friend of the emperor Julian. Priscus was a pupil of Aedesius in Pergamon, and later went to teach in Athens, where he taught Julian. When Julian was in Gaul, he wrote to Priscus in the hope of acquiring the writings of Iamblichus on the Chaldean Oracles. When Julian was proclaimed Caesar he summoned Priscus to Gaul, and he took him with him to Constantinople when he became Augustus in 361. Priscus and Maximus travelled with Julian on campaign in Persia, and they were with him when he died in 363.
Rank of relevant passage with dense query	N/A

Table 5: Specifics of the case (b) in Figure 4.

Key	**Value**
Raw query	how many states in india ruling by bjp
Top retrieved passage with raw query	As of March 2020, the BJP holds a majority of Legislative Assembly in 12 states - Arunachal Pradesh, Assam, Goa, Gujarat, Himachal Pradesh, Karnataka, Madhya Pradesh, Manipur, Tripura, Uttar Pradesh and Uttarakhand. In Haryana, BJP shares power as the senior partner (Chief Minister of BJP) with Jannayak Janta Party.
Rank of relevant passage with raw query	1
Top retrieved passage with rewritten query	In its verdict on 15 November 1995, the district and sessions court in Jaipur dismissed the case and acquitted all the five accused. Five judges were changed, and it was the sixth judge who ruled that the accused were not guilty, stating "inter alia" that Bhanwari's husband couldn't have passively watched his wife being gang-raped. Under pressure from women's groups, the State Government decided to appeal against the judgment. The judgement led to a nationwide campaign for justice for Bhanwari Devi. However, by 2007, 15 years after the incident, the Rajasthan High Court held only one hearing on the case and two of the accused were dead.
Rank of relevant passage with dense query	N/A

Table 6: Specifics of the case (c) in Figure 4.