Combined Transformer for Conversational Question Answering

Yuanzhi Zhu
yuazhu@ethz.ch
ETH Zurich

Bartosz Dzionek
bdzionek@ethz.ch
ETH Zurich

Abstract

Conversational question answering (ConvQA) has been one of the most important question answering tasks. In most of the state-of-the-art models, a passage and the past Q&A session are sent as an input to a neural network like BERT. However, only few models are concerned how to combine and use the input as additional information to the current question. In this task, we would try to propose better methods for extracting information from the conversation history as input to a BERT model to generate answers. Our project would first build our framework using the combination of Bert and T5 encoder. A T5 model can be used to generate rewritten question from the conversational history and current history. We expect the encoder of the T5 model can learn the embedding of the conversation history and use this embedding as additional information to improve the performance of Bert. All of our codes can be found on GitHub[1].

1 Introduction

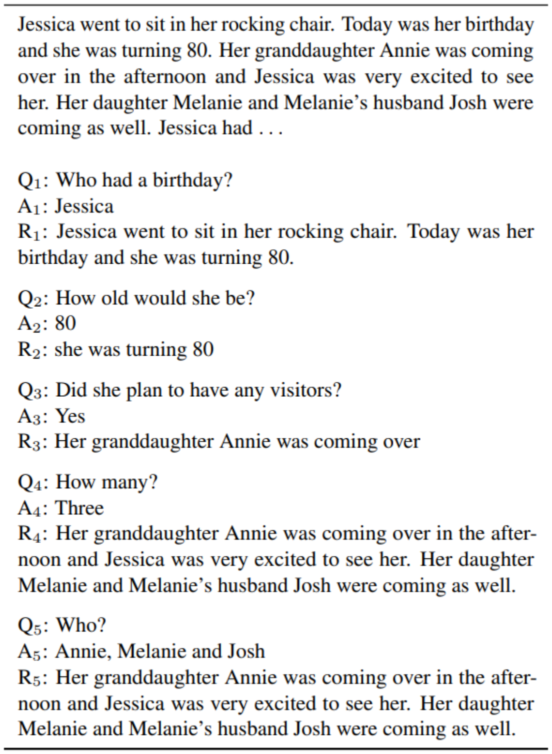
1.1 Background

Question Answering (QA) has been a challenging task in natural language understanding. QA is a task of automatic answer extraction for questions asked in a natural language [5]. Thus, the key components in QA require the capability of understanding the question and the passage in which the question is generated.

ConvQA techniques form the building blocks of QA dialog systems, and the idea behind it is to let the machine generate an answer to a question based on the provided passage and historical conversation [23].

In ConvQA task, the system is asked a sequence of questions in a conversational manner. As shown in Figure 1, every question after the first one is built on what was asked before [16]. The dependence is often of the form of coreference or ellipsis [2]. ConvQA task can be based on a retriever using a large collection of documents of diversified topics, which make it an open domain conversational question answering task [3]. But the task was also explored in the simplified setting, where the system is directly given a reference text [12].

One potential solution to the ConvQA task is question rewriting (QR). QR is a technique that allows to convert a conversational QA task into non-conversational QA. This is done by generating questions that are semantically equivalent to the original one and contain the whole contextual information [2]. For instance, the question "How old would **she** be?" can be rewritten as "How old would **Jessica** be?" based on the conversation history. ConvQA task is also important as it is the base for many modern chatbots [6] [11].



Jessica went to sit in her rocking chair. Today was her birthday and she was turning 80. Her granddaughter Annie was coming over in the afternoon and Jessica was very excited to see her. Her daughter Melanie and Melanie's husband Josh were coming as well. Jessica had . . .

Q$_1$: Who had a birthday?
A$_1$: Jessica
R$_1$: Jessica went to sit in her rocking chair. Today was her birthday and she was turning 80.

Q$_2$: How old would she be?
A$_2$: 80
R$_2$: she was turning 80

Q$_3$: Did she plan to have any visitors?
A$_3$: Yes
R$_3$: Her granddaughter Annie was coming over

Q$_4$: How many?
A$_4$: Three
R$_4$: Her granddaughter Annie was coming over in the afternoon and Jessica was very excited to see her. Her daughter Melanie and Melanie's husband Josh were coming as well.

Q$_5$: Who?
A$_5$: Annie, Melanie and Josh
R$_5$: Her granddaughter Annie was coming over in the afternoon and Jessica was very excited to see her. Her daughter Melanie and Melanie's husband Josh were coming as well.

Figure 1. A sample of dataset for training (*CoQA*) [16]

1.2 Historical Development

The development of systems allowing a natural langauge conversation with computers can be traced back to *ELIZA* in 1966. It was a simple program based only on keywords and fixed rules [21]. In the next decades, various architectures were developed but they were still based on pattern matching. A notable example is *ALICE* from 2007, working on XML-based AIML files with patterns and templates [17]. The significant progress came around the year 2015 with the

[1] https://github.com/yuanzhi-zhu/CSNLP-Project-ETH

advent of large datasets and deep learning that outperformed traditional rule-based methods [23].

In 2018, the invention of BERT revolutionized many NLP tasks including ConvQA[7]. Most of the state-of-the-art models in the leaderboard of both *CoQA* and *QuAC* are variants of BERT [16] [4]. In [2], researchers adapted different models for QR including *AllenAI Coref* [9], *Transformer++*[18] and others. In [19], the researchers compared and combined several QR methods like *Transformer++*[18] and *QuReTeC* [20].

Except QR, there are also other methods to extract information from the conversational history. In [1], the topic for each round of QA is extracted as additional information to improve the overall model performance.

Vakulenko et al. (2021) combined QR methods of different types (sequence generation or term classification) and found it improved upon individual QR methods and achieved state-of-the-art retrieval performance on *CAsT 2019* [19]. However, they simply append terms from *QuReTeC* to the rewritten question produced by one of the generative models like *Transformer++* [18], which left huge improvement space for modification to better combine different ideas.

In this task, we would try to combine state-of-the-art QR approaches and evaluate their performances on ConvQA task. Then we would modify the QA model to further improve the performance.

1.3 Goal

As part of the research project, we will build a framework that consists of two modules. One module will use T5 [14] for question rewriting, and the second will use BERT [7] for question answering. The framework is shown in Figure 2 and described in more detail in the next sections.

2 Baseline Model

2.1 Task Formulation

We first define here our task. Given a source paragraph P and a question Q, the task is to find an answer A to the question and can the pipeline be formulated as follows:

Input: The input of the model consists of current question Q_i, given passage P, history questions Q_{i-1}, ..., Q_{i-k} answers A_{i-1}, ..., A_{i-k}

Output: The answer A_i identified using start span and end span generated by the model. (even for CoQA dataset, there are more generative questions, unlike for SQuAD most of the questions are extractive, here we still use start and end position to extract the answer.) where i and k represents the indices of turn and the number of dialogue history considered, respectively.

2.2 Datasets

For question rewriting, we use CANARD [8] and CoQAR [13], which are adapted from QuAC [4] and CoQA [16], respectively.

QuAC dialogs often switch topics while CoQA dialogs include more queries for details. QuAC focuses on information that could plausibly be in context material, and CoQA does not significantly cover unanswerable questions. Our analysis strongly implies that beyond yes/no questions, abstractive behavior is not a significant component in either QuAC or CoQA. As such, QuAC models can be trivially adapted to CoQA [22]. We will have deeper analysis on the predictions of these datasets in the following sections.

Evaluation for CoQA is the same as SQuAD [15], exact match and F1 score. We will analyze the predictions based on these metrics.

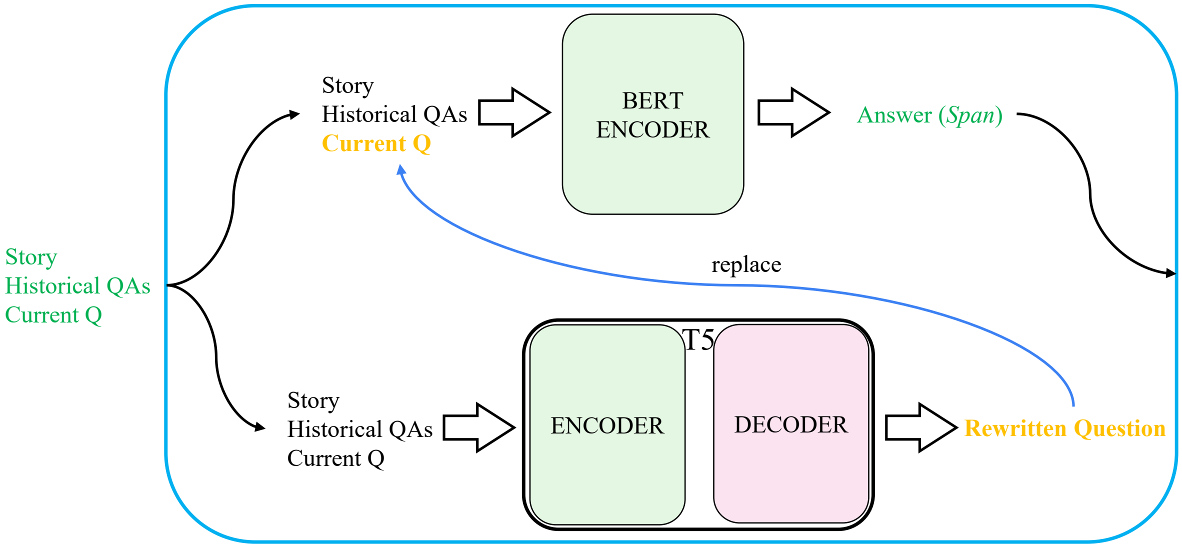


Figure 2. Our Framework

2.3 Fine-Tuning Bert as Baseline

The baseline here is the Bert base uncased model fine-tuned on the CoQA dataset. We show the scores for Bert trained on different epochs in Table 1:

epochs	em	f1
1	65.0	74.9
2	**67.2**	**77.1**
4	66.9	77.0

Table 1. Baseline model for different epochs

From the above table, we use the 2 epochs fine-tuned Bert base model as our baseline for comparison.

The other important training parameters are listed here and we will keep them unchanged unless specified.

Bert model	batch size	max_seq_length	learning rate
base	2	512	1e-5

scheduler	warmup_steps	history_len	weight_decay
linear	2000	2	0.01

Table 2. Parameters for training

3 Model Prediction Analysis

3.1 Distribution of Answers

After fine-tuning BERT, we generated its predictions on the CoQA dev set. For evaluating answers, we used the official script provided by the authors of the dataset. The answers are evaluated by comparing them to human answers. The evaluation metric is exact match score and F1 score of word overlap.

We split the answers of BERT into three categories according to the evaluation metrics:

- *correct* answers are those with exact match with one of human answers,
- *partially* correct answers do not have exact match but have non-zero F1 score,
- *incorrect* answers have zero F1 score.

The dataset is conversational with the maximum turn of 25. It's easily to find that the depth of conversations in the dataset start to decrease after turn 10 and is rare at depth larger than 20, we normalized the first 20 turns. The result is Figure 3. We see that in the first turn the percent of correct answers is 68.8%, and for turns 2-20 it is on average 58.3% (95% confidence interval = 57.5% – 59.1% [2]). In the first turn, incorrect answers comprise 10.4% of answers. In turns 2-20, they comprise on average 17.3% (95% confidence interval = 16.4% – 18.3% [3]).

Such a difference between the first turn and the rest means BERT underperforms in the multi-turn setting. This allows us to experiment with question rewriting from T5 to decrease the gap between single-turn and multi-turn QA.

Even [1] suggest that with all history, the open retrieval QA model performs better than with only the rewritten question. We argue here that with the correct rewritten question, we can turn multi-turn setting into single-turn and improve the performance.

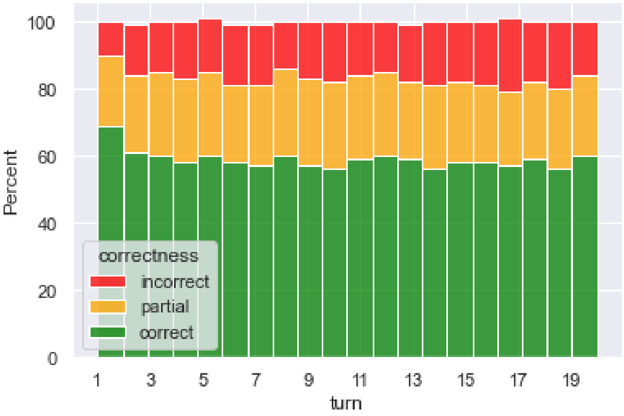


Figure 3. Normalized correctness for turns ≤ 20.

3.2 Failure Case Analysis

As a next step, we looked closer into the answers to find patters connected to incorrect answers. Since we want to get an improvement in the conversational setting, we will only consider examples from turn 2 onward.

We first define the answers with exact match score equals 1 as correct, f1 score equals 0 as incorrect and the rest as partial. We found the ratio of correct/partial/incorrect answers for answers composed of 1 up to 10 words (longer answers comprised very small part) and presented it in Table 3. For answers that are single words it is much easier to get an exact match with the ground truth. As the answers start to be longer, it is less common to get an exact match but more common to have a word overlapping with ground truth answers, thus being partially correct. It is worth noting that ratio of incorrect answers is much bigger for single-word answers (0.22). For comparison, there are 205 single-word answers to a question in the first turn and their incorrect ratio is 0.13. Given that single words have the highest count among the generated answers, they constitute a significant bottleneck hurting performance of the model.

Finally, we checked the correctness with respect to the domain of a question. The public part of CoQA consists of five different domains, every domain appears in approximately similar number of questions. Again, we compared the number of incorrect answers in the first turn against

[2] Computed with t-test, but we also obtained almost identical results with bootstrapping.

[3] See footnote 1.

#words	count	correct	partial	incorrect
1	**3518**	**0.70**	0.08	**0.22**
2	1360	0.66	0.21	0.13
3	880	0.56	0.29	0.14
4	511	0.45	0.46	0.09
5	326	0.37	0.51	0.13
6	227	0.34	0.56	0.10
7	138	0.26	0.61	0.13
8	114	0.21	0.66	0.13
9	91	0.26	0.59	0.14
10	46	0.15	**0.74**	0.11

Table 3. Distribution vs length of the answer. (turn ≥ 2)

turns	domain				
	cnn	gutenberg	mctest	race	wikipedia
turn = 1	0.08	0.19	0.10	0.10	0.05
turn ≥ 2	0.15	0.19	0.18	0.20	0.14
abs diff	**0.07**	**0.00**	**0.08**	**0.10**	**0.09**

Table 4. Ratio of incorrect answers.

the next turns. The result is shown as Table 4. We observed that 4 domains have very similar gap between the first turn and other turns. However, for the *gutenberg* part, we got approximately the same ratio. We plan to examine questions in this domain more closely as our goal is to minimize the gap for other four domains.

3.3 Limitations of CoQA

During exploration of the answers, we noted that CoQA under-evaluates some of our answers. That happens when the generated answer is more detailed than ground truth or vice versa. For example:

- model: *"in a barn near a farm house"*, CoQA: *"in a barn"* (EM: 0.0, F1: 0.71)
- model: *"Joe Fontana"*, CoQA: *"Detective Joe Fontana"* (EM: 0.0, F1: 0.8)

Fortunately, this effect is mitigated by a high F1 score. However, final evaluation might be noisy due to the preference of human contributors to shorter or more detailed answers.

The worse problem is that F1 score does not capture sematics of the answer. This can lead perfect answers to be classified as incorrect (e.g. model: *"fifty"*, CoQA: *"50"*, (EM: 0.0, F1: 0.0)), and adds more noise to the final evaluation.

4 Our Framework

Our framework is a combination of BERT [7] and T5 [14]. As you can see in Figure 2, the common input to both models is the story and conversations (including the historical question answers and current question). These inputs are first sent to a T5 small model to generate the rewritten question. Then, we can replace the original question or even the original conversation with the rewritten question to form updated inputs for BERT.

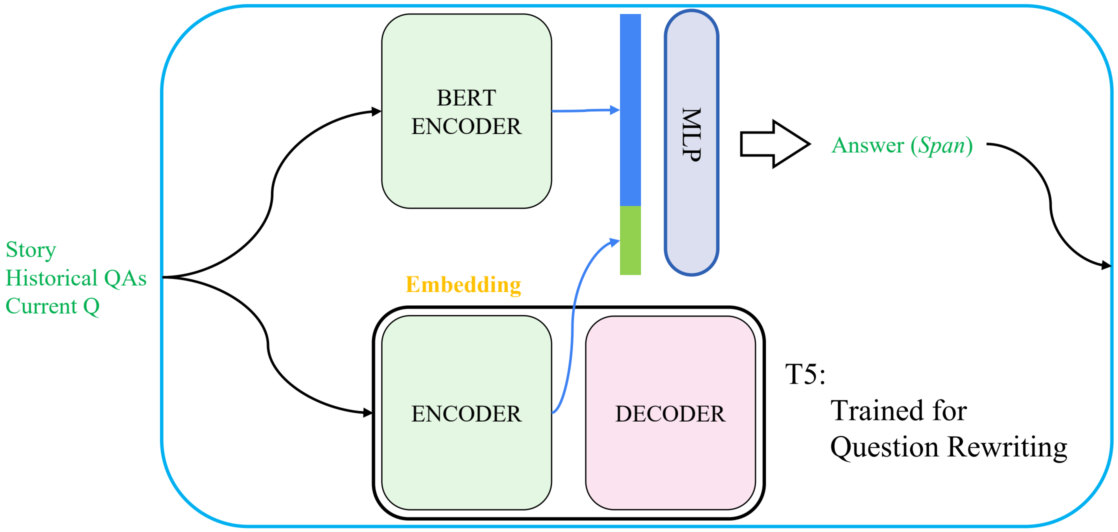


Figure 4. Our Framework (new)

Since both the T5 encoder and the Bert base have the exactly same architecture and we believe that the decoder of T5 will lead to information loss when output rewritten question. In [10], the authors also derived the conclusions that for downstream tasks, utilizing only the T5 encoder is better than using the full T5 encoder-decoder model; and encoder-only mean (pooled) embedding is better than encoder-only first (special token embedding [CLS]). As a result, we proposed another way to combine information from these two models, as illustrated in Figure 4, where the raw output of Bert and T5 encoder are concatenated into a long vector then sent to MLP layers. In our experiment, we will try both the token embedding and the sentence embedding from the T5 encoder.

The final output from BERT is the answer span, and we expect an improvement of more than 5% according to the analysis in section 3.

4.1 T5 Component

We first fine-tuned BERT base on CoQA dataset and T5 small on a mixed dataset of CANARD and CoQAR. The dev set of both CoQAR and CoQA is the same, which makes it easier to build our framework.

The BERT version with the best performance in our case is BERT base trained on CoQA training dataset for 4 epochs. For T5 question rewriting model, we did several experiments because the performance of question rewriting is the key in our framework. The result is shown in Table 5

It is interesting that even without the story, the model can generate a good rewritten question. This is because one can do coreference using only the historical conversation. It is also worth mentioning that the last two experiments show that with the additional training samples from CoQAR, the BLEU score on CANARD is not improved. Indeed, CANARD is adapted from QuAC and CoQAR is adapted from CoQA, and these two datasets have different focus.

According to this result, we select the last model as the component in our framework.

model	story	batch size	hist size	dataset	BLEU CoQA	BLEU CANARD
$T5_{small}$	w/o	16	20	canard	0.3068	0.4665
$T5_{small}$	w/	16	20	canard	0.3304	0.4866
$T5_{base}$	w/	4	20	canard	0.3563	**0.5065**
$T5_{base}$	w/o	4	20	canard	0.3276	0.4953
$T5_{small}$	w/	4	3	canard	0.3261	0.4855
$T5_{small}$	w/	16	3	canard	0.3281	0.4881
$T5_{small}$	w/	16	3	mixed	**0.4039**	0.4756

Table 5. T5 experiments

4.2 Bert Component

Here the Bert part is nothing but the same Bert base model as baseline. However, since the input has changed, we have to re-fine-tune the model to get desired result.

In this case, without modified the Bert itself, we are pushing the limit of this Bert model. And what we want is to fill the gap between first turn and rest turns.

Besides, the additional layers may also be rewritten to better extract the answer from the Bert's raw output.

5 Results

5.1 Preliminary Results

At the beginning, we used the fine-tuned T5 model to generate rewritten questions, and use them to form new inputs the fine-tuned Bert model to get desired answers. When we implement this design, we found that it's rather inconvenient to use the T5 in real time. In order to tackle this issue, we generated a new dataset, named *coqa-v1.0-append_with_T5.json* in our GitHub repository, feeding to the Bert model. The preliminary results are shown below:

Inputs	em	f1
Baseline	67.2	77.1
Replaced with Rewritten Q	61.4	71.5
Concatenated with Rewritten Q	61.8	71.9

Table 6. Preliminary Results without Fine-Tuning

At a result, we observe a much worse score after using the new inputs. To mitigate this issue, we decided to fine-tune the Bert model with the new formed inputs. However, to fine-tune the Bert again, we can't use the T5 model fine-tuned on the mixed dataset as the CoQA train dataset will be seen by the model in advance. So we select the T5 base model fine-tuned only on CANARD dataset. On top of that, we investigated that some of the questions are ill-defined(3%), after the adjustment, we get a updated results as in Table 8

Inputs	em	f1
Baseline	67.2	77.1
Replaced with Rewritten Q (4 epochs)	64.8	74.8
Replaced all (2 epochs)	62.4	71.8
Concatenated w/o history (2 epochs)	63.8	73.5
Concatenated with Rewritten Q (4 epochs)	66.5	76.7
Concatenated with Rewritten Q (2 epochs)	**67.3**	**77.2**

Table 7. Preliminary Results with Fine-Tuning

Here we also tried replace all historical conversation and current question with the rewritten question from T5 (replace all), and concatenated the rewritten question but without the historical conversation (Concatenated w/o history).

We expected to get a higher score with 'Concatenated with Rewritten', but actually not. The reason here could be that:

- append: the Bert model get two 'repeated' question to answer, which somehow distracts the model.
- replace: T5 could generate correct rewritten questions, which leads to a information loss to the Bert.

5.2 Further Results

As proposed above, we also tried to concatenate the embedding extracted from t5 encoder. There are two types of embedding in our case, the token embedding with size $(Batch, max\_seq\_length, hidden\_size)$, and the pooled embedding with size $(Batch, hidden\_size)$

Inputs	em	f1
Baseline	67.2	77.1
embedding (2 epochs)	67.3	77.2
embedding (4 epochs)	**67.4**	**77.2**
pooled embedding (2 epochs)	66.7	76.6
pooled embedding (3 epochs)	66.9	76.8

Table 8. Preliminary Results with Fine-Tuning

This further results, sadly, are still barely above the baseline, far away from our expectation. The reason behind this could be that the vocabulary of Bert and T5 are different and the way they process the tokens are different (tokenizor). In this study, we didn't make it such that the pre-process and other setting of these two models all the same, this may also lead to difficulty for Bert to understand T5's embedding.

6 Conclusion and Future Work

In this work, we did a fair number of experiment to investigate the performance of different Bert setting on conversational question answering.

We first analysis the prediction distribution of standard Bert model and found that in general the first round F1 score is higher than the rest round of conversation. Inspired by this, we plan to inject the rewritten question from a T5 (generative) model to help transfer the conversational QA setting into a normal QA setting and improve the overall scores by 5-10 percents.

To do this, we proposed two different methods. The first is to combine the rewritten question from T5 with original input to Bert (as in Figure 2), and the other is to extract the output embedding from a T5 encoder and combine it with embedding of Bert output and send them concatenated to some MLP layers (as in Figure 4).

However, through extensive experiments (much more failed didn't show here), we show that we didn't gain expected improvement. The most possible reason behind could be that the power of T5 is limited and the output from T5 is not always helpful.

In the future, we still believe there is a way to improve the conversational QA performance (like joint training two models) and hope this conversational to one-turn transformation can benefit all conversational QA tasks.

Due to the limit of time and human resource, we did all the experiments on only CoQA dataset and didn't explore the dataset as much as we want. We would like to encourage others to fine-tune this model and try it also on QuCA and give more comprehensive analysis on both the dataset itself and the predictions given by the model.

7 Acknowledgement

We thank prof. Sachan for offer this course and the chance to do this project (Also I have to thank him for keep my grade from last year, which make this project possible.). We also thank the TAs, especially Shehzaad for his insightful discussion and suggestion.

References

[1] Vaibhav Adlakha, Shehzaad Dhuliawala, Kaheer Suleman, Harm de Vries, and Siva Reddy. 2021. TopiOCQA: Open-domain Conversational Question Answering with Topic Switching. https://doi.org/10.48550/ARXIV.2110.00768

[2] Raviteja Anantha, Svitlana Vakulenko, Zhucheng Tu, Shayne Longpre, Stephen Pulman, and Srinivas Chappidi. 2021. Open-Domain Question Answering Goes Conversational via Question Rewriting. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (2021).

[3] Danqi Chen and Wen-tau Yih. 2020. Open-Domain Question Answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*. Association for Computational Linguistics, Online, 34–37. https://doi.org/10.18653/v1/2020.acl-tutorials.8

[4] Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC : Question Answering in Context. arXiv:1808.07036 [cs.CL]

[5] Philipp Cimiano, Christina Unger, and John McCrae. 2014. *Ontology-Based Interpretation of Natural Language*. https://doi.org/10.2200/S00561ED1V01Y201401HLT024

[6] Lei Cui, Shaohan Huang, Furu Wei, Chuanqi Tan, Chaoqun Duan, and Ming Zhou. 2017. SuperAgent: A Customer Service Chatbot for E-commerce Websites. In *Proceedings of ACL 2017, System Demonstrations*. Association for Computational Linguistics, Vancouver, Canada, 97–102. https://aclanthology.org/P17-4017

[7] 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, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[8] Ahmed Elgohary, Denis Peskov, and Jordan Boyd-Graber. 2019. Can You Unpack That? Learning to Rewrite Questions-in-Context. In *Empirical Methods in Natural Language Processing* (Hong Kong, China). http://umiacs.umd.edu/~jbg//docs/2019_emnlp_sequentialqa.pdf

[9] 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, Volume 2 (Short Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 687–692. https://doi.org/10.18653/v1/N18-2108

[10] Jianmo Ni, Gustavo Hernández Ábrego, Noah Constant, Ji Ma, Keith B Hall, Daniel Cer, and Yinfei Yang. 2021. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. *arXiv preprint arXiv:2108.08877* (2021).

[11] Minghui Qiu, Feng-Lin Li, Siyu Wang, Xing Gao, Yan Chen, Weipeng Zhao, Haiqing Chen, Jun Huang, and Wei Chu. 2017. AliMe Chat: A Sequence to Sequence and Rerank based Chatbot Engine. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Vancouver, Canada, 498–503. https://doi.org/10.18653/v1/P17-2079

[12] 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*. 1133–1136.

[13] Gwenole Lecorve Quentin Brabant and Lina Rojas-Barahona. 2022. CoQAR Question Rewriting on CoQA. To be published in LREC2022. https://github.com/Orange-OpenSource/COQAR

[14] 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. http://jmlr.org/papers/v21/20-074.html

[15] Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. https://doi.org/10.48550/ARXIV.1806.03822

[16] 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.

[17] Bayan Shawar and Eric Atwell. 2007. Chatbots: Are they Really Useful? *LDV Forum* 22 (01 2007), 29–49.

[18] 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* (Virtual Event, Israel) *(WSDM '21)*. Association for Computing Machinery, New York, NY, USA, 355–363. https://doi.org/10.1145/3437963.3441748

[19] Svitlana Vakulenko, Nikos Voskarides, Zhucheng Tu, and Shayne Longpre. 2021. A Comparison of Question Rewriting Methods for Conversational Passage Retrieval. https://arxiv.org/pdf/2101.07382.pdf

[20] Nikos Voskarides, Dan Li, Pengjie Ren, Evangelos Kanoulas, and Maarten de Rijke. 2020. *Query Resolution for Conversational Search with Limited Supervision*. Association for Computing Machinery, New York, NY, USA, 921–930.

[21] Joseph Weizenbaum. 1966. ELIZA—a Computer Program for the Study of Natural Language Communication between Man and Machine. *Commun. ACM* 9, 1 (Jan. 1966), 36–45. https://doi.org/10.1145/365153.365168

[22] Mark Yatskar. 2018. A qualitative comparison of coqa, squad 2.0 and quac. *arXiv preprint arXiv:1809.10735* (2018).

[23] Munazza Zaib, Wei Emma Zhang, Quan Z Sheng, Adnan Mahmood, and Yang Zhang. 2021. Conversational question answering: A survey. *arXiv preprint arXiv:2106.00874* (2021).