A Comparison of Question Rewriting Methods for Conversational Passage Retrieval

Svitlana Vakulenko[1(✉)], Nikos Voskarides[1], Zhucheng Tu[2],
and Shayne Longpre[2]

[1] University of Amsterdam, Amsterdam, The Netherlands
{s.vakulenko,n.voskarides}@uva.nl
[2] Apple Inc., Cupertino, USA

Abstract. Conversational passage retrieval relies on question rewriting to modify the original question so that it no longer depends on the conversation history. Several methods for question rewriting have recently been proposed, but they were compared under different retrieval pipelines. We bridge this gap by thoroughly evaluating those question rewriting methods on the TREC CAsT 2019 and 2020 datasets under the same retrieval pipeline. We analyze the effect of different types of question rewriting methods on retrieval performance and show that by combining question rewriting methods of different types we can achieve state-of-the-art performance on both datasets (Resources can be found at https://github.com/svakulenk0/cast_evaluation.)

1 Introduction

Conversational search aims to provide automated support for natural and effective human–information interaction [1]. The TREC Conversational Assistance Track (CAsT) introduced the task of conversational (multi-turn) passage retrieval (PR) [3], where the goal is to retrieve short passages of text from a large passage collection that answer the information need at the current turn.

One prominent challenge in conversational PR is that the question at the current turn often requires information from the conversation history (questions and passages retrieved in previous turns) to be interpreted correctly. A proposed solution to this challenge is question rewriting (or resolution, QR), i.e., modifying the question such that it no longer depends on the conversation history. For instance, the question "What did he work on?" can be rewritten into "What did Bruce Croft work on?" based on the conversation history (see Table 4 for the complete example).

Recently proposed methods for QR in conversational PR can be categorized into two types, namely sequence generation and term classification. Sequence generation QR methods generate natural language sequences using the conversation history [7,9], while term classification QR methods add terms from the conversation history to the current turn question [5,8]. The former can be trained

using human generated rewrites or data obtained from search sessions and heuristics [7,9], while the latter are either heuristic-based [5], or trained using human generated rewrites or distant supervision [8].

In this paper, we conduct a systematic evaluation of the state-of-the-art QR methods under the same retrieval pipeline on the CAsT 2019 and 2020 datasets. While CAsT 2019 only depends on the previous questions in the conversation, CAsT 2020 also includes questions that depend on the previously retrieved passages. Our results provide insights on the ability of the QR methods to account for the conversation history, as well as on the potential of combining QR methods of different types for improving retrieval effectiveness.

2   Task Definition

We model the conversational PR task as a sequence of two subtasks: (1) question rewriting (QR) and (2) passage retrieval (PR) [7–9]. In this paper, we focus on the QR subtask and investigate the impact of QR on PR performance.

In the QR subtask, we are given the current turn question Q_i and a sequence of question-answer pairs $H := [Q_1, A_1, \ldots, Q_{i-1}, A_{i-1}]$ (the conversation history). The current turn question Q_i may depend on the conversation history H and thus some information in H is required to correctly interpret Q_i. The goal of QR is to generate a question rewrite Q_i' that no longer depends on H.

In the PR subtask, we are given the question rewrite Q_i' and a passage collection C, and the goal is to retrieve a list of passages R sorted by their relevance to Q_i' from C. If Q_i' is semantically equivalent to $\langle Q_i, H \rangle$, we expect R to constitute relevant passages for $\langle Q_i, H \rangle$.

3   Experimental Setup

We aim to answer the following research questions:

RQ1. How do different QR methods perform on the two datasets we consider (CAsT 2019 and CAsT 2020)?

RQ2. Can we combine different QR models to improve retrieval performance?

Following previous work, we perform both intrinsic and extrinsic evaluation [2,8]. In intrinsic evaluation, we compare rewrites produced by QR methods with manual rewrites produced by human annotators using ROUGE-1 Precision (P), Recall (R) and F-measure (F) [2].[1] In extrinsic evaluation, we measure PR performance when using different QR methods using standard ranking metrics: NDCG@3, MRR and Recall@1000.

---

[1] We use ROUGE-1 to measure unigram overlap after punctuation removal, lower casing and Porter stemming. We use the following ROUGE implementation: https://github.com/google-research/google-research/tree/master/rouge.

3.1   Question Rewriting Methods

We compare the following question rewriting methods:

- **Original** The original current turn question without any modification.
- **Human** The gold standard rewrite of the current turn question produced by a human annotator.
- **Rule-Based** and **Self-Learn** model question rewriting as a sequence generation task and use GPT-2 to perform generation [9]. In order to gather training data, these methods convert ad-hoc search sessions to conversational search sessions either by using heuristic rules (**Rule-Based**) or by using self-supervised learning (**Self-Learn**).
- **Transformer++** [7] is a GPT-2 sequence generation model. It was trained on CANARD, a conversational question rewriting dataset [4].
- **QuReTeC** [8] models question rewriting as term classification, i.e., predicting which terms from the conversation history to add to the current turn question. It uses BERT to perform term classification and can be trained using human rewrites or distant supervision obtained from query-passage relevance labels. In this paper, we use the model trained on CANARD [4] to be comparable with **Transformer++**. Since **QuReTeC** does not generate natural language text but rather appends a bag-of-words (BoW) to the original question, we also introduce an oracle **Human-BoW** as an upper-bound for **QuReTeC** performance.

Table 1. Datasets statistics.

Dataset	#Topics	#Questions	#Copy	(%)
CAsT 2019	50	479	88	(21)
CAsT 2020	25	216	5	(3)

3.2   Datasets

We use the recently constructed TREC CAsT 2019 and CAsT 2020 datasets [3]. Table 1 shows basic statistics of the datasets. **Copy** indicates the number of questions for which the human rewrite is exactly the same as their corresponding original question. This statistic shows that in contrast to CAsT 2019, in CAsT 2020, only a very few questions can be copied verbatim and the majority of questions require extra terms.

Another major difference between the two datasets is that the current turn question in CAsT 2020 may also depend on the answer passage to the previous turn question (A_{i-1}), while in CAsT 2019 the current turn question depends only on the questions of the previous turns in the conversation history ($Q_1, Q_2, \ldots, Q_{i-1}$). Therefore, we experiment with two variations of input

to the QR models: (1) all previous questions (indicated as **Q**) and (2) all previous questions and the answer passage to the previous turn question (indicated as **Q&A**).[2]

Table 2. Evaluation of question rewriting methods on CAsT 2019.

QR Method	Recall@1000	NDCG@3		ROUGE-1		
	Initial	Initial	Reranked	P	R	F
Original	0.417	0.131	0.266	0.92	0.76	0.82
Transformer++ Q	0.743	0.265	**0.525**	**0.96**	0.88	**0.91**
Self-Learn Q	0.725	0.261	0.513	0.93	0.89	0.90
Rule-Based Q	0.717	0.248	0.487	0.94	0.89	0.91
QuReTeC Q	**0.768**	**0.296**	0.500	0.89	**0.90**	0.89
Transformer++ Q + QuReTeC Q	**0.791**	0.300	**0.546**	**0.93**	0.91	**0.91**
Self-Learn Q + QuReTeC Q	0.785	0.293	0.519	0.90	**0.93**	**0.91**
Rule-Based Q + QuReTeC Q	0.783	**0.301**	0.534	0.91	**0.93**	**0.91**
Human-BoW Q	0.769	0.297	0.524	0.91	0.90	0.90
Human	0.803	0.309	0.577	1.00	1.00	1.00

3.3 Passage Retrieval Pipeline

All QR methods described in Sect. 3.1 were previously evaluated on CAsT 2019 using different retrieval pipelines. For a fair comparison, we evaluate the QR methods on both CAsT 2019 and CAsT 2020 using the same passage retrieval pipeline.

We use a standard two-stage pipeline for passage retrieval, consisting of an unsupervised ranker for initial retrieval performing efficient lexical match (BM25) and a supervised reranker (BERT) over the top-1000 passages returned by initial retrieval [6].[3] Both components were fine-tuned on a subset of the MS MARCO dataset ($k_1 = 0.82, b = 0.68$).[4]

4 Results

4.1 QR Methods Comparison

Here we answer **RQ1**: How do different QR methods perform on the two datasets we consider?

---

[2] We use the answer passage to the previous turn question retrieved by the *automatic* rewriting system provided by the TREC CAsT 2020 organizers.

[3] Note that our pipeline outperforms the official baseline provided by the TREC CAsT organizers for both 2019 and 2020 datasets for all query rewriting methods they considered. Since our focus is on comparing different query rewriting methods, we do not report those results for brevity.

[4] https://github.com/nyu-dl/dl4marco-bert.

CAsT 2019. In Table 2, we observe that QuReTeC outperforms all other methods in initial retrieval (Recall@1000 and NDCG@3). However, we see that Transformer++ Q outperforms QuReTeC in reranking (NDCG@3). This may indicate that the reranking component (BERT) is more sensitive to rewritten questions that do not resemble natural language text (produced by QuReTeC) than the initial retrieval component (BM25). This is also reflected in the ROUGE-1 metric variations: ROUGE-1 R is generally in agreement with initial retrieval performance. This is expected since our initial retrieval component is BoW and does not get substantially affected by missing or incorrect terms such as pronouns and stopwords, which are usually insignificant for lexical matching (see Human-BoW in Table 2). ROUGE-1 P, however, favours the sequence generation methods, and penalizes QuReTeC, since QuReTeC does not have a mechanism to delete or replace such terms from the original question.

Table 3. Evaluation of question rewriting methods on CAsT 2020.

QR Method	Recall@1000	NDCG@3		ROUGE-1		
	Initial	Initial	Reranked	P	R	F
Original	0.251	0.068	0.193	**0.87**	0.66	0.74
Transformer++ Q& A	0.351	0.098	0.252	0.75	0.69	0.70
Self-Learn Q& A	0.462	0.156	0.342	0.84	0.73	0.76
Rule-Based Q& A	0.455	0.137	0.339	0.84	0.75	**0.78**
QuReTeC Q& A	**0.531**	**0.171**	**0.370**	0.82	**0.77**	**0.78**
Transformer++ Q + QuReTeC Q& A	0.525	0.160	0.351	**0.83**	0.77	0.78
Self-Learn Q + QuReTeC Q& A	**0.567**	0.168	**0.375**	0.82	**0.79**	**0.79**
Rule-Based Q& A + QuReTeC Q& A	0.519	**0.173**	0.362	0.80	**0.79**	0.78
Human-BoW Q	0.579	0.189	0.465	0.89	0.81	0.84
Human-BoW Q& A	0.649	0.226	0.465	0.88	0.85	0.86
Human	0.707	0.240	0.531	1.00	1.00	1.00

CAsT 2020. In Table 3, we observe that the retrieval performance of Original and Human is much lower than in Table 2, which indicates that CAsT 2020 is more challenging than CAsT 2019.[5] We observe that QuReTeC outperforms all other methods in all ranking metrics. This indicates that QuReTeC better captures relevant terms both from the previous turn questions and the answer passage to the previous turn question than the other QR methods. Similarly to Table 2, ROUGE-1 R is in agreement with initial retrieval performance. As for ROUGE-1 P, we observe that it is not as important for retrieval as in Table 2. Next, we assess the contribution of the answer passage to the previous turn question on QR performance. In Fig. 1, we observe that most QR methods (except Transformer++) do benefit from using the answer passage, with QuReTeC having the biggest gain in initial retrieval. Table 4 shows examples of question rewrites produced by Rule-Based and QuReTeC.

---

[5] Recall that questions in CAsT 2020 may depend on the answer of the previous turn question, but this is not the case in CAsT 2019.
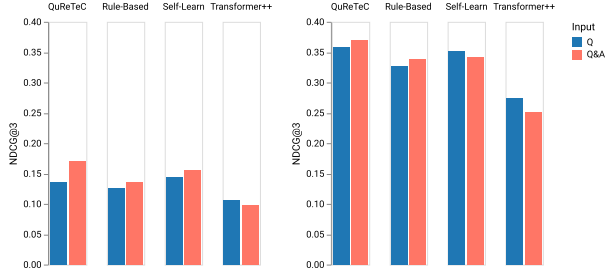
Fig. 1. Initial retrieval (left) and reranking (right) performance on CAsT 2020 when the answer passage to the previous turn question is used (Q&A) or not used (Q) as input to the QR methods.

Table 4. Example question rewrites for the topic in CAsT 2020 starting with "Who are some of the well-known Information Retrieval researchers?".

Answer passage	Original	Rule-Based Q& A	QuReTeC Q& A
Bruce Croft formed the Center	What did he work on?	What did Bruce Croft work on?	What did he work on? croft bruce
Karpicke and Janell R. Blunt (2011) followed up	Who are some important British ones?	Who are some important British ones?	Who are some important British ones? information retrieval

4.2 Combining QR Methods

Next we answer **RQ2**: Can we combine different QR models to improve performance? In order to explore whether combining QR methods of different types (sequence generation or term classification) can be beneficial, we simply append terms from the conversation history predicted as relevant by QuReTeC to the rewrite produced by one of the sequence generation methods. We found that by doing this we can improve upon individual QR methods and achieve state-of-the-art retrieval performance on CAsT 2019 by combining Transformer++ Q with QuReTeC Q (see Table 2), and on CAsT 2020 by combining Self-Learn Q and QuReTeC Q&A (see Table 3); however the gains on CAsT 2020 are smaller.

5 Conclusion

We evaluated alternative question rewriting methods for conversational passage retrieval on the CAsT 2019 and CAsT 2020 datasets. On CAsT 2019, we found that QuReTeC performs best in terms of initial retrieval, while Transformer++ performs best in terms of reranking. On CAsT 2020, we found that QuReTeC

performs best both in terms of initial retrieval and reranking. Moreover, we achieved state-of-the-art ranking performance on both datasets using a simple method that combines the output of QuReTeC (a term classification method) with the output of a sequence generation method. Future work should focus on developing more advanced methods for combining term classification and sequence generation question rewriting methods.

References

1. Anand, A., Cavedon, L., Joho, H., Sanderson, M., Stein, B.: Conversational search (dagstuhl seminar 19461). Dagstuhl Reports (2019)
2. Anantha, R., Vakulenko, S., Tu, Z., Longpre, S., Pulman, S., Chappidi, S.: Open-domain question answering goes conversational via question rewriting. arXiv preprint arXiv:2010.04898 (2020)
3. Dalton, J., Xiong, C., Callan, J.: Cast 2019: The conversational assistance track overview. In: TREC (2019)
4. Elgohary, A., Peskov, D., Boyd-Graber, J.: Can you unpack that? Learning to rewrite questions-in-context. In: EMNLP-IJCNLP (2019)
5. Mele, I., Muntean, C.I., Nardini, F.M., Perego, R., Tonellotto, N., Frieder, O.: Topic propagation in conversational search. In: SIGIR (2020)
6. Nogueira, R., Cho, K.: Passage re-ranking with BERT. arXiv preprint arXiv:1901.04085 (2019)
7. Vakulenko, S., Longpre, S., Tu, Z., Anantha, R.: Question rewriting for conversational question answering. In: WSDM (2021)
8. Voskarides, N., Li, D., Ren, P., Kanoulas, E., de Rijke, M.: Query resolution for conversational search with limited supervision. In: SIGIR (2020)
9. Yu, S., et al.: Few-shot generative conversational query rewriting. In: SIGIR (2020)