**Table 1: Summary statistics of QReCC and OR-QuAC**

| QReCC / OR-QuAC | Train | Dev | Test |
|---|---|---|---|
| # dialogues | 8.7k / 4.4k | 2.2k / 0.5k | 2.8k / 0.7k |
| # questions (Qs) | 50.8k / 31.5k | 12.7k / 3.4k | 16.4k / 5.6k |
| avg Qs in dialogue | 6.0 / 7.2 | 6.0 / 7.0 | 6.0 / 7.2 |

*Question Rewriting Models.* We implement GQR [22] and EQR [24] using Huggingface[1]. Since the QR models described in Section 3 are not publicly available, we implemented and finetuned both models with the training and validation set of QReCC and OR-QuAC, respectively. Hyperparameters are determined by grid search through epochs {2, 3, 4}, learning rate {1e-3, 1e-4, 1e-5}, and warm-up steps {600, 800, 1000}. All other hyperparameters follow the original implementation in their respective papers. We generate the rewrite of a question using all the previous questions and answers in the dialogues following the setup introduced by Anantha et al. [1]. We have performed an intrinsic evaluation of the rewrites produced by our models, and obtained ROUGE-1R of 0.85 and 0.84 for GQR and EQR, respectively. These results are in line with those reported by Vakulenko et al. [23], showing that the quality of the rewrites is comparable to the ones produced by the state-of-the-arts.

*QA systems.* For the experiments in Section 4.2, we use BERTserini [25] with default parameters. BERTserini is an end-to-end open-domain QA system that leverages BERT architecture for reading, and implements the retrieval with the open-source Pyserini toolkit [11]. The retriever is implemented with a sparse model, BM25, which is by far the most common setting in ODCQA [23, 25]. For the experiments in Section 4.3, we use the state-of-the-art Transformer-based ORConvQA model [14].[2] In this case, top K relevant passages are retrieved by the retriever, and fed first to the re-ranker then to the reader to extract the final answer span. In all its components, the dialogue history is modeled by concatenating previous questions to the current question.

*Evaluation.* We evaluate the end-to-end QA system performance using the two metrics provided by the QuAC challenge [2]: the word-level F1 and the human equivalence score (HEQ). F1 is a primary QA performance metric. It measures the word-overlap between the predicted and the ground-truth answer span. HEQ measures if the system can output good answers as an average human. It computes the percentage of questions for which F1 exceeds or matches human F1. The metric can be computed at the question level (HEQ-Q) and the dialogue level (HEQ-D). Additionally, we use recall to evaluate the performance of the retriever, which is defined as the fraction of relevant passages that are retrieved by the system [14].

## 4.2 Combining QR Approaches

In this section, we assess how the combination of different QR approaches for retriever and reader affects the end-to-end results in ODCQA. All the retrieval metrics are computed for the top 10 passages. Table 2 summarizes the end-to-end QA performance (F1), together with the intermediate retriever results (recall), on QReCC and OR-QuAC. Besides the results obtained with automatic rewrites (E and G), we also report the results obtained with the manual rewrites included in the dataset (M), which set the upper-bound for the task. Our results confirm that both QR approaches, GQR and EQR, in general do help improve the end-to-end performance of ODCQA. As we focus on the effect of QR on the two components of the system, we observe that QR helps improve the performance of the reader, as shown by the improvement of OE and OG compared to OO. However, it is at the stage of the retriever that we observe the main effect of QR, as shown by the large improvement in F1 of EX and GX compared to OX (where X ∈ {O, E, G, M} is the QR strategy used at reader). The importance of QR for the retriever is also confirmed by the recall values reported in Table 2, showing that correct answers are retrieved much more often when using any rewrite compared to when using the original question.

**Table 2: Results of different QR setups with sparse retriever.**

| Input Question | | QReCC | | OR-QuAC | |
|---|---|---|---|---|---|
| Retriever | Reader | F1 | Recall | F1 | Recall |
| O | O | 6.15 (3) | 5.56 | 7.01 (4) | 7.57 |
| O | E | 5.93 (4) | | 7.61 (3) | |
| O | G | 6.69 (2) | | 8.42 (2) | |
| O | M | 6.95 (1) | | 10.32 (1) | |
| E | O | 11.82 (3) | 25.34 | 11.62 (3) | 24.12 |
| E | E | 10.37 (4) | | 11.21 (4) | |
| E | G | 11.87 (2) | | 11.95 (2) | |
| E | M | 12.21 (1) | | 13.82 (1) | |
| G | O | 11.75 (2) | 23.63 | 10.92 (2) | 20.39 |
| G | E | 10.28 (4) | | 10.51 (4) | |
| G | G | 11.43 (3) | | 11.12 (1) | |
| G | M | 12.00 (1) | | 10.76 (3) | |
| M | O | 12.97 (1) | 29.07 | 13.56 (2) | 27.84 |
| M | E | 11.54 (4) | | 13.17 (3.5) | |
| M | G | 12.78 (3) | | 13.17 (3.5) | |
| M | M | 12.93 (2) | | 14.92 (1) | |

*Notes*: O – original question without rewrite, E – EQR rewritten question, G – GQR rewritten question, M – manual rewrite. Results are grouped under QR strategies used in retriever. The number in parentheses is the rank of the result within the group. Recall measures the intermediate retriever results.

The best QR setup (excluding the use of manual rewrites, which is unavailable in applications) is to apply EQR at retriever and GQR at reader. In terms of F1, EG achieves the best performance on both datasets. Moreover, among all 16 paired comparisons of EX vs. GX or OX across datasets, EX outperforms all 16 times; Among all 16 paired comparisons of XG vs. XE or XO, XG outperforms 12 times. According to two-tailed sign tests [4], it suggests EQR and GQR are significantly better at retriever and reader with $p < 0.05$, respectively. Such conclusion is in line with [23] and it is expected, considering (i) EQR tends to append more keywords to the original question, compared to those generated by GQR (the average number of tokens in the questions returned by EQR and GQR is 11.3 and

[1] https://huggingface.co/
[2] https://github.com/prdwb/orconvqa-release

**Table 3: History embedding vs. QR on OR-QuAC with the ORConvQA model.**

| Retriever | Reader | History | HEQ-Q | HEQ-D | F1 |
|---|---|---|---|---|---|
| O | O | 6 | 24.33 | 0.65 | 29.59 |
| O | O | 0 | 11.82 | 0 | 17.89 |
| O | E | 0 | 14.99 | 0 | 20.09 |
| O | G | 0 | 14.55 | 0 | 19.87 |
| E | O | 0 | 15.70 | 0.13 | 22.26 |
| E | E | 0 | 15.98 | 0.13 | 22.03 |
| E | G | 0 | 16.40 | 0.13 | 22.25 |
| G | O | 0 | 16.98 | 0.39 | 23.25 |
| G | E | 0 | 17.28 | 0.26 | 23.40 |
| G | G | 0 | 16.98 | 0.13 | 23.37 |

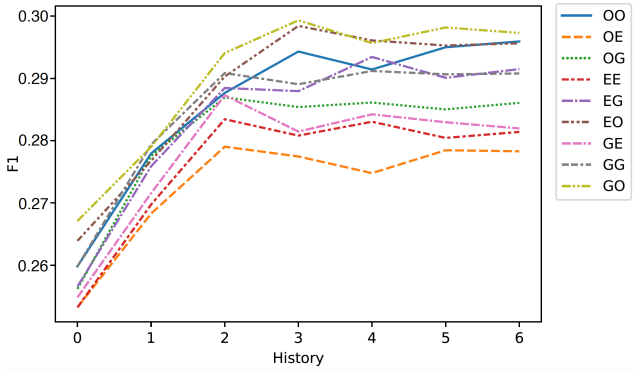


**Figure 1: Results of different QR setups with dense retriever on OR-QuAC, including the impact when embedding history of varied window sizes.**

9.08, respectively); (ii) the sparse retriever such as BM25 and TF-IDF is less affected by the non-fluent rewrites produced by EQR; (iii) the rewrites of GQR are fluent, and QA models are trained with natural questions. In contrast, XE is almost always the worst strategy, as EQR negatively affects the reader's performance.

Some additional observations are also worth mentioning. First, while MX slightly outperforms both EX and GX, the performance of the automatic rewrites are close to those obtained with the manual ones. This result attests the quality of the rewrites produced by GQR and EQR. Second, when the retriever performance improves, the effectiveness of QR at reader diminishes, as shown by similar performances between, e.g., EG and EO, GG and GO: At the reader stage, the more relevant passages are provided as input, the less it needs QR. This is a relevant finding from a practical point of view, since it provides an opportunity to trade off marginal performance gain against efficiency.

## 4.3 QR vs. History Embedding

We address our second research question, *how does history embedding compare to various QR models for improving ODCQA performance*, with the OR-QuAC dataset and the state-of-the-art ORConvQA model by Qu et al. [14]. In Table 3, we evaluate the ORConvQA model using different rewrites at retriever and reader. The setup OO with history window $w = 6$ represents the state-of-the-art performance of ORConvQA. Unlike [14], since we are interested in comparing the models with and without conversational history, when $w = 0$ we do not include the first dialogue question in the retrieval. We find that: (i) In line with [14], OO without history ($w = 0$) has poor performance; (ii) Using only QR to model the conversational history always improves upon OO, but it does not perform as well as history embedding that directly embeds previous turns in system components; (iii) When used for a dense retriever, GQR is better than EQR; (iv) Similar to the results of sparse retrieval in Table 2, the overall performance boost primarily comes from the retrieval.

Finally, we investigate whether QR can complement history embedding. Figure 1 shows results when using different rewrite combinations with ORConvQA. In this experiment we follow the setup in [14], where the first question of the dialogue is always included in retrieval, even for $w = 0$. We note that: (i) The use of QR (e.g., GO) enables a reduction of the history window size by half in the reader, without impacting the performance. This is a useful result as it makes possible to reduce the max length of input questions to the reader, and potentially enables the use of smaller models; (ii) However, with larger window sizes, e.g., $w = 6$, the effect of rewrites is mitigated and performance converges to the ones using original questions. These observations suggest that QR alone has a subpar performance comparing to state-of-the-arts that embed previous turns into the system components [14–16], but to obtain the best end-to-end performance, applying QR and history embedding jointly is the best bet.

## 5 CONCLUSION

In this paper, we present an empirical study on question rewriting techniques for open-domain conversational QA. Our investigation has two goals: (i) Assessing how the combination of different QR strategies affects the performance of an ODCQA system based on the retriever-reader architecture; (ii) Comparing the effectiveness of QR to the one of history modeling with dense representations.

For (i), with a standard retriever-reader architecture, where the retrieval is done via a sparse vector space model, our results suggest the existing practice to apply the same QR approach for both retriever and reader is suboptimal, and recommend the expansive QR model for the retriever and the generative QR model for the reader. Furthermore, we have identified that the primary contribution of QR is at the retriever stage; its impact to the reader, while still positive, is considerably smaller. It also reflects the fact that a relatively simple model such as BM25 may benefit more from question rewriting compared to the neural network used for the reader. For (ii), while modeling conversation history with dense representations does lead to better performance compared to applying QR only, our results suggest that using QR and history embedding jointly is still beneficial, especially when few previous history turns are considered.

## REFERENCES


[1] Raviteja Anantha, Svitlana Vakulenko, Zhucheng Tu, Shayne Longpre, Stephen Pulman, and Srinivas Chappidi. 2021. Open-Domain Question Answering Goes Conversational via Question Rewriting. *CoRR* arXiv:2010.04898 (2021).

[2] Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question Answering in Context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. ACL, Brussels, Belgium, 2174–2184.

[3] Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2019. TREC CAsT 2019: The Conversational Assistance Track Overview. In *Proceedings of the Twenty-Eighth Text Retrieval Conference*. NIST, Gaithersburg, MD, USA, 13–15.

[4] Janez Demšar. 2006. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research* 7 (2006), 1–30.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. ACL, Minneapolis, Minnesota, 4171–4186.

[6] 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*. ACL, Hong Kong, China, 5918–5924.

[7] Siddhant Garg, Thuy Vu, and Alessandro Moschitti. 2020. TANDA: Transfer and adapt pre-trained transformer models for answer sentence selection. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*. AAAI, New York, NY, USA, 7780–7788.

[8] 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*. ACL, 6769–6781.

[9] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: a Benchmark for Question Answering Research. *Transactions of the Association of Computational Linguistics* 7 (2019), 453–466.

[10] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. ACL, Florence, Italy, 6086–6096.

[11] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Virtual Event, Canada, 2356–2362.

[12] Sheng-Chieh Lin, Jheng-Hong Yang, Rodrigo Nogueira, Ming-Feng Tsai, Chuan-Ju Wang, and Jimmy Lin. 2021. Multi-Stage Conversational Passage Retrieval: An Approach to Fusing Term Importance Estimation and Neural Query Rewriting. *CoRR* arXiv:2005.02230 (2021).

[13] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* arXiv:1907.11692 (2019).

[14] Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W.B̃ruce 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*. ACM, Xi'an, China, 539–548.

[15] Chen Qu, Liu Yang, Minghui Qiu, W. Bruce Croft, Yongfeng Zhang, and Mohit Iyyer. 2019. BERT with History Answer Embedding for Conversational Question Answering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Paris, France, 1133–1136.

[16] Chen Qu, Liu Yang, Minghui Qiu, Yongfeng Zhang, Cen Chen, W Bruce Croft, and Mohit Iyyer. 2019. Attentive history selection for conversational question answering. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. ACM, Beijing, China, 1391–1400.

[17] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* 21, 140 (2020), 1–67.

[18] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. ACL, Austin, Texas, US, 2383–2392.

[19] Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A Conversational Question Answering Challenge. *Transactions of the Association for Computational Linguistics* 7 (2019), 249–266.

[20] Alon Talmor and Jonathan Berant. 2018. The Web as a Knowledge-Base for Answering Complex Questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. ACL, New Orleans, Louisiana, US, 641–651.

[21] Svitlana Vakulenko, Shayne Longpre, Zhucheng Tu, and Raviteja Anantha. 2020. A Wrong Answer or a Wrong Question? An Intricate Relationship between Question Reformulation and Answer Selection in Conversational Question Answering. In *Workshop on Search-Oriented Conversational AI*.

[22] Svitlana Vakulenko, Shayne Longpre, Zhucheng Tu, and Raviteja Anantha. 2021. Question Rewriting for Conversational Question Answering. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. ACM, Jerusalem, Israel, 355–363.

[23] Svitlana Vakulenko, Nikos Voskarides, Zhucheng Tu, and Shayne Longpre. 2021. A Comparison of Question Rewriting Methods for Conversational Passage Retrieval. *CoRR* arXiv:2101.07382 (2021). ECIR short paper.

[24] 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*. ACM, Xi'an, China, 921–930.

[25] Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-End Open-Domain Question Answering with BERTserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*. ACL, Minneapolis, Minnesota, 72–77.

[26] Shi Yu, Jiahua Liu, Jingqin Yang, Chenyan Xiong, Paul Bennett, Jianfeng Gao, and Zhiyuan Liu. 2020. Few-Shot Generative Conversational Query Rewriting. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Xi'an, China, 1933–1936.