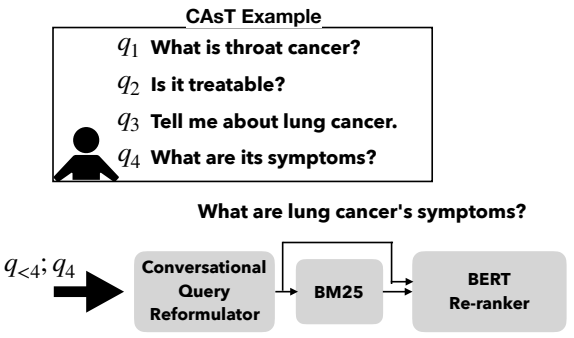
Contextualized Query Embeddings for Conversational Search

Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin

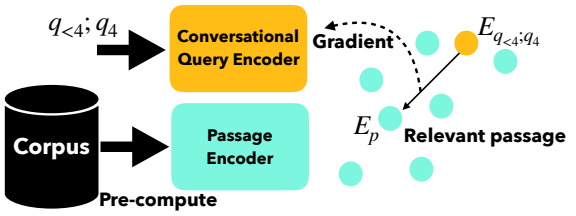
David R. Cheriton School of Computer Science
University of Waterloo

Abstract

This paper describes a compact and effective model for low-latency passage retrieval in conversational search based on learned dense representations. Prior to our work, the state-of-the-art approach uses a multi-stage pipeline comprising conversational query reformulation and information retrieval modules. Despite its effectiveness, such a pipeline often includes multiple neural models that require long inference times. In addition, independently optimizing each module ignores dependencies among them. To address these shortcomings, we propose to integrate conversational query reformulation directly into a dense retrieval model. To aid in this goal, we create a dataset with pseudo-relevance labels for conversational search to overcome the lack of training data and to explore different training strategies. We demonstrate that our model effectively rewrites conversational queries as dense representations in conversational search and open-domain question answering datasets. Finally, after observing that our model learns to adjust the L_2 norm of query token embeddings, we leverage this property for hybrid retrieval and to support error analysis.



(a) Multi-stage pipeline for conversational search: fine-tuned language models are used to reformulate user utterances for the downstream IR pipeline.



(b) End-to-end conversational search: query reformulation is directly incorporated into the IR pipeline, thus enabling end-to-end training.

Figure 1: A comparison between (a) a multi-stage pipeline and (b) our proposed method for conversational search.

1 Introduction

With the growing popularity of virtual assistants (e.g., Alexa and Siri), information seeking through dialogues has attracted many researchers' attention. To facilitate research on conversational search (ConvS), Dalton et al. (2019) organized the TREC Conversational Assistance Track (CAsT) and defined ConvS as the task of iteratively retrieving passages in response to user queries in a conversation session. An example conversation in the CAsT dataset is shown at the top of Figure 1(a).

There are two main challenges for the task of conversational search: (1) User utterances are often ambiguous when treated as stand-alone queries since omission, coreference, and other related linguistic phenomena are common in natural human dialogues. Hence, directly feeding the utterances into IR systems would lead to poor retrieval effectiveness. Understanding queries through conversational context is required. (2) There is limited data regarding conversational search for model training. To address the aforementioned challenges, existing papers (Lin et al., 2021c; Yu et al., 2020; Voskarides et al., 2020; Kumar and Callan, 2020) take a multi-stage pipeline approach. They train a conversational query reformulation (CQR) model using publicly available datasets (Elgohary et al., 2019; Quan et al., 2019) and feed the automatically decontextualized queries to an off-the-shelf IR pipeline (Nogueira and Cho, 2019). However, such ConvS pipelines can be slow (i.e., over 10s per

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 1004–1015
November 7–11, 2021.

query on GPUs). Furthermore, this design assumes that the reformulated queries are independent of the downstream IR pipeline, which may not be true.

In this paper, we study a low-latency end-to-end approach to ConvS. Specifically, we adopt a bi-encoder model and incorporate CQR into the query encoder, illustrated in Figure 1(b). To overcome the challenge of limited training data, we create a dataset with pseudo-relevance labels to guide the query encoder to rewrite conversational queries in latent space directly. One may consider this approach as throwing conversational queries into a black box since the reformulated queries are represented as dense vectors. However, we find that the fine-tuned contextualized query embeddings (CQE) are easily interpretable. They can be transformed into text for failure analysis and can facilitate dense–sparse hybrid retrieval.

Our contributions are summarized as follows: (1) We integrate two tasks in ConvS, query reformulation and dense passage retrieval, into our dense representation learning framework. Due to the lack of human labeled data, we create a dataset with pseudo-relevance labels for model training. We empirically show that our model successfully learns to reformulate conversational queries in a latent representation space. (2) We uncover how CQE learns to reformulate conversational queries in a latent space. Based on this finding, we can easily transform CQE into a text (sparse) representation. We demonstrate that the CQE text representation also performs well on sparse retrieval and can further improve CQE retrieval effectiveness using a hybrid of sparse and dense retrieval. The CQE text also helps us understand why the technique fails or succeeds. (3) We show that the query latency of CQE (without re-ranking) is at least an order of magnitude lower than existing multi-stage ConvS pipelines while yielding competitive retrieval effectiveness. Hence, CQE is superior for integration with other models in downstream tasks. (4) We empirically demonstrate its effectiveness in open-domain conversational question answering in a zero-shot setting.

2 Preliminaries

Let us define a sequence of conversational queries $Q = (q_1, \cdots, q_{i-1}, q_i)$ for a topic-oriented session s, where q_i stands for the i-th user query ($i \in \mathbb{N}^+$) in the session. The goal of conversational search is to find the set of relevant passages P_i^+ for the user query q_i at each turn, given the conversational context $q_{<i}$. Thus, the task can be formulated as the objective:

$$\arg\max_{\theta} \sum_{p \in P_i^+} \mathcal{F}_\theta^{\text{ConvS}}(q_{<i}; q_i, p) \qquad (1)$$

where $\mathcal{F}_\theta^{\text{ConvS}}$ is the function (parameterized by θ) to compute a relevance score between the conversational query $(q_{<i}; q_i)$ and passage p.

Since end-to-end training data for conversational search is extremely limited, a common approach is to factorize $\mathcal{F}_\theta^{\text{ConvS}}$ into a multi-stage pipeline. In a multi-stage pipeline, the components can be tuned with data collected at different stages:

$$\mathcal{F}_\theta^{\text{ConvS}} \triangleq \mathcal{F}_\phi^{\text{ir}}(q_i^*, p) \cdot \mathcal{F}_\varphi^{\text{cqr}}(q_i^* | q_{<i}; q_i), \qquad (2)$$

where q_i^* is the stand-alone oracle query that best represents the user's information need given the context $q_{<i}; q_i$. $\mathcal{F}_\phi^{\text{ir}}$ and $\mathcal{F}_\varphi^{\text{cqr}}$ denote the components of information retrieval (IR) and conversational query reformulation (CQR), respectively. Thus, Eq. (1) can be approximated by separately maximizing $\mathcal{F}_\phi^{\text{ir}}$ and $\mathcal{F}_\varphi^{\text{cqr}}$. For example, we can reuse the representative *ad hoc* retrieval pipeline comprised of BM25 + BERT re-ranking for $\mathcal{F}_\phi^{\text{ir}}$, then conduct the CQR task for $\mathcal{F}_\varphi^{\text{CQR}}$.

Specifically, the most common current approach (Lin et al., 2021c; Voskarides et al., 2020; Vakulenko et al., 2020; Kumar and Callan, 2020; Yu et al., 2020) is to fine-tune a pretrained language model (LM) supervised by decontextualized queries manually rewritten by humans, and then use the fine-tuned LM to reformulate user queries for BM25 retrieval and BERT re-ranking, as illustrated in Figure 1(a). While effective, this approach has two limitations: (1) although mimicking the way humans rewrite queries is reasonable, it hypothesizes that the optimal decontextualized queries are manually rewritten queries, which may not be true; (2) the CQR and IR modules rely on computation-demanding pretrained LMs; thus, when combined together, they are often too slow for practical applications.

3 Our Approach

In this section, we first explain why a bi-encoder design is a good fit for conversational search, and then introduce our contextualized query embeddings (CQE) for conversational search.

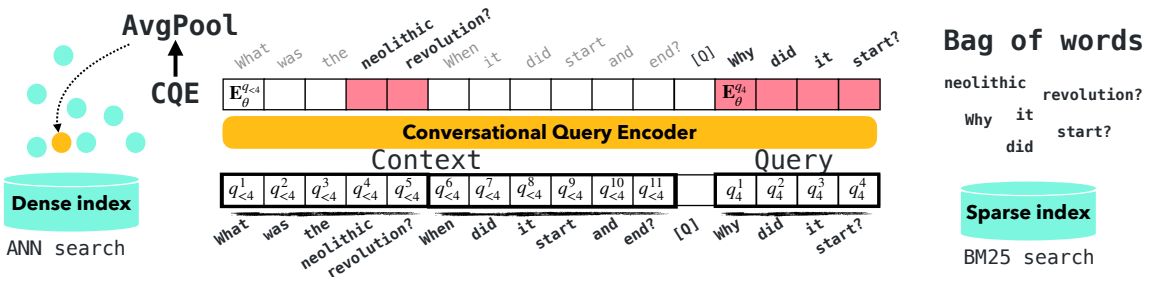


Figure 2: Our contextualized query token embeddings can be used both for dense and sparse retrieval. The left side illustrates CQE for dense retrieval by average pooling of token embeddings. The right side shows that the token embeddings can be used to select tokens from the context to form a decontextualized bag-of-words query for sparse retrieval.

3.1 Bi-encoder Model

Recently, dense passage retrieval based on bi-encoders (Reimers and Gurevych, 2019; Karpukhin et al., 2020; Xiong et al., 2021; Lin et al., 2021b) has attracted the attention of researchers due to its good balance between efficiency and effectiveness. Bi-encoder models are trained to encode queries and passages in a shared latent space. At query time, only query texts are encoded to search for the nearest passage embeddings, which are precomputed by the passage encoder. Formally speaking, the relevance score ϕ of a given query q_i (with its context $q_{<i}$) and a passage p is computed as the dot product of their embeddings:

$$\phi\left((q_{<i}; q_i), p\right) = \langle \mathbf{E}_\theta^{(q_{<i};q_i)}, \mathbf{E}_\theta^{p} \rangle, \qquad (3)$$

where $\mathbf{E}_\theta^{(\cdot)} \in \mathbb{R}^h$ is the BERT representation of the input texts, which can be the average or maximum pooling over token embeddings or a specific token embedding (e.g., the [CLS] embedding in BERT), and θ represents the parameters of BERT.

In this study, we adopt average pooling over token embeddings, which lets us interpret CQE easily, as we will discuss later. Thus, we can formulate conversational search as maximizing the following log likelihood:

$$\mathcal{F}_\theta^{\text{ConvS}}\left((q_{<i}; q_i), p\right) \triangleq \log \frac{\exp\left(\langle \mathbf{E}_\theta^{(q_{<i};q_i)}, \mathbf{E}_\theta^{p} \rangle / \tau\right)}{\sum_{p' \in \mathcal{D}} \exp\left(\langle \mathbf{E}_\theta^{(q_{<i};q_i)}, \mathbf{E}_\theta^{p'} \rangle / \tau\right)}, \qquad (4)$$

where τ denotes the temperature parameter and $\mathcal{D}$ is the set of passages comprising the corpus. In practice, $\mathcal{D}$ is replaced by the subset $\mathcal{D}_\mathcal{B}$, consisting of the passages in a training batch, i.e., the positive and negative passages from all the queries in the same batch. With Eq. (4), the optimization problem of Eq. (1) can be approached by end-to-end representation learning, which can be interpreted as projecting a conversational query $\mathbf{E}_\theta^{(q_{<i};q_i)}$ into the latent space such that it has maximum dot product with its relevant passage p^+.

3.2 Contextualized Query Embeddings

Given the conversational context and query tokens $(q_{<i}^1 \cdots q_{<i}^j, q_i^1 \cdots q_i^k)$, we define contextualized query embeddings (CQE) formally as $\mathbf{E}_\theta^{\text{cqe}} \in \mathbb{R}^{(j+k)\times h}$ based on BERT's contextualized token embeddings:

$$(\overbrace{\mathbf{E}_\theta^{q_{<i}}(1) \cdots \mathbf{E}_\theta^{q_{<i}}(j)}^{\text{context}}, \overbrace{\mathbf{E}_\theta^{q_i}(j+1) \cdots \mathbf{E}_\theta^{q_{<i}}(j+k)}^{\text{query}}). \qquad (5)$$

Here, we take the last layer's hidden representations from BERT. From $\mathbf{E}_\theta^{\text{cqe}}$, a single vector query embedding can be computed by average pooling the query token embeddings:

$$\mathbf{E}_\theta^{(q_{<i};q_i)} = \frac{1}{j+k} \sum_{l=1}^{j+k} \mathbf{E}_\theta^{\text{cqe}}(l) \in \mathbb{R}^h. \qquad (6)$$

We then use $\mathbf{E}_\theta^{(q_{<i};q_i)}$ to conduct nearest neighbor search for the top-k passages in the corpus using an off-the-shelf library (Facebook's Faiss, in our case), as shown in Figure 2 (left).

3.3 Interpreting CQE

While condensing a multi-stage pipeline into single-stage dense retrieval is attractive, it may be difficult for interpretation (i.e., we cannot examine the reformulated queries). In this subsection, we explain how to interpret CQE. With Eq. (6), we rewrite Eq. (3) as the average dot product of each token

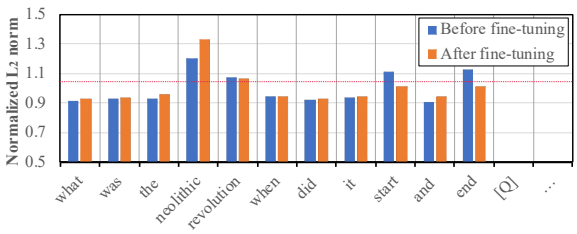


Figure 3: L_2 norm distribution of context token embeddings, normalized by the mean of L_2 norms among the context tokens. After fine-tuning, L_2 norms of context tokens that are non-relevant (e.g., start, end) decrease, and the relevant ones (e.g., neolithic) increase.

embedding $\mathbf{E}_\theta^{\text{cqe}}(l)$ and a single-vector passage embedding $\mathbf{E}_\theta^p$:

$$\phi\left((q_{<i}; q_i), p\right) = \frac{1}{j+k} \sum_{l=1}^{j+k} \|\mathbf{E}_\theta^{\text{cqe}}(l)\|_2 \langle \hat{\mathbf{E}}_\theta^{\text{cqe}}(l), \mathbf{E}_\theta^p \rangle, \quad (7)$$

where $\hat{\mathbf{E}}_\theta^{\text{cqe}}(l)$ is a unit vector. Intuitively, to maximize Eq. (4), CQE can learn to adjust the L_2 norm of $\mathbf{E}_\theta^{\text{cqe}}(l)$ when we freeze the passage embeddings. To be more specific, it appears that CQE learns to increase the L_2 norm for relevant query–passage pairs and decrease it otherwise. Thus, we can consider the L_2 norm of each token embedding as its term importance for retrieving relevant passages.

For the example in Figure 2, we empirically analyze the query token embeddings of our CQE model. Figure 3 shows the normalized L_2 norm for the context of the user query ("*why did it start?*"). We observe that after fine-tuning, the terms "*neolithic*" and "*revolution*" show greater L_2 norms than the others. On the other hand, the L_2 norms for the terms "*start*" and "*end*" decrease.

With this observation, we can use CQE to generate decontextualized queries. Specifically, inspired by the term weighting ideas of Dai and Callan (2020), we conduct query expansion by selecting the terms ($\|\mathbf{E}_\theta^{q<i}(\cdot)\|_2 \geq \gamma$, where γ is a hyperparameter) from the context using CQE and concatenate them to the user query q_i, illustrated on the right side of Figure 2. Note that the decontextualized queries generated by CQE are bag-of-words sets rather than fluent natural language queries. However, in Section 5, we show that the decontextualized queries can be used for sparse retrieval and even for conducting failure analysis.

Table 1: CANARD dataset statistics.

CANARD	Training	Dev	Test
# Queries	31,526	3,430	5,571
# Dialogues	4,383	490	771

Table 2: CAsT dataset statistics.

	CAsT19		CAsT20
	Training	Eval	Eval
# Queries	108	173	208
# Dialogues	13	20	25
# Passages		38M	

4 Training Data and Strategies

In this section, we first introduce how we create weakly supervised training data for conversational search. Then, we discuss some possible strategies to fine-tune CQE.

Weakly supervised training data. By taking the idea of pseudo-labeling, we create our weakly supervised training data for end-to-end conversational search. There are human rewritten queries that help models learn to decontextualize them in conversation; however, only limited labels are available for end-to-end conversational search, as shown in Table 2. Hence, we combine three existing resources to train our model with weak supervision: (1) CANARD (Elgohary et al., 2019), a conversational query reformulation dataset; (2) ColBERT (Khattab and Zaharia, 2020), a strong text ranking model trained on MS MARCO for passage ranking; and (3) the passage collection provided by the TREC CAsT Tracks (Dalton et al., 2019).

To combine the three resources, we make a simple assumption: decontextualized queries can be paired with their relevant passages selected by "good enough" *ad hoc* retrieval models. Thus, for each human reformulated query in the CANARD dataset, we retrieve 1000 candidate passages from the CAsT collection using BM25, and then re-rank them using ColBERT. We assume the top-3 passages are relevant for each query, while treating the rest as non-relevant.

Bi-encoder warm-up. Training a bi-encoder model for dense retrieval requires lots of data, not to speak of conversational search. Following previous work on conversational search (Yu et al., 2020; Lin et al., 2021c; Vakulenko et al., 2020), we adopt MS MARCO as our bi-encoder warm-up training

dataset, where the training procedure is adopted from the work of Lin et al. (2020).

CQE fine-tuning. After bi-encoder warm-up, we fine-tune the *query* encoder to consume conversational queries and generate contextualized query embeddings. Specifically, for each query q_i in our training data, we sample a triplet $([q_{<i}; q_i], p^+, p^-)$ for fine-tuning, where p^+ and p^- are sampled from positive passages (labeled by ColBERT) and top-200 BM25 passages (without replacement), respectively. Note that, at this stage, we freeze our passage encoder and only fine-tune the query encoder; thus, we can precompute all the passage embeddings in the CAsT corpus, and only encode queries for evaluation. In this work, we further explore different strategies to better train CQE using our weakly supervised training data.

Hard negative mining. Although sampling negatives from BM25 top-k candidates is effective for dense retrieval training, Xiong et al. (2021) demonstrate that hard negatives bring more useful information for training dense retrievers. In this work, we explore whether hard negatives benefit the fine-tuning of our CQE model. Instead of using asynchronous index refreshing, as in the work of Xiong et al. (2021), we sample hard negatives p^- from the top-200 passages re-ranked by ColBERT.

Training with soft labels. Due to the strong assumptions we make for weak supervision, using cross entropy for one-hot pseudo-label training may be sub-optimal because our model could be overconfident about its predictions. To address this issue, we use the logits of ColBERT as soft labels to fine-tune CQE to have similar confidence predictions, i.e., knowledge distillation. It is worth noting that we only minimize the KL divergence of softmax normalized dot products with respect to in-batch query–passage pairs without using cross entropy for interpolation, as in the traditional (strongly) supervised setting.

5 Experimental Setup

Datasets. We conduct experiments on TREC CAsT datasets. TREC organized the Conversational Assistance Tracks (Dalton et al., 2019), aiming to collect reusable collections for conversational search. The organizers have created relevance judgments (relevance grades 0–4) for each query using assessment pools from participants. In total, there are three datasets available, CAsT19 (training and eval) and CAsT20 (eval).[1] The dataset statistics are listed in Table 2. All relevant passages come from the CAsT corpus (consisting of 38M passages). In addition, we demonstrate the generalization of CQE on an open-domain conversational question answering dataset ORConvQA in a zero-shot setting, detailed in Section 6.2.

Query reformulation baselines. A reasonable setting to compare CQE with existing query reformulation models is to directly feed the reformulated queries into a bi-encoder model for dense retrieval. For a fair comparison, we encode the reformulated queries into query embeddings using our pretrained bi-encoder model, which is suitable for stand-alone queries. Note that the passage embeddings in the corpus for CQE and the other models are the same since we freeze the passage encoder while fine-tuning CQE. We compare CQE with three state-of-the-art conversational query reformulation models and a human baseline, described below:

- **Few-Shot Rewriter**: Yu et al. (2020) fine-tune the pretrained sequence-to-sequence LM GPT2-medium on CAsT manually reformulated queries and synthetic queries created by a rule base. For the CAsT19 and CAsT20 eval sets, we directly use their publicly released queries.[2]
- **QuReTeC**: Voskarides et al. (2020) conduct query expansion using BERT-large as a term classifier, which is fine-tuned on the CANARD dataset. We directly use the reformulated queries provided by the authors.[2]
- **NTR (T5)**: Lin et al. (2021c) fine-tune the pretrained sequence-to-sequence LM, T5-base, on the CANARD dataset. Following their work, we use the released model[2] with beam-search inference (setting width to 10).
- **Humans**: We also conduct experiments on the manually reformulated queries provided by the TREC CAsT organizers as a reference.

Since CQE can be used to decontextualize conversational queries, as discussed in Section 3.3, we also apply CQE reformulated queries (denoted CQE-sparse) to sparse retrieval (after conversion to text). The optimal L_2 threshold γ (10.5) is tuned on the CAsT19 training set.

[1] https://github.com/daltonj/treccastweb
[2] Few-Shot Rewriter, QuReTeC, NTR (T5)

Table 3: CAsT passage retrieval effectiveness comparisons. The best automatic approach in the same comparison group (sparse/dense) is bolded. Superscripts denote significant differences with respect to CQE (paired t-test $p < 0.05$); * denotes significant difference between CQE-hybrid and all the other automatic approaches. W/T denotes # of queries win/tie against human queries.

		# Params	Latency	CAsT19 Eval				CAsT20 Eval			
	Query	millions	ms/q	Recall	nDCG	nDCG@3	W / T	Recall	nDCG	nDCG@3	W / T
Sparse	(0) Humans	-	-	.803	.510	.309	-	.707	.423	.240	-
	(1) Few-Shot Rewriter	355	582	.717	.438	.248	10 /126	.490	.284	.145	23 /121
	(2) QuReTeC	340	29	.768	**.485**	**.296**	27 /117	.508	.291	.136	18 /115
	(3) NTR (T5)	220	307	.753	.471	.295	15 /136	.514	.303	.159	24 /115
	(4) CQE-sparse	110	11	**.773**^{1}	.462	.272	**45** / 67	**.582**$^{1-3}$	**.332**$^{1-2}$	**.172**^{2}	**38** / 89
Dense	(0) Humans	-	-	.797	.571	.507	-	.804	.558	.460	-
	(1) Few-Shot Rewriter	465	593	.723	.510	.449	18 /117	.611	.378	.256	10 /108
	(2) QuReTeC	450	40	.773	.545	.473	38 / 82	.600	.390	.288	26 / 78
	(3) NTR (T5)	330	318	.762	.543	.495	23 /125	.635	.421	**.323**	28 / 90
	(4) CQE	110	11	**.784**^{1}	**.559**^{1}	**.499**^{1}	**60** / 45	**.699**$^{1-3}$	**.447**$^{1-3}$	.312$^{1-2}$	**40** / 46
	CQE-hybrid	110	11	.823*	.598*	.515	-	.730*	.475*	.338	-

Model details. We fine-tune CQE using BERT-base for 10K steps with batch size 96 and learning rate 7×10^{-6} on all queries (training, dev, and test) in the CANARD dataset (see Table 1), and use the CAsT19 training set as our development set. In our main experiments, we use our best training strategy, combining hard negative mining and soft labeling (see the ablation study in Section 6.3). We perform dense retrieval using Faiss (Johnson et al., 2017) (Faiss-GPU, brute force) and sparse retrieval using Pyserini (Lin et al., 2021a) (BM25, $k_1 = 0.82, b = 0.68$). In addition, we measure the latency of conversational query reformulation for each model (Latency). For CQE, we report the latency of generating the contextualized query embeddings. Note that for encoder-only models (BERT), we set maximum input length to 150, while for decoder-only and encoder-decoder models (GPT and T5), we further set maximum output length to 32 and use greedy search decoding. All latency measurements are from Google Colab using a single GPU (12GB NVIDIA Tesla K80). Finally, we report the size of each model (# Params).

Evaluation metrics. Following Dalton et al. (2020), for each approach, we compare overall retrieval effectiveness using nDCG and recall (at cutoff 1000), and top-ranking accuracy using nDCG@3. For recall, we take relevance grade ≥ 2 as positive. The evaluation is conducted using the `trec_eval` tool. In addition, for each model, we report the number of queries win (tie) against manual queries on nDCG@3. All significance tests are conducted with paired t-tests ($p < 0.05$).

6 Results

6.1 Results on CAsT

First-stage retrieval comparisons. Table 3 reports the sparse and dense retrieval effectiveness of various methods. Overall, dense retrieval yields better effectiveness than BM25 retrieval. Observing the first block in Table 3, CQE-sparse yields reasonable effectiveness compared to the other CQR models, indicating that CQE can be well represented with text. As for dense retrieval, CQE is able to beat the other CQR models. Although NTR (T5) and CQE yield comparable top-ranking accuracy, it is worth mentioning that unlike CQE, the other CQR modules are built independently. Thus, when incorporated with dense retrieval, the overall memory and latency required increase, i.e., # params of NTR (T5) increases from 220M to 330M and is much slower.

Finally, we also conduct CQE dense–sparse hybrid retrieval using their linear score combination (denoted by CQE-hybrid); see Appendix A for detailed settings. CQE-hybrid retrieval effectiveness shows significant gains over CQE dense only. The gains from the dense–sparse hybrid suggest that the textual interpretation of CQE not only helps us understand the query reformulation mechanism in dense retrieval but also improves effectiveness, all using a single, unified model.

A comparison of win (tie) entries shows that CQE has more wins against human queries than all the other CQR models. On the other hand, the other CQR models have relatively more ties against human queries than CQE. The difference between

Title: Bronze Age collapse (CAsT19 session 34)
Turn1: Tell me about the Bronze Age collapse. Turn2: What is the evidence for it? Turn3: What are some of the possible causes? Turn4: Who were the Sea Peoples? Query: What was their role in it?
Humans: What was their role in the Bronze Age collapse?
CQE: ***bronze age collapse*** ***sea peoples*** What was their role in it?
Title: prison psychology studies (CAsT19 session 37)
Turn1: What was the Stanford Experiment? Turn2: What did it show? Turn3: Tell me about the author of the experiment. Turn4: Was it ethical? Turn5: What are other similar experiments? Query: What happened in the Milgram experiment?
Humans: What happened in the Milgram experiment?
CQE: ***Stanford*** What happened in the Milgram experiment?

(a) Cases where CQE wins over humans

Title: Washington DC tourism (CAsT19 session 54)
Turn1: What is worth seeing in Washington D.C.? Turn2: Which Smithsonian museums are the most popular? Turn3: Why is the National Air and Space Museum important? Turn4: Is the Spy Museum free? Query: What is there to do in DC after the museums close?
Humans: What is there to do in Washington D.C. after the museums close?
CQE: ***washington*** ***smithsonian space spy*** What is there…museums close?
Title: Avengers and Superhero Universes (CAsT19 session 61)
Turn1: Who are The Avengers? Turn2: Tell me about their first appearance. Turn3: Who is the most powerful and why? Turn4: What is the relationship of Spider-Man to the team? Turn5: Why is Batman not a member? Query: What is an important team in the DC universe?
Humans: What is an important team in the DC universe?
CQE: ***avengers batman*** What is an important team in the DC universe?

(b) Cases where CQE loses against humans

Figure 4: Case studies. We choose cases based on nDCG dense retrieval scores; the CQE text shown is for sparse retrieval. Underline denotes terms *not* appearing in human queries.

Table 4: Comparisons to SOTA multi-stage pipelines.

CAsT19 Eval	nDCG@3
BERT-base: latency = 314 ms	
CQE	.499
CQE-hybrid	.515
CQR + BM25 + BERT-base: latency = 5,350 ms	
QuReTec (Voskarides et al., 2020)	.476
Few-Shot Rewriter (Yu et al., 2020)	.492
3CQR + BM25 + BERT-base: latency = 8,025 ms (est.)	
MVR (Kumar and Callan, 2020)	.565
CQR + BM25 + BERT-large: latency = 16,450 ms	
Transformer++ (Vakulenko et al., 2020)	.529
NTR (T5) (Lin et al., 2021c)	.556
HQE + NTR (T5) (Lin et al., 2021c)	.565

the queries is probably because CQE learns to reformulate conversational queries through the guide of pseudo-relevant passages, meaning that CQE approaches the task in a different way from the other CQR models, which are trained to mimic the way humans reformulate queries. This observation indicates that CQE provides different "views" from other CQR models and could further benefit from fusion with state-of-the-art CQR models, which we demonstrate in Appendix B.

Multi-stage pipeline comparisons. We compare our CQE method with other multi-stage pipelines in terms of top-ranking effectiveness, reported in Table 4. All of these pipelines consist of conversational query reformulators (CQR), BM25 retrieval, and BERT re-ranking. Here, we also list systems that use a BERT-large re-ranker for reference. As for the retrieval latency, since the CAsT corpus requires 55 GiB for the dense vector index, we measure the latency of CQE on two V100 GPUs. For the other BERT re-ranking pipelines, we divide the numbers reported in Khattab and Zaharia (2020), which is measured on a single V100 GPU, by two for a fair comparison. We observe that *single-stage* CQE (with much lower latency) can compete with all the multi-stage pipelines that use a BERT-base re-ranker, except for MVR, which fuses three re-ranked lists from three different neural CQR models. As expected, re-ranking with BERT-large can yield higher effectiveness, but is also much slower. Of course, we can take CQE results and further re-rank them also.

Case studies. We demonstrate how CQE reformulates queries by comparing CQE and human reformulated queries on the CAsT19 eval set. Figure 4(a) shows cases where CQE beats humans in terms of nDCG (in the dense retrieval setting). The first example shows that humans mistakenly rewrite the query by omitting "*sea peoples*" in the context. The second example shows that humans reformulate the query correctly; however, CQE further adds the key term "*Stanford*" to the original query and obtains better ranking accuracy. These cases tell us that manually reformulated queries may not be optimal for the downstream IR pipeline, and CQE can actually do better. On the other hand, Figure 4(b) illustrates cases where CQE performs worse than humans. In both cases, we observe that CQE adds related terms (i.e., "*avengers*" and "*batman*"), but these terms degrade retrieval effectiveness. This suggests that a better negative sampling strategy may be required to guide CQE to select key terms and generate more accurate embeddings under such challenging contexts.

Table 5: Results on the ORConvQA eval set.

Model	Retriever		Reader
	Recall	MRR	F_1
BERTserini (Yang et al., 2019b)	.251	.178	26.0
Qu et al. (2020)	.314	.225	29.4
CQE	.365	.266	30.5
CQE-hybrid	.415	.310	32.0

Table 6: Ablation study on CAsT19.

Cond.	Strategy		CAsT19 Train	
	Soft label	Hard neg.	Recall	nDCG
(0)	No training		.160	.050
(1)			.670	.300
(2)	✓		.723	.344
(3)		✓	.660	.306
(4)	✓	✓	**.734**	**.353**

6.2 Zero-shot Transfer to ORConvQA

In this section, we examine the effectiveness of CQE on a downstream task: open-domain conversational question answering. The ORConvQA dataset is built on QuAC (Choi et al., 2018), a conversational question answering (ConvQA) dataset. To better approximate open-domain ConvQA, Qu et al. (2020) share an extensive corpus using the English Wikipedia dump from Oct. 20th, 2019. Then, they split 5.9 million Wikipedia articles into passage chunks with at most 384 BERT WordPiece tokens, resulting in a corpus of 11M passages. Thus, the task is to first retrieve passages from the corpus using conversational queries and then extract answer spans from the retrieved passages. Since the task shares the same conversational queries as our created dataset (both are built on CANARD), we fine-tune CQE only on the training set listed in Table 1. For a fair comparison between the retrievers, we directly use the reader provided by Qu et al. (2020),[3] which extracts the answer span from the top-5 retrieved passages.

We first compare our CQE retrieval effectiveness to baselines, where the numbers are from Qu et al. (2020). To fairly compare with the dense retriever of Qu et al. (2020) (with 128 dimensions), we first conduct unsupervised dimensionality reduction using Faiss (OPQ128, 1VF1, PQ128) from 768 to 128 dimensions. As shown in Table 5, CQE beats the other models in terms of retrieval effectiveness. It is worth noting that the baselines are fine-tuned on ORConvQA, with the passages containing answer spans as positives. In contrast, CQE is only fine-tuned on our weakly supervised training data. This difference suggests that CQE has a degree of generalization capability. More importantly, we observe that the retrieval effectiveness gain from CQE directly benefits F_1 scores. Finally, with hybrid retrieval, a unique feature of CQE, we further improve both retrieval and the downstream task.

[3] https://github.com/prdwb/orconvqa-release

6.3 Fine-Tuning Ablation

We explore different training strategies with our weakly supervised training data. We use the CAsT19 training set for evaluation and the results are reported in Table 6. Recall that, while training without hard negative sampling, we use the negatives randomly sampled from BM25 top-200 candidates. First, we observe that simply training with our pseudo-labeled data can effectively guide model training; see condition (1) vs. (0). In addition, training with ColBERT's soft labels brings substantial effectiveness gains, as shown by condition (4) vs. (3) and condition (2) vs. (1). Finally, although hard negative samples cannot directly enhance CQE's retrieval effectiveness, from condition (3) vs. (1), by combining soft labels, a modest effectiveness gain can still be observed, from condition (4) vs. (2). Thus, the best strategy for fine-tuning CQE on our weakly supervised training data is to combine hard negative sampling and soft labeling.

7 Related Work

Conversational search. Radlinski and Craswell (2017) define conversational search as addressing users' information needs through multi-turn conversational interactions, which can be classified into two scenarios: (1) A user is searching for a single item through multi-turn query clarifications, which has been studied by Aliannejadi et al. (2019); Ahmad et al. (2018); Hashemi et al. (2020). (2) A user is searching for multiple items surrounding a topic. For example, when planning a vacation, a user would query some source of knowledge (possibly, even a human expert) to find information about destinations, hotels, transportation, etc. through conversational interactions. Our work belongs to the latter search scenario.

Query reformulation. TREC organizers have built standard benchmark datasets, CAsT (Dalton et al., 2019), to facilitate research on conversational

search. Existing work built on CAsT mainly focuses on conversational query reformulation, previously studied by Ren et al. (2018); Rastogi et al. (2019). For example, Voskarides et al. (2019); Yang et al. (2019a) perform rule-based query expansion from dialogue context. Yu et al. (2020); Voskarides et al. (2020); Vakulenko et al. (2020); Lin et al. (2021c) fine-tune pretrained language models to mimic the way humans rewrite conversational queries. These papers demonstrate that building a CQR model on top of IR systems works well. However, Lin et al. (2021c); Kumar and Callan (2020) point out that human reformulated queries may not be optimal for downstream IR modules. They further address this problem by fusing the ranked lists retrieved using different CQR models; however, these solutions still rely on multi-stage pipelines. In contrast, this work explores a *single-stage*, *end-to-end* approach to conversational search.

Conversational question answering. Another related thread of research is conversational question answering (ConvQA) (Reddy et al., 2019; Choi et al., 2018), a downstream task of conversational search. Most related work (Qu et al., 2019a,b) focuses on improving answer span extraction using dialogue context information. Qu et al. (2020) first create an open-domain ConvQA dataset on top of QuAC (Choi et al., 2018) and then tackle this dataset with a pipeline consisting of a retriever and a reader. In this work, we demonstrate that weakly supervised CQE can directly serve as a strong retriever without further fine-tuning, and it improves the accuracy of answer span extraction. Furthermore, different from Qu et al. (2020), CQE provides a single model that supports dense–sparse hybrid retrieval for conversational search, which further improves retrieval effectiveness.

8 Conclusions

In this paper, we study how to simplify the multi-stage pipeline for conversational search and propose to integrate modules for conversational query reformulation (CQR) and dense passage retrieval into our dense representation learning framework. To address the lack of training data for conversational search, we create a dataset with pseudo-relevance labels and explore different training strategies on this dataset. Experiments demonstrate that our model learns to reformulate conversational queries in a latent space and generates contextualized query embeddings (CQE) for conversational search. In addition, our analyses provide insight into how CQE learns to rewrite conversational queries in this latent space. Finally, we show that there are two main advantages of CQE: First, the effectiveness of CQE is on par with state-of-the-art multi-stage pipelines for conversational search, but with much lower query latency. Second, CQE serves as a strong dense retriever for open-domain conversational question answering.

Limitations and future work. Our work shows the feasibility of integrating conversational query reformulation and *ad hoc* retrieval into a bi-encoder dense representation learning framework. However, it is unclear whether the same strategy can be applied to a cross-encoder re-ranker, which, although much slower, still achieves the highest levels of effectiveness. Another limitation of our work is that only historical queries are considered as context; nevertheless, in a real conversational scenario, other types of contexts should also be considered, e.g., system responses and conversations between multiple speakers (if present). There is still much to explore around dense representations in these scenarios, which we leave to future work. Finally, as shown in Gao et al. (2020), incorporating sparse retrieval signals into the training of dense retrieval improves dense–sparse fusion effectiveness. We suspect that there is more to be gained from better fusion of dense and sparse results for conversational search.

Acknowledgements

This research was supported in part by the Canada First Research Excellence Fund and the Natural Sciences and Engineering Research Council (NSERC) of Canada. Additionally, we would like to thank the support of Cloud TPUs from Google's TPU Research Cloud (TRC).

References

Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. 2018. Multi-task learning for document ranking and query suggestion. In *Proc. ICLR*.

Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. 2019. Asking clarifying questions in open-domain information-seeking conversations. In *Proc. SIGIR*, pages 475–484.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettle-

moyer. 2018. QuAC: Question answering in context. In *Proc. EMNLP*, pages 2174–2184.

Zhuyun Dai and Jamie Callan. 2020. Context-aware term weighting for first stage passage retrieval. In *Proc. SIGIR*, page 1533–1536.

Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2019. CAsT 2019: The conversational assistance track overview. In *Proc. TREC*.

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

Ahmed Elgohary, Denis Peskov, and Jordan Boyd-Graber. 2019. Can you unpack that? Learning to rewrite questions-in-context. In *Proc. EMNLP*, pages 5917–5923.

Luyu Gao, Zhuyun Dai, Zhen Fan, and Jamie Callan. 2020. Complementing lexical retrieval with semantic residual embedding. *arXiv:2004.13969*.

Helia Hashemi, Hamed Zamani, and W. Bruce Croft. 2020. Guided transformer: Leveraging multiple external sources for representation learning in conversational search. In *Proc. SIGIR*, page 1131–1140.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with GPUs. *arXiv:1702.08734*.

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 *Proc. EMNLP*, pages 6769–6781.

Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *Proc. SIGIR*, page 39–48.

Vaibhav Kumar and Jamie Callan. 2020. Making information seeking easier: An improved pipeline for conversational search. In *Proc. EMNLP Findings*.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021a. Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proc. SIGIR*, pages 2356–2362.

Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2020. Distilling dense representations for ranking using tightly-coupled teachers. *arXiv:2010.11386*.

Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2021b. In-batch negatives for knowledge distillation with tightly-coupled teachers for dense retrieval. In *Proc. RepL4NLP*, pages 163–173.

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 Trans. Inf. Syst.*, 39(4).

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with BERT. *arXiv:1901.04085*.

Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W. Bruce Croft, and Mohit Iyyer. 2020. Open-retrieval conversational question answering. In *Proc. SIGIR*.

Chen Qu, Liu Yang, Minghui Qiu, W. Bruce Croft, Yongfeng Zhang, and Mohit Iyyer. 2019a. BERT with history answer embedding for conversational question answering. In *Proc. SIGIR*, page 1133–1136.

Chen Qu, Liu Yang, Minghui Qiu, Yongfeng Zhang, Cen Chen, W. Bruce Croft, and Mohit Iyyer. 2019b. Attentive history selection for conversational question answering. In *Proc. CIKM*, page 1391–1400.

Jun Quan, Deyi Xiong, Bonnie Webber, and Changjian Hu. 2019. GECOR: An end-to-end generative ellipsis and co-reference resolution model for task-oriented dialogue. In *Proc. EMNLP*, pages 4547–4557.

Filip Radlinski and Nick Craswell. 2017. A theoretical framework for conversational search. In *Proc. CHIIR*, pages 117–126.

Pushpendre Rastogi, Arpit Gupta, Tongfei Chen, and Mathias Lambert. 2019. Scaling multi-domain dialogue state tracking via query reformulation. In *Proc. NAACL*, pages 97–105.

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

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proc. ACL*.

Gary Ren, Xiaochuan Ni, Manish Malik, and Qifa Ke. 2018. Conversational query understanding using sequence to sequence modeling. In *Proc. WWW*, pages 1715–1724.

Svitlana Vakulenko, Shayne Longpre, Zhucheng Tu, and Raviteja Anantha. 2020. Question rewriting for conversational question answering. *arXiv:2004.14652*.

Nikos Voskarides, Dan Li, Andreas Panteli, and Pengjie Ren. 2019. ILPS at TREC 2019 conversational assistant track. In *Proc. TREC*.

Nikos Voskarides, Dan Li, Pengjie Ren, Evangelos Kanoulas, and Maarten de Rijke. 2020. Query resolution for conversational search with limited supervision. In *Proc. SIGIR*, pages 921–930.

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 *Proc. ICLR*.

Jheng-Hong Yang, Sheng-Chieh Lin, Chuan-Ju Wang, Jimmy Lin, and Ming-Feng Tsai. 2019a. Query and answer expansion from conversation history. In *Proc. TREC*.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019b. End-to-end open-domain question answering with BERTserini. In *Proc. ACL (Demonstrations)*, pages 72–77.

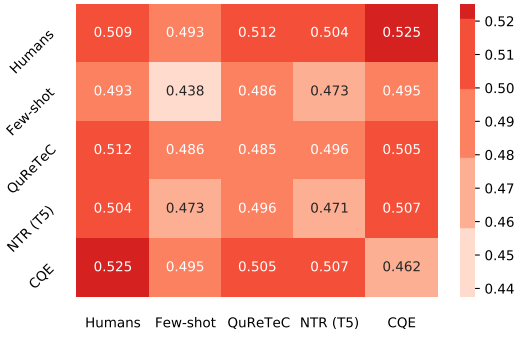
Shi Yu, Jiahua Liu, Jingqin Yang, Chenyan Xiong, Paul Bennett, Jianfeng Gao, and Zhiyuan Liu. 2020. Few-shot generative conversational query rewriting. In *Proc. SIGIR*, pages 1933–1936.

A Dense–Sparse Hybrid Settings

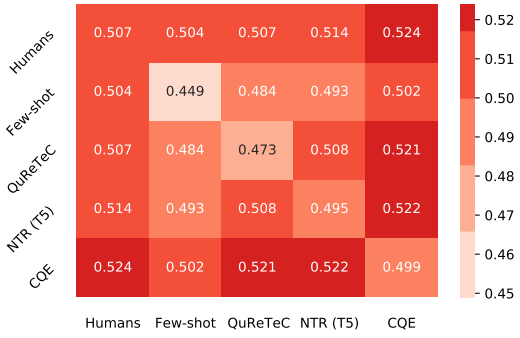
For each query q, we use sparse and dense representations to retrieve top-1000 passages, $\mathcal{D}_{sp}$ and $\mathcal{D}_{ds}$, with their relevance scores, $\phi_{sp}(q, p \in \mathcal{D}_{sp})$ and $\phi_{ds}(q, p \in \mathcal{D}_{ds})$, respectively. Then, we compute the score for each retrieved passage, $p \in \mathcal{D}_{sp} \cup \mathcal{D}_{ds}$, as follows:

$$\phi(q,p) = \begin{cases} \alpha \cdot \phi_{sp}(q,p) + \min\limits_{p \in \mathcal{D}_{ds}} \phi_{ds}(q,p), & \text{if } p \notin D_{ds} \\ \alpha \cdot \min\limits_{p \in \mathcal{D}_{sp}} \phi_{sp}(q,p) + \phi_{ds}(q,p), & \text{if } p \notin D_{sp} \\ \alpha \cdot \phi_{sp}(q,p) + \phi_{ds}(q,p), & \text{otherwise.} \end{cases} \tag{8}$$

Eq. (8) is an approximation of a linear combination of sparse and dense relevance scores. If $p \notin \mathcal{D}_{sp}$(or $\mathcal{D}_{ds}$), we directly use the minimum score of $\phi_{sp}(q, p \in \mathcal{D}_{sp})$ or $\phi_{ds}(q, p \in \mathcal{D}_{ds})$ as a substitute. For the sparse and dense retrieval combination, we select the best hyperparameters α (0.1) and γ (12) optimizing nDCG@3 on the CAsT19 training set.



(a) Sparse retrieval fusion



(b) Dense retrieval fusion

Figure 5: Ranked list fusion on the CAsT19 eval set, reporting nDCG@3.

B Model Fusion Study

We conduct experiments on model fusion to see whether CQE can complement other CQR models in terms of retrieval effectiveness. Specifically, we use reciprocal rank fusion (RRF) of ranked lists from different queries. Figure 5 shows the effectiveness (nDCG@3) of different fusion combinations on the CAsT19 eval set. We observe that CQE better fuses with all the other CQR models, even in sparse retrieval, where CQE does not perform as well. In addition, CQE shows even better fusion results than human queries in dense retrieval.