Few-Shot Conversational Dense Retrieval
Shi Yu1*, Zhenghao Liu1*, Chenyan Xiong2, Tao Feng1, and Zhiyuan Liu1
Tsinghua University1, Microsoft Research2
{yus17, liu-zh16}@mails.tsinghua.edu.cn; chenyan.xiong@microsoft.com;
liuzy@tsinghua.edu.cn

ABSTRACT
Dense retrieval (DR) has the potential to resolve the query understanding challenge in conversational search by matching in the learned embedding space. However, this adaptation is challenging due to DR models’ extra needs for supervision signals and the long-tail nature of conversational search. In this paper, we present a Conversational Dense Retrieval system, ConvDR, that learns contextualized embeddings for multi-turn conversational queries and retrieves documents solely using embedding dot products. In addition, we grant ConvDR few-shot ability using a teacher-student framework, where we employ an ad hoc dense retriever as the teacher, inherit its document encodings, and learn a student query encoder to mimic the teacher embeddings on oracle reformulated queries. Our experiments on TRECCAST and OR-QuAC demonstrate ConvDR’s effectiveness in both few-shot and fully-supervised settings. It outperforms previous systems that operate in the sparse word space, matches the retrieval accuracy of oracle query reformulations, and is also more efficient thanks to its simplicity. Our analyses reveal that the advantages of ConvDR come from its ability to capture informative context while ignoring the unrelated context in previous conversation rounds. This makes ConvDR more effective as conversations evolve while previous systems may get confused by the increased noise from previous turns. Our code is publicly available at https://github.com/thunlp/ConvDR.

CCS CONCEPTS
• Information systems → Information retrieval.

KEYWORDS
Conversational Retrieval; Dense Retrieval; Knowledge Distillation

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGIR ’21, July 11–15, 2021, Virtual Event, Canada.
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8037-9/21/07...
https://doi.org/10.1145/3404835.3462856

* indicates equal contribution.

1 INTRODUCTION
Recent advances in language understanding, voice recognition, and availability of edge devices provide search engines an opportunity to better serve users through conversational search [14]. Instead of ad hoc queries, search in the conversational format enables the system to interact with users through multi-round natural dialogues and take initiatives, to support more complicated information needs and assist with more complex tasks [1, 8].

The multi-round conversational queries yield a unique challenge in conversational search. Human conversations are contextualized, concise, and assume prior knowledge. The conversational search usually implies contexts from previous turns with omissions, references, and ambiguities, making it harder for search systems to understand the underlying information needs compared with ad hoc retrieval [10, 33, 36]. Recent conversational search systems address this challenge by reformulating the conversational queries into fully-grown ad hoc queries [26, 37, 40, 46], Learning a reformulated query to mimic the ad hoc query inevitably introduces another noise channel that further aggravates the vocabulary mismatch problem and underperforms their ad hoc counterpart [46]. A potential solution to fundamentally address the query understanding and vocabulary mismatch challenges in conversational search is to leverage the dense retrieval technique [15, 20, 21, 24, 27, 43] recently developed in ad hoc search. As illustrated in Figure 1, such a system can first encode each conversational query into an embedding that conveys user’s information needs, and then directly match with the documents in the learned embedding space [31]. Using dense retrieval in conversational search is intriguing. Conversational search aims to model user’s information needs from conversation contexts. Thus, it has a more severe vocabulary mismatch problem [8], which is where dense retrieval thrives on ad hoc search [43].

In practice, however, there is a sharp contradiction between the extreme data-hungriness of dense retrieval [43] and the long-tail
nature of conversational search [14]. It is known in ad hoc search that compared to other Neu-IR methods [28], dense retrieval requires more relevance labels, more fine-tuning iterations, and more sophisticated fine-tuning strategies to be useful [15, 27, 43]. On the other hand, the conversational search may never enjoy the large-scale relevance supervision available in ad hoc search. Its information needs are more complex, more personalized, and often require multiple rounds of queries to develop [8]. Every conversational search session is tall and there are not sufficient training signals to fully supervise current dense retrieval models [14].

In this paper, we address this discrepancy with a new few-shot learning approach for conversational dense retrieval (ConvDR). We first identify the bottleneck of ConvDR is still at the question representation side, whereas the documents can inherit the same dense representation learned in ad hoc search—their information is still the same. Then we develop a teacher-student framework to train a student conversational query encoder to "mimic" the representation of the oracle query rewrite from the teacher, which is an ad-hoc dense retrieval model [43]. This knowledge distillation (KD) is combined with the standard ranking loss in multi-task learning to enhance the few-shot ability of ConvDR.

In our experiments on TREC CAsT benchmark [8, 9], ConvDR shows strong empirical advantages in conversation search. It outperforms the previous state-of-the-art query rewriting based model [46] by 9% and 48% in retrieval accuracy, achieving even better accuracy than manual question reformulations on CAsT-19. In addition, our experiments on QR-QuAC [31], a crowd-sourced conversational QA dataset [3, 13], demonstrate that our knowledge distillation also benefits fully-supervised ConvDR and helps ConvDR almost double the accuracy of previous published state-of-the-arts [31].

Our studies find that, compared with question reformulation in the sparse space, the embeddings from our conversational question encoder better approximate the oracle teacher, capture the context in previous turns, and align with the relevant document encodings. Eliminating the extra question reformulation step and directly operating in the dense space not only simplify the pipeline, but also improve online latency as the costly question generation is no longer needed.

Furthermore, we conduct a series of analyses to understand the effectiveness of ConvDR. We observe that ConvDR effectively and robustly captures the necessary and informative context from previous conversation rounds, while the query rewriter can have dramatically different behavior with alterations in the contexts. The ability to handle rich and noisy context information ensures ConvDR to maintain retrieval quality throughout the conversation, while other automatic approaches can get confused in later turns.

2 RELATED WORK

Conversational search is considered as one of the most important research frontiers in information retrieval [7]. With the rapid adoption of smartphones and speakers, handling user’s search requests in the conversational format is a pressing need in commercial systems [5, 14]. Meanwhile, it also provides search engines new opportunities to better support users’ complicated information needs with multi-turn interactions and active suggestions [1, 5, 35, 47].

The TREC Conversational Assistant Track (CAsT) initiative constructed an evaluation benchmark for conversational search [8, 9]. Following the voice search scenario in Bing, CAsT formulates a conversational search task where a user’s information need is presented by multiple turns of dialogue queries. The task is to retrieve relevant passages for each turn. The queries are manually curated by the organizers to mimic user behaviors in the voice search scenario (minus the voice recognition part), thus include various human dialogue phenomena: subtle topic shift, reference to the previous context, assuming common knowledge, etc. [10].

Recent research on the CAsT benchmark identified the main challenge is on the query side: the document ranking parts can mostly reuse techniques developed in ad hoc search, while the concise conversational queries require new techniques to resolve their context dependency and vocabulary mismatch [8]. The solutions developed in recent research are to reformulate conversational queries into de-contextualized and fully-grown and ad hoc queries that include all necessary information to represent the user’s information needs, and then perform ad hoc retrieval with reformulated queries.

The query reformulation can be conducted via query expansion, i.e., selecting context terms in previous turns to add to the current query’s bag-of-word representation, using rules [26] or supervised classifiers [40]. It can also use query rewriting, which leverages natural language generation models to directly rewrite the conversational queries to de-contextualized ones. One of the top-performing systems in TREC CAsT 2019 feeds the previous and current queries to GPT-2 [32] and uses it to generate fully-grown ad hoc queries. Yu et al. [46] further fine-tune GPT-2 using synthetic weak supervision data to improve the rewriting accuracy in few-shot scenarios.

These solutions aim to reformulate the conversational query to an ad hoc query in the sparse bag-of-words space and then leverage standard sparse retrieval pipelines such as BM25 and BERT ranker [8, 22, 29]. By design, the vocabulary mismatch problem [6] in these conversational search systems is more severe than their corresponding ad hoc search systems, as the extra query reformulation step is an additional source of errors. Empirically, there is a notable gap between solutions using automatic query reformulations and those using manual oracle query rewrites [8, 46].

Recently, dense retrieval has shown strong empirical performance in various ad hoc search and open domain query answering (QA) benchmarks [2, 23]. Instead of sparse word representations, dense retrieval leverages pretrained models, e.g., BERT [11], to encode query and documents into embeddings and conduct retrieval purely in the dense representation space [18]. With effective fine-tuning strategies, dense retrieval methods, such as DPR [20], ME-BERT [27], and ANCE [43], significantly outperform sparse retrieval. On the other hand, learning an effective dense representation space requires more relevance labels and the effectiveness of dense retrieval in few-shot scenarios is less observed [42].

Using the QuAc conversational QA dataset [3, 13], where crowd-source workers are employed to ask multi-turn questions about a given Wikipedia entity and its description, Qu et al. [31] constructed a conversational search task OR-QuAc, using the crowded-sourced questions as the query, and the evidence passages as the retrieval targets. The synthetic nature of OR-QuAc results in all the relevant passages of a dialog reside in the same section of a
Wikipedia document, which is not the case in real scenarios. However, OR-QuAC still provides a large number of synthetic labels to showcase the advantage of fully supervised dense retrieval in conversational search [31].

3 METHODOLOGY

In this section, we first recap preliminaries in conversational search. Then we present our Conversational Dense Retrieval (ConvDR) framework and its teacher-student few-shot learning strategy.

3.1 Preliminary

The conversational search task [8] is formulated as to find document \( d \) from a collection \( D \) for each query turn in a multi-round conversation \( Q = \{q_i\}_{i=1}^t \). The unique part is that each conversational query \( q_k \) can be context-dependent, under-specified, and requires more sophisticated query understanding techniques that infer hidden context from previous queries \( Q_{1:k−1} \) [10, 33, 36].

A standard solution for conversational search is to reformulate the conversational query into a de-contextualized, fully-grown, ad hoc query [8, 25, 26, 37, 46]:

\[
q_k; Q_{1:k−1} \xrightarrow{\text{Reformulate}} q'_k, (1)
\]

where the ad hoc query \( q'_k \) is expected to fully represent the underlying information needs. The reformulation can be done by query expansion, for example, selecting terms from \( Q_{1:k−1} \) and expanding \( q_k \) to \( q'_k \) [26, 40], or rewriting queries, which directly generates the target query \( q'_k \) using generation models, e.g., GPT-2 [37, 46].

Recent conversational search benchmarks often provide a manual ad hoc query \( q'_k \) that fully represents the information needs in turn \( k \) [10, 13]. The manual oracles are valuable supervision signals to train the query reformulation model in Eq. 1.

After reformulating the conversational query to an ad hoc query, either as a bag-of-words (expansion) or a natural language query (rewriting), the conversational search problem is re-framed to an ad hoc search problem. Then we can reuse the standard retrieval-and-rerank IR pipeline, for example, first use BM25 to retrieval top \( K \) documents \( D^* \) from the collection \( D \):

\[
D^* = \text{BM25}(q'_K, D). (2)
\]

After that, we can use BERT to rerank the candidates \( d \) in \( D^* \) by computing the ranking score \( f^\text{ad}(\cdot) \) as [29]:

\[
H_0(q_k, d) = \text{BERT}([\text{CLS}] \circ q'_k \circ [\text{SEP}] \circ d \circ [\text{SEP}]), (3)
\]

\[
f^\text{ad}(q'_k, d) = \text{MLP}(H_0(q'_k, d)). (4)
\]

It uses an MLP layer on the [CLS] embedding \( (H_0) \) of BERT that encodes the concatenation \( (\circ) \) of reformulated \( q'_k \) and \( d \).

Reformulating conversational search queries to ad-hoc search queries provides a jump start for conversational search [8], but it also bounds the effectiveness of conversational search systems by ad-hoc search, rather than leveraging the opportunity of multi-turn conversations to provide search experiences of the next generation [5]. In practice, the query reformulation step is not perfect and further reduces the conversational search accuracy [46].

3.2 Conversational Dense Retrieval

Instead of retrofitting to ad hoc retrieval, ConvDR provides an end-to-end solution dedicated to conversational search using dense retrieval [20, 43]. In general, ConvDR first maps the conversational search queries and documents into one embedding space:

\[
q_k \leftarrow g_{\text{ConvDR}}(q_k; Q_{1:k−1}), (5)
\]

\[
d \leftarrow g_{\text{ConvDR}}(d). (6)
\]

Then ConvDR calculates the retrieval score simply by the dot product of the query embedding and the document embedding:

\[
f^\text{DR}(q'_k, d) = q'_k \cdot d. (7)
\]

Top \( K \) retrieval in the embedding space is efficiently supported by various libraries [18].

As the retrieval function is a simple dot product, the main model capacity is in query and document encoders. We use the common BERT-Siamese/Dual-Encoder architecture in dense retrieval [24] to encode the conversational turns and documents:

\[
q'_k = H_0(q_k) = \text{BERT}([\text{CLS}] \circ q_1 \circ \ldots \circ [\text{SEP}] \circ q_k \circ [\text{SEP}]), (8)
\]

\[
d = H_0(d) = \text{BERT}([\text{CLS}] \circ d \circ [\text{SEP}]). (9)
\]

We use the BERT [CLS] embeddings as the encodings for the query and document. Note that we feed in the concatenation of the current query and all previous queries to the query encoder, a prerequisite to capture the information needs in one contextualized embedding. ConvDR can be optimized to learn retrieval-oriented representations using the negative log likelihood (NLL) ranking loss:

\[
L_{\text{Rank}} = -\log \frac{\exp(q'_k \cdot d^*)}{\exp(q'_k \cdot d^*) + \sum_{d' \in D^-} \exp(q'_k \cdot d')}. (10)
\]

where \( d^* \) and \( d^- \) are relevant and irrelevant documents for \( q_k \).
In the reranking stage, instead of reformulating to query $q'_k$, we directly apply BERT on the concatenation of all queries $Q_{1:k}$ and document $d$:

$$H_0(q_k, d) = \text{BERT}([\text{CLS}] q_1 \ldots [\text{SEP}] q_k \ldots [\text{SEP}] d = [\text{SEP}]), \quad (11)$$

and then add an MLP layer (Eq. 4.) to calculate the ranking score.

### 3.3 Few-Shot from Ad Hoc Teacher

A challenge of applying dense retrieval is that, dense retrieval models require more relevance-oriented supervision signals to be effective, even in ad hoc retrieval [20, 43, 48]. In conversational search, ConvDR needs to construct a contextualized query representation using not only the current query but also previous queries, figuring out the context dependency would require additional training signals than encoding an ad hoc query. Besides, it is harder to accumulate relevance labels in conversational search than in ad hoc search, the information needs in conversational search are expected to be more complicated and personalized [33].

To overcome this limitation, we develop a teacher-student framework that improves the few-shot ability of ConvDR by learning from an ad hoc dense retriever. Given the query embedding $q_k$ obtained from an ad hoc dense retrieval encoder $g_{\text{ad hoc}}$ on the manual oracle query $q_k$ and the document $d$, we use the following teacher-student learning framework to obtain the query and document embeddings in ConvDR:

$$q_k \leftarrow g_{\text{ad hoc}}(q_k), \quad L_{\text{KD}} = \text{MSE}(q_k', q_k). \quad (12)$$

$$d \leftarrow g_{\text{ad hoc}}(d). \quad (13)$$

We use a state-of-the-art ad hoc dense retriever ANCE [43] as the teacher. The teacher-student framework introduces two assumptions:

1. The underlying information needs in the manual oracle $q_k$ and the conversation query $q_k$ are the same thus their embeddings should be the same.
2. The meaningful information in a document $d$ is the same when serving ad hoc search and conversational search, thus its embedding in the two scenarios can be shared. Following the first assumption, we use MSE loss to distill the knowledge from ANCE to ConvDR (Eq. 12) [16]. Following the second assumption, we inherit the document embedding from ANCE to ConvDR (Eq. 13).

The knowledge distillation (KD) loss (Eq. 12) is combined with the ranking loss (Eq. 10) in multi-task training:

$$L = L_{\text{KD}} + L_{\text{Rank}}. \quad (14)$$

To form the negative document set $D^-$ for $L_{\text{Rank}}$, we retrieve negatives ranked top by ANCE for the manual oracle query $q_k$.

The teacher-student training reduces the need for a large number of relevance labels of ConvDR by inheriting the document representations from ad hoc dense retrieval and mimicking the query embeddings of the manual oracle queries. The ranking loss uses the available conversational search labels to supervise ConvDR, rather than just retrofitting the ad hoc dense retriever. The same few-shot paradigms can also be used to train the BERT Reranker.

### 4 EXPERIMENTAL METHODOLOGIES

This section describes our experimental settings, baselines, and implementation details.

**Datasets.** We use three conversational search benchmarks in our experiments: TREC CAsT 2019 [10] & 2020 [9] include a small amount of TREC-quality relevance labels (few-shot); OR-QuAC [31] where the search task is synthetic but includes a large amount of labels ("supervised"). Their statistics are presented in Table 1.

**CAS-T-19.** The TREC Conversational Assistance Track (CAsT) 2019 benchmark [8] provides a series of dialogues (topics) for conversational search. There are 30 training dialogues and 50 test dialogues. Each dialogue contains an average of 9 to 10 queries in natural language forms. Queries in a dialogue are constructed manually to mimic a “real” dialogue on a certain topic, with common natural language characteristics including coreferences, abbreviations, and omissions. The later query turns often depend on previous contexts. The corpus includes passages from MS MARCO [2] and TREC Complex Answer Retrieval (CAR) [12]. It provides manual oracle de-contextualized queries for all test queries and relevance judgments for 173 queries in 20 test dialogues.

**CAS-T-20.** CAsT 2020 [9] is the dataset of second year of Conversational Assistance Track. It contains 25 evaluation dialogues and uses the same document collection as CAS-T-19. The queries can refer to previous answers from system responses, making the dataset more realistic. The organizers included two versions of canonical responses returned by a baseline system: the manual canonical response which uses the manually rewritten query, and the automatic canonical response which uses queries rewritten by a conversation query rewriter [46]. CAS-T-20 provides manually rewritten queries for all queries and relevance judgments for most of them.

**OR-QuAC** [31] is a large-scale synthetic conversational retrieval dataset built on a conversational QA dataset, QuAC [3]. In QuAC, authors employed crowd-source workers to ask multi-turn questions about a given Wikipedia entity and its description, to synthesize a conversational QA task. CANARD [13] recruited crowd-source workers to manually write de-contextualized queries for QuAC. OR-QuAC extends them to the open-retrieval setting by building a corpus from English Wikipedia (10/20/2019 dump) and using the passages that contain the golden answer as relevant ones. Note that in OR-QuAC, the answer passages of all turns in a conversation topic are guaranteed to be from the Wikipedia page of the given entity. And the entity is added the first conversation turn [31].

**Evaluation Metrics.** We evaluate model performance on TREC CAsT in terms of MRR and NDCG@3 (main). We also report the average hole rate at cutoff 10 (fraction of the top 10 ranked results without judgments) on CAsT to reveal the uncertainty of the evaluation. For OR-QuAC, we report Recall@5 and MRR@5 following previous work [31] as well as MAP@10. We conduct

| Statistics | CAsT-19 | CAsT-20 | OR-QuAC |
|------------|---------|---------|---------|
| # Conversations | 50 (20) | 25 (20) | 31,526 (208) |
| # Questions | 479 (173) | 216 (208) | 31,526, 3,430, 5,571 |
| # Avg. Question Tokens | 6.1 | 6.8 | 6.7 |
| # Avg. Questions / Conversation | 9.6 | 8.6 | 7.2, 7.0, 7.2 |
| # Documents | 36M | 11M | |
### Table 2: Overall results on TREC CAsT and OR-QuAC. Superscripts †, ‡, §, and ¶ indicate statistically significant improvements over BM25-Raw\(^1\), ANCE-Raw\(^2\), BM25-Query Rewriter\(^3\), and ANCE-Query Rewriter\(^4\) or their corresponding reranking results. Unavailable and inapplicable results are marked by “n.a.” and “-”. Best results in each group are marked Bold.

| Method | CAsT-19 | CAsT-20 | OR-QuAC |
|--------|---------|---------|---------|
|        | MRR     | NDCG@3  | Hole@10 | MRR     | NDCG@3  | Hole@10 | MAP@10 | R@5 | MRR@5 | |
| **Public Baselines** | | | | | | | | | | |
| pgbert (from [8]) | 0.665 | 0.413 | n.a. | – | – | – | – | – | – | – |
| h2olo\_RUN2 (from [8]) | 0.714 | 0.434 | n.a. | – | – | – | – | – | – | – |
| CFDA\_CLIP\_RUN7 (from [8]) | **0.715** | 0.436 | n.a. | – | – | – | – | – | – | – |
| RRF (Reranking) (from [26]) | n.a. | 0.536 | n.a. | – | – | – | – | – | – | – |
| Auto-Rewriter (from [46]) | n.a. | 0.492 | n.a. | – | – | – | – | – | – | – |
| quretecQR (from [38]) | – | – | – | 0.476 | 0.340 | n.a. | – | – | – | – |
| h2olo\_RUN2\* (from [9]) | – | – | – | **0.621** | 0.493 | n.a. | – | – | – | – |
| ORConvQA reranker (from [31]) | – | – | – | – | – | – | n.a. | 0.314 | 0.313 | – |
| **First Stage Retrieval Only** | | | | | | | | | | |
| BM25-Raw | 0.322 | 0.134 | 32.7% | – | – | – | – | – | – | – |
| ANCE-Raw | 0.420\(†\) | 0.247\(†\) | 60.6% | 0.234\(†\) | 0.150\(†\) | 68.9% | 0.114\(†\) | 0.149\(†\) | 0.113\(†\) | – |
| BM25-Query Rewriter | 0.581\(‡\) | 0.277\(‡\) | 19.0% | 0.250 | 0.159\(§\) | 39.7% | 0.200\(‡\) | 0.302\(‡\) | 0.202\(‡\) | – |
| ANCE-Query Rewriter | 0.665\(‡\) | 0.409\(‡\) | 43.1% | 0.375\(‡\) | 0.253\(‡\) | 49.6% | 0.451\(‡\) | 0.584\(‡\) | 0.457\(‡\) | – |
| ConvDR | **0.740**\(‡\) | **0.466**\(‡\) | 40.3% | **0.501**\(‡\) | **0.340**\(‡\) | 42.2% | **0.607**\(‡\) | **0.750**\(‡\) | **0.616**\(‡\) | – |
| **With BERT Reranker** | | | | | | | | | | |
| BM25-Raw→BERT-Raw | 0.503\(§\) | 0.306\(§\) | 36.1% | 0.220 | 0.154 | 48.7% | 0.154 | 0.171 | 0.157 | – |
| ANCE-Raw→BERT-Raw | 0.449 | 0.275 | 50.7% | 0.249 | 0.184\(§\) | 56.9% | 0.174\(†\) | 0.207\(†\) | 0.177\(†\) | – |
| BM25-OR→BERT-Query Rewriter | 0.786\(‡\) | 0.495\(‡\) | 21.1% | 0.383\(‡\) | 0.264\(‡\) | 30.9% | 0.537\(‡\) | 0.615\(‡\) | 0.554\(‡\) | – |
| ANCE-OR→BERT-Query Rewriter | 0.747\(‡\) | 0.481\(‡\) | 28.4% | 0.383\(‡\) | 0.298\(‡\) | 36.7% | 0.664\(‡\) | 0.710\(‡\) | 0.615\(‡\) | – |
| ConvDR→BERT | **0.802**\(¶\) | 0.526\(¶\) | 33.0% | 0.506\(¶\) | 0.362\(¶\) | 32.1% | **0.758**\(¶\) | **0.850**\(¶\) | **0.773**\(¶\) | – |
| ConvDR→BERT (RRF) | 0.799\(¶\) | **0.541**\(¶\) | **31.2**% | **0.545**\(¶\) | **0.392**\(¶\) | **31.1**% | 0.698\(¶\) | 0.821\(¶\) | 0.718\(¶\) | – |
| **Manual for Reference** | | | | | | | | | | |
| BM25-Manual | 0.671 | 0.347 | 10.5% | 0.445 | 0.301 | 17.7% | 0.316 | 0.430 | 0.330 | – |
| ANCE-Manual | 0.740 | 0.461 | 37.1% | 0.591 | 0.422 | 32.2% | 0.507 | 0.642 | 0.515 | – |
| BM25-Manual→BERT-Manual | 0.842 | 0.548 | 16.0% | 0.632 | 0.470 | 7.1% | 0.645 | 0.719 | 0.665 | – |
| ANCE-Manual→BERT-Manual | **0.835** | **0.566** | **23.1**% | **0.663** | **0.483** | **14.9**% | 0.683 | 0.778 | 0.698 | – |

the statistical significance test between our implemented methods using permutation test with \(p < 0.05\).

**Baselines.** We include two groups of baselines, which come from published research and our implementation ones.

The published baselines all use the query reformulation and retrieval-rerank pipeline. We first include three top performing systems in CAsT-19, pgbert, h2olo\_RUN2, and CFDA\_CLIP\_RUN7.

pgbert [8] fine-tunes GPT-2 [32] for query rewriting using the manual rewrites from CANARD [15]. It uses Anserini BM25 [45] and BERT reranker. h2olo\_RUN2 [44] adds topic title in its BM25 retrieval, uses heuristics to expand queries and BERT to rerank. CFDA\_CLIP\_RUN7 [44] adds doc2query [30] to h2olo\_RUN2.

In addition, we include two recent state-of-the-art systems in CAsT-19. RRF (Reranking) [26]\(^3\) fusions results from h2olo\_RUN2 and Neural Transfer Reformulation (NTR). The latter uses a T5 [34] based query rewriter. Auto-Rewriter [46] fine-tunes GPT-2 as an automatic query rewriter. Its retrieval and reranking use the standard BM25-BERT pipeline.

Besides above baselines, on CAsT-20, we include two top performing baselines using the automatic canonical responses: quretecQR and h2olo\_RUN2\(^3\). quretecQR [38] leverages the QuReTeC [40] model to classify historical query terms to be added to the current query. h2olo\_RUN2\* [9] uses heuristics to extract contents from system responses and uses T5 to rewrite queries for ranking. On OR-QuAC, we compare with ORConvQA, the current state-of-the-art [31]. It is a pipeline neural QA system including a dense retriever, a reranker, and a reader. The dense retriever concatenates the current and historical utterances for retrieval.

In addition to published methods, we implement baselines using different retrieval models, Anserini BM25 [45] or ANCE retrieval [43], and BERT reranker [29]. The queries fed into these models can be raw queries (Raw), automatic query rewrites from a trained GPT-2 [46] (Query Rewriter), and manual oracle rewrites (Manual, for reference). On CAsT-19, we train GPT-2 following Yu et al. [46], using the Rule-based+CV method that trains the query rewriter on rule-based weak supervision data. On CAsT-20, we train GPT-2 query rerewriter on CANARD, with the answer for the previous query. On OR-QuAC, we train the query rewriter on its training set.

**Implementation Details.** For ConvOR and BERT reranker, we conduct three training paradigms discussed in Section 3 and include three variants for both: KD, which only distills knowledge from ad hoc models; Rank, which only uses ranking loss; and Multi-Task, the combination of the two. In addition, we also compare with their zero-shot variants that only trained in ad hoc retrieval.
Table 3: Efficiency of ConvDR and baselines in our university-standard computing environment. ANCE retrieval is measured on GPUs by FAISS [18].

| Operation                | CAsT-19  | OR-QuAC |
|--------------------------|----------|---------|
| Query Rewriting          | 245ms    | 400ms   |
| Query Encoding           | 15ms     | 15ms    |
| BM25 Retrieval           | 645ms    | 266ms   |
| ANCE Retrieval           | 545ms    | 312ms   |
| ANCE Retrieval (Batched) | 2.4ms    | 1.1ms   |
| BM25—Query Rewriter Total| 890ms    | 666ms   |
| ANCE—Query Rewriter Total| 805ms   | 727ms   |
| ConvDR Total             | 560ms    | 327ms   |
| ConvDR Total (Batched)   | 17.4ms   | 16.1ms  |

In the first stage retrieval, we use the open-sourced ANCE checkpoints⁴ as the teacher [43]. On CAsT, we use its checkpoint on MARCO passages (at the 600k-th step). On OR-QuAC, we use its multi-task checkpoint trained on Natural Questions [23] and TriviaQA [19]; in addition, we continuously train ANCE on the training set of OR-QuAC with manually rewritten queries to adapt it to OR-QuAC. The continuous training uses 3 GPUs, batch size 6, and Adam with a learning rate of 1e-6.

All document embeddings are encoded by ANCE and fixed in our experiments. The query encoder in ConvDR is warmed up from ANCE and fine-tuned. For the data scarcity on CAsT-20, we further warm up ConvDR on CANARD. We concatenate historical queries to the current query and discard early turns if exceed 256 tokens. On CAsT-20, we follow the automatic canonical setting and prepend the automatic canonical response to the query of the previous conversational turn. All variants of ConvDR are trained with 8 epochs on CAsT-19, 16 epochs on CAsT-20 and 1 epoch on OR-QuAC. For the ConvDR (Rank) and ConvDR (Multi-Task), we use the manual rewrites to retrieve documents with vanilla ANCE and sample 9 negative documents per query.

**BERT Reranker.** We train BERT rerankers [29] on MS MARCO Passage and OR-QuAC tasks with manually reformulated queries. Then we fine-tune BERT rankers with the same three training paradigms, yielding BERT (KD), BERT (Rank) and BERT (Multi-Task). We use Reciprocal Rank Fusion (RRF) [4] to combine the results from the best BERT ranker and ConvDR, yielding BERT (RRF).

All training on CAsT uses five-fold cross-validation. All training processes on OR-QuAC use their official training splits to ensure a fair comparison with previous research [31]. To train ConvDR, we use 1 GPU, set batch size to 4, and use Adam optimizer with learning rate 1e-5. BERT reranker training uses 4 GPUs, set batch size to 32, and use Adam optimizer with a learning rate of 3e-6.

### 5 EVALUATION RESULTS

In this section, we present five groups of experiments on the overall performance of ConvDR, its few-shot strategies, the learned query representations, its ability to model contexts, and case studies.

### 5.1 Overall Performance

The results on CAsT and OR-QuAC are listed in Table 2.

**Few-Shot Accuracy.** On CAsT, ConvDR outperforms all baselines in first stage retrieval. It improves BM25—Query Rewriter by 68% and 113%, showing the advantage of dense retriever over BM25 sparse retrieval. ConvDR is more effective than ANCE—Query Rewriter and also simpler: It directly maps queries to the embedding space, rather than intermediate reformulating queries in the sparse space. Noteworthy, ConvDR shows high hole rates on CAsT, indicating that the performance may be underestimated.

Encouragingly, ConvDR performs on par with ANCE-Manual on CAsT-19, and the gap between automatic and manual conversational search systems on CAsT-20 is also greatly eliminated. In later experiments, we study how ConvDR mimics the manual oracle query in the representation space (Sec. 5.3) while also effectively capturing salient context information (Sec. 5.4).

Solely using embedding dot products, ConvDR already outperforms the best participating system in CAsT-19, CFDA_CLIP_RUN7. The latter is a well-designed system with state-of-the-arts sparse retrieval and Neu-IR techniques [44]. We have made promising progress since the CAsT initiative.

The effectiveness of ConvDR in first stage retrieval can translate to the reranking stage with BERT reranker. On CAsT-19, ConvDR→BERT outperforms the query rewriting baselines. Combined with the results from ConvDR using RRF, it further outperforms the previous state-of-the-art, RRF (Reranking) [26], which uses additional query rewriting supervision from CANARD [13] and a more powerful pretrained model, T5. On CAsT-20, ConvDR→BERT outperforms every baseline by a large margin except h2o100_RUN2, which uses a dense-sparse hybrid retrieval model followed by a powerful T5 ranking model with T5-based query reformulation. We also observe a larger automatic-manual gap in reranking compared to the first stage retrieval when reranking dense retrieval candidates. Reranking on the top of dense retrieval require more future research.

**Supervised Accuracy.** On OR-QuAC where a large amount of synthetic relevance labels are available, our fully-supervised ConvDR outperforms all previous methods by huge margins. With first stage retrieval only, ConvDR almost doubles the accuracy of ORConvQA reranker, the best system from Qu et al. [31]; ConvDR→BERT even outperforms ANCE-Manual→BERT-Manual.

Empirical gains on OR-QuAC should be read with caution because the synthetic data includes various artifacts that over-simplify the retrieval problem [31]⁵. For example, all relevant passages for a dialog are from the Wikipedia page of the entity in the first round. Nevertheless, the synthetic data helps us take a peek at the potential of ConvDR when we have large-scale user feedback signals when search engines become conversational.

**Efficiency.** We measure the online inference efficiency of ConvDR and compare with the spare solutions in Table 3.

In production systems, one can aggressively batch queries (Batched) in dense retrieval and enjoy the speed up with tensor operations [43]. Nevertheless, we also measure the efficiency when queries are processed individually for a more complete picture. In per query setting, BM25 and ANCE have similar latency, but query rewriting [46] is

---

⁴https://github.com/microsoft/ANCE

⁵Such artifacts exist in many benchmarks. They do not disqualify the value of these benchmarks in progressing research as long as we are aware of their limitations.
Table 4: Results of different training paradigms. Candidates from ConvDR (KD) and ConvDR (Multi-Task) are reranked with BERT on CAsT-19 and OR-QuAC. Win/Tie/Loss (%) are compared with ANCE-Manual or ANCE-Manual→BERT-Manual.

| Method            | CAsT 2019 | OR-QuAC |
|-------------------|-----------|---------|
| First Stage Retrieval Only |           |         |
| ConvDR (Zero-Shot) | 0.202     | 0.202   |
| ConvDR (KD)       | 0.466     | 0.450   |
| ConvDR (Rank)     | 0.084     | 0.078   |
| ConvDR (Multi-Task) | 0.157     | 0.157   |
| With Rerank       |           |         |
| BERT (Zero-Shot)  | 0.407     | 0.407   |
| BERT (KD)         | 0.497     | 0.497   |
| BERT (Rank)       | 0.513     | 0.513   |
| BERT (Multi-Task) | 0.526     | 0.526   |

Figure 3: Query representations on CAsT-19. Query embeddings are from ConvDR (Zero-Shot) (0S), ANCE-Raw (Raw), ANCE-Query Rewriter (QR), ConvDR (KD) and ANCE-Manual (MQ). The similarity (a) is the average dot product score of two representations of the same query. The Figure (b) plots the query embeddings at the 6-th turn of topic 79. Q1-Q6 are query embeddings from ANCE (Raw).

Figure 4: Similarities of query and document embeddings on CAsT-19. Query embeddings are from ConvDR (Zero-Shot) (0S), ANCE-Query Rewriter (QR), ANCE-Manual (MQ) and ConvDR (KD). The similarity (a) is the average dot product score of queries and their nearest relevant documents. The representation distribution of queries and documents from turn 6, topic 79 of CAsT-19 is plotted through t-SNE in (b).

Figure 5: Model behavior at different conversation turns on CAsT-19, including: ANCE-Raw (Raw), ANCE-Query Rewriter (QR), ConvDR (KD) and ANCE-Manual (MQ). We evaluate their ranking accuracy at different turns (x-axes) in (a) and the similarity of query embeddings with previous round in (b).

5.2 Effectiveness of Training Strategies

In this experiment, we study the performance of different training strategies in ConvDR and BERT Reranker on CAsT-19 and OR-QuAC in Table 4. The models include Zero-Shot, which uses no specific training on conversational search; KD, which only uses the knowledge distillation (MSE Loss, Eq. 12); Rank, which only uses the ranking loss (Eq. 10); and Multi-Task which uses both.

Effectiveness in Few-Shot Scenarios. On CAsT 2019, KD is the only one that effectively trains ConvDR. The other three perform worse than plain ANCE results on CAsT 2019. The relevancy supervision, either used individually in Rank or combined in Multi-Task, downgrades the model to worse performance than Zero-Shot. Current dense retrievers require a large amount of relevance supervision signals or may perform worse than BM25. Our teacher-student few-shot learning is necessary for the effectiveness of ConvDR. In comparison, the BERT reranker seems to be more robust and can benefit from the labels from TREC CAsT. Combined with the KD loss, BERT (Multi-Task) can achieve the best accuracy. Similar trends are also observed on the per query comparisons with Manual runs. Notably, in the retrieval stage, KD is the closest to ANCE-Manual with most ties. ConvDR mimics its teacher well.

Effectiveness in Supervised Setting. With a large number of training signals from OR-QuAC, training directly using Rank works well with both ConvDR and BERT reranker. Still our teacher-student learning with KD provides addition signals when combined in Multi-Task and helps ConvDR become the best on OR-QuAC.

5.3 Learned Query Representations

In this experiment, we analyze the learned query and document embeddings from ConvDR and ANCE variants.

Query Embeddings. We first plot the average similarity (dot product) between the embeddings of the same query learned by different methods in Figure 3a. As expected, ANCE-Raw and ANCE-Manual
are very different. \( \theta S \) is different from all the others including \( \text{Raw} \). Directly applying the ad hoc dense retriever on the concatenation of multi-turn queries does not lead to meaningful embeddings. ANCE–Query Rewriter is similar to both ANCE–Manual (MQ) and ANCE–Raw. The automatic query rewrites are in between the raw queries and the manual oracle ones. Without explicitly re-formulating a new query, \( \text{ConvDR} \) (KD) effectively approximates ANCE–Manual by end-to-end learning in the embedding space.

We further demonstrate this effect by using t-SNE [39] to plot a conversation topic in Figure 3b. The Zero–Shot embedding resides in the middle of the raw embeddings of the current (Q6) and previous turns; treating the concatenation of all queries as one ad hoc query yielded average embeddings. \( \text{ConvDR} \) (KD) is as close to the current query Q6 as its teacher MQ, but balances its location with previous queries. Rather than the manual oracle query, \( \text{ConvDR} \) (KD) has access to the raw queries and can capture information of previous turns. We provide an example of this effect in Sec. 5.5.

Similarities with Document Embeddings. The effects on the query embeddings also influence their similarities with relevant documents. As shown in Figure 4a, \( \text{ConvDR} \) (KD) has the highest similarity, even higher than ANCE–Manual (MQ). The automatic query rewriting, however, has lower similarity compared with MQ. Mimicking the manual oracle queries in the sparse word space is a harder task than in the dense space.

We also visualize the learned query embedding of the 6th turn with their document embeddings in Figure 4b. The zero-shot query embedding resides in the middle of irrelevant documents. The other three all locate closer to relevant documents in the space. Among them, ANCE–Query Rewriter seems to introduce some biases and favors a subset of relevant documents. The rewritten query might only capture a part of the full information needs. ANCE–Manual and \( \text{ConvDR} \) (KD), in comparison, position the query closer to the middle of the relevant documents. In representation learning, such a more uniformed representation allocation often indicates better generalization ability [41].

5.4 Effectiveness in Capturing Contexts

In this experiment, we study the effectiveness of \( \text{ConvDR} \) on learning query representations from multi-turn conversations.

Accuracy Through Multiple Turns. We first study the behavior of dense retrieval models at different conversation turns, using their per turn retrieval accuracy and their query embedding similarities between conversation rounds.

As shown in Figure 5a, the accuracy of ANCE variants drops as the conversation goes on, the same with previous conversational search systems [8, 46]. The queries in later conversation turns are more likely to depend on previous turns, while the increased reasoning space—conversation turns—makes the context dependency harder to resolve. However, \( \text{ConvDR} \) (KD) maintains its accuracy throughout the conversation and performs on par with the Manual Oracle (MQ) on all turns. This further confirms that our few-shot strategy helps \( \text{ConvDR} \) capture the context information required to understand user’s information needs.

\( \text{ConvDR} \)’s better query understanding ability is also reflected in Figure 5b. The query representations from \( \text{ConvDR} \) (KD) are more similar with each other in adjacent rounds, indicating a smooth transition during conversation turns in the embedding space. In comparison, the de-contextualized manual queries (MQ) are less correlated, as ANCE–Manual does not use previous turns. The embeddings from automatic rewritten queries change more dramatically during the conversation. Adding query rewriting as an intermediate step introduces new source of error. Learning the contextualized query embeddings directly is more effective and efficient.

Dependency on Previous Conversation Turns. To further study how \( \text{ConvDR} \) models contexts in previous turns, we conduct an intrusion test by randomly discarding a previous turn in the conversation, and track the changes of query embeddings before and after. We compare this embedding change with the term overlap between the discard turn and the manual oracle query of the current turn. The term overlap is an approximation of the context importance in the discarded turn. The results are plotted in Figure 6.

\( \text{ConvDR} \) (Zero–Shot) is the least impacted by this intrusion. It does not capture much context information anyway and removing a previous turn does not make it worse. ANCE–Query Rewriter is significantly impacted by this intrusion test. Language generation models are known to be fragile and a small change of its input can lead to dramatically different outputs [17, 32, 46].

Figure 6: The overlap between query representations before and after discarding a previous conversation turn in the model input. The X-axes denote the similarity (dot product) score between the model’s query embedding with and without that context turn. The Y-axes mark the token overlap ratio of the discarded previous turn with the manual oracle query of the current turn. All results are conducted on CAsT-19.
Table 5: Winning examples of ConvDR on CAS-T-19. We show the historical queries, the current query (underlined), its manual and automatic rewrites, and the first passages different methods disagree in the ranking results. The [bracketed] numbers are the score drops in ConvDR (KD) when the query is discarded. We manually find notable context terms and mark them italic.

| Queries | First Disagreed Passages |
|---------|--------------------------|
| **CAS-T Topic-59** | |
| Which weekend sports have the most injuries? [-1.07] | ConvDR: An ACL injury is the tearing of the anterior cruciate ligament ... ACL injuries most commonly occur during sports ... |
| What are the most common types of injuries? [-0.53] | ANCE-Manual & Query Rewriter: ACL Short for Access Control List, an ACL is a list containing one or more ACE which are used by ... |
| What is the ACL? | |
| **Manual & Auto Rewrite:** What is the ACL? | |
| **CAS-T Topic-37** | |
| What was the Stanford Experiment? [-1.51] | ConvDR: In 1961, ... the Yale psychologist Stanley Milgram performed ... also known as the Milgram Experiment, in order to determine ... |
| What did it show? [-0.10] | The Milgram Experiment raised queries about the ethics of ... |
| Tell me about the author of the experiment. [-0.22] | ANCE-Manual: The Milgram experiment was a study ... that helped measure a person's moral character. ... The Milgram experiment caused a huge amount of criticism among individuals. ...many were completely convinced of the wrongness of what they were doing. |
| Was it ethical? [-0.19] | |
| What are other similar experiments? [-0.16] | |
| What happened in the Milgram experiment? [-3.92] | |
| Why was it important? | |
| **Auto Rewrite:** What was the Stanford Experiment important? | |
| **Manual Rewrite:** Why was the Milgram experiment important? | |
| **CAS-T Topic-56** | |
| What is Darwin's theory in a nutshell? [-0.77] | ConvDR: Speciation is the evolutionary process by which reproductively isolated biological populations evolve to become distinct species. ... Charles Darwin was the first to describe the role of natural selection in speciation in his 1859 book The Origin of Species. ... |
| How was it developed? [-0.19] | ANCE-Query Rewriter: ... Speciation is defined by ... as the distribution of an element amongst defined chemical species in a system. Chemical species include ... |
| How do sexual and asexual reproduction affect it? [-0.10] | ANCE-Manual: Darwin’s Theory. In 1859, Charles Darwin set out his theory of evolution by natural selection as an explanation for adaptation and speciation. ... |
| How can fossils be used to understand it? [-0.01] | |
| What is modern evidence for it? [-0.12] | |
| What is the impact on modern biology? [-0.18] | |
| Compare and contrast microevolution and macroevolution. [-0.37] | |
| What is the relationship to speciation? | |
| **Auto Rewrite:** What is the relationship to speciation? | |
| **Manual Rewrite:** ... relationship of Darwin’s theory to speciation? | |

ConvDR (KD) shows a smooth correlation between the embedding changes and the term overlap of the context turn and the manual oracle. It learns to capture the important context (indicated by high term overlap with oracle) and attend less on unrelated context. Its embedding changes are also more subtle: a lot of query embeddings are changed but most only by a little, showing its robustness to context removal. A context term may appear in multiple previous turns and an effective model should be able to use duplicated contexts to recover from removing one turn if possible. ConvDR (Multi-Task) has more variances in this intrusion test. The ranking loss with limited relevant labels is not reliable.

5.5 Case Study

Our last study shows some winning examples of ConvDR in Table 5. In the first case, the raw query is not explicitly context-dependent and can be an ad hoc query itself. The manual oracle is the same with the raw query. However, the acronym "ACL" is ambiguous and by encoding with previous conversational turns, ConvDR disambiguates it correctly. The second case is a typical coreference error made by automatic query rewriters. ConvDR correctly identifies the right context in its latent space and retrieves the correct passage.

The last case is interesting as it reflects the “conservative” behavior of the automatic query rewriter. When it is not able to resolve the context dependency, the rewriter tends to withdraw to and repeat the raw query [46]. By modeling all previous conversational turns, ConvDR provides a more meaningful query representation by attending more to the first turn, as indicated by the score drops, and capturing the salient context terms “Darwin” in its retrieval.

6 CONCLUSION

In this paper, we present a conversational dense retriever, ConvDR, that conducts the first stage retrieval of conversational search purely in the dense embedding space. We propose a few-shot strategy that trains ConvDR’s query encoder to mimic the embeddings of manual oracle queries from a well-trained ad hoc dense retriever. Our experiments on TREC CAS-T and OR-QuAC demonstrate that ConvDR achieves state-of-the-art performance in first stage retrieval. Detailed analysis shows that our simple approach can better capturing salient context information from the previous conversation turns in few-shot conversational search setting, as well as greatly improve online latency.

7 ACKNOWLEDGMENTS

Part of this work is supported by the National Key Research and Development Program of China (No. 2020AAA0106501) and Beijing Academy of Artificial Intelligence (BAAI).
