Explicit Query Rewriting for Conversational Dense Retrieval

Hongjin Qian, Zhicheng Dou*
Gaoling School of Artificial Intelligence, Renmin University of China
{ian,dou}@ruc.edu.cn

Abstract

In a conversational search scenario, a query might be context-dependent because some words are referred to previous expressions or omitted. Previous works tackle the issue by either reformulating the query into a self-contained query (query rewriting) or learning a contextualized query embedding from the query context (context modelling). In this paper, we propose a model CRDR that can perform query rewriting and context modelling in a unified framework in which the query rewriting's supervision signals further enhance the context modelling. Instead of generating a new query, CRDR only performs necessary modifications on the original query, which improves both accuracy and efficiency of query rewriting. In the meantime, the query rewriting benefits the context modelling by explicitly highlighting relevant terms in the query context, which improves the quality of the learned contextualized query embedding. To verify the effectiveness of CRDR, we perform comprehensive experiments on TREC CAsT-19 and TREC CAsT-20 datasets, and the results show that our method outperforms all baseline models in terms of both quality of query rewriting and quality of context-aware ranking.

1 Introduction

The recent rising of intelligent assistants triggers the transition of information retrieval systems from ad-hoc search to conversational search (Croft, 2019; Gao et al., 2020; Ren et al., 2021). Searching in a conversational manner provides interactive information exchange between users and the system. Such interactive format is usually achieved by multi-turn dialogues, with which the system can understand the user's complex information needs (Dalton et al., 2020b,a).

In a conversational system, a user often asks follow-up questions about something referred to previous dialogue context (Vakulenko et al., 2021a; Radlinski and Craswell, 2017; Ren et al., 2021). Therefore, a conversational search query is likely to be context-dependent and not self-contained. The real user intent is hidden underneath not only the current query but the whole query context. Figure 1 shows an example of conversational search, from which we can find two most typical linguistic phenomenons that undermine the semantic completeness of a single query in the conversation, namely *anaphora* and *ellipsis* (Vakulenko et al., 2020). Specifically, *anaphora* refers to the phenomenon of an expression that depends on an expression in the previous context. For example, in Figure 1, the word "its" in q_2 depends on the expression "the Phoenix city" in q_1. Meanwhile, *ellipsis* refers to the phenomenon of the omission of expressions in the previous context. For example, the complete form of q_3 in Figure 1 should be "How about the population of New York?". However, the expression "the population of" is omitted.

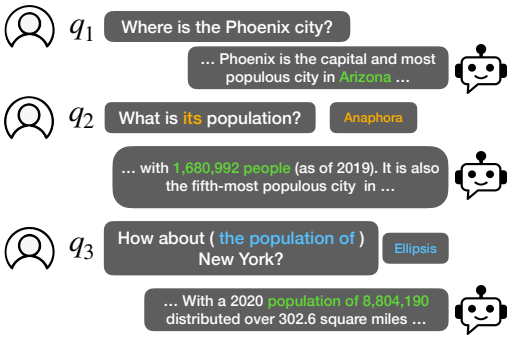

Figure 1: A case of Conversational Search

Due to these two key characteristics, a single query in conversational search is inherently incomplete and ambiguous (Qu et al., 2020; Reddy et al., 2019). There are a few attempts that aim to alleviate such problems. Generally, these methods can be divided into two groups. The first group of methods formalize the problem as context-aware

*Corresponding author.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 4725 - 4737

query rewriting (Ren et al., 2018; Vakulenko et al., 2021b; Lin et al., 2021c; Vakulenko et al., 2021a; Yu et al., 2020). Most of these methods finetune a pretrained language model (*e.g.*GPT2 and T5) to generate a new query via an auto-aggressive decoder (Vakulenko et al., 2021b,a; Yu et al., 2020).

The second group of methods are on the top of dense retrieval models (Lee et al., 2019; Karpukhin et al., 2020; Luan et al., 2021; Xiong et al., 2020; Yu et al., 2021; Lin et al., 2021b). These methods seek to tackle the query ambiguity by learning a contextualized dense representation from the query context, which can be then used for dense retrieval (Yu et al., 2021; Lin et al., 2021b). Though these dense retrieval methods booster the performances in many information retrieval tasks, we argue that the current dense retrieval methods have two limitations in conversational search: (1) previous conversational dense retrieval models fail to reformulate the original query into a well-formed readable query, which is useful for re-ranking, providing explainability and benefit other conversational scenarios (*e.g.*query suggestion or query clarification) ; (2) most of the current dense retrieval models are trained on ad-hoc queries (Xiong et al., 2020; Yates et al., 2021). And we speculate that these models lack the ability to highlight important terms in the conversational query context accurately (will discuss in Section 4). The two limitations of current conversational dense retrieval methods undermine the potentiality of a conversational search system.

In this paper, we propose a **C**onversational Query **R**writing method for **D**ense **R**etrieval (CRDR). It explores enhancing the context modelling ability of a dense retrieval model with term supervision signals learned during query rewriting. In other words, **CRDR simultaneously learns a term-enhanced contextualized query embedding for a better ranking and reformulates the original query into a well-formed query**. The advantages of CRDR are two-fold. First, CRDR can generate a well-formed query which is beneficial as mentioned above. Second, with the explicitly enhanced term weighting inspired by the query reformulation, the learned conversational query embedding can better represent the user's intent and will lead to better retrieval quality.

CRDR achieves the goal via two modules, namely the *Query Rewriting* module and the *Dense Retrieval* module. The *Query Rewriting* module treats the task as *query modification*. Specifically, it directly performs token-level modifications (replace or insert) on the original query with relevant terms in the query context. The idea is directly motivated by the linguistic definition of *anaphora* and *ellipsis* which are two key characteristics of conversational search (see Section 3.3). Instead of generating a new query, our method directly use information in the query context to modify the current query. Besides, the *Dense Retrieval* module learns a contextualized query embedding from the query context in which it considers the relevant terms detected during query writing. During the query rewriting, we measure the relevance between the terms in the query context and the current query. For the terms with high relevant scores, we will dynamically fuse their semantics into the current query embedding.

To verify the quality of the reformulated query and contextualized query embedding generated by CRDR, we conduct experiments on both traditional inverted index-based retrieval and dense retrieval models. Experiment results show that CRDR outperforms all baseline models.

Our contributions can be summarized as: (1) We seek to unify query rewriting with dense retrieval in conversational search and reveal that explicit query rewriting is beneficial to conversational dense retrieval. (2) We propose CRDR, which integrates the query rewriting and dense retrieval in a unified framework in which the learned contextualized query embedding is enhanced by the relevant terms detected during query rewriting. (3) CRDR can simultaneously learn a term-enhance contextualized query embedding for dense retrieval and generate an explicit query which provides explanability and benefits other tasks. (4) Extensive experiments show that our method outperforms the state-of-the-art methods on both traditional inverted index-based retrieval and dense retrieval.

2 Related Work

Conversational search is considered as one of the most promising searching paradigm in the information retrieval domain (Gao et al., 2020). With the release of TREC Conversational Assistant Track (CAsT), many researchers pay much attention into the task (Dalton et al., 2020b,a). The CAsT tasks contain multi-turn dialouge turns seeking information from the web search engine. The tasks are evaluated by measuring the accuracy of retrieved

documents given the dialogue queries. Along the direction, previous works either leveraging a query rewriting model to generate a new query or learn a contextualized query embedding to represent the whole query context.

Query rewriting is a widely applied technique to alleviate query ambiguity in conversational search. It aims to reformulate the original query into a self-contained query with the query context. Query rewriting task was first introduced by Elgohary et al. (2019). They released the CANARD dataset deriving from the QuAC dataset and provides human reformulated queries (Choi et al., 2018). Recent TREC CAsT tasks greatly promoted the development of conversational query writing research (Dalton et al., 2020b,a). They defined conversational search as a task in which effective passage retrieval requires understanding a query context. Follow the CAsT tasks, Mele et al. (2020) proposed a set of heuristic methods including part-of-speech, dependency parse and co-reference resolution to rewrite the original query. Besides, Voskarides et al. (2020) treated the task as relevance term classification in which each term in the query context is assigned a binary label. They then added the relevant terms at the end of the original query. To generate a well-formed natural language query, many methods treat the task as text generation in which they input the query context into a text generation model and predict a new query (Vakulenko et al., 2021a; Yu et al., 2020; Pradeep et al., 2021; Lin et al., 2021c; Ren et al., 2018). Yu et al. (2020) proposed a rule-based query simplification method generating a large number of distantly supervised data to train a text generation model. To increase the generative ability of the text generation model, Vakulenko et al. (2021a) proposed a model that considers several individual word distributions during decoding. Besides, Lin et al. (2021c) proposed a method that considers term importance.

A dense retrieval system usually applies a bi-encoder architecture to encode the query and document into dense vector separately (Karpukhin et al., 2020; Luan et al., 2021; Xiong et al., 2020). It then computes the query-document relevance with similarity functions (*e.g.*dot product) and performs retrieval via dense searching algorithms (*e.g.*approximate nearest neighbor) (Andoni and Indyk, 2008). In conversational search, the query embedding is learned from a set of follow-up queries instead of an ad-hoc query. As labelling ground truth query in conversational search is exhausted, it is difficult to train a dense retrieval model dedicated to the conversational system. Therefore, Lin et al. (2021b) directly concatenated the follow-up queries into a long sequence and fed them to the dense retrieval model finetuned on the CANARD dataset. Through this way, the model can learn a query embedding from the whole query context and then perform dense retrieval. Besides, Yu et al. (2021) proposed a few-shot framework to continue training an ad-hoc dense retrieval model. Mao et al. (2022) utilize a curriculum contrastive method to denoise the conversational context for better representations. Similarly, they then learn a query embedding which is used in the dense retrieval system.

In this paper, we first obtain the well-formed query via an Encode-Tag-Modify framework. Then, we use the relevant terms detected in the query context as supervision signals to enhance the robustness of the query encoder.

3 Methodology

3.1 Preliminary

The goal of a search system is to retrieve a document d^* from a large document repository D for a query q that maximizes the following objective:

$$d^* = \arg\max_{d \in D} f(q, d), \tag{1}$$

where $f(q, d)$ denotes the relevance score for a query-document pair (q, d).

In the conversational search scenario, the query $q_k, k \leq n$ belongs to multi-turn queries $Q = \{q_i\}, i \in [1, n]$. Thus, the query q_k can be ambiguous and context-dependent. More specifically, the current query q_k is conditioned on the query history $Q_{1:k-1}$. As a result, the real user intent is hidden under not only the current query but the whole query history. Formally, in conversational search, Eq. (1) can be modified as :

$$d^* = \arg\max_{d \in D} f(Q_{1:k}, d), \tag{2}$$

where $Q_{1:k} = \{q_1, \cdots, q_k\}$ is the query context which can be very long and noisy. To this end, we need to find a surrogate query q^* that represents the real intent of the current query q_k:

$$q^* = g(Q_{1:k}), \tag{3}$$

where $g(\cdot)$ denotes the mapping function.

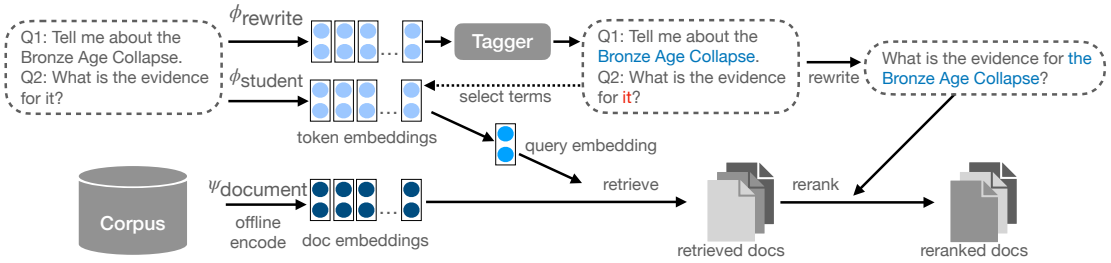

Figure 2: The overall Architecture of the proposed model CRDR which is comprised of the Query Rewriting module and the Dense Retrieval module. The detail of Query Rewriting module is at Figure 3.

As mentioned in Section 1, many works define the mapping function $g(\cdot)$ as a text generator or a term classifier. They obtain the reformulated query q^* by query generation or expansion. The obtained query is then feed into the traditional retrieval-rerank pipeline.

Another direction, namely conversational dense retrieval, aims to encode the query context into a contextualized query embedding $\mathbf{q}^*$ via a query encoder $\phi(\cdot)$:

$$\mathbf{q}^* = \phi(Q_{1:k}). \tag{4}$$

The query context encoder $\phi(\cdot)$ is usually initialized from a pretrained language model (*e.g.*BERT) and the contextualized query embedding $\mathbf{q}^*$ can be used for end to end dense retrieval (Yu et al., 2021; Lin et al., 2021b). The dense retrieval step is usually performed by computing similarity scores for the (query, document) pairs, which can be efficient via various tools (*e.g.*Faiss (Johnson et al., 2019)). We will further introduce the conversational dense retrieval in Section 3.4.

3.2 CRDR: The Proposed Model

The query ambiguity in conversational search undermines the search system's performance. Previous works seek to solve such an issue by (1) rewriting the context-dependent query into a self-contained query; (2) learning a contextualized query embedding for dense retrieval. Intuitively, we can unify the two methods to boost the search system's performance by first reformulating the query and then learning a dense representation from the reformulated query. Nevertheless, in practice, the reformulated query is not perfect, and therefore the performance of dense search system is bounded (Vakulenko et al., 2021a; Yu et al., 2020).

To this end, we propose CRDR, which is comprised of the *Query Rewrite* module and the *Dense Retrieval* module. Instead of learning a query embedding from the reformulated query, CRDR dynamically choose to utilize the relevant terms detected during query rewriting to enhance the query embedding learned from the query context. Specifically, the *Query Rewrite* module generates a self-contained query via an *Encode-Tag-Modify* framework. It first identifies the relevant terms in the query context, and then modifies the tokens of the original query with the identified terms. The *Dense Retrieval* module learns a term-enhanced contextualized query representation from the query context which considers the semantic connections between the current query and relevant terms in the query context. The *Dense Retrieval* module is enhanced by the prior knowledge provided by the *Query Rewrite* module. Figure 2 shows the overall architecture of CRDR.

3.3 Query Rewriting via Modification

Instead of generating a new query or expanding query with relevant terms, CRDR formalizes the query rewriting task as *query modification*. We directly modify the current query q_k by either replacing or inserting tokens at a proper position in the current query q_k. CRDR achieves the goal in a *Encode-Tag-Modify* framework. Figure 3 illustrates the query rewrite module in CRDR.

In the *Encode-Tag-Modify* framework, CRDR first encodes the query context $Q_{1:k}$ into contextualized representations. It then performs token-level classification to assign a label $l \in \{\mathbf{O}, \mathbf{REL}, \mathbf{IN}\}$ to each token in $Q_{1:k}$. The labels refer to: (1) **REL**: mentions terms in the previous query context $Q_{1:k-1}$ that might be a referred or omitted expressions for the current query; (2) **IN**: mentions the tokens that might be an entry point for query modification (*e.g.*"its" in the q_2 of the Figure 1); (3) **O**: mentions other irrelevant terms. In the *modify*

step, CRDR rewrites the current query q_k by two heuristic methods: (1) replace the **IN** token with the **REL** tokens, and (2) insert the **REL** tokens after the **IN** token. Take the conversation in Figure 1 as an example, the word "its" in q_2 is a **IN** token which will be replaced by the **REL** token "Arizona". To achieve so, we apply three simple rules: (1) replace: when the **IN** token is a pronoun[1] or possessive pronoun[2], we replace it with the **REL** tokens or its possessive form; (2) insert: when the **IN** token is other words, we insert the **REL** tokens after the **IN** token; (3) add: when no **IN** token is detected, we add the **REL** tokens at the end of the current query. Especially when no **IN** token and **REL** token is detected, the current query will keep unmodified.

Precisely, for the i-th query $q_i, i \in [1, k]$, its tokens $\{w_1^i, \cdots, w_{n_i}^i\}$, and the query context $Q_{1,k} = \{q_1, \cdots, q_k\}$, we have $Q = \{\{w_1^1, \cdots, w_{n_1}^1\}, \cdots, \{w_1^k, \cdots, w_{n_k}^k\}\}$. In the *Encode* step, we concatenate the query context $Q = \{q_1, \cdots, q_k\}$ into a token sequence S:

$$S = w_{\text{CLS}}, w_1^1, \cdots, w_{\text{SEP}}, \cdots, w_{n_k}^k. \quad (5)$$

Then, we use a pretrained language model (*e.g.*BERT) to encode the token sequence S into contextualized token representations $\mathbf{S}$:

$$\mathbf{S} = \phi_{\text{rewrite}}(S) = \{\mathbf{e}_1, \cdots, \mathbf{e}_n\}, \quad (6)$$

where ϕ_{rewrite} denotes the query encoder in the query rewriting module. And $\mathbf{S} \in \mathbb{R}^{n \times d}$, n is the total number of tokens in the sequence S, d is the size of the contextualized token representation.

In the *Tag* step, we feed the contextualized token representations $\mathbf{S}$ into a multi-layer perceptron (MLP) with Softmax to get the label probability distributions:

$$\tilde{\mathbf{S}} = \text{Softmax}(\text{MLP}(\mathbf{S})) = \{\tilde{\mathbf{e}}_1, \cdots, \tilde{\mathbf{e}}_n\}, \quad (7)$$

where $\tilde{\mathbf{S}} \in \mathbb{R}^{n \times 3}$. We then assign each token a label that has the highest probability to get the label sequence:

$$L = \arg\max(\tilde{\mathbf{S}}) = \{l_1, \cdots, l_n\}, \quad (8)$$

where $l_i, i \in [1, n]$ represents the label for the i-th token.

After obtaining the labels $\{l_1, \cdots, l_n\}$, we can rewrite the current query q_k into a self-contained query q_k^* via the *modify* step which is introduced above. We will further discuss the benefits and limitations of the CRDR's query rewriting method in Section 5.

[1]it, he, she, they etc.

[2]its, his, her, their etc.

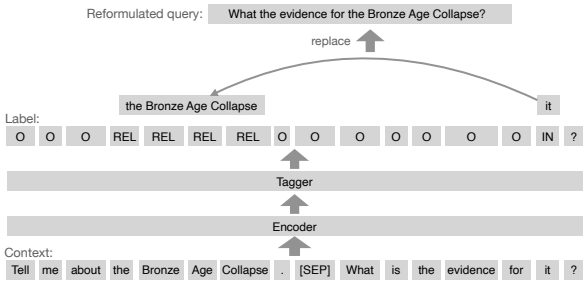

Figure 3: The Query Rewrite module of CRDR

3.4 Dense Retrieval in CRDR

Recently, the dense retrieval methods draw much attention (Karpukhin et al., 2020; Luan et al., 2021; Xiong et al., 2020; Yu et al., 2021; Lin et al., 2021b). These methods usually utilize an deep neural model (*e.g.*Bi-Encoder) to separately encodes the query q and the document d into dense representations. It then obtain the relevance score:

$$f(q, d) = \text{sim}\langle \phi(q), \psi(d) \rangle, \quad (9)$$

where $\text{sim}\langle \cdot \rangle$ denotes computing the similarity between two vectors. $\phi(\cdot)$ and $\psi(\cdot)$ denote the query encoder and document encoder, respectively. The encoders are usually initialized from a pretrained language model (*e.g.*BERT). They encode the text sequence into a dense embedding sequence and then perform pooling operations (*e.g.*the [CLS] embedding in BERT) to get a single dense embedding to represent the query or the document.

Typically, the optimizing goal for a dense retrieval model is to maximize the probability:

$$\mathbf{P}(q, d^+, \mathcal{D}) = \frac{\mathbf{e}^{f(q,d^+)}}{\mathbf{e}^{f(q,d^+)} + \sum_{d^- \in \mathcal{D}} \mathbf{e}^{f(q,d^-)}}, \quad (10)$$

where d^+ and d^- represent the relevant document and the negative document given the query q. $\mathcal{D}$ represents the entire document corpus.

Besides, the document embeddings can be offline computed and indexed. Thus, many dense searching algorithms (*e.g.*approximate nearest neighbor) can be used when performing inference (Andoni and Indyk, 2008).

In this paper, we focus on dense retrieval for conversational search. We investigate how to obtain a query embedding that faithfully represents all the

necessary information of a set of follow-up queries. In conversational search, a query $q_k, k \in [1, n]$ belongs to multi-turn queries $Q = \{q_i\}, i \in [1, n]$. The current query q_k is context-dependent and might be semantically incomplete to represent the real user intent. Therefore, solely encoding the current query q_k would omit the context information, which is essential to represent real user intent. To tackle the issue, Yu et al. (2021) propose a teacher-student framework to train the query encoder $\phi(\cdot)$. In the framework, the student model encodes the query context $Q_{1:k}$ into a dense representation $\mathbf{q}'_k$:

$$\mathbf{q}'_k = \phi_{\text{student}}(Q_{1:k}), \tag{11}$$

where $\phi_{\text{student}}(\cdot)$ represents the student query encoder.

The teacher model use a pretrained ad-hoc query encoder to encode the manual oracle query q^*_k into a dense representation $\mathbf{q}^*_k$:

$$\mathbf{q}^*_k = \phi_{\text{teacher}}(q^*_k), \tag{12}$$

And the teacher model distill its knowledge to the student model by using the MSE loss.

The contextualized query embedding $\mathbf{q}'_k$ aims to summarize the semantic information of the whole query context. We find that when the query context becomes long, it is difficult for the contextualized query embedding $\mathbf{q}'_k$ to highlight all the important information accurately. In such a situation, we propose the contextualized query representation $\mathbf{q}'_k$ can be enhanced by considering the relevant terms in the query context. As mentioned in Section 3.3, the *Query Rewrite* module in CRDR modifies the original query with detected relevant terms. Intuitively, we can enhance the contextualized query embedding $\mathbf{q}'_k$ by explicitly introduce these detected relevant terms as supplementing information. Formally speaking, given the query context $Q_{1:k} = \{\{w^1_1, \cdots, w^1_{n_1}\}, \cdots, \{w^k_1, \cdots, w^k_{n_k}\}\}$, we concatenate the query context the same as Eq. (5) to get a text sequence S which is then fed to the query encoder $\phi(\cdot)$:

$$\mathbf{S} = \phi(S) = \{\mathbf{e}_1, \cdots, \mathbf{e}_n\}, \tag{13}$$

where the $\mathbf{e}_1$ corresponds to the $[CLS]$ token, which is usually used the query embedding (Lin et al., 2021b; Yu et al., 2021). Thus, Eq. (11) become:

$$\mathbf{q}'_k = \phi_{\text{student}}(Q_{1:k}) = \mathbf{e}_1. \tag{14}$$

As mentioned above, the [CLS] token's embedding might fail to summarize all the necessary information from the query context. Therefore, except for the [CLS] embedding, CRDR integrates the relevant term's embeddings into the [CLS] token's embedding, yielding an enhanced query embedding $\hat{\mathbf{q}}'_k$:

$$\hat{\mathbf{q}}'_k = \alpha \cdot \mathbf{e}_1 + (1 - \alpha) \cdot \underset{\text{dim}=0}{\text{mean}}(\mathbf{e}^r_1, \cdots, \mathbf{e}^r_m), \tag{15}$$

where α decides the proportions of relevant embeddings and $\text{mean}(\cdot)$ defines the mean pooling operation. And $(\mathbf{e}^r_1, \cdots, \mathbf{e}^r_m)$ denote m relevant token's embeddings.

We think that only when the contextualized query embedding $\mathbf{q}'_k = \mathbf{e}_1$ fails to highlight the important information in the query context, we need to enhance the contextualized query representation. Therefore, we use the attention scores from the [CLS] token to measure whether the relevant terms are highlightened. Specifically, for the [CLS] token, we have its attention scores $Z = \{z_1, \cdots, z_n\}$ (averaged on attention heads) to n tokens in the sequence. We compute the α value by:

$$\alpha = 1 - \frac{\text{mean}(z^r_1, \cdots, z^r_m)}{\max(Z)}, \tag{16}$$

where $\text{mean}(z^r_1, \cdots, z^r_m)$ represent the average of the m relevant token's attention scores. The $\max(Z)$ represents the highest attention score in Z, which is used to normalize the averaged attention score of the relevant tokens. Eq. (16) shows that when the [CLS] token already put enough attention on the relevant terms, the α value becomes small, and therefore, the impacts of relevant term's embeddings decrease. Otherwise, when the [CLS] token fail to attend to the relevant terms, the impacts of the relevant embeddings increase and therefore, the query embedding $\hat{\mathbf{q}}'_k$ is enhanced.

3.5 Training and Inference

In the training phase, we train the *Query Rewriting* module and the *Dense Retrieval* module separately as the optimization goal of the two tasks are rather different. Specifically, the optimization goal of the *Query Rewriting* module is to correctly assign labels to the tokens in the query context and objective function of the module is the *Cross-Entropy Loss*. The goal of the *Dense Retrieval* in the conversational search is to obtain a contextualized query embedding that faithfully represent the query context.

To train the *Dense Retrieval* module, we apply the teacher-student training framework with the MSE Loss. In the framework, according to the Eq (15), we can rewrite the Eq. (11) as $\hat{\mathbf{q}}'_k = \phi_{\text{student}}(Q_{1:k})$. Thus, the loss function of the *Dense Retrieval* module is:

$$\mathcal{L} = \text{MSE}(\hat{\mathbf{q}}'_k, \mathbf{q}^*_k). \quad (17)$$

In the inference phase, as shown in Figure 2, the two modules synchronously encode the query context into dense embedding sequences. Then, the *Query Writing* module will first assign labels to the tokens, and the *Dense Retrieval* module will obtain the term-enhanced contextualized query embedding $\hat{\mathbf{q}}'_k$ according to Eq. (15), which is then used for dense retrieval. In practice, CRDR can output the reformulated query and the query embedding approximately the same time. We can either use the reformulated query for traditional inverted index-based retrieval or query embedding for dense retrieval. Furthermore, the reformulated query can be used to perform document rerank.

4 Experiments

4.1 Settings

We conduct experiments regarding three types of search systems: the traditional inverted-index retrieval (BM25), dense retrieval, and hybrid retrieval. For each type, we choose previous SOTA models as baselines. The details of the baseline models can be referred to Appendix B.

To fairly compare with the baselines, we evaluate CRDR with the following settings: (1) we use the CRDR's reformulated queries to perform the traditional inverted-index retrieval (sparse retrieval); (2) we use the CRDR's query embeddings to perform dense retrieval; (3) we fuse the CRDR'S results of sparse retrieval and dense retrieval, yielding hybrid search. The implementation details of CRDR and all baseline models can be referred to Appendix A.

We evaluate the ranking performance of models that apply the inverted-index retrieval with cutoff@1000 and report Recall@1000, MRR[3] and nDCG@3 (official evaluation metric of TREC CAsT). We also report the reformulated query's F1-score (F) to compare the token-level rewriting quality. For dense retrieval, following Yu et al. (2021), we report cutoff@100 and report MRR and nDCG@3 to evaluate their ranking performance.

[3]we follow the TREC CAsT-20's official setting, using relevance scale ≥ 2 as positive for MRR.

In CRDR, the query rewriting module is trained on CANARD dataset (Elgohary et al., 2019) of which the token labels are automatically generated. We evaluate the models by performing five-fold cross validation on the benchmark datasets, TREC CAsT-19 and TREC CAsT-20 (Dalton et al., 2020b,a). More information of the used datasets can be referred to Appendix C.

4.2 Results

Table 1 shows the results from which we find that **CRDR achieves the best performance across most evaluation metrics on the two datasets regarding both sparse retrieval and dense retrieval.** Regarding sparse retrieval, CRDR obtains the best performance on the two datasets across all the evaluation metrics except for reranked NDCG@3 on CAsT-20, which verifies the high quality of CRDR's reformulated queries. Regarding dense retrieval, CRDR consistently outperforms the previous method, which verifies the effectiveness of the term-enhanced query embedding learned by CRDR. Regarding hybrid search, which fuses the BM25 results and the dense retrieval results, we find that the performances increase in the CAsT-19 dataset but not in the CAsT-20 dataset. The potential reason might be: in the CAsT-20 dataset, the BM25 results can not provide orthogonal information that the dense retrieval system fails to capture. As mentioned by Dalton et al. (2020a), queries in the CAsT-20 dataset refer to previous responses given by a system, which makes the dependencies in the query context more complex. Hence, the ability of query rewriting with ad-hoc search is undermined as it can not fully represent the query context. In such situations, the potentiality of a dense retrieval system greatly increases.

5 Discussion

In this section, we first conduct a breakdown analysis to explore the pros and cons of sparse retrieval and dense retrieval in conversational search. Then, we illustrate how the query rewriting in our CRDR can be used to denoise the query context when learning the conversational context representation. In Appendix D, we use a case study to explicitly walk through why our CRDR performs better than baseline models.

Table 1: The results of all models on two datasets. '†' indicates the model outperforms all baselines significantly with paired t-test at $p < 0.05$ level. '♢' indicates the model uses cutoff@100. Otherwise, the model uses cutoff@1000. Best results in each block are denoted in bold. We also list the performance of manual oracle query as reference.

Search	Model	CAsT-19						CAsT-20					
		Recall	MRR		NDCG@3		F1	Recall	MRR		NDCG@3		F1
		Initial	Initial	Reranked	Initial	Reranked		Initial	Initial	Reranked	Initial	Reranked	
Sparse	Original	0.419	0.321	0.418	0.136	0.267	0.82	0.252	0.100	0.252	0.069	0.197	0.74
Sparse	Transformer++	0.755	0.557	0.805	0.267	0.525	**0.91**	0.351	0.162	0.349	0.100	0.254	0.70
Sparse	Self-Learn	0.729	0.541	0.768	0.311	0.501	0.90	0.463	0.240	0.453	0.158	0.343	0.76
Sparse	Rule-based	0.736	0.519	0.745	0.280	0.492	**0.91**	0.458	0.210	0.436	0.136	0.340	0.78
Sparse	QuReTeC	0.778	0.605	0.810	0.338	0.507	0.89	0.559	0.262	0.489	0.171	**0.370**	0.78
Sparse	CRDR	**0.795**†	**0.664**†	**0.811**	**0.340**	**0.528**†	**0.91**	**0.574**†	**0.267**†	**0.497**†	**0.174**	0.363	**0.80**†
Sparse	Human	0.812	0.802	0.866	0.312	0.580	1.00	0.711	0.378	0.691	0.242	0.532	1.00
Dense	ConvDR♢	-	0.740	0.802	0.466	0.526	-	-	**0.501**	0.506	0.340	0.362	-
Dense	CRDR♢	-	**0.765**†	**0.819**†	**0.472**†	**0.553**†	-	-	**0.501**	**0.517**†	**0.350**†	**0.381**†	-
Dense	Human	-	0.740	0.835	0.461	0.566	-	-	0.591	0.663	0.422	0.483	-
Hybrid	ConvDR (RRF)♢	-	-	0.799	-	0.541	-	-	-	0.545	-	**0.392**	-
Hybrid	CRDR	**0.853**†	**0.837**†	**0.852**†	**0.538**†	**0.578**†	-	**0.702**†	**0.501**	**0.552**	**0.350**	0.381	-

Table 2: Breakdown Analysis of the Result on CAsT

	CAsT-19				CAsT-20			
NDCG@3	=1	>0.5	>0	=0	=1	>0.5	>0	=0
Sparse Retrieval	5	45	79	44	2	26	73	107
Dense Retrieval	16	63	64	30	10	58	71	69
Hybrid	17	77	59	20	10	58	71	69

Table 3: Performance comparison between sparse retrieval and dense retrieval.

Search	Model	CAsT-19		CAsT-20	
		MRR	NDCG@3	MRR	NDCG@3
Sparse	Transformer++	0.557	0.267	0.162	0.100
Sparse	QuReTeC	0.605	0.338	0.262	0.171
Sparse	CRDR	0.664	0.340	0.267	0.174
Dense	Transformer++	0.696	0.441	0.296	0.185
Dense	QuReTeC	0.709	0.443	0.430	0.287
Dense	ConvDR	0.740	0.466	**0.501**	0.340
Dense	CRDR	**0.765**	**0.472**	**0.501**	**0.350**

5.1 Sparse Retrieval v.s. Dense Retrieval

We conduct a breakdown analysis on the retrieval results of CRDR, which is shown in Table 2. First, the dense retrieval can rank positive document higher in the documents list than sparse retrieval as the overall NDCG@3 scores greatly increase; Second, the advantages of dense retrieval system are more obvious on CAsT-20 in which the query might refer to the system's answer. It illustrates that the strong context modelling ability of dense retrieval system enables handling complex query context Besides, we find that the hybrid results greatly improve on the CAsT-19 but not on the CAsT-20, which implies that, for complex query context, the dense retrieval system can fully cover the retrieval ability of sparse retrieval. In conclusion, the dense retrieval system is superior to sparse retrieval in conversational search due to its strong context modeling ability, especially for complex query context.

5.2 Context Denoise via Query Rewriting

As mentioned in Section 5.1, generating a self-contained query might suffer from information missing. And learning a contextualized query embedding from the whole query context might be biased by the context noise. Table 3 shows the performance comparison between sparse retrieval and dense retrieval, from which we can find that: (1) the Transformer++ and QuReTec are the previous SOTA query rewriting models. Applying them into a dense retrieval system would improve their performance by a large margin, but cannot compete with ConvDR and CRDR which takes the whole query context as the input. This verifies our claim about the information missing issue for query rewriting models; (2) regarding sparse retrieval, our CRDR performs better than previous SOTAs, which implies that CRDR can detect more necessary information from the query context. Benefiting from the detected necessary information, CRDR learns the query embedding by selectively extracting useful information from the query context.

6 Conclusion

In this paper, we explore how to unify query rewriting and dense retrieval in conversational search in one framework. We propose CRDR, a model that can simultaneously generate a self-contained query and model the query context into a contextualized query embedding that considers the relevant terms detected during query rewriting. Specifically, the *Query Rewriting* module directly modifies the original query via an *Encode-Tag-Modify* framework. The *Dense Retrieval* module leverage the relevant

terms detected by the *Query Rewriting* module as supervision signal to enhance the contextualized query embedding. We conduct comprehensive experiments on the CAsT-19 and CAsT-20 datasets. The experiment results show that CRDR outperforms all baseline models on both sparse retrieval and dense retrieval.

Acknowledgments

Zhicheng Dou is the corresponding author. This work was supported by the National Natural Science Foundation of China No. 61872370, Beijing Outstanding Young Scientist Program NO. BJJWZYJH012019100020098, the Fundamental Research Funds for the Central Universities, the Research Funds of Renmin University of China NO. 22XNKJ34, Public Computing Cloud, Renmin University of China, Intelligent Social Governance Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China, and the Outstanding Innovative Talents Cultivation Funded Programs 2022 of Renmin Univertity of China. The work was partially done at Beijing Key Laboratory of Big Data Management and Analysis Methods, and Key Laboratory of Data Engineering and Knowledge Engineering, MOE.

Limitations

In this paper, we explore enhancing the representative ability of a conversational dense retriever by explicitly modelling the relevant terms in the query context. In the meantime, the proposed method, CRDR, can generate a self-contained query. The limitations of proposed method are threefold. First, though the conversational dense retriever in CRDR can benefit from the supervision signal provided by its query rewriting module, the query rewriting module fails to benefit from the retrieval process. We will explore a better strategy to unify the two tasks for conversational search in the future; Second, the query rewriting module in CRDR is trained on an auto-labelled dataset which may contain label noise and therefore bias the query rewriting module. We think CRDR's query rewriting module can be largely improved if trained on a manually-labelled dataset. The *Modify* step involves heuristic rules, which might limit the query rewrite module generalize to broader usage; Third, though TREC CAsT-19 and TREC CAsT-20 are the most widely-used benchmark datasets for conversational search, their dataset size is relatively small, which may bias the robustness of the evaluation results. We believe that with the popularity of conversational search, there will be large-scale datasets released so that we can evaluate the models in a more robust way.

References

Alexandr Andoni and Piotr Indyk. 2008. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Commun. ACM*, 51(1):117–122.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184, Brussels, Belgium. Association for Computational Linguistics.

Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, page 758–759, New York, NY, USA. Association for Computing Machinery.

W. Bruce Croft. 2019. The importance of interaction for information retrieval. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 1–2, New York, NY, USA. Association for Computing Machinery.

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

Jeffrey Dalton, Chenyan Xiong, Vaibhav Kumar, and Jamie Callan. 2020b. *CAsT-19: A Dataset for Conversational Information Seeking*, page 1985–1988. Association for Computing Machinery, New York, NY, USA.

Ahmed Elgohary, Denis Peskov, and Jordan Boyd-Graber. 2019. Can you unpack that? learning to rewrite questions-in-context. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5918–5924, Hong Kong, China. Association for Computational Linguistics.

Edward A Fox and Joseph A Shaw. 1994. Combination of multiple searches. *NIST special publication SP*, 243.

Jianfeng Gao, Chenyan Xiong, and Paul Bennett. 2020. *Recent Advances in Conversational Information Retrieval*, page 2421–2424. Association for Computing Machinery, New York, NY, USA.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

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

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021a. Pyserini: An easy-to-use python toolkit to support replicable ir research with sparse and dense representations. *arXiv preprint arXiv:2102.10073*.

Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2021b. 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. 2021c. Multi-stage conversational passage retrieval: An approach to fusing term importance estimation and neural query rewriting. *ACM Transactions on Information Systems (TOIS)*, 39(4):1–29.

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

Kelong Mao, Zhicheng Dou, and Hongjin Qian. 2022. Curriculum contrastive context denoising for few-shot conversational dense retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 176–186.

Ida Mele, Cristina Ioana Muntean, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, and Ophir Frieder. 2020. *Topic Propagation in Conversational Search*, page 2057–2060. Association for Computing Machinery, New York, NY, USA.

Rodrigo Nogueira and Kyunghyun Cho. 2020. Passage re-ranking with bert.

Ronak Pradeep, Rodrigo Nogueira, and Jimmy Lin. 2021. The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models. *arXiv preprint arXiv:2101.05667*.

Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W. Bruce Croft, and Mohit Iyyer. 2020. *Open-Retrieval Conversational Question Answering*, page 539–548. Association for Computing Machinery, New York, NY, USA.

Filip Radlinski and Nick Craswell. 2017. A theoretical framework for conversational search. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval*, CHIIR '17, page 117–126, New York, NY, USA. Association for Computing Machinery.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Gary Ren, Xiaochuan Ni, Manish Malik, and Qifa Ke. 2018. Conversational query understanding using sequence to sequence modeling. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 1715–1724, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Pengjie Ren, Zhongkun Liu, Xiaomeng Song, Hongtao Tian, Zhumin Chen, Zhaochun Ren, and Maarten de Rijke. 2021. Wizard of search engine: Access to information through conversations with search engines. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 533–543.

Svitlana Vakulenko, Shayne Longpre, Zhucheng Tu, and Raviteja Anantha. 2020. A wrong answer or a wrong question? an intricate relationship between question reformulation and answer selection in conversational question answering. In *Proceedings of the 5th International Workshop on Search-Oriented Conversational AI (SCAI)*, pages 7–16.

Svitlana Vakulenko, Shayne Longpre, Zhucheng Tu, and Raviteja Anantha. 2021a. Question rewriting for conversational question answering. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, WSDM '21, page 355–363, New York, NY, USA. Association for Computing Machinery.

Svitlana Vakulenko, Nikos Voskarides, Zhucheng Tu, and Shayne Longpre. 2021b. Leveraging query resolution and reading comprehension for conversational passage retrieval. *arXiv preprint arXiv:2102.08795*.

Nikos Voskarides, Dan Li, Pengjie Ren, Evangelos Kanoulas, and Maarten de Rijke. 2020. *Query Resolution for Conversational Search with Limited Supervision*, page 921–930. Association for Computing Machinery, New York, NY, USA.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

Andrew Yates, Rodrigo Nogueira, and Jimmy Lin. 2021. Pretrained transformers for text ranking: Bert and beyond. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, WSDM '21, page 1154–1156, New York, NY, USA. Association for Computing Machinery.

Shi Yu, Jiahua Liu, Jingqin Yang, Chenyan Xiong, Paul Bennett, Jianfeng Gao, and Zhiyuan Liu. 2020. *Few-Shot Generative Conversational Query Rewriting*, page 1933–1936. Association for Computing Machinery, New York, NY, USA.

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.

A Implementation Details

In our CRDR, the *Query Rewriting* module is finetuned on a pretrained Bert model (bert-large-uncased[4]) with the CANARD dataset. The dataset is preprocessed as introduced in Appendix C. We set the batch size to 4, the max sequence length to 300. We use the AdamW optimizer with a learning rate of 5e-5. We train the model on the training set of CANARD for 8 epochs and apply the early stop strategy when the loss on the dev set does not decrease.

For the sparse retrieval, we use the Pyserini tool with its default parameter setting (Lin et al., 2021a). For the dense retrieval, we initialize our model with the ANCE-FirstP[5] checkpoints at 600k-th step (Xiong et al., 2020). For the CAsT-19 and CAsT-20 datasets, we perform five-fold cross-validation for 8 epochs with the Adam optimizer (learning rate 1e-5). We set the batch size as 4. The max sequence length is 256 and 512 for the CAsT-19 and CAsT-20 datasets, respectively. To facilitate the training of CRDR's *Dense Retrieval* module, we generate the relevant term labels using the human-labelled bag-of-word data[6]. During evaluation, the relevant terms are predicted by the *Query Rewriting* module. Following Yu et al. (2021), we first warm up the ANCE model by training one epoch on the CANARD dataset for the CAsT-20 dataset. Besides, all document embeddings are encoded by ANCE and fixed in all experiments. For the rerank phase, we use the BERT-large[7] trained on MS MARCO as the reranker (Nogueira and Cho, 2020). We did not fine-tune the reranker and only replaced the original queries with the CRDR's reformulated queries. For the hybrid search, we use the CombSUM algorithm with $depth = 1000$ (Fox and Shaw, 1994). All experiments are conducted on a Tesla V100 16G GPU. The results of all baselines are either implemented by us or from the original paper. All baseline results are equal to or higher than reported in the original papers.

[4]https://huggingface.co/bert-large-uncased/tree/main
[5]https://github.com/microsoft/ANCE
[6]https://github.com/svakulenk0/cast_evaluation
[7]https://github.com/nyu-dl/dl4marco-bert

B Baseline Models

For traditional inverted-index retrieval, we use the following baselines: (1) Original: the original query without modification; (2) Human: the manual oracle query annotated by humans; (3) Self-learn (Yu et al., 2020): it trains a query simplifier to generate conversational session queries and then fine-tune a GPT-2 with the generated data; (4) Rule-based (Yu et al., 2020): it applies heuristic rules to generate conversational session queries and then fine-tune a GPT-2 with the generated data; (5) Transformer++ (Vakulenko et al., 2021a): it fine-tune GPT-2 with the CANARD dataset. The decoder of the GPT-2 considers multiple token distributions when predicting the new query. (6) QuReTec (Voskarides et al., 2020): it trains a binary tagger to find relevant terms in the query context and then adds these relevant terms at the end of the original query. We use the following baselines for dense retrieval and hybrid search: (7) ConvDR (Yu et al., 2021): it inputs the whole query context into a pretrained query encoder to get a contextualized query embedding. (8) ConvDR (RRF) (Yu et al., 2021): it fuses the results from the best BERT ranker and ConvDR via RRF algorithm (Cormack et al., 2009).

C Dataset

In this paper, we conduct experiments on the two benchmark datasets TREC CAsT-19 and TREC CAsT-20, which are the most widely-used datasets to evaluate conversational search models. Besides, we apply the CANARD dataset to train the CRDR's *Query Rewriting* module. The details of the three datasets are as follow: (1) CAsT-19 (Dalton et al., 2020b): The TREC Conversational Assistance Track (CAsT) 2019 dataset is a benchmark dataset for conversational search. It provides 30 training dialogues and 50 test dialogues in which each dialogue is comprised of 9 to 10 queries. The corre-

Table 4: The statistics of the CAsT datasets.

	CAsT-19		CAsT-20
	Train	Eval	Eval
# Query	108	173	208
# Dialog	13	20	25
# Assessment	2,399	29,571	40,451
Fails to meet (0)	1,759	21,451	33,781
Slightly meet (1)	329	2,889	2,697
Moderately meet (2)	311	2,157	1,834
Highly meet (3)	0	1,456	1,408
Fully meet (4)	0	1,618	731

sponding passage corpus is a combination of MS MARCO and TREC Complex Answer Retrieval (CAR). All of the test queries in the CAsT 19 have manual oracle reformulated queries. (2) CAsT-20 (Dalton et al., 2020a): The CAsT-20 dataset is released in the following year of CAsT-19. It contains 25 test dialogues and shares the same passage corpus with CAsT-19. Except for query context, CAsT-20 also provides corresponding answers to each query. Thus, the query might also refer to previous answers. Table 4 shows the statistic information of the CAsT-19 and CAsT-20 datasets. (3) CANARD (Elgohary et al., 2019): The CANARD dataset is derived from the QuAC dataset (Choi et al., 2018). It contains 40,527 questions that have gold resolutions annotated by humans. The CANARD dataset does not have relevant term labels. We follow Voskarides et al. (2020) to label relevant terms in the original CANARD dataset with **REL** label. Moreover, we label the **IN** token in the original query by comparing the token-level differences between the manual oracle query and the original query. For example, for the original query "What is its population" and an manual oracle query "What is the Phoenix city's population?", by comparing we can find the word "its" is replaced by the words "he Phoenix city's". Then, we label the "its" word as **IN**.

D Case Study

We conduct case study on the CAsT-19 dataset to explore the quality of CRDR's reformulated queries and how the relevant terms effect the retrieval system's performance. We show the case study results in Table 5.

First, regarding the quality of query rewriting, CRDR generates the most similar query to the manual oracle query in the "sharks" topic and therefore outperforms other models in sparse retrieval. In the "Bronze Age collapse" topic, CRDR generates a query that does not follow the manual oracle query's word order as it chooses the token "their" as the entry word. However, CRDR still outperforms the other two query rewriting methods regarding the sparse retrieval's performances. The potential reason might be that the sparse retrieval is not sensitive to word orders. **Second**, regarding the performance of dense retrieval system, the two dense retrieval methods outperform all sparse retrieval methods in the "sharks" topic, which implies the [CLS] token succeed to summarize the important information in the query context accurately. Nevertheless, in the "Bronze Age collapse" topic, the two dense retrieval methods are inferior to the sparse retrieval methods, which implies the [CLS] token fails to highlight the important information in the context. Compared to ConvDR, CRDR further considers the relevant terms in the context and therefore outperform ConvDR in this case. Besides, we also find that after fusing the sparse retrieval and dense retrieval's results, the ranking performance decreases in the "sharks" but increases in the "Bronze Age collapse" topic, which implies that fusing multi-source retrieval results cannot consistently lead to improvement.

Table 5: Case study from CAsT-19. The blue and red words represent relevant and entry words detected by the CRDR's query rewriting module, respectively. The values in $(\cdot)$ are the nDCG@3 scores.

Topic: sharks
Query Context
q_1: What are the different types of sharks?
q_7: Tell me about makos.
q_8: What are their adaptations?
q_9: Where do they live?
q_{10}:What do they eat?
Query Rewriting
Human: What do Mako sharks eat?
QuReTec: What do they eat? sharks makos (0.074)
Transformer++: What do makos eat sharks (0.074)
CRDR: What do sharks makos eat? (0.309)
Dense Retrieval
ConvDR: Dense (0.765)
CRDR: Dense (0.765)
CRDR: Fused (0.296)

Topic: Bronze Age collapse
Query Context
q_1: Tell me about the Bronze Age collapse.
q_2: What is the evidence for it?
q_3: What are some of the possible causes?
q_4: Who were the Sea Peoples?
q_5: What was their role in it?
Query Rewriting
Human: What was their role in the Bronze Age collapse?
QuReTec: What was their role in it? peoples bronze collapse age sea (0.531)
Transformer++: What was Sea Peoples role in the bronze age collapse (0.531)
CRDR: What was sea peoples bronze age collapse's role in it? (0.704)
DenseRetrieval
ConvDR: Dense (0.0)
CRDR: Dense (0.235)
CRDR: Fused (0.852)