Saving Dense Retriever from Shortcut Dependency in Conversational Search

Anonymous ACL submission

Abstract

In conversational search (CS), it needs holistic understanding over conversational inputs to retrieve relevant passages. In this paper, we demonstrate the existence of a retrieval shortcut in CS, which causes models to retrieve passages solely relying on partial history while disregarding the latest question. With in-depth analysis, we first show naively trained dense retrievers heavily exploit the shortcut and hence perform poorly when asked to answer history-independent questions. To prevent models from solely relying on the shortcut, we explore iterative hard negatives mined by pre-trained dense retrievers (Xiong et al., 2020). Experimental results show that training with the iterative hard negatives effectively mitigates the dependency on the shortcut and makes substantial improvement on recent CS benchmarks. Our retrievers achieve new state-of-the-art results, outperforming the previous best models by 9.7 in Recall@10 on QReCC and 10.8 in Recall@5 on TopiOCQA.

1 Introduction

Conversational search (CS) is a task of retrieving relevant passages from a large amount of web text given the current question and its conversational history, which consists of previously asked questions and their answers (Dalton et al., 2019). Unlike open-domain question answering (ODQA) taking a single question (Voorhees and Tice, 2000; Chen et al., 2017), CS assumes a sequence of questions interactively taken from information seekers. Hence, the questions need to be understood with the conversational history to find relevant evidences at each turn.

Most of recent studies in CS propose a pipeline approach that decomposes CS task into conversational query rewriting (CQR) and passage retrieval, to encourage retrievers to focus on the retrieval task. In the approach, CQR models generate a standalone question from conversational input (Yu et al., 2020; Voskarides et al., 2020; Lin et al., 2021b; Qu et al., 2020; Wu et al., 2021), such that it could be directly fed into the off-the-shelf retrievers (e.g., BM25, dense retrievers). They achieve competitive performances on several benchmarks without re-training dense retrievers on the CS benchmarks. However, these retrievers do not leverage the conversational history, which should be considered to better understand the conversational context and accurately retrieve relevant passages.

To build a dense retriever that properly makes use of the conversational history, we first analyze a simple dense retriever baseline trained on one of the CS datasets. Our analysis shows us the exis-
tence of a retrieval shortcut in recent CS datasets, indicating dense retrievers heavily rely on the shortcut and retrieve irrelevant passages. Specifically, these shortcuts represent the spurious correlation between the conversational history and the relevant passages, pushing the dense retrievers to ignore current questions. For example, as illustrated in Figure 1, a dense retriever retrieves wrong passages only paying attention to ‘Russia’ and ‘World Cup’ mentioned in the previous history \((a_1, a_2)\) while ignoring the crucial cue ‘win the World Cup’ in the current question \(a_3\).

Motivated by our observation, we further test how much the shortcut contributes to the performance of current retrievers. First, we build a simple BM25 baseline, which only takes the previous conversational history as inputs, but still performs surprisingly well on QReCC (Anantha et al., 2021). Similarly, a dense retriever trained by feeding the conversational history without current question keeps 70-80% of the original performance. It implies significant effect of the shortcut dependency on dense retriever. From our analysis, we find the shortcut is more likely to be learned when the topic of conversation is constant. In other words, performance of the models drops especially when they are asked to answer history-independent questions.

To alleviate the overreliance on the shortcut, we explore using hard negative mining strategies, which have been recently proposed in ODQA (Xiong et al., 2020). In particular, we first warm up the dense retrievers with in-batch negatives. Then, we iteratively train our model with hard negatives mined from the previous checkpoint until its performance gain is saturated. Experimental results show the iterative hard negatives make remarkable improvements in various CS benchmarks and are especially helpful to history-independent questions, mitigating the dependency on the shortcut effectively. Our retrievers outperform baselines by 9.7 in Recall@10 on QReCC and 10.8 in Recall@5 on TopiOCQA (Anantha et al., 2021; Adlakha et al., 2021).

Our contributions are summarized in three folds:

- We reveal the presence of a retrieval shortcut in the conversational search and dense retriever dependent on the shortcut is poor at generalizing towards real scenario
- We show training the dense retriever with iterative hard negatives effectively mitigates the shortcut dependency by in-depth analysis
- We achieve a new state-of-the-art on recent CS benchmarks, QReCC and TopiOCQA, in both retrieval and end-to-end tasks.

## 2 Background

Let \(X_t = \{q_1, a_1, ..., a_{t-1}, q_t\}\) is a conversation input up to turn \(t\) where the \(q_t\) and \(a_t\) are the question and answer at turn \(t\). We assume \(M\) number of pre-chunked passages collection \(C = \{p_1, p_2, ..., p_M\}\) for the retrieval. Then, the formal objective of conversational passage retrieval is learning function \(f : (X_t, C) \rightarrow P_t\), where the \(P_t = \{p_1, p_2, ..., p_k\} \subset C\) and \(k \ll |C|\). In training time, \(p_t^+\), a ground-truth relevant passage corresponding to the \(X_t\), is used for supervision with its negative passages \(P_t^-\) in a contrastive manner. After the relevant passage set, \(P_t\), is retrieved, extractive or generative reader \(g\) can be followed to output answer \(a_t\) from the \(P_t\), i.e. \(g : (X_t, P_t) \rightarrow a_t\).

On the other hand, conversational query rewriting (CQR) is a generative task that rewrites the conversational input \(X_t\) into a standalone question \(q_t^*\) (Yu et al., 2020; Voskarides et al., 2020; Lin et al., 2021b; Qu et al., 2020; Wu et al., 2021). In training time, the CQR model learns to minimize negative log-likelihood between \(q_t^* \leftarrow CQR(X_t)\) and \(q_t^*\), where the \(q_t^*\) is a ground-truth human rewrite. Then, existing retrieval systems such as BM25 take the standalone question \(q_t^*\) to find \(P_t\) at inference time, i.e. \(f(CQR(X_t), C) \rightarrow P_t\). As a result, the CQR approaches do not require to re-train additional retriever in a conversational manner. However, the approach is limited in triggering information loss while rewriting whole conversational inputs into the standalone question.

## 3 Retrieval Shortcut

First, we demonstrate the presence of the shortcut in CS datasets. Formally, we define the shortcut as where \(p_t^+\) can be predicted in top-k predictions even without the current question \(q_t\). Then, we show how heavily dense retriever relies on the shortcut and how its overall performances are overestimated.

### 3.1 Lexical Analysis

We investigate whether there are lexical cues to predict relevant gold passages in CS. Specifically, we input \(X_t\backslash\{q_t\} = \{q_1, a_1, ..., a_{t-1}\}\) to the BM25 to measure the shortcut. Figure 2 (a) shows the result. Surprisingly, we observe the BM25(\(X_t\backslash\{q_t\}, C\)) achieves 58.4 for R@10 on
QReCC dataset (Anantha et al., 2021) even without the current question \( q_t \). The performance drop compared to BM25\((X_t, C)\) is only 6.7 in R@10, indicating \( X_t \setminus \{ q_t \} \) contains enough lexical cues to predict \( p_t^+ \). However, a model taking only current question \( q_t \) does not predict the gold passage well since it does not contain enough lexical cues. Instead, the previous answer \( a_t-1 \) is more responsible for the performance, achieving 46.4 of R@10.

### 3.2 Comparison with Biased Model

To examine how dense retriever trained on the dataset behave, we train two Dense Passage Retriever (DPR) models with in-batch negatives (Karpukhin et al., 2020) by feeding \( X_t \) and \( X_t \setminus \{ q_t \} \) as input query to each model. We refer each models as DPR and DPR\(_{\text{shortcut}}\), respectively. Surprisingly, we find the DPR\(_{\text{shortcut}}\) performs 78% of R@10 and 85% of R@100 compared to DPR as shown in Figure 2 (b). Thus, we conclude the DPR model is highly dependent on the shortcut. Following Wu et al. (2021), we also compare the DPR models with a CQR-based baseline, GPT2QR (Anantha et al., 2021). Note that the GPT2QR is less likely to be exposed to the shortcut since it uses the standalone question \( q_t \) without the history. DPR performs better than GPT2QR\(_{\text{BM25}}\) in R@10 despite the shortcut dependency. Thus, we hypothesize that the performances of DPR are overestimated due to the shortcut in the dataset.

### 3.3 Breakdown by Question Types

To probe when and how models take the shortcut, we break down the evaluation results by question types as in Wu et al. (2021). Specifically, we define three question types, first, no-switch, and switch. The first question is literally first question of conversation without any history. The no-switch and switch questions can be distinguished by whether \( p_t^+ \) contains similar or same topics as \( p_{t-1}^+ \), where \( t > 1 \).

Figure 2 (c) shows the breakdown result of R@10 on each baseline model. As hypothesized, the shortcut-dependent model DPR\(_{\text{shortcut}}\) achieves competitive performance with the DPR in no-switch questions, which can benefit from previous conversational history. However, the performances in other two types, first and switch, drop significantly. Similarly, when we compare DPR with the GPT2QR\(_{\text{BM25}}\), we find the performance at no-switch turn largely contributes to the gain while degraded in first and switch types. As a result, its overreliance on the shortcut prevents the model from generalizing towards real scenarios where a large proportion of topic-switching questions could appear. Thus, we claim that the proper ways to take the shortcut could improve the overall score with performance gains at the first and switch turns while keeping them at the no-switch.

---

1More details are in Appendix A.
4 Approach

We hypothesize the DPR model is poorly representative of the conversational input, which results in learning the shortcut since it is trained by using only uninformative in-batch negatives (Xiong et al., 2020). Thus, we examine iterative hard negative mining recently proposed in ODQA (Xiong et al., 2020) to push retrievers to learn better representations. In this section, we demonstrate a detailed method for the hard negative mining in conversational search.

4.1 Training

We employ Dense Passage Retriever (DPR) (Karpukhin et al., 2020) to perform the conversational passage retrieval. DPR consists of two transformer encoders (Vaswani et al., 2017; Devlin et al., 2019), $E_Q$ and $E_P$, for encoding conversational input and passages, respectively. Each encoder takes the $X_t$ and $p$, a passage in the $C$, to represent a $d$ dimensional vector. Then, we can compute the similarity between the representations via dot product.

$$sim(X_t, p) = E_Q(X_t)^T E_P(p)$$ (1)

Given the input $X_t$, the encoders are trained in a contrastive manner with the negative passages $P_t^- = \{p_{t1}, p_{t2}, \ldots, p_{tp-1}\}$ and its corresponding positive passage $P_t^+$. $L = -log \frac{e^{sim(X_t, p_t^+)} }{ e^{sim(X_t, p_t^+)} + \sum_{j} e^{sim(X_t, p_{tj}^-)}}$ (2)

Basically, we adopt in-batch negatives (IBN) to obtain the $P_t^-$ (Karpukhin et al., 2020). For each query representation, it computes the similarity score with other $(B - 1)$ number of passage representations except for its gold relevant passage in the same batch, where the $B$ is batch size. The IBN could be one of the intuitive options to construct the negative examples while reducing memory consumption. However, they are often easy to be distinguished from the $p_t^+$. As a result, training encoders with IBN could lead to suboptimal results (Xiong et al., 2020).

4.2 Inference

In inference time, we perform approximate nearest neighbor (ANN) search between $X_t$ and $C$ (Johnson et al., 2017a). Specifically, the function $F_{\theta}(X_t, C, k)$ returns top-$k$ relevant passages from the large corpus $C$ with given input query $X_t$, where the $\theta$ is trained model checkpoint$^2$.

$$F_{\theta}(X_t, C, k) = \text{argsort}(sim(X_t, p))[:k]$$ (3)

4.3 Iterative Hard Negative

To enhance the representation of DPR, we follow the approach of model-based hard negative mining introduced by ANCE (Xiong et al., 2020). However, there are two differences compared to the ANCE in implementation details. First, we sample negatives iteratively before training model while they are simultaneously done in ANCE in an asynchronous manner. Second, we use the mined hard negatives along with in-batch negatives. It is an efficient approximation of the ANCE in reducing engineering burdens. We refer to described our method as iterative hard negatives (IHN) mining.

In particular, we assume $I$ number of training iterations and denote $i$-th trained model checkpoint as $\theta_{iter=i}$. At each $i$-th iteration, we train for a specific number of epochs from the same initial checkpoint $\theta_{iter=0}$ not the $\theta_{iter=i-1}$. We construct hard negative set $N_{iter=i-1}$ by retrieving top-$k$ passages using $\theta_{iter=i-1}$ for training $\theta_{iter=i}$. Then, we randomly sample $n$ hard negatives from the $N_{iter=i-1}$. Finally, we use the $n$ number of hard negatives in addition to the in-batch negatives. Note that the $P_t^-$ is $(B - 1 + n)$ number of negatives per a training example.

$$N_{iter=i}^-= F_{\theta_{iter=i}}(X_t, C, k)$$ (4)

At the first training iteration ($iter = 1$), we can not take the iterative hard negatives since there is no prior checkpoint. Thus, we conduct warm-up training solely relying on in-batch negatives.

For $iter > 1$ iterations, we train our $i$-th model with the iterative hard negatives, which leverages $N_{iter=i-1}^-$. We repeat these procedures until the metric scores for each downstream task are saturated, which means we select the whole iteration numbers $I$ in empirically (Appendix D). The implementation details of our models are in Appendix C.

5 Experimental Setup

5.1 Dataset

We mainly conduct experiments on recent CS benchmarks, OR-QuAC, QReCC, and Topi-

$^2$The $\text{argsort}_{j \in I}(f(j))$ is a function that returns sorted $J$ in descending order of $f(j) \in \mathbb{R}^1$. 

4
OCQA (Qu et al., 2020; Anantha et al., 2021; Adlakha et al., 2021). We briefly describe the procedures of data construction and features of each dataset. Please see Appendix B for more details including dataset statistics (Table 5).

**OR-QuAC** Qu et al. (2020) extend one of the popular CQA datasets, QuAC (Choi et al., 2018) to the open-domain setting by aligning relevant passages with the questions in QuAC5. Moreover, they facilitate CQR as a subtask by reusing examples in CANARD (Elgohary et al., 2019). For retrieval, they construct passage collections from Wikipedia. However, the dataset has limitations in that all questions in the same conversation share the same gold passage. In other words, most of the questions in OR-QuAC are no-switch type. Thus, it is vulnerable to the shortcut. Even though it is far from the real world scenario, we include OR-QuAC to compare previous dense retrieval approaches (Lin et al., 2021a; Yu et al., 2021).

**QReCC** Anantha et al. (2021) construct QReCC dataset based on three existing datasets, QuAC, Natural Questions (NQ), and TREC (Choi et al., 2018; Kwiatkowski et al., 2019; Dalton et al., 2020)4. To annotate gold passage, they reuse conversational questions in QuAC and CAsT as in Qu et al. (2020), while collecting new questions for the NQ dataset. Given a question randomly selected from NQ, each crowdworker alone generates not only the following questions but also their corresponding answers. Even though it contains more diverse and realistic questions than the OR-QuAC, most of the questions (78%) still belong to the QuAC, causing models to exploit the shortcut.

**TopiOCQA** Adlakha et al. (2021) introduce a new challenge to promote topic-switching in conversational search, assuming a more realistic scenario of information seeking 5. On the contrary to QReCC, Adlakha et al. (2021) divide the annotator’s roles into questioner and answerer. They do not allow the questioners to access the relevant passages, which encourages them to ask information seeking questions. Moreover, they enable topic-switching by providing hyperlinks for switching to related documents. As a result, it has about four topic-switching in a conversation on average, but still contains lots of no-switch questions (64% of train set).

### 5.2 Baselines

For the passage retrieval task, we include three groups of baselines, BM25, conversational query rewriting (CQR) based models, and dense retriever models when they are available. Especially, the BM25($X^t \{q_t\}, C$) largely depends on lexical cues in conversational history without the current turn question $q_t$ (Section 3). Thus, it represents a lower bound that uses the shortcut.

CQR-based models (Anantha et al., 2021; Adlakha et al., 2021; Wu et al., 2021) are baselines that use the standalone question instead of directly encoding a conversation for the input of off-the-shelf retriever such as BM25 or Dense Encoder (DE) (Ni et al., 2021) finetuned on ODQA dataset. Anantha et al. (2021) propose GPT2QR as baseline model which is GPT-2 (Radford et al., 2019) based CQR model. T5QR and CONQRR (mix, RL) are based on T5 (Raffel et al., 2020) for the CQR (Adlakha et al., 2021; Wu et al., 2021). Especially, Wu et al. (2021) train the CONQRR using reinforcement learning (RL) against retrieval metrics (MRR, Recall) as reward signals. The mix and RL indicate that the training is mixed with supervision and relies purely on RL.

For dense retriever baselines, we include DPRshortcut (IBN), which encodes only previous history $X^t \{q_t\}$ used in Section 3 and DPR (IBN) trained with in-batch negatives. We also include DPR model reported by Adlakha et al. (2021) on TopioCQA. On OR-QuAC, we compare our models with previously proposed dense retrievers, CQE (Lin et al., 2021a) and ConvDR (Yu et al., 2020). They both require ground-truth human rewrite $q^*_t$ for hard negative mining based on off-the-shelf retrievers.

### 6 Experimental Results

For the passage retrieval, we report scores among Mean Reciprocal Rank (MRR) and Recall (R@K where the $K \in \{5, 10, 20, 100\}$) is selected following previous works6. For end-to-end experiments (retrieve-and-read), we report Exact Match (EM) and F1 Scores.

---

1github.com/prdwbo/convoqa-release
2zenodo.org/record/5115890#.YgCWNfVBxhF
3github.com/McGill-NLP/topiocqa
4For QReCC, we use normalized evaluation script, which does not consider when there is no ground truth, by authors.
Table 1: Experimental results on QReCC (test), and TopiOCQA (dev). The IBN and IHN indicate the model is trained by using In-Batch Negatives and Iterative Hard Negatives, for respectively. The DPR \( \text{shortcut} \) is model regarding previous history \( X_t \{q_t\} \) as input query discussed in Section 3. Note that most CQR approaches are not tested on TopiOCQA since the dataset does not contain the human rewrites. The \( \dagger \) indicates the performances on TopiOCQA are based on zero-shot inference by the CQR model trained on QReCC. The \( \ast \) means our evaluation based on released model predictions by Anantha et al. (2021) and Adlakha et al. (2021).

### 6.1 Main Results

Table 1 shows the retrieval performances of baseline models and our methods on CS benchmarks, QReCC and TopiOCQA. Our models trained with iterative hard negatives (IHN) achieve remarkable performances and become a state-of-the-art method advancing the state-of-the-art by 9.7 (R@10) and 10.8 (R@5), respectively. In QReCC, our DPR (IHN) model outperforms all baselines, including BM25 and CQR-based models, while DPR trained with only in-batch negatives (IBN) does not. Especially, the IHN significantly improves the performances compared to DPR (IBN) that would suffer from overreliance on the shortcut as hypothesized in Section 3.

On TopiOCQA, the lexical baselines (BM25) and shortcut-dependent model (DPR\(\ast\)shortcut (IBN)) do not show strong performances as in QReCC. It might be because they could not handle switch questions well, while the TopiOCQA contains lots of topic-switching questions. However, there still remains a danger of exposing to the shortcut in that the DPR\(\ast\) shortcut (IBN) achieves approximately 34% and 53% of R@20 and R@100 scores of the DPR (IBN). Our training with IHN improves 10.8 in R@5 and 6.1 in R@20 compared to the DPR\(\ast\) (Adlakha et al., 2021). We further discuss the effectiveness of the IHN in mitigating the dependency on the shortcut in the following sections.

Table 2 shows results on OR-QuAC. Please note that all models take only multi-round questions \( Q_t = \{q_1, q_2, ..., q_t\} \) instead of \( X_t \) as input following previous works. The \( \dagger \) indicates the CQE model performs zero-shot inference and dimensionality reduction (Lin et al., 2021a).

### Retriever Finetuning

<table>
<thead>
<tr>
<th>Model</th>
<th>MRR</th>
<th>R@5</th>
</tr>
</thead>
<tbody>
<tr>
<td>BM25(q_t, C)</td>
<td>0.043</td>
<td>5.6</td>
</tr>
<tr>
<td>BM25(Q_{t-1}, C)</td>
<td>0.170</td>
<td>21.3</td>
</tr>
<tr>
<td>BM25(Q_t, C)</td>
<td>0.198</td>
<td>24.9</td>
</tr>
<tr>
<td>ALBERT (Qu et al., 2020)</td>
<td>0.225</td>
<td>31.4</td>
</tr>
<tr>
<td>CQE(\dagger) (Lin et al., 2021a)</td>
<td>0.266</td>
<td>36.5</td>
</tr>
<tr>
<td>ConvDR (Yu et al., 2021)</td>
<td>0.616</td>
<td>75.0</td>
</tr>
<tr>
<td>DPR (IBN) (Ours)</td>
<td>0.525</td>
<td>63.9</td>
</tr>
<tr>
<td>DPR (IHN) (Ours)</td>
<td>0.633</td>
<td>75.9</td>
</tr>
</tbody>
</table>

Table 2: Experimental result on OR-QuAC. Please note that all models take only multi-round questions \( Q_t = \{q_1, q_2, ..., q_t\} \) instead of \( X_t \) as input following previous works. The \( \dagger \) indicates the CQE model performs zero-shot inference and dimensionality reduction (Lin et al., 2021a).
Table 3: Breakdown results by three question types on the QReCC dataset. The $P^-$ indicates hard negative mining strategy.

<table>
<thead>
<tr>
<th>Type</th>
<th>Model $\rightarrow$</th>
<th>$P^-$</th>
<th>R@10</th>
<th>R@100</th>
</tr>
</thead>
<tbody>
<tr>
<td>first</td>
<td>GPT2QR</td>
<td>56.1</td>
<td>99.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DPR$_{\text{shortcut}}$</td>
<td>IBN</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>DPR</td>
<td>IBN</td>
<td>51.7</td>
<td>75.6</td>
</tr>
<tr>
<td></td>
<td>DPR</td>
<td>IHN</td>
<td>69.3</td>
<td>85.5</td>
</tr>
<tr>
<td>switch</td>
<td>GPT2QR</td>
<td>52.8</td>
<td>88.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DPR$_{\text{shortcut}}$</td>
<td>IBN</td>
<td>26.0</td>
<td>48.0</td>
</tr>
<tr>
<td></td>
<td>DPR</td>
<td>IBN</td>
<td>49.0</td>
<td>70.9</td>
</tr>
<tr>
<td></td>
<td>DPR</td>
<td>IHN</td>
<td>59.9</td>
<td>81.3</td>
</tr>
<tr>
<td>no-switch</td>
<td>GPT2QR</td>
<td>65.7</td>
<td>88.9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DPR$_{\text{shortcut}}$</td>
<td>IBN</td>
<td>73.5</td>
<td>88.6</td>
</tr>
<tr>
<td></td>
<td>DPR</td>
<td>IBN</td>
<td>76.5</td>
<td>90.0</td>
</tr>
<tr>
<td></td>
<td>DPR</td>
<td>IHN</td>
<td>88.5</td>
<td>93.4</td>
</tr>
<tr>
<td>all</td>
<td>GPT2QR</td>
<td>50.5</td>
<td>82.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DPR$_{\text{shortcut}}$</td>
<td>IBN</td>
<td>44.9</td>
<td>60.9</td>
</tr>
<tr>
<td></td>
<td>DPR</td>
<td>IBN</td>
<td>58.5</td>
<td>77.2</td>
</tr>
<tr>
<td></td>
<td>DPR</td>
<td>IHN</td>
<td>74.8</td>
<td>87.5</td>
</tr>
</tbody>
</table>

(IHN) consistently outperforms previous baselines. While ConvDR (Yu et al., 2020) shows competitive performances with ours, ConvDR requires ground-truth human rewrite $q^*$ for hard negative mining. On the other hand, our IHN achieves better performances without any usage of query rewriting.

Thus, we conclude our method improves the performances when only no-switch questions are given.

6.2 What does the IHN actually improve?

To examine the actual effects of the IHN training, we break down the results by three question types as discussed in Section 3. Table 3 shows the breakdown results on QReCC dataset. Our DPR (IHN) model consistently outperforms in R@10 on all three question types. Unlike our DPR models, the GPT2QR is not likely to be exposed the shortcut since it passes generated standalone question $q^*$ to BM25 as input query. We observe the DPR (IBN) underperforms the GPT2QR in first and switch questions. Also, there are tiny gaps between DPR$_{\text{shortcut}}$ and DPR (IBN) in no-switch type of questions which can benefit from the shortcut. The IHN significantly improves performances of first and switch questions with improvements of 17.6 and 10.9 in R@10. We then conclude that our retrievers perform well regardless of question types by reducing the heavy dependency on the shortcut.

Figure 3 shows the breakdown results on the TopiOCQA dataset by question types and training iterations. The model only trained with IBN ($i = 1$) shows large performance gap between switch and no-switch questions. Training with IHN ($i = 3$) improves 20.0 and 22.0 of R@5 for first and switch type of questions while it still improves 16.3 of no-switch type. In summary, as the iteration progresses, all scores increase regardless of their question types. In particular, scores in first questions have soared drastically. Moreover, performance gap between switch and no-switch questions is getting smaller. It thus implies the IHN effectively relaxes the shortcut dependency of dense retrievers in conversational search.

6.3 End-to-End Experiments

To examine how our retrievers improve the end-to-end QA performance, we report final QA scores on two benchmarks, QReCC and TopiOCQA. Following Adlakha et al. (2021), DPR reader (extractive) or FiD (generative) (Karpukhin et al., 2020; Izacard and Grave, 2021) architectures are sequentially combined to our retriever. As shown in Table 4, when DPR readers encode the passages retrieved from our retrievers, DPR (IBN) and DPR (IHN), they achieve comparable performance with the baseline model DPR$^*$ on TopiOCQA dataset. However, our readers significantly outperform previous SOTA models relying on CQR-based retrieval, GPT2QR and T5QR (Anantha et al., 2021; Vakulenko et al., 2022) by 3.7 and 11.5 in EM and F1.
Table 4: Result of end-to-end experiment on QReCC dataset. It is notable that our FiD readers with DPR (IHN) retriever outperform the best performing baseline by 3.7 and 0.7 EM scores in QReCC and TopiOCQA, respectively. The training details of readers are in Appendix F.

### 7 Related Works

**Dataset Bias in NLP**  Our research is conducted by referring analyses of dataset biases in the machine learning field and ways to tackle them (Geirhos et al., 2020). Models in sentence classification such as natural language inference (NLI) suffer from statistics skewed for some keywords or ignorance of part of inputs (Poliak et al., 2018; Belinkov et al., 2019; McCoy et al., 2019; Lin and Su, 2021). On visual question answering, models often disregard evidence images and solely depend on the language prior when answering the questions (Agrawal et al., 2016; Goyal et al., 2017; Johnson et al., 2017b; Agrawal et al., 2018; Niu et al., 2021). Several studies in QA also demonstrated that QA models only exploit shallow features without leveraging the entire information in the given document (Chen et al., 2016; Min et al., 2018, 2019; Ko et al., 2020). Recent studies proposed methods to alleviate unknown biases (Utama et al., 2020). In this paper, we define and thoroughly analyzes a novel shortcut that often exists in the CS task, and propose the method to use it properly.

**Existing methods in Conversational Search**  CQR approaches are also widely studied in CS since standalone questions could be directly fed into the off-the-shelf retrievers (i.e., BM25) (Yu et al., 2020; Voskarides et al., 2020; Lin et al., 2021b; Kumar and Callan, 2020; Anantha et al., 2021; Wu et al., 2021). As a result, there is no need to train a new retriever in a conversational manner. However, the CQR approaches yield long latency at inference time since its autoregressive nature of the generation. To overcome the limitation, Yu et al. (2021); Lin et al. (2021a) attempt to train dense retrievers to directly represent the multi-round questions into a single dense vector. As a result, efficient search like maximum inner product becomes convenient (Johnson et al., 2017a). They usually focused on few-shot adaptation or weakly supervision utilizing other accessible resources. Both of them utilize the standalone questions to mine hard negatives and knowledge distillation from off-the-shelf retrievers trained on ODQA, regarding it as a teacher model. On the other hand, we explore various schemes for better retrievers, without using any CQR process.

**Hard Negative Mining for Dense Retrieval**  Karpukhin et al. (2020) try to finetune dense retriever on downstream QA dataset without pretraining. They show the effectiveness of hard negative based on BM25. Xiong et al. (2020) argue that local in-batch negatives are suboptimal since their easy-to-distinguish features. They propose ANCE training with approximated global negatives from pre-trained checkpoint and report its effectiveness in training dense retrievers. Qu et al. (2021) present optimized training techniques, including the model-based hard negative mining and de-noising process over the hard negatives using cross-encoder.

**8 Conclusion**  In this work, we show the presence of the shortcut in conversational search, which causes dense retriever often heavily relies on it when trained on in-batch negatives. We find the shortcut dependency hurts the generalization ability of dense retrievers. To save the model from relying on the shortcut, we examine iterative hard negative mining recently proposed in ODQA. The retriever trained with iterative hard negatives appropriately takes beneficial information of the shortcut only when needed, and achieves the state-of-the-art performance on multiple CS benchmarks.
References


Aishwarya Agrawal, Dhruv Batra, and Devi Parikh. 2016. Analyzing the behavior of visual question answering models. In EMNLP.

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2018. Don’t just assume; look and answer: Overcoming priors for visual question answering. In CVPR.


A Details of Question Types

We classify the no-switch and switch questions using dot product score between BM25 vectors of \( p_{i-1}^t \) and \( p_i^t \) as threshold in QReCC dataset. This is similar with division of topic-concentrated and topic-shifted questions in Wu et al. (2021) while we take them only when \( t > 1 \) to distinguish them from first questions. The number of subsets is 267, 279, and 573 for the first, no-switch, and switch respectively. Please note that the sum of each subset is not equal to the number of all (8209) since we take the question types from only NQ and TREC subdomains in the QReCC dataset as in Wu et al. (2021). In TopiOCQA dataset, we divide the questions according to whether title of \( p_i^t \) and \( p_{i-1}^t \) are same or not, following Adlakha et al. (2021).

B Details of Dataset Statistics

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Train</th>
<th>Dev</th>
<th>Test</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>OR-QuAC</td>
<td># C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4,383</td>
<td>490</td>
<td>771</td>
<td>11M</td>
</tr>
<tr>
<td>QReCC</td>
<td># C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>8,823</td>
<td>2,000</td>
<td>2,775</td>
<td>54M</td>
</tr>
<tr>
<td>TopiOCQA</td>
<td># C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3,509</td>
<td>205</td>
<td>-</td>
<td>25M</td>
</tr>
<tr>
<td></td>
<td># Q</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>45,450</td>
<td>2,514</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Dataset statistics used in our experiments. The # C and # Q indicate the number of conversations and questions, respectively.

Table 5 shows dataset statistics we used. For QReCC, we newly select the development set by sampling 2k conversations from the train set, since Anantha et al. (2021) combined them into the train set when the dataset is released. We mainly evaluate our model on the development set since test split is not available in the version for now. For model development, we use smaller collections (6.9k) provided by the authors for OR-QuAC (Qu et al., 2020). Hence, we choose 7.3k and 10.8k corresponding passages for development on QReCC and TopiOCQA, respectively.

C Implementation Details: Retriever

DPR pre-trained on NQ dataset (Kwiatkowski et al., 2019) of Karpukhin et al. (2020) is used for the initial checkpoint \( \theta_{iter=0} \). We set maximum sequence length to 128 and 384 for \( X_t \) and \( p \), respectively. The whole number of training iterations \( I \) is set to 2 for OR-QuAC and QReCC, but 3 for TopiOCQA. Basically, we train each iterations for 10 epochs with 3e-5 of learning rate (lr). However, we choose 40 for the number of epochs at the last training iteration of TopiOCQA. For optimization, AdamW is used with 0.1 warming up ratio for linear lr decay scheduling (Kingma and Ba, 2017). We first construct top 100 passages for the iterative hard negatives and select 1 from them per each training steps (\( k = 100 \) and \( n = 1 \)). Batch size is set to 128 for OR-QuAC and TopiOCQA, and 256 for QReCC in the warming up training (\( i = 1 \)) and becomes half when \( i > 1 \). We choose the best performing model based on dev set (dev examples and \( C_{dev} \)) and report \( I \)-th checkpoint as final model for each datasets. We use IndexFlatIP index of FAISS (Johnson et al., 2017a) for ANN search in the inference time.

D Performance by Training Iterations

Figure 4 shows Recall@100 of our models on TopiOCQA and QReCC datasets by their training iterations.

<table>
<thead>
<tr>
<th>Hard negatives from</th>
<th>MRR</th>
<th>R@10</th>
<th>R@100</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPR_{iter=i-1}(X_t,C)</td>
<td>0.516</td>
<td>74.8</td>
<td>87.5</td>
</tr>
<tr>
<td>BM25(q^*_t,C)</td>
<td>0.498</td>
<td>72.1</td>
<td>85.9</td>
</tr>
<tr>
<td>DPR_{iter=0}(q^*_t,C)</td>
<td>0.482</td>
<td>70.8</td>
<td>85.6</td>
</tr>
</tbody>
</table>

Table 6: Comparison over various hard negative mining strategies on the QReCC dataset.

E Other Hard Negatives

We compare our iterative hard negative mining with other alternative methods based on off-the-
shelf retrievers, BM25 or DPR\textsubscript{iter}=0 trained on NQ dataset, when the ground-truth query rewrite $q^*_{t}$ is available. Even though we leverage the $q^*_{t}$ for hard negative mining like Lin et al. (2021a); Yu et al. (2020), training within iterative hard negatives using DPR\textsubscript{iter}=i−1 still performs better on the QReCC dataset as shown in Table 6.

F Implementation Details: Reader

For both DPR and FiD readers, we mainly follow hyper-parameter setups that are suggested in the previous work (Adlakha et al., 2021), otherwise specified. We use base model size for all readers. We choose the model checkpoint that shows the best EM score on the development set and use it as the final model. On QReCC we set the maximum input sequence length and the maximum answer length as 512 and 30 for DPR reader, respectively, following Vakulenko et al. (2022). Our FiD models are trained on top 50 passages selected by retrieval models. Each encoder of FiD reader has the maximum length of 384 and 100 for input and output, respectively. To circumvent the memory intensive computation, we use the checkpoint option that is provided in Izacard and Grave (2021).

G More Experimental Details

For input representation of dense retrieval, all history is concatenated with a [SEP] token in between. We retrain the first question and truncate tokens from the left side up to the maximum length of 128 for conversational inputs. We concatenate title information to the passage only when they are available (OR-QuAC, TopiOCQA). For QReCC, we only regard the examples that contain ground truth relevant passages. Thus, the actual number of training examples is 24,283. We also find weakly supervision harms the final performances (Karpukhin et al., 2020). We use 0.82 and 0.68 for parameters of BM25, $k_1$ and $b$, respectively in QReCC and set them to 0.9 and 0.4 for other datasets (OR-QuAC, TopiOCQA). All our experiments is based on Transformers library (Wolf et al., 2020) using \{4, 8\} 32GB V100 GPUs.

H Computational Cost

Overall computational cost is summarized in Table 7. Please note that the number of passages collection and test set of QReCC is much larger than others. Thus, we allocate 8 GPUs for QReCC and 4 GPUs for others to perform training and indexing. The intensive computational cost for the mining procedures, i.e., indexing and inference of dense retriever over huge passages collection, is one of our limitations.

<table>
<thead>
<tr>
<th>Data</th>
<th>Training</th>
<th>Indexing</th>
<th>Inference</th>
</tr>
</thead>
<tbody>
<tr>
<td>OR-QuAC</td>
<td>2h</td>
<td>8h</td>
<td>40m</td>
</tr>
<tr>
<td>QReCC</td>
<td>2h</td>
<td>28h</td>
<td>11h</td>
</tr>
<tr>
<td>TopiOCQA</td>
<td>3h</td>
<td>19h</td>
<td>37m</td>
</tr>
</tbody>
</table>

Table 7: Summarized computational cost (run-time) for each training, indexing, and inference of dense retrieval. The target of each function is train set, passages collection, and test or dev set.