Saving Dense Retriever from Shortcut Dependency in Conversational Search

Sungdong Kim1,2 Gangwoo Kim3⋄
NAVER AI Lab1 KAIST AI2 Korea University3
sungdong.kim@navercorp.com gangwoo_kim@korea.ac.kr

Abstract
Conversational search (CS) needs a holistic understanding of 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 that naively trained dense retrievers heavily exploit the shortcut and hence perform poorly when asked to answer history-independent questions. To build more robust models against shortcut dependency, we explore various hard negative mining strategies. Experimental results show that training with the model-based hard negatives (Xiong et al., 2020) effectively mitigates the dependency on the shortcut, significantly improving dense retrievers on recent CS benchmarks. In particular, our retriever outperforms the previous state-of-the-art model by 11.0 in Recall@10 on QReCC (Anantha et al., 2021).1

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 evidence at each turn.

To build a retriever that properly makes use of the conversational history, we first analyze a simple dense retriever baseline trained on one of the CS datasets, QReCC (Anantha et al., 2021). Our analysis shows us the existence 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, a2 (Red dashed line). We show the shortcut dependency is harmful to robust retrieval.

Figure 1: An example of a retrieval shortcut in conversational search. While we expect the retriever to predict relevant passages by using all conversational inputs up to q3 (Blue solid line), a dense retriever often ignores current turn question q3 and only exploits previous history, a2 (Red dashed line). We show the shortcut dependency is harmful to robust retrieval.

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 input, but still performs

1 Work done while interning at NAVER AI Lab
2 The code is available at github.com/naver-ai/cs-shortcut.
surprisingly well on QReCC. Similarly, a dense retriever trained by feeding the conversational history without the current question keeps 70-80% of the original performance. It implies a significant effect of the shortcut dependency on dense retrievers. 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 and CS (Karpukhin et al., 2020; Xiong et al., 2020; Yu et al., 2020; Lin et al., 2021b). Experimental results show the model-based 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 11.0 in Recall@10 on QReCC.

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 toward a real scenario.
- We show training the dense retriever with hard negatives effectively mitigates the heavy shortcut dependency by in-depth analysis.
- We achieve a new state-of-the-art of recent CS benchmarks, QReCC and OR-QuAC.

2 Background and Related Work

Let \( X_t = \{q_1, a_1, ..., a_{t-1}, q_t\} \) is a conversation up to turn \( t \) where the \( q_t \) and \( a_t \) are the question and answer at turn \( t \). We assume pre-chunked passages collection \( C = \{p_1, p_2, ..., p_{|C|}\} \) for the retrieval. Then, the formal objective of conversational search is learning function \( f : (X_t, C) \rightarrow P_t \), where the \( P_t = \{p_1, p_2, ..., p_k\} \subset C \) and \( k \ll |C| \).

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., 2021c; Kumar and Callan, 2020; Anantha et al., 2021; Wu et al., 2021). Then, existing retrieval systems such as BM25 take the standalone question \( q'_t \) to find \( P_t \) at inference time, i.e. \( f(q'_t, C) \rightarrow P_t \). 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 and long latency while rewriting the conversation into the standalone question.

To overcome the limitations, Yu et al. (2021); Lin et al. (2021b) attempt to train dense retrievers to directly represent the multi-round questions into a single dense vector. They usually focused on few-shot adaptation or weak supervision utilizing other accessible resources including the standalone questions for hard negative mining.

3 Retrieval Shortcut

First, we demonstrate the presence of the shortcut in CS datasets. Formally, we define the shortcut as where gold passage \( p^+_t \) 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 performance is overestimated.

3.1 Lexical Analysis

We investigate whether there are spurious lexical cues to predict relevant gold passages in CS. Specifically, we input \( X_t \setminus \{q_t\} = \{q_1, a_1, ..., a_{t-1}\} \) to the BM25 to measure the shortcut. Figure 2 (a) shows the result. Surprisingly, we observe the BM25 taking \( X_t \setminus \{q_t\} \) achieves 58.4 for R@10 on QReCC (Anantha et al., 2021) even without the current question \( q_t \). It retains about 90% of its original performance from BM25 \( (X_t \text{ as an input}) \), 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 Lower and Upper bounds Analysis

To examine how dense retriever trained on the dataset behave, we contrast a dense retriever with its lower and upper bound models in terms of dependency on the retrieval shortcut. For this, 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.\(^2\) We denote the latter one as DPR\(^\oplus\), and it represents the lower bound model that does not consider the current question \( q_t \) at all. Surprisingly, we find the DPR\(^\oplus\) performs 78% of R@10 and 85%

\(^2\)Please see § 4.1 for the training details.
of R@100 compared to DPR as shown in Figure 2 (b). Thus, we presume the original DPR model is also likely to depend on the shortcut. Next, we introduce the upper bound model, GPT2QR (Anantha et al., 2021). It is less likely to be exposed to the shortcut since it first generates standalone question $q'_t$, and then its BM25 retriever only takes the decontextualized $q'_t$ as input. We also find that the DPR$^\otimes$ is comparable with GPT2QR in R@10 despite the heavy shortcut dependency. It reminds us the overall score is not enough to identify robust retrieval methods.

### 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 $p_{t-1}$ contains similar or same topics as $p_t$, where the $p_t$ is a gold passage at turn $t$ and $t > 1$.

Figure 2 (c) shows the breakdown result of R@10. The DPR$^\otimes$ 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, 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 toward real scenarios where a large proportion of topic-switching questions could appear (Adlakha et al., 2022). 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.

### 4 Experiments

We hypothesize random in-batch negatives promote the shortcut dependency of the vanilla DPR model because of their easy-to-distinguish nature. Thus, we examine hard negative mining as one of the solutions to push the retriever to exploit the shortcut properly. We mainly evaluate it on two CS benchmarks, QReCC and OR-QuAC (Anantha et al., 2021; Qu et al., 2020).

#### 4.1 Training Dense Retriever

DPR consists of two encoders, $E_Q$ and $E_P$, for encoding conversational input and passages, respectively. Each encoders takes the $X_t$ and $p$, a passage in the $C$, to represent $d$ dimensional vector.

Then, we can compute the similarity between the representations via dot product.

$$\text{sim}(X_t, p) = E_Q(X_t)^T E_P(p)$$

Given the input $X_t$, the encoders are trained in a contrastive manner with the negative passages...
$P^-_t = \{p^-_{t1}, p^-_{t2}, \ldots, p^-_{t|P^-|}\}$ and its corresponding positive passage $p^+_t$.

$$L = -\log \frac{e^{\text{sim}(X_t, p^+_t)}}{e^{\text{sim}(X_t, p^-_t)} + \sum_j e^{\text{sim}(X_t, p^-_j)}}$$

Basically, we adopt in-batch negatives 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.

4.2 Hard Negative Mining

The in-batch negative is one of the intuitive options to construct the negative examples while reducing memory consumption. However, it is often easy to distinguish from the $p^-_t$ and consequently encourages shortcut dependency. To reduce the dependency, we include a hard negative passage $p^-_{t_j}$ in the $P^-_t$. We first construct $k$ number of negative passages $N^-_i$ for each training instance. Then, we randomly sample a passage from the $N^-_i$ to include it in $P^-_t$ as the $p^-_{t_j}$. We denote off-the-shelf retriever to obtain top-k passages in $C$ from given input query $x$ as $F(x, C, k)$. Specifically, we compare three strategies for hard negative mining:

**BM25 Negs** De-facto strategy is BM25-based negative mining following Karpukhin et al. (2020). We mine the $N^-_i$ using whole conversational input $X_t$, i.e., $N^-_i \leftarrow$ BM25$(X_t, C, k)$.

**CQR Negs** If gold standalone question $q'_t$ is available for each $X_t$, we can leverage it to find the negative passages with off-the-shelf retriever as in Yu et al. (2020); Lin et al. (2021b), i.e., $N^-_i \leftarrow F(q'_t, C, k)$. For this, we employ another DPR pre-trained on Natural Questions (NQ) (Kwiatkowski et al., 2019) as the $F$.

**Model Negs** Lastly, we explore model-based hard negative mining proposed by Xiong et al. (2020). First, we train vanilla DPR model on the target dataset using only in-batch negative as in § 3. Then, we employ the model as $F$ to select negative passages, i.e., $N^-_i \leftarrow F(X_t, C, k)$.

4.3 Implementation Details

DPR pre-trained on NQ dataset (Kwiatkowski et al., 2019) of Karpukhin et al. (2020) is used for the initial checkpoint of our dense retrievers. It consists of two BERT encoders and 220M of learnable parameters (Devlin et al., 2019). We set maximum sequence length to 128 and 384 for $X_t$ and $p$, respectively. 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 $X_t$.

We train the models for 10 epochs with 3e-5 of learning rate (Ir). For optimization, AdamW is used with 0.1 warming up ratio for linear Ir decay scheduling (Kingma and Ba, 2017). We build top 100 passages for the hard negatives $N^-_i$, i.e., $k = 100$. Batch size is set to 128 for OR-QuAC and 256 for QReCC. We choose the best performing model based on dev set. We use Pyserini (Lin et al., 2021a) to implement BM25 and IndexFlatIP index of FAISS (Johnson et al., 2019) to perform dense retrieval.\(^5\)

4.4 Baselines

In QReCC, we include BM25 and BM25\(^5\) take $X_t$ and $X_t \setminus \{q_t\}$ as input query, respectively. For CQR baselines less dependent on the shortcut, we include GPT2QR and CONQRR (Anantha et al., 2021; Wu et al., 2021). They use standalone question instead of directly encoding a conversation for the input of off-the-shelf retriever such as BM25 or T5-DE (Ni et al., 2022) finetuned on ODQA dataset. Anantha et al. (2021) propose GPT2QR as baseline model which is GPT-2 (Radford et al., 2019) based CQR model. We only perform BM25 inference based on released model predictions by authors instead of re-training it. CONQRR is based on T5 (Raffel et al., 2020) for the CQR (Wu et al., 2021). Especially, Wu et al. (2021) train the CONQRR using reinforcement learning against retrieval metrics (MRR, Recall) as reward signals. We also include DPR and DPR\(^5\) without hard negative mining to represent shortcut-dependent model.

In OR-QuAC, we compare our models with previously proposed dense retrieval approaches in conversational search, CQE (Lin et al., 2021b) and ConvDR (Yu et al., 2020). Both of them utilize the standalone question $q'_t$ to mine hard negatives and knowledge distillation from off-the-shelf retrievers trained on ODQA, regarding it as a teacher model. Although they were not tested on QReCC, we can indirectly compare them with others using DPR with CQR Negs instead.

\(^5\)All our experiments is based on NSML platform (Sung et al., 2017; Kim et al., 2018) and Transformers library (Wolf et al., 2020) using \{4, 8\} 32GB V100 GPUs.
Table 1: Experimental results on QReCC test set (All) and its sub-splits by three question types discussed in §3. The ⊗ indicates the model takes only $X_t\{q_t\}$ as input.

| Model         | All MRR | R@10 | R@100 | first MRR | R@10 | R@100 | switch MRR | R@10 | R@100 | no-switch MRR | R@10 | R@100 |
|---------------|---------|------|-------|-----------|------|-------|------------|------|-------|----------------|------|-------|
| BM25          | 0.47    | 65.1 | 82.8  | 0.32      | 56.1 |       | 0.18       | 36.3 | 70.5  | 0.78           | 90.8 | 97.9  |
| BM25⊗         | 0.43    | 58.4 | 63.9  | -         | -    | -     | 0.16       | 32.8 | 65.3  | 0.76           | 90.3 | 96.5  |
| DPR           | 0.28    | 46.5 | 65.9  | -         | -    | -     | 0.14       | 30.2 | 56.7  | 0.54           | 75.5 | 88.4  |
| GPT2QR        | 0.32    | 50.5 | 82.3  | 0.32      | 56.1 |       | 0.30       | 52.8 | 88.0  | 0.46           | 65.7 | 88.9  |
| CONQRR        | 0.42    | 65.1 | 84.7  | -         | -    | -     | -          | -    | -     | -              | -    | -     |
| DPR w. CQR Negs | 0.39   | 59.1 | 77.6  | 0.36      | 55.7 | 77.6  | 0.29       | 50.3 | 71.5  | 0.60           | 80.8 | 90.8  |
| w. BM25 Negs  | 0.50    | 71.6 | 86.0  | 0.42      | 64.0 | 82.2  | 0.34       | 57.9 | 80.8  | 0.70           | 86.7 | 94.2  |
| w. Model Negs | 0.53    | 76.1 | 88.3  | 0.48      | 70.9 | 87.1  | 0.40       | 63.0 | 84.1  | 0.72           | 88.1 | 94.1  |

4.5 Results

We report scores among Mean Reciprocal Rank (MRR) and Recall (R@K, K ∈ {5, 10, 100}). Table 1 shows the retrieval performances of baseline models and hard negative mining methods on QReCC, and our findings are summarized:

Overall performances are not enough to distinguish robust methods in CS. We find lexical baselines, BM25 and BM25⊗, outperform CQR-based models, GPT2QR and CONQRR (Anantha et al., 2021; Wu et al., 2021) and vanilla DPR in MRR of overall retrieval performances (All). However, as we discussed in §3, the most performances are from no-switch questions which can benefit from the shortcut.

Hard negatives could mitigate shortcut dependency of dense retrievers. We observe the vanilla DPR underperforms the GPT2QR in first and switch questions. Also, there is a relatively smaller gap between DPR⊗ and DPR in no-switch type of questions. Compared to the vanilla DPR, all three negatives effectively improve the overall performance. Especially, the history-independent types, first and switch, are improved at most 12.7-15.2 in R@10 indicating relaxed shortcut dependency of the model. Figure 3 shows T-SNE visualizations (Van der Maaten and Hinton, 2008) to compare DPR models with and without hard negative training. The shortcut (blue multiply) passages are obtained by BM25⊗. The example is from 5th turn of conversation 1935 in QReCC test set, which is one of switch questions. Please see Appendix E for the corresponding qualitative example.

Among the negative mining methods, the model-based hard negative consistently outperforms others. We observe consistent results in other CS dataset, OR-QuAC (Qu et al., 2020) compared to previous works (Please see Appendix C). Moreover, our model achieves a new state-of-the-art with improvements of 11.0% point R@10.

5 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 that shortcut dependency hurts the generalization ability of dense retrievers. To save the model from relying on the shortcut, we study various hard negative mining strategies. The retriever trained with hard negatives appropriately takes beneficial information of the shortcut only when needed and achieves the state-of-the-art performance on multiple CS benchmarks.

Limitations

Even if we explain the existence of shortcut in conversational search, we could not suggest specific solutions to the shortcut dependency of dense retrievers. In the preliminary study, we tried other meth-
ods, e.g., history masking to promote the model attending more to the current question, but we found those methods are not effective as the hard negative mining in terms of shortcut dependency. However, we believe our work is an important step toward more robust conversational search.

Another limitation is the implementation cost to perform the model-based hard negative mining, i.e., indexing and inference of dense retriever over huge passages collection. Please see Appendix D for the details. Especially, the cost is increased notoriously according to the number of passage collections. We expect a more efficient method to balance shortcut dependency in future works.

Acknowledgements

The authors would like to thank to Kyunghyun Cho, Jinhyuk Lee, Minjoon Seo, Sang-Woo Lee, Hwaran Lee and other members of NAVER AI for their constructive comments.

References

Vaibhav Adlakha, Shehzaad Dhuliawala, Kaheer Suleman, Harm de Vries, and Siva Reddy. 2022. Topicoqa: Open-domain conversational question answering with topic switching. Transactions of the Association for Computational Linguistics, 10:468–483.

Raviteja Anantha, Svitlana Vakulenko, Zhucheng Tu, Shayne Longpre, Stephen Pulman, and Srinivas Chappidi. 2021. Open-domain question answering goes conversational via question rewriting. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL), pages 520–534.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL), pages 1870–1879.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentao Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 2174–2184.

Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2019. Cast 2019: The conversational assistance track overview. In Proceedings of the Twenty-Eighth Text REtrieval Conference, TREC, pages 13–15.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In In North American Chapter of the Association for Computational Linguistics (NAACL).

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

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

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

Hanjoo Kim, Minkyu Kim, Dongjoo Seo, Jinwoong Kim, Heungsok Park, Soeun Park, Hyunwoo Jo, KyungHyun Kim, Youngil Yang, Youngkwon Kim, et al. 2018. Nsml: Meet the mlaas platform with a real-world case study. arXiv preprint arXiv:1810.09957.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

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

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Ilia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: A benchmark for question answering research. Transactions of the Association for Computational Linguistics (TACL), 7:452–466.

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

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

Sheng-Chieh Lin, Jheng-Hong Yang, Rodrigo Nogueira, Ming-Feng Tsai, Chuan-Ju Wang, and Jimmy Lin. 2021c. Multi-stage conversational passage retrieval: An approach to fusing term importance estimation and neural query rewriting. ACM Transactions on Information Systems (TOIS), 39(4):1–29.
Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. In Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland. Association for Computational Linguistics.

Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W Bruce Croft, and Mohit Iyyer. 2020. Open-retrieval conversational question answering. In Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval (SIGIR), pages 539–548.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. OpenAI blog, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. (JMLR), 21(140):1–67.

Nako Sung, Minkyu Kim, Hyunwoo Jo, Youngil Yang, Jingwoong Kim, Leonard Lausen, Youngkwan Kim, Gayoung Lee, Donghyun Kwak, Jung-Woo Ha, et al. 2017. Nsml: A machine learning platform that enables you to focus on your models. arXiv preprint arXiv:1712.05902.

Svitlana Vakulenko, Johannes Kiesel, and Maik Fröbe. 2022. Scai-qrecc shared task on conversational question answering. arXiv preprint arXiv:2201.11094.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. Journal of machine learning research (JMLR), 9(11).

Ellen M. Voorhees and Dawn M. Tice. 2000. The TREC-8 question answering track. In Proceedings of the Second International Conference on Language Resources and Evaluation (LREC’00), Athens, Greece. European Language Resources Association (ELRA).

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP demo), pages 38–45.

Zeqiu Wu, Yi Luan, Hannah Rashkin, David Reitter, and Gaurav Singh Tomar. 2021. Conqr: Conversational query rewriting for retrieval with reinforcement learning. arXiv preprint arXiv:2112.08558.
### A Details of Question Types

We classify the no-switch and switch questions using dot product score between BM25 vectors of \( p_{t-1} \) and \( p_{t} \) 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).

### B Details of Dataset

| Dataset  | Train | Dev       | Test      | C       |
|----------|-------|-----------|-----------|---------|
| OR-QuAC  | # C   | 4,383     | 490       | 771     | 11M     |
|          | # Q   | 31,526    | 3,430     | 5,571   |         |
| QReCC    | # C   | 8,823     | 2,000     | 2,775   | 54M     |
|          | # Q   | 51,928    | 11,573    | 16,451  |         |

Table 2: Dataset statistics used in our experiments. The # C and # Q indicate the number of conversations and questions, respectively.

We mainly conduct experiments on recent CS benchmarks, OR-QuAC and QReCC (Qu et al., 2020; Anantha et al., 2021). We briefly describe the procedures of data construction and features of each dataset. Table 2 shows dataset statistics we used.

**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 QuAC. Moreover, they facilitate CQR as a subtask by reusing examples in CANARD (Elghory 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., 2021b; Yu et al., 2021). We use smaller collections \( C_{dev} \) (6.9k) provided by the authors for the development.

**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., 2019). 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. 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 also choose 7.3k number of corresponding dev passages for the development collections \( C_{dev} \). We only regard the examples that contain ground truth relevant passages. Thus, the actual number of training examples is 24,283.

### C Experimental Results on OR-QuAC

Table 3 shows results on OR-QuAC where most of the questions are no-switch type. First, we observe another retrieval shortcut on the first question, which is not observed in QReCC. Even if we input only first question to BM25, BM25\((q_1, C)\), it achieves competitive results with ALBERT baseline by Qu et al. (2020). We presume the lexi-

| Model                  | MRR | R@5 |
|------------------------|-----|-----|
| BM25\((q_1, C)\)       | 0.216 | 30.6 |
| BM25\((q_t, C)\)       | 0.043 | 5.6  |
| BM25\((q_{t-1}, C)\)   | 0.170 | 21.3 |
| BM25\((q_t, C)\)       | 0.198 | 24.9 |
| ALBERT (Qu et al., 2020) | 0.225 | 31.4 |
| CQE (Lin et al., 2021b) | 0.266 | 36.5 |
| ConvDR (Yu et al., 2021) | 0.616 | 75.0 |
| DPR                    | 0.525 | 63.9 |
| w. Model Negs          | **0.633** | **75.9** |

Table 3: 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 ⋄ indicates the CQE model performs zero-shot inference and dimensionality reduction (Lin et al., 2021b).
Data Training Indexing Inference

|          | OR-QuAC | QReCC |
|----------|---------|-------|
| Training | 2h      | 2h    |
| Indexing | 8h      | 28h   |
| Inference| 40m     | 11h   |

Table 4: 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.

cal cues from the first question are caused by pre-processing for the questions, rewriting to the standalone questions (Qu et al., 2020).

Our DPR with model-based hard negatives consistently outperforms the previous dense retrievers (Yu et al., 2020; Lin et al., 2021b). Even though it is not fair comparison since their different backbones and setups, we can compare the models in terms of hard negative mining strategies. Both CQE and ConvDR utilize CQR-based negatives requiring gold human rewrite $q'_t$ (CQR Negs). Similar to result in Table 1, our model with model-based negatives (Model Negs) achieves better performances without any usage of query rewriting.

## D Computational Cost

Overall computational cost is summarized in Table 4. Please note that the number of passages collection and test set of QReCC is much larger than the other. Thus, we allocate 8 GPUs for QReCC and 4 GPUs for OR-QuAC to perform training and indexing. We conduct training and inference once for all experiments because of the expensive computational cost.
E  Qualitative Examples

**Conversation (id: 1935_5)**

$q_1$: when was a view to a kill made?

$a_1$: a view to a kill is a 1985 spy film and the fourteenth in the James Bond series produced by Eon Productions.

$q_2$: who directed the film?

$a_2$: a view to kill was the third James Bond film and was directed by John Glen.

$q_3$: who played the main character?

$a_3$: Roger Moore played James Bond, MI6 agent 007, in a view to kill.

$q_4$: how old was the actor?

$a_4$: Roger Moore starred in his final Bond film, a View to a Kill, at the age of 58.

$q_5$: in how many films did the actor play the character?

$q_5$: In the Bond movie universe, [Moore](#) is the proud holder of three noteworthy records:

1. At age 45 in 1973, he was the oldest actor to take over the role.
2. At 57, he was again the oldest actor to play the role in 1985’s A View To a Kill, his seventh and final Bond film.
3. With seven appearances as 007, he starred in the most official Bond films.

---

**DPR without Negatives**

In the Bond movie universe, **Moore** is the proud holder of three noteworthy records:

1. At age 45 in 1973, he was the oldest actor to take over the role.
2. At 57, he was again the oldest actor to play the role in 1985’s A View To a Kill, his seventh and final Bond film.
3. With seven appearances as 007, he starred in the most official Bond films.

---

**DPR with Negatives**

**---** British actor Sir [Roger Moore](#) KBE Moore in 1973 Born Roger George Moore (1927-10-14)

14 October 1927 Stockwell, London, England Died 23 May 2017 (2017-05-23) (aged 89) Crans-Montana, Switzerland [1] Burial place Monaco Cemetery Alma mater Royal Academy of Dramatic Art Occupation Actor Years active 1945–2013 Known for [James Bond in seven feature films](#) from 1973 to 1985

---

Table 5: An example of top-1 predictions from vanilla DPR (without Negatives) and DPR trained with model-based hard negatives (with Negatives). The vanilla DPR without hard negatives fails to predict a gold passage since it heavily relies on shortcut, i.e., previous answer $a_4$. On the other hand, the DPR successfully predicts a gold passage with comprehending whole conversational context up to $q_5$, when the retriever is trained with hard negatives.