Question Rewriting for Conversational Question Answering

Svitlana Vakulenko*
University of Amsterdam
s.vakulenko@uva.nl

Shayne Longpre
Apple Inc.
slongpre@apple.com

Zhucheng Tu
Apple Inc.
zhucheng_tu@apple.com

Raviteja Anantha
Apple Inc.
raviteja_anantha@apple.com

ABSTRACT

Conversational question answering (QA) requires the ability to correctly interpret a question in the context of previous conversation turns. We address the conversational QA task by decomposing it into question rewriting and question answering subtasks, and conduct a systematic evaluation of this approach on two publicly available datasets (QuAC and TREC CAsT). Question rewriting is designed to reformulate ambiguous questions, which depend on the conversational context, into unambiguous questions that can be correctly interpreted outside of the context of a conversation. Thereby, standard QA components can answer such explicit questions without the need to modify their architecture, which allows to leverage the pre-trained models and datasets available for the standard QA task. Another practical benefit of our approach is that the rewritten questions allow to use 3rd-party QA services without the need to share the conversation history with them. To the best of our knowledge, we are the first to evaluate question rewriting on the conversational question answering task and show its improvement over the end-to-end baselines that have direct access to the conversational context. Furthermore, a large part of our study is devoted to examining an intricate relation between question formulation and question answering performance. We demonstrate the use of an evaluation framework that takes advantage of the intermediate output produced by the question rewriting and allows us to automatically distinguish errors in question formulation from errors in question answering. The results of our analysis reveal sensitivity of all our QA models to the variance in question formulation, as well as the flaws in the QA evaluation setup that does not correct for the dataset bias, which allows models to guess correct answers even when given incorrect questions.

CCS CONCEPTS

• **Information systems** → **Question answering**; **Query reformulation**; **Information retrieval**; **Information extraction**.

*This research was completed during the internship at Apple Inc.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

KEYWORDS

conversational search, question answering, question rewriting

ACM Reference Format:
Svitlana Vakulenko, Shayne Longpre, Zhucheng Tu, and Raviteja Anantha. 2020. Question Rewriting for Conversational Question Answering. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Extending question answering systems to a conversational setting is a natural development that allows smooth interactions with a digital assistant [9]. This transition requires taking care of complex linguistic phenomena characteristic of a human dialogue, such as anaphoric expressions and ellipsis [27]. Such contextual dependencies refer back to the previous conversational turns, which are necessary to correctly interpret the question.

In this paper, we propose to address the task of conversational QA by decomposing it into two sub-tasks: (1) a *question rewriting* (QR) model that given previous conversational turns produces an explicit (contextually independent) question, which can then be used as an input to (2) a standard *question answering* (QA) model that can process explicit questions outside of the conversation context. Thereby, an explicit question serves as an intermediate output that connects QR with QA components. This setup offers a wide range of advantages in comparison to training a single conversational QA model end-to-end:

(1) **Traceability**: Due to the intermediate output produced by the QR model, errors in QA can be traced back to the individual components. Our error analysis makes full use of this feature by investigating different sources of error and their correlation.
(2) **Reuse**: Since QR produces an explicit question that can be answered by non-conversational QA models, we can leverage existing QA models and datasets without the need to re-design architectures to handle conversational context. Additionally, a QR model can be pre-trained once, and reused with a variety of standard QA models.
(3) **Modularity**: QR setup makes it possible to query multiple remote 3rd-party APIs suitable for QA over different collections. This is a realistic scenario, in which information is distributed across a network of heterogenous nodes that do not share internal representations. Natural language provides a suitable communication protocol between these components [23].

Two dimensions on which QA tasks vary are: the type of data source used to retrieve the answer (e.g., a paragraph, a document collection, or a knowledge graph); and the expected answer type (a text span, a ranked list of passages, or an entity). In this paper, we experiment with two variants of the QA task: *retrieval QA*, the task of finding an answer to a given natural-language question as a ranked list of relevant passages given a document collection; and *extractive QA*, the task of finding an answer to a given natural-language question as a text span within a given passage. Though the two QA tasks are complementary to each other, in this paper we focus on the QR task and its ability to enable different types of QA models within a conversational setting. We experiment with both retrieval and extractive QA models to examine the effect of the QR component on the end-to-end QA performance.

We demonstrate how the proposed composite architecture allows us to trace errors back to the indivudal components. This allows us to pinpoint the exact cases where an incorrect answer is either due to the incorrect context interpretation or the incorrect question-answer matching. Our results indicate that the majority of error cases are due to the errors in QA rather than QR, and QR performs on par with humans on this task. Thereby, we show that question rewriting is, indeed, a very effective method for extending standard question answering (QA) approaches to the conversational QA setting, and the future work in this direction should focus primarily on improving QA performance.

Moreover, we confirmed that the QR metric provides a good indicator for the end-to-end QA performance. Our analysis results show that the QR performance metric is able to correctly predict the model that performs consistently better across both QA tasks. Thereby, this metric can be used for QR model selection to avoid more costly end-to-end evaluation in the future. We also demonstrate how correlation in the performance metrics and question rewrites can be used as a tool for diagnosing QA models. QR helps to reveal sensitivity of the QA models to the question formulation, which should be exploited for training more robust QA models in the future. The contributions of this work are two-fold:

(1) decomposing the task of conversational QA into question rewriting (QR) and question answering (QA) for both retrieval and extractive settings, and
(2) a systematic end-to-end evaluation of the proposed approach, along with the detailed analysis of the interactions between the QR and QA components.

2 RELATED WORK

Conversational QA is an extension of the standard QA task that introduces contextual dependencies between the input question and the previous dialogue turns. Several datasets were recently proposed extending different QA tasks to a conversational setting including extractive [2, 24], retrieval [4] and knowledge graph QA [3, 11]. One common approach to conversational QA is to extend the input of a QA model by appending previous conversation turns [3, 12, 20]. Such approach, however falls short in case of retrieval QA, which requires a concise query as input to the candidate selection step, such as BM25 [19]. Results of the recent TREC CAsT track demonstrated that co-reference models are also not sufficient to resolve the missing context in the follow-up questions [4]. A considerable gap between the performance of automated rewriting approaches and manual human annotations call for new architectures that are capable of retrieving relevant answers from large text collection using conversational context.

In this paper we explore an alternative to training an end-to-end conversational QA model by separating it into QR and QA subtasks. We present the results of our experiments with the state-of-the-art text generation approach applied to resolve missing conversation context in the follow-up questions. QR is designed to handle conversation context and produce an equivalent question that no longer depends on the conversation context, i.e., with anaphoras and other contextual ambiguities resolved. The output of QR model can then be used by a standard QA model pretrained on non-conversational datasets, such as SQuAD [22], WikiQA [31] or TrecQA [28].

QR has been already shown effective for multi-turn chitchat and task-oriented dialogues [23, 26]. These settings are dissimilar to ours since they use different evaluation criteria. Instead of measuring slot-matching and intent detection performance as in task-oriented dialogue systems or user engagement as in chit-chat, we evaluate open-domain QA. As we show in our experimental evaluation, this directly correlates with the QR performance. In this paper, we show that QR is well suited for open-domain conversational QA and demonstrate that QR helps to achieve superior performance on this task. The setup we propose also provides a convenient framework for measuring performance of the individual components.

Question rewriting for conversational QA was initially proposed by Elgohary et al., who released the CANARD dataset that contains human rewrites of questions from QuAC. However, the evaluation of the question rewriting approaches trained on this task was limited to the intrinsic metrics reported in the original paper [15]. No evaluation of the end-to-end performance using the generated question rewrites or the impact of the errors propagating from this stage to the question answering stage was reported to date. We provide evidence that an improvement in a QR model, as evaluated by these metrics, will translate to better end-to-end performance in conversational QA. Finally, we show that the same QR model trained on the question rewrites for extractive QA can as well extend a standard retrieval QA model trained on the passage ranking task [17] to perform conversational QA on the TREC CAsT dataset.

3 APPROACH

QR allows to apply existing retrieval and extractive QA models in a conversational setting by introducing a question rewriting component that generates explicit questions interpretable outside of the conversation context (see Figure 1 for an illustrative example). To evaluate question rewriting for conversational QA on the end-to-end task, we set up two different QA models independently from each other: one of them is designed for passage retrieval and one for answer extraction from a passage. This setup allows us to better examine performance of the individual components and analyze similarities and differences between retrieval and extractive models that can provide insights on the potential for a better integration of all three components together.

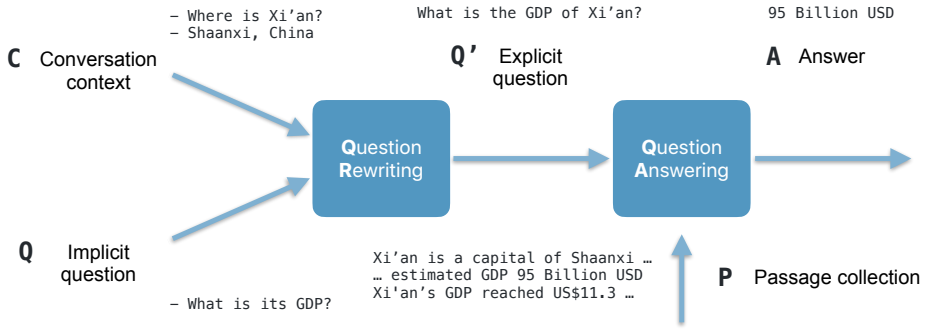

Figure 1: Our approach for end-to-end conversational QA relies on the question rewriting component to handle conversation context and produce an explicit question that can be fed to standard, non-conversational QA components.

3.1 Question Rewriting

Given a conversation context C and a potentially implicit question Q, a question which may require the conversation context C to be fully interpretable, the task of a question rewriting (QR) model is to generate an explicit question Q' which is equivalent to Q under conversation context C and has the same correct answer A. We use a model for question rewriting, which employs a unidirectional Transformer decoder [21] for both encoding the input sequence and decoding the output sequence.

The input to the model is the question with previous conversation turns (we use 5 previous turns in our experiments) turned into token sequences separated with a special [*SEP*] token. The training objective is to predict the output tokens provided in the ground truth question rewrites produced by human annotators. The model is trained via teacher forcing approach, which is a standard technique for training language generation models, to predict every next token in the output sequence given all the preceding tokens. The loss is calculated as negative log-likelihood (cross-entropy) between the output distribution $D' \in \mathbb{R}^{|V|}$ over the vocabulary V, and the one-hot vector $y_{Q'} \in \mathbb{R}^{|V|}$ for the correct token from the ground truth: $loss = -y_{Q'} \log D'$. At training time the output sequence is shifted by one token and is used as input to predict all next tokens of the output sequence at once. At inference time, the model uses maximum likelihood estimation to select the next token from the final distribution D' (greedy decoding), as shown in Figure 2.

We further increase capacity of our generative model by learning to combine several individual distributions (D'_1 and D'_2 in Figure 2). The final distribution D' is then produced as a weighted sum of the intermediate distributions: $D' = \sum_{i=0}^{m} \alpha_i D'_i$ ($m = 2$ in our experiments). To produce $D'_i \in \mathbb{R}^{|V|}$ we pass the last hidden state of the Transformer Decoder $h \in \mathbb{R}^d$ through a separate linear layer for each intermediary distribution: $D'_i = W_i^H h + b^H$, where W_i^H is the weight matrix and b^H is the bias. For the weighting coefficients α_i we use the matrix of input embeddings $X \in \mathbb{R}^{n \times d}$, where n is the maximum sequence length and d is the embedding dimension, and the output of the first attention head of the Transformer Decoder $G \in \mathbb{R}^{n \times d}$ put through a layer normalization function: $\alpha_i = W_i^G norm(G) + W_i^X X + b_i^\alpha$, where all W are the weight matrices and b_i^α is the bias.

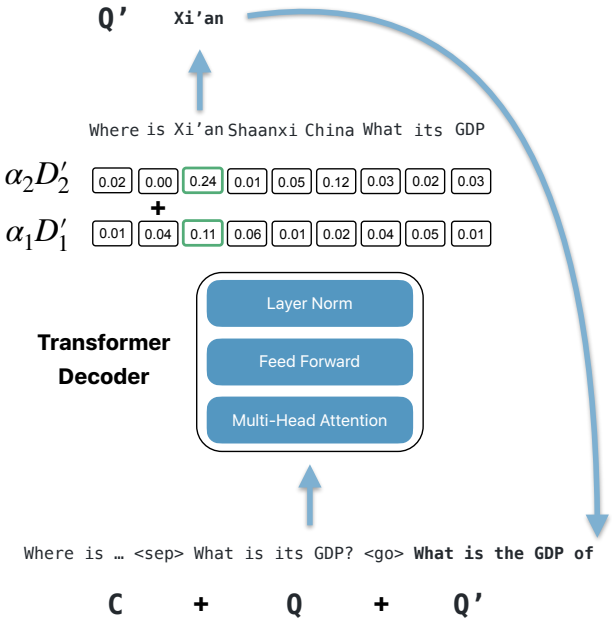

Figure 2: The question rewriting component uses the Transformer Decoder architecture, to recursively generate the tokens of an "explicit" question. At inference time, the generated output is appended to the input sequence for the next timestep in the sequence.

3.2 Retrieval QA

In the retrieval QA settings, the task is to produce a ranked list of text passages from a collection, ordered by their relevance to a given a natural language question [6, 17]. We employ a state-of-the-art approach to retrieval QA, which consists of two phases: candidate selection and passage re-ranking.

In the first phase, a traditional retrieval algorithm (BM25) is used to quickly sift through the indexed collection retrieving top-k passages ranked by relevance to the input question Q'. In the second phase, a more computationally-expensive model is used to re-rank all question-answer candidate pairs formed using the previously retrieved set of k passages.

For re-ranking, we use a binary classification model that predicts whether the passage answers a question, i.e., the output of the model is the relevance score in the interval [0, 1]. The input to the re-ranking model is the concatenated question and passage with a separation token in between (see Figure 3 for the model overview). The model is initialized with weights learned from unsupervised pre-training on the language modeling (masked token prediction) task (BERT) [5]. During fine-tuning, the training objective is to reduce cross-entropy loss, using relevant passages and non-relevant passages from the top-k candidate passages.

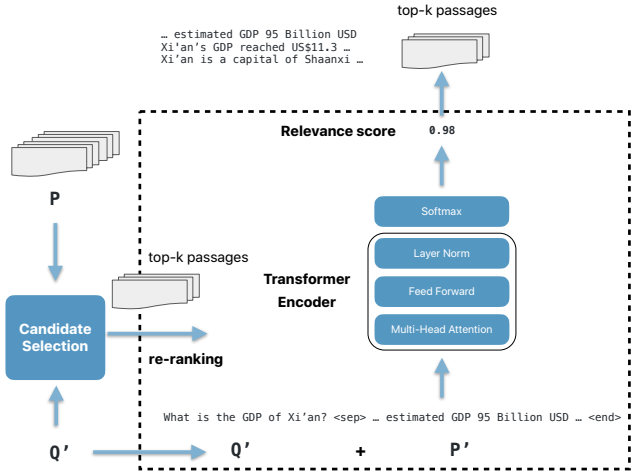

Figure 3: Retrieval QA component includes two sequential phases: candidate selection (BM25) followed by passage re-ranking (Transformer Encoder).

3.3 Extractive QA

The task of extractive QA is given a natural language question and a single passage find an answer as a contiguous text span within the given passage [22]. Our model for extractive QA consists of a Transformer-based bidirectional encoder (BERT) [5] and an output layer predicting the answer span. The input to the model is the sequence of tokens formed by concatenating a question and a passage separated with a special [*SEP*] token. The encoder layers are initialized with the weights of a Transformer model pre-trained on an unsupervised task (masked token prediction). The output of the Transformer encoder is a hidden vector T_i for each token i of the input sequence. For fine-tuning the model on the extractive QA task, we add weight matrices W^s, W^e and biases b^s, b^e that produce two probability distributions over all the tokens of the given passage separately for the start (S) and end position (E) of the answer span. For each token i the output of the Transformer encoder T_i is passed through a linear layer, followed by a softmax normalizing the output logits over all the tokens into probabilities:

$$S_i = \frac{e^{W^s \cdot T_i + b^s}}{\sum_{j=1}^{n} e^{W^s \cdot T_j + b^s}} \qquad E_i = \frac{e^{W_e \cdot T_i + b_e}}{\sum_{j=1}^{n} e^{W^e \cdot T_j + b^e}} \tag{1}$$

The model is then trained to minimize cross-entropy between the predicted start/end positions (S_i and E_i) and the correct ones from the ground truth (y_S and y_E are one-hot vectors indicating the correct start and end tokens of the answer span):

$$loss = -\sum_{i=1}^{n} y_S \log S_i - \sum_{i=1}^{n} y_E \log E_i \tag{2}$$

At inference time all possible answer spans from position i to position j, where $j \geq i$, are scored by the sum of end and start positions' probabilities: $S_i + E_j$. The output of the model is the maximum scoring span (see Figure 4 for the model overview).

21% of the CANARD (QuAC) examples are Not Answerable (NA) by the provided passage. To enable our model to make No Answer predictions we prepend a special [*CLS*] token to the beginning of the input sequence. For all No Answer samples we set both the gold truth start and end positions spans to this token's position (0). Likewise, at inference time, predicting this special token is equivalent to a No Answer prediction for the given example.

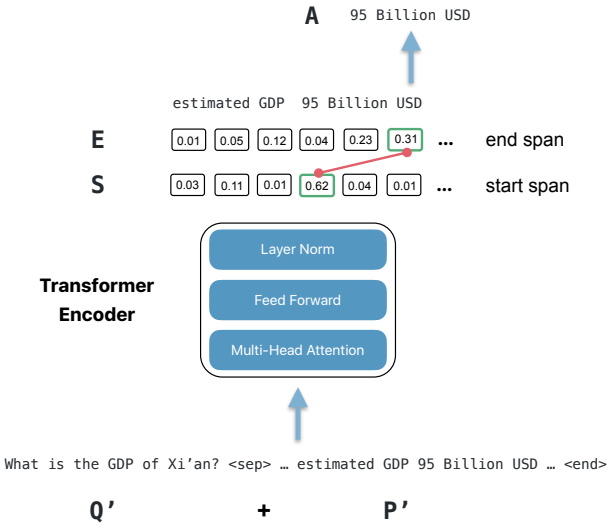

Figure 4: Extractive QA component predicts a span of text in the paragraph P', given an input sequence with the question Q' and passage P'.

4 EXPERIMENTAL SETUP

In the following subsections we describe the datasets used for training and evaluation, the set of metrics for each of the components, our baselines and details of the implementation.

4.1 Datasets

We chose two conversational QA datasets for the evaluation of our approach: (1) CANARD, derived from Question Answering in

Table 1: Datasets used for training and evaluation (with the number of questions).

Tasks	Question Rewriting	Retrieval QA	Extractive QA
Train	CANARD (35k)	MS MARCO (399k)	MultiQA (75k) CANARD (35k)
Test	TREC CAsT (173) CANARD (5.5k)	TREC CAsT (173)	CANARD (5.5k)

Context (QuAC) for extractive conversational QA [2], and (2) TREC CAsT for retrieval conversational QA [4].

Following the setup of the TREC CAsT 2019, we use the MS MARCO Passage Ranking [17] and the TREC CAR [6] paragraph collections. After de-duplication, the MS MARCO collection contains 8.6M documents and the TREC CAR – 29.8M documents. We evaluated on the test set with relevance judgements for 173 questions across 20 dialogues (topics).

CANARD [7] is built upon the QuAC dataset [2] by employing human annotators to rewrite original questions from QuAC dialogues into explicit questions. CANARD contains 40.5k pairs of question rewrites that can be matched to the original answers in QuAC. We use CANARD splits for training and evaluation. Each answer in QuAC is annotated with a Wikipedia passage from which it was extracted alongside the correct answer spans within this passage. We use the question rewrites provided in CANARD and passages with answer spans from QuAC. In our experiments, we refer to this joint dataset as CANARD for brevity.

See Table 1 for the overview of the datasets. Since TREC CAsT is relatively small we use only CANARD for training QR. The same QR model trained on CANARD is evaluated on both CANARD and TREC CAsT. The model for retrieval QA is tuned on a sample from the MS MARCO passage ranking dataset, which includes relevance judgements for 12.8M query-passage pairs with 399k unique queries [18]. The model for extractive QA is pre-trained on MultiQA dataset, which contains 75k QA pairs from six standard QA benchmarks [8].

4.2 Metrics

Mean average precision (*MAP*), mean reciprocal rank (*MRR*), normalized discounted cumulative gain (*NDCG@3*) and precision on the top-passage (*P@1*) evaluate quality of passage ranking. 1000 documents are evaluated per query with a relevance judgement value cut-off level of 2. We use *F1* and Exact Match (*EM*) for extractive QA, which measure word token overlap between the predicted answer span and the ground truth. We also report accuracy for questions without answers in the given passage (*NA Acc*).

Our analysis showed that *ROUGE* recall calculated for unigrams (ROUGE-1 recall) correlates with the human judgement of the question rewriting performance (Pearson 0.69), which we adopt for our experiments as well. ROUGE [14] is a standard metric of lexical overlap, which is often used in text summarization and other text generation tasks. We also calculate question similarity scores with the Universal Sentence Encoder (*USE*) model [1] (Pearson 0.71).

4.3 QR Baselines

The baselines were designed to challenge the need for a separate QR component by incorporating previous turns as direct input to custom QA components. Manual rewrites by human annotators provide the upper-bound performance for a QR approach and allows for an ablation study of the down-stream QA components.

Original. Original questions from the conversational QA datasets without any question rewriting.

Original + k-DT.. Our baseline approach for extractive QA prepends the previous k questions to the original question to compensate for the missing context. The questions are separated with a special token and used as input to the Transformer model. We report the results for $k = \{1, 2, 3\}$.

Original + k-DT*. Since in the first candidate selection phase we use BM25 retrieval function which operates on a bag-of-words representation, we modify the baseline approach for retrieval QA as follows. We select keywords from k prior conversation turns (not including current turn) based on their inverse document frequency (IDF) scores and append them to the original question of the current turn. We use the keyword-augmented query as the search query for **Anserini** [30], a Lucene toolkit for replicable information retrieval research, and if we use BERT re-ranking we concatenate the keyword-augmented query with the passages retrieved from the keyword-augmented query. We use the keywords with IDF scores above the threshold of 0.0001, which was selected based on a 1 million document sample of the MS MARCO corpus.

Human. To provide an upper bound (skyline), we evaluate all our models on the question rewrites manually produced by human annotators.

4.4 QR Models

In addition to the baselines described above, we chose several alternative models for question rewriting of the conversational context: (1) co-reference resolution as in the TREC CAsT challenge; (2) PointerGenerator proposed in the related work for CANARD but not evaluated on the end-to-end conversational QA task [7]; (3) CopyTransformer extension of the PointerGenerator model that replaces the bi-LSTM encoder-decoder architecture with a Transformer Decoder model. All models, except co-reference, were trained on the train split of the CANARD dataset. Question rewrites are generated turn by turn for each dialogue recursively using already generated rewrites as previous turns. This is the same setup as in the TREC CAsT evaluation.

Co-reference. Anaphoric expressions in original questions are replaced with their antecedents from the previous dialogue turns. Co-reference dependencies are detected using a publicly available neural co-reference resolution model that was trained on the OntoNotes corpus [13].[1]

PointerGenerator. A sequence-to-sequence model for text generation with bi-LSTM encoder and a pointer-generator decoder [25].

[1] https://github.com/kentonl/e2e-coref

CopyTransformer. The Transformer decoder, which, similar to pointer-generator model, uses one of the attention heads as a pointer [10]. The model is initialized with the weights of a pre-trained GPT2 model [21, 29] (Medium-sized GPT-2 English model: 24-layer, 1024-hidden, 16-heads, 345M parameters) and then fine-tuned on the question rewriting task.

Transformer++. The Transformer-based model described in Section 3.1. Transformer++ is initialized with the weights of the pre-trained GPT2 model, same as in CopyTransformer.

4.5 QA Models

Our retrieval QA approach is implemented as proposed in [18] using Anserini for the candidate selection phase with BM25 (top-1000 passages) and $BERT_{LARGE}$ for the passage re-ranking phase (Anserini + BERT). Both components were fine-tuned only on the MS MARCO dataset ($k_1 = 0.82, b = 0.68$).[2]

We train several models for extractive QA on different variants of the training set based on the CANARD training set [7]. All models are first initialized with the weights of the $BERT_{LARGE}$ model pre-trained using the whole word masking [5].

CANARD-O. The baseline models were trained using original (implicit) questions of the CANARD training set with a dialogue context of varying length (Original and Original + k-DT). The models are trained separately for each $k = \{0, 1, 2, 3\}$, where $k = 0$ corresponds to the model trained only on the original questions without any previous dialogue turns.

CANARD-H. To accommodate input of the question rewriting models, we train a QA model that takes human rewritten question from the CANARD dataset as input without any additional conversation context, i.e., as in the standard QA task.

MultiQA → CANARD-H. Since the setup with rewritten questions does not differ from the standard QA task, we experiment with pretraining the extractive QA model on the MultiQA dataset with explicit questions [8], using parameter choices introduced by Longpre et al. [16]. We further fine-tune this model on the target CANARD dataset to adopt the model to a different type of QA samples in CANARD (see Figure 6).

5 RESULTS

Our proposed approach, using question rewriting for conversational QA, consistently outperforms the baselines that use previous dialogue turns, in both retrieval and extractive QA tasks. The PointerGenerator network that was previously proposed for the QR task in [7] is the weakest rewriting model according to the end-to-end QA results (MAP 0.100 F1 57.37). This result was not apparent from the QR evaluation metric reported in Table 2.

Passage re-ranking with BERT always improves ranking results (almost a two-fold increase in MAP, see Table 3). The best question rewriting model, Transformer++, when used together with BERT re-ranking and Anserini retrieval, sets the new state of the art on the TREC CAsT dataset with a 28% improvement in MAP and 21% in NDCG@3 over the top automatic run [4]. Keyword-based baselines (Original + k-DT*) prove to be very strong outperforming both Co-reference and PointerGenerator models on all three performance metrics. Both MRR and NDCG@3 are increasing with the number of turns used for sampling keywords, while MAP is slightly decreasing, which indicates that it brings more relevant results at the very top of the rank but non-relevant results also receive higher scores. In contrast, the baseline results for Anserini + BERT model indicate that the re-ranking performance for all metrics decreases if the keywords from more than 2 previous turns are added to the original question. Similarly, we observe a performance peak of the F1 measure at $k = 2$ in the extractive QA settings (see Table 4).

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

Table 2: Evaluation results of the QR models. *Human performance is measured as the difference between two independent annotators' rewritten questions, averaged over 100 examples. This provides an estimate of the upper bound.

Test Set	Question	ROUGE	USE	EM
CANARD	Original	0.51	0.73	0.12
	Co-reference	0.68	0.83	0.48
	PointerGenerator	0.75	0.83	0.22
	CopyTransformer	0.78	0.87	0.56
	Transformer++	**0.81**	**0.89**	**0.63**
	Human*	0.84	0.90	0.33
TREC CAsT	Original	0.67	0.80	0.28
	Co-reference	0.71	0.80	0.13
	PointerGenerator	0.71	0.82	0.17
	CopyTransformer	0.82	0.90	0.49
	Transformer++	**0.90**	**0.94**	**0.58**
	Human*	1.00	1.00	1.00

Table 3: Retrieval QA results on the TREC CAsT test set.

QA Input	QA Model	MAP	MRR	NDCG@3
Original	Anserini	0.089	0.245	0.131
Original + 1-DT*		0.133	0.343	0.199
Original + 2-DT*		0.130	0.374	0.213
Original + 3-DT*		0.127	0.396	0.223
Co-reference		0.109	0.298	0.172
PointerGenerator		0.100	0.273	0.159
CopyTransformer		0.148	0.375	0.213
Transformer++		0.190	0.441	0.265
Human		0.218	0.500	0.315
Original	Anserini	0.172	0.403	0.265
Original + 1-DT*	+BERT	0.230	0.535	0.378
Original + 2-DT*		0.245	0.576	0.404
Original + 3-DT*		0.238	0.575	0.401
Co-reference		0.201	0.473	0.316
PointerGenerator		0.183	0.451	0.298
CopyTransformer		0.284	0.628	0.440
Transformer++		**0.341**	**0.716**	**0.529**
Human		0.405	0.879	0.589

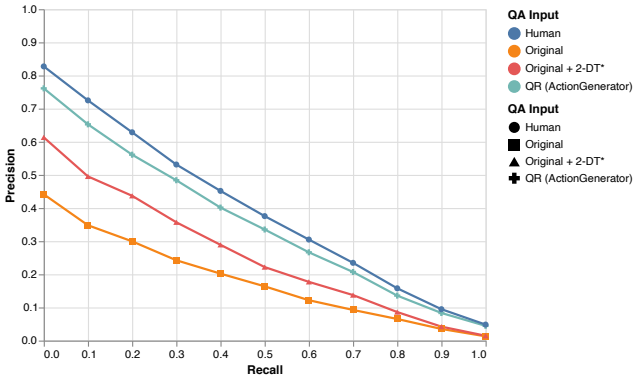

Figure 5: Precision-recall curve illustrating model performance on the TREC CAsT test set for Anserini + BERT.

Table 4: Extractive QA results on the CANARD test set.

QA Input	Training Set	EM	F1	NA Acc
Original	CANARD-O	38.68	53.65	66.55
Original + 1-DT		42.04	56.40	66.72
Original + 2-DT		41.29	56.68	68.11
Original + 3-DT		42.16	56.20	**68.72**
Original	CANARD-H	39.44	54.02	65.42
Original	MultiQA →	41.32	54.97	65.84
Co-reference	CANARD-H	42.70	57.59	66.20
PointerGenerator		41.93	57.37	63.16
CopyTransformer		42.67	57.62	68.02
Transformer++		**43.39**	**58.16**	68.29
Human		45.40	60.48	70.55

Training an extractive QA model on rewritten questions improves the performance even when the original questions are used as input at inference time (CANARD-H in Table 4). The tendency of the extractive model to hit correct answers even with incomplete information from ambiguous questions is also observable in Figure 7. In comparison with the retrieval QA performance across different question formulations on the left plot, extractive model answers twice as much questions without rewriting correctly, i.e. the impact of rewriting original questions is much more pronounced in retrieval QA (middle layers in the left plot). The anomalous behaviour of the extractive model to answer the original implicit question but not when it was explicitly reformulated by a human annotator (green and purple in the right plot) is almost absent in the retrieval model analysis (see also the sparse rows in Table 3).

Pre-training on MultiQA improves performance of the extractive QA model. The style of questions in CANARD dataset is rather different from other QA tasks and the Figure 6 shows that even a little training data for CANARD helps to immediately boost the performance of a pre-trained model.

The precision-recall trade-off curve in Figure 5 shows that question rewriting performance is close to the performance achieved by manually rewriting implicit questions. Precision is decreasing

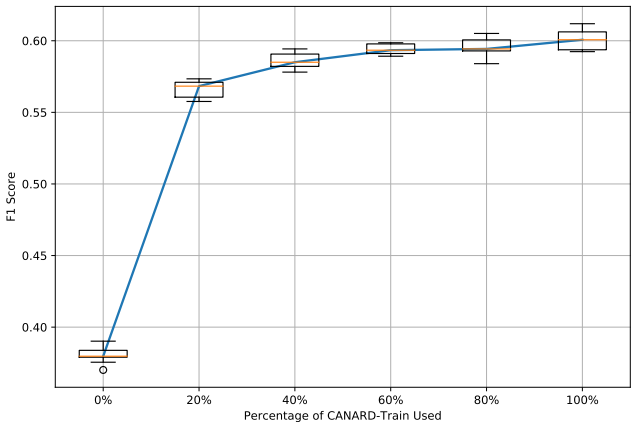

Figure 6: Effect from fine-tuning the MultiQA model on a portion of the target CANARD-H dataset, visualizing domain shift between the datasets. Median, quartiles, and min/max are given by the red lines, boxes and whiskers, respectively.

rapidly and is below 0.1 when optimised for full recall. At the same time, the performance results of the extractive QA suggest that the model can not discriminate well the passages that do not have an answer to the question (71% accuracy on the human rewrites). Since the proportion of irrelevant passages produced by the retrieval component is twice the proportion of non-answerable questions in CANARD, the error rate is expected to increase when these components are combined.

6 OUR ERROR ANALYSIS FRAMEWORK BASED ON QUESTION REWRITES

We introduce a novel evaluation framework based on question rewrites, which helps to automatically distinguish the errors that originate from the incorrect context interpretation from the errors in question answering. We demonstrate the use of the framework by performing a detailed analysis of our evaluation results to better understand the impact of question rewriting on the performance of the retrieval and extractive QA components.

For every answer in a dataset we have 3 types of question formulation: (1) an original, possibly implicit, question (**Original**), (2) rewrites produced by one of our QR models (we take the rewrites generated by the best QR model – Transformer++ (**QR**) and (3) rewrites produced by a human annotator (**Human**). By observing how the answer is changing with the change in the question formulation we can estimate the effectiveness of question rewriting and the robustness of the question answering model.

Table 5 illustrates our approach. Notice, that we can apply the same approach for both retrieval and extractive QA evaluation (see Table 6 for the results on extractive QA). Each row of the table represents one of the combinations of the possible QA results for 3 types of question formulation. For example, the first row of the table indicates the situation, when neither the original question, nor ther generated or human rewrite were able to solicit the correct answer (× × ×). We then can automatically calculate how many

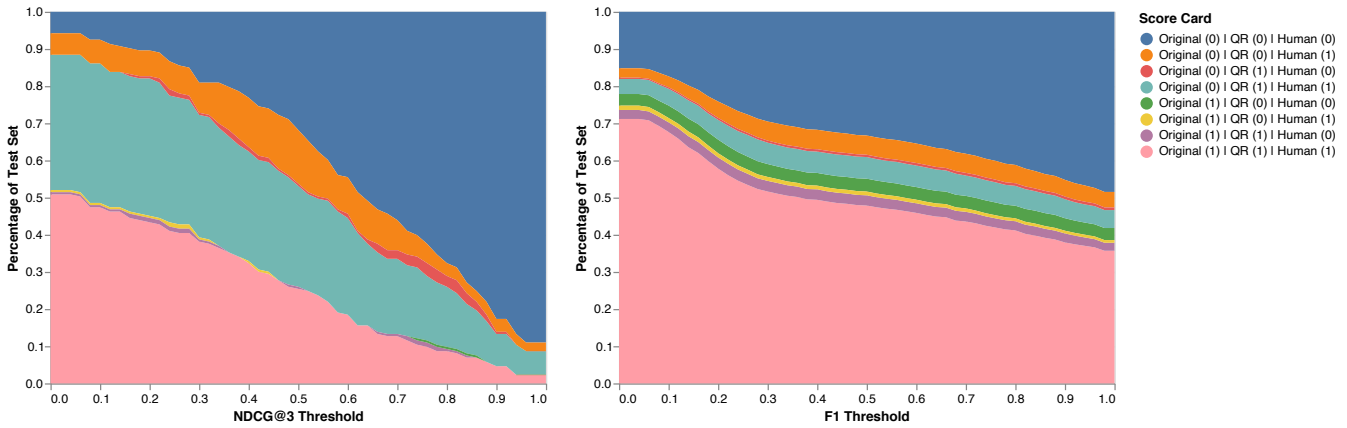

Figure 7: Break-down analysis (best in color) with a sliding threshold for both retrieval (left) and extractive QA (right) results. The plot shows the difference between error distributions in retrieval and extractive settings. Most of the correct spans can be extracted given a relevant passage even with an original ambiguous question (the pink region in the bottom).

samples in our results fall into each of these bins. Since there is no single binary measure for the answer correctness, we can pick different cut-off thresholds using our QA metrics. For example, P@1=1 will consider the answer correct if it came up at the top of the ranking; or F1=1 will consider the answer correct only in cases with full span overlap, i.e, exact matches only. Figure 7 extends this analysis by considering all thresholds in the range [0; 1] with 0.02 intervals for NDCG@3 in retrieval and F1 in extractive QA. This figure shows the proportion of different error types as well as the results sensitivity to the choice of the performance threshold.

Assuming that humans always produce correct question rewrites, we can attribute all cases in which these rewrites did not result in a correct answer as errors of the QA component (rows 1-4 in Tables 5-6). The next two rows 5-6 show the cases, where human rewrites succeeded but the model rewrites failed, which we consider to be a likely error of the QR component. The last two rows are true positives for our model, where the last row combines cases where the original question was just copied without rewriting (numbers in brackets) and other cases when rewriting was not required.

The majority of errors stem from the QA model: 29% of the test samples for retrieval and 55% for extractive estimated for P@1 and F1, comparing to 11% and 5% for QR respectively. It is a rough estimate since we can not tell whether the cases failing QA did not fail QR as well. Another interesting observation is that 10% of the questions in TREC CAsT were rewriten by human annotators that did not need rewriting to retrieve the correct answer. For CANARD the majority of questions (62%) can be correctly answered without question rewriting even when the questions are ambiguous.

There are two anecdotal cases where our QR component was able to generate rewrites that helped to produce better ranking than the human-written questions. The first example shows that the re-ranking model does not handle paraphrases well. Original question: "What are good sources in food?", human rewrite: "What are good sources of *melatonin* in food?", model rewrite: "What are good sources in food for *melatonin*". In the second example the human annotator and our model chose different context to disambiguate the original question. Original question: "What about environmental factors?", human rewrite: "What about environmental factors during the *Bronze Age collapse*?", model rewrite: "What about environmental factors that lead to led to a *breakdown of trade*". Even though both model rewrites are not grammatically correct they solicited correct top-answers, while the human rewrites failed, which indicate flaws in the QA model performance.

Table 5: Break-down analysis of all retrieval QA results for the TREC CAsT dataset. Each row represents a group of QA samples that exhibit similar behaviour. ✓ indicates that the answer produced by the QA model was correct or × – incorrect, according to the thresholds provided in the right columns. We consider three types of input for every QA sample: the question from the test set (Original), generated by the best QR model (Transformer++) or rewritten manually (Human). The numbers correspond to the count of QA samples for each of the groups. The numbers in parenthesis indicate how many questions do not require rewriting, i.e., should be copied from the original.

			P@1	NDCG@3		
Original	QR	Human	= 1	> 0	≥ 0.5	= 1
×	×	×	49 (14)	10 (1)	55 (20)	154 (49)
✓	×	×	0	0	0	0
×	✓	×	2	0	1	0
✓	✓	×	0	1	1	0
×	×	✓	19	10	25	4
✓	×	✓	0	1	0	0
×	✓	✓	48	63	47	11
✓	✓	✓	55 (37)	88 (52)	44 (33)	4 (4)
			Total			173 (53)

Metrics correlation. Our evaluation results in Section 5 show that QR results often correlate with the QA results on the model level, which can be used for model selection using QR metrics which can be easier to compute than the end-to-end QA performance. In this section, we check whether QR metrics can also predict the QA performance for the individual questions.

To discover the correlation between QR and QA metrics, we discarded all samples, where the human rewrites do not lead to the correct answers (top 4 rows in Tables 5-6). The remaining subset contains only the samples in which the QA model was able to find the correct answer. We then compute ROUGE for the pairs of human and generated rewrites, and measure its correlation with P@1 to check if rewrites similar to the correct question will also produce correct answers, and vice versa.

There is a strong correlation for ROUGE = 1, i.e., when the generated rewrite is very close to the human one, but when ROUGE < 1 the answer is less predictable. Even for rewrites that have a relatively small lexical overlap with the ground-truth (ROUGE ≤ 0.4) it is possible to retrieve a correct answer, and vice versa.

We further explore the effect of the QR quality on the QA results by comparing differences of the answer sets for different rewrites. We compare answers produced separately for human and model rewrites of the same input question. However, this time we look at all the answers produced by the QA model irrespective of whether the answers were considered correct or not. This setup allows us to better observe how much a change in the question formulation can trigger the change in the produced answer.

Figure 8 demonstrates strong correlation between the question similarity, as measured by ROUGE, and the answer set similarity. We measure the similarity between the top-1000 answers returned for the human rewrites and the generated rewrites by computing recall (R@1000). Points in the bottom right of this plot show sensitivity of the QA component, where similar questions lead to different answer rankings. The data points that are close the top center area indicate weakness of the QR metric as a proxy for the QA results: often questions do not have to be the same as the ground truth questions to solicit the same answers. The blank area in the top-left of the diagonal shows that a lexical overlap is required to produce the same answer set, which is likely due to the candidate filtering phase based on the bag-of-word representation matching.

We also compared ROUGE and Jaccard similarity for the tokens in extractive QA results but they showed only a weak correlation (Pearson 0.31). This result confirms our observation that the extractive model tends to be rather sensitive to a slight input perturbation but will also often provide the same answer to very distinct questions.

QA sensitivity. We showed that QA results can provide an estimate of the question similarity. However, this property is directly dependent on the ability of the QA component to match equivalent questions to the same answer. Alternative question rewrites allow us to evaluate robustness and consistency of the QA models. Our analysis indicates that small perturbations of the input question, such as anaphora resolution, has a considerable impact on the answer ranking, e.g., the pair of the original question: "Who are the Hamilton Electors and what were *they* trying to do?", and the human rewrite: "Who are the Hamilton Electors and what were the

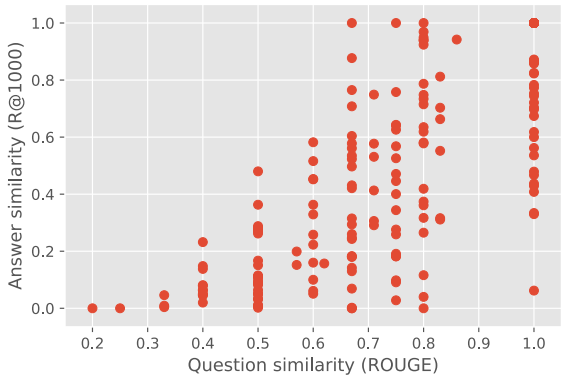

Figure 8: Strong correlation (Pearson 0.77) between question similarity (ROUGE) and the passage rankings produced by the retrieval QA model (Recall).

Hamilton Electors trying to do?" produce ROUGE = 1 but R@1000 = 0.33. We identified many cases in which inability of the QR component to generate apostrophes jeopardized relevance matching. These results demonstrate the challenges in the quality control for this task. When asked for a human judgment of the quality of the generated rewrites independent from the QA results, little deviations may not seem important, but from the pragmatic point of view they have an impact on the overall performance of the end-to-end system.

QR completeness. The level of detail required to answer a particular question is often not apparent and depends on the collection. There are cases in which original questions without rewriting were already sufficient to retrieve the correct answers from the passage collection (see last row of the Table 5). For example, original question: "What is the functionalist theory?", human rewrite: "What is the functionalist theory in *sociology*?" However, in another question from the same dialogue, omitting the same word from the rewrite leads to retrieval of an irrelevant passage, since there are multiple alternative answers. This class of errors also corresponds to the variance evident from the Figure 8, since a one-word difference between two questions may have a very little effect on the answer ranking as well as a dramatic change in the question interpretation. This effect, however, interacts with the size and diversity of the collection content. Some of the questions were correctly answered even with underspecified questions, e.g., original question: "What are some ways to avoid injury?", human rewrite: "What are some ways to avoid *sports* injuries?", because of the collection bias. The idea behind the question rewriting approach is to learn patterns that correct for such semantic differences independent from the collection content, similar to how humans resolve such cases, based on their knowledge of language and the world.

7 CONCLUSION

Question rewriting (QR) is a challenging task that attempts to learn linguistic patterns that signal and resolve ambiguity in question formulation. The core idea to develop QR as a separate component

Table 6: Break-down analysis of all extractive QA results for the CANARD dataset, similar to Table 5.

Original	QR	Human	F1 > 0	F1 ≥ 0.5	F1 = 1
×	×	×	847 (136)	1855 (235)	2701 (332)
✓	×	×	174	193	181
×	✓	×	19	35 (2)	40 (1)
✓	✓	×	135	153	120
×	×	✓	141	288	232
✓	×	✓	65 (1)	57 (1)	40
×	✓	✓	226	324	269
✓	✓	✓	3964 (529)	2666 (428)	1988 (333)
			Total		5571 (666)

is that the context understanding and question formulation are independent from the knowledge collection process, i.e. the way humans approach this by relying on their linguistic and world knowledge. Our experimental results show that human intuition about correct rewrites is sub-optimal but it allows to establish a mechanism that performs well across different datasets.

We showed in an end-to-end evaluation that question rewriting is an effective method to extend existing question answering (QA) approaches to conversational settings. Our experiments provide a comprehensive evaluation of the impact of question rewriting on the conversational QA performance. By producing explicit representations of the question interpretation, question rewriting makes conversational QA results more explicable and the contribution of the individual components more transparent, which enable us to conduct a thorough performance analysis. The experimental evaluation and the detailed analysis we provide increases our understanding of the task and the main sources of errors. Finally, our analysis demonstrates sensitivity of the both QA models to different question formulations. We argue that this evaluation setup is more adequate since it is able to reflect upon the model robustness. Future work should explore joint training that could use this setup also for fine-tuning QR and QA models.

We compared two QA models side by side and discovered major differences in their interaction with the QR component. The role of question rewriting is especially prominent in the case of retrieval QA where the candidate answer space is so large that any ambiguity in question formulation results in a very different answer. In contrast, the extractive QA model setup is optimized for recall and tends to produce answers for questions that are ambiguous or unanswerable given the passage. In future work we would like to inregrate both retrieval and extractive QA models, which requires collection of an appropriate dataset that covers all three tasks.

It is important to note that our QR-QA architecture is generic enough to incorporate other types of context, such as a user model or an environmental context obtained from multi-modal data (deictic reference). Experimental evaluation of QR-QA performance augmented with such auxiliary inputs is a promising direction for future work.

REFERENCES

[1] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 169–174.

[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*. 2174–2184.

[3] Philipp Christmann, Rishiraj Saha Roy, Abdalghani Abujabal, Jyotsna Singh, and Gerhard Weikum. 2019. Look before you Hop: Conversational Question Answering over Knowledge Graphs Using Judicious Context Expansion. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 729–738.

[4] Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. 2019. CAsT 2019: The Conversational Assistance Track Overview. In *Proceedings of the 28th Text REtrieval Conference*. 13–15.

[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*. 4171–4186.

[6] Laura Dietz, Ben Gamari, Jeff Dalton, and Nick Craswell. 2018. TREC Complex Answer Retrieval Overview. TREC.

[7] 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*. 5920–5926.

[8] Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. MRQA 2019 Shared Task: Evaluating Generalization in Reading Comprehension. *arXiv preprint arXiv:1910.09753* (2019).

[9] Jianfeng Gao, Michel Galley, and Lihong Li. 2019. Neural Approaches to Conversational AI. *Foundations and Trends in Information Retrieval* 13, 2-3 (2019), 127–298.

[10] Sebastian Gehrmann, Yuntian Deng, and Alexander M Rush. 2018. Bottom-Up Abstractive Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 4098–4109.

[11] Daya Guo, Duyu Tang, Nan Duan, Ming Zhou, and Jian Yin. 2018. Dialog-to-Action: Conversational Question Answering Over a Large-Scale Knowledge Base. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018*. 2946–2955.

[12] Ying Ju, Fubang Zhao, Shijie Chen, Bowen Zheng, Xuefeng Yang, and Yunfeng Liu. 2019. Technical report on Conversational Question Answering. *CoRR* abs/1909.10772 (2019).

[13] Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-Order Coreference Resolution with Coarse-to-Fine Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 687–692.

[14] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.

[15] Sheng-Chieh Lin, Jheng-Hong Yang, Rodrigo Nogueira, Ming-Feng Tsai, Chuan-Ju Wang, and Jimmy Lin. 2020. Conversational Question Reformulation via Sequence-to-Sequence Architectures and Pretrained Language Models. *arXiv preprint arXiv:2004.01909* (2020).

[16] Shayne Longpre, Yi Lu, Zhucheng Tu, and Chris DuBois. 2019. An Exploration of Data Augmentation and Sampling Techniques for Domain-Agnostic Question Answering. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*. 220–227.

[17] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches*.

[18] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).

[19] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document Expansion by Query Prediction. *CoRR* abs/1904.08375 (2019).

[20] 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*. 1391–1400.

[21] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog* 1, 8 (2019).

[22] 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*. 2383–2392.

[23] Pushpendre Rastogi, Arpit Gupta, Tongfei Chen, and Lambert Mathias. 2019. Scaling Multi-Domain Dialogue State Tracking via Query Reformulation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 97–105.

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

[25] Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. 1073–1083.

[26] Hui Su, Xiaoyu Shen, Rongzhi Zhang, Fei Sun, Pengwei Hu, Cheng Niu, and Jie Zhou. 2019. Improving Multi-turn Dialogue Modelling with Utterance ReWriter. *arXiv preprint arXiv:1906.07004* (2019).

[27] Andrew L Thomas. 1979. Ellipsis: The Interplay of Sentence Structure and Context. *Lingua Amsterdam* 47, 1 (1979), 43–68.

[28] Mengqiu Wang, Noah A Smith, and Teruko Mitamura. 2007. What is the Jeopardy model? A quasi-synchronous grammar for QA. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. 22–32.

[29] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Transformers: State-of-the-art Natural Language Processing. *arXiv preprint arXiv:1910.03771* (2019).

[30] Peilin Yang, Hui Fang, and Jimmy Lin. 2017. Anserini: Enabling the Use of Lucene for Information Retrieval Research. In *Proceedings of the 40th International Conference on Research and Development in Information Retrieval*. 1253–1256.

[31] Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A Challenge Dataset for Open-Domain Question Answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2013–2018.