From Rewriting to Remembering: Common Ground for Conversational QA Models

Anonymous ACL submission

Abstract

In conversational QA, models have to leverage information in previous turns to answer upcoming questions. Current approaches, such as Question Rewriting, struggle to extract relevant information as the conversation unwinds. We introduce the Common Ground (CG), an approach to accumulate conversational information as it emerges and select the relevant information at every turn. We show that CG offers a more efficient and human-like way to exploit conversational information compared to existing approaches, leading to improvements on Open Domain Conversational QA.

1 Introduction

Speakers involved in a conversation continuously share new information, and build on it to achieve their communicative goals. In human communication, this process takes place effortlessly. As QA systems become conversational, efforts were made to make them able to mimic human behaviour, and to interpret the question at a turn in a conversation, based on the information in the previous turns. An approach to this task is to concatenate the previous turns to the current question (Christmann et al., 2019; Ju et al., 2019; Qu et al., 2019b). The approach has a main shortcoming, namely, it introduces a great amount of noise, since not everything in the previous turns is relevant. An alternative approach is Question Rewriting (QR), in which the question is rewritten in a self-contained form based on the previous conversational information (Vakulenko et al., 2021a; Anantha et al., 2020). QR selects only the relevant information in previous turns, thus improving over concatenation. However, as the conversation progresses and the amount of information grows, QR models often fail to compress it in a rewrite. We argue that this is not only a limitation of the models, but an intrinsic limit of this approach, since producing informative rewrites is often unnatural also for humans (see Section 4).

In this work, we address the shortcomings above. Inspired by the studies of Clark (1996), we propose a methodology to represent conversational information as a set of propositions, named the *Common Ground* (CG): At each turn, the relevant information is distilled in one or more propositions, which are added to the CG. As a new question comes in, the model selects the relevant information in the CG, and uses it to answer the question. The CG can thus be considered as an *optimized* summary, which returns the relevant information at every turn while keeping all the information discussed so far.

We use the QReCC dataset (Anantha et al., 2020) to test CG on the task of Open-Domain Conversational QA (ODCQA) - in which answers to questions in a conversation have to be found in a large collection of documents - and show that it improves over existing approaches for modelling conversational information. We show that this is due to the fact that CG implements a more efficient and human-like way to account for previous information, which takes the best of existing approaches while avoiding their shortcomings: on the one hand, CG can access and maintain the full previous conversational context, but it avoids the noise issue; on the other, it can distill relevant information, but it is not forced to compress it in a single rewrite.

2 Common Ground

We now detail how we created a dataset for CG, and the model we implemented to generate the CG.

2.1 Building the CG

We devise the CG as a set of propositions summarizing the information in a conversation. Since no dataset annotated for CG is available for QA, we created it. We use QReCC (Anantha et al., 2020), a dataset for QR consisting in a set of conversations. For each turn in a conversation, the original question q and its rewrite r are provided. Intuitively, the rewrite makes explicit the entities discussed in the

conversation. If q is self-contained, then q=r. We define a proposition in the CG as any sequence of words in the rewrite which are nouns, adjectives or entities.[1] For example, given q_1 'how old is Messi?', the rewrite r_1 is equal to q_1, and CG_1 is {'*Messi*'}. Given q_2 'which position does he play?', r_2 is 'which position does Messi play?' and CG_2 is {'*Messi*', '*position*'}. We use this approach to enrich each turn in QReCC with the gold CG.

Importantly, ∼70% of the conversations in QReCC were collected by showing the speaker the title and first sentence of a Wikipedia article (Anantha et al., 2020). This information is often crucial to understand a question, especially at turn 1 (e.g., title: 'Albert Camus', q_1: 'When was he born?'), but, potentially, also at subsequent turns (q_2: 'What did he write?'). We therefore collect the relevant Wikipedia information (which we call *doc*), and use it to further enrich QReCC conversations.[2] Note that *doc* is the same at every turn in the conversation. We refer to the union of conversational and Wikipedia information as *contextual* information. Finally, since QReCC only includes train and test split, we randomly sample 20% of the train and use it as validation set.

2.2 Predicting the CG

We introduce a model to produce the CG, which consists of two modules: *Generator* and *Selector*.

Generator At turn t_n, the Generator is trained to generate the gold CG CG_n given $doc \| conv_{[0:n-1]} \| q_n$, where $\|$ indicates concatenation, *doc* is the information from Wikipedia, $conv_{[0:n-1]}$ is the concatenation of questions and answers from turn t_0 to t_{n-1}, and q_n is the current question. We implement the Generator using a T5-base model.[3] We train the generator using the enriched QReCC.

Selector The propositions returned by the Generator for every turn are stacked in the CG. However, as the conversion goes on, some of the propositions are no longer relevant. The role of the Selector is to select only the relevant propositions in the CG.

We implement the Selector as a binary classifier. To create the data to train the model, we use again QReCC: given the full CG available at turn n, we label as 1 the propositions in it that occur in the gold answer span, 0 otherwise. The rationale behind this approach is: an item in the CG is relevant if it is mentioned in the answer. We train the model on the QReCC train split. At test time, we label the propositions in the CG, and keep only those labelled as 1. Figure 1 shows an example of CG.

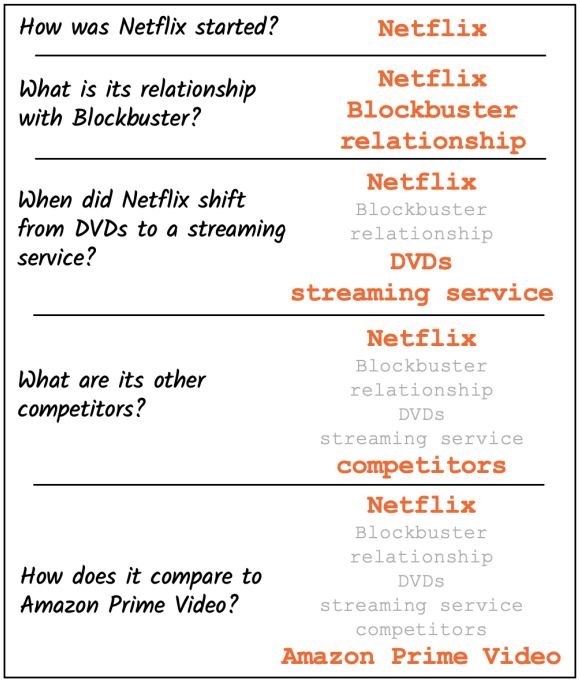

Figure 1: On the left, the questions from the user; on the right, the CG generated by the Generator: highlighted the propositions selected by the Selector at each turn, in grey those kept in the CG but not selected.

3 Experiments

The goal of accounting for contextual information is to improve the performance on a downstream task. Hence, we compare CG to existing approaches on the task of ODCQA.

Data We use again QReCC, as it meets the requirements of the task: it is conversational, and it allows to experiment in an Open-Domain scenario.

Pipeline We use a retriever-reader pipeline. The retriever returns the top n most relevant candidates from the set of documents; these are passed to the reader, which extracts the final answer. We use BERTserini (Yang et al., 2019), using BM25 as a retriever and a BERT-Large as a reader. Each candidate returned by the retriever has a score s_{ret}; the answer extracted from that candidate by the reader has a score s_{rea}. The final score s for the answer is defined as: $(1-\mu) \cdot s_{ret} + \mu \cdot s_{rea}$.

For the retriever, we set n to 20, and we follow Anantha et al. (2020) in setting k_1=0.82 and b=0.68. We tune the value of μ on the validation set inde-

[1] Identified using Spacy: https://spacy.io/.
[2] The details about the enriched dataset are in Appendix A.
[3] The details of Generator and Selector are in Appendix B.

pendently for each approach (see Section 3.1). We do not finetune the reader, as we want to assess how much the CG can directly benefit any QA model, without the need to finetune it.

3.1 Setups

We test the pipeline's performance when provided, at turn n, with each of the following inputs:
original: the original question q_n.
concat.: the concatenation $doc \parallel conv_{n-1} \parallel q_n$.[4]
rewrite: the rewrite r_n produced with a T5-base model. The model generates the rewrite based on $doc \parallel conv_{[0:n-1]} \parallel q_n$.
summary: the concatenation $summ_{[0:n-1]} \parallel q_n$, where $summ_{[0:n-1]}$ is the summary of $doc \parallel conv_{[0:n-1]}$, created with a T5-base model pre-trained for summarization (Raffel et al., 2019).[5]
CG: The CG predicted using our approach, concatenated with the current question: $CG_n \parallel q_n$.
CG-full: The full CG generated up to turn n, i.e., we do not use the Selector module: $CG_n\text{-}full \parallel q_n$.

4 Results and Analysis

We show the results of our experiments in Table 1. We measure the performance on the target task in terms of F1, and use MRR and Recall@10/20 to assess the performance of the retriever.[6] We also report the results obtained with gold (-g) rewrites and CG, where the latter is defined, at turn n, as gold $CG\text{-}full_n$ for the retriever and gold CG_n for the reader - i.e., the best combination observed in our experiments (see below).

As expected, approaches leveraging contextual information improve over the original question. Among these approaches, CG is the best: it improves the performance over rewrite, and, remarkably, it matches the results obtained with *gold* rewrites. A further improvement in F1 is observed when using CG-full at the retriever and CG at the reader (CG-full/CG), while using only CG-full degrades the performance. This shows that using the more informative but potentially noisier CG-full improves retrieval, but one needs to feed the filtered information from CG to the reader to see improvements in F1, as also observed by Del Tredici et al. (2021). The different response to noise also explains the results of concatenation, which obtain high performance in retrieval, but drops in F1.

Approach	F1	MRR	R@10	R@20
original	6.23	2.89	5.56	6.65
concat.	8.95	21.67	37.55	41.51
rewrite	12.46	13.73	24.52	28.6
summary	12.02	21.81	34.72	38.33
CG	13.41	15.66	27.67	32.09
CG-full	12.18	16.52	29.47	34.06
CG-full/CG	14.2	16.52	29.47	34.06
rewrite-g	13.42	17.16	29.07	33.26
CG-g	15.17	17.95	31.18	35.65

Table 1: Results on the QReCC test set. CG-full/CG indicates that we used CG-full for the retriever and CG for the reader.

CG vs. QR In Table 2, we show examples from QR and CG. In row 1, both approaches extract the relevant information from the previous turns - in a conversation about physician assistants. In the next turn (2), QR fails to expand the question and to substitute 'about' with the contextual information, due to the large amount of information required ('the average starting salary for a physician's assistant in the US'). We often observe this limitation for the QR model. This is not the case for CG, since here the information grows *incrementally*, i.e., the information from the current turn ('the US') is added *on top* of the one already present, while non relevant information ('the UK') is discarded.

In the previous case, the QR model fails to produce a rewrite; in others, this is just not possible. In the 6th turn of a conversation about different kinds of data network architectures (row 3), the user asks a general question about flaw types which encompasses all the previous information: there is so much information to compress, here, that *not even* humans manage to do it, and the gold rewrite is the same as the original question.[7] CG sidesteps this problem simply by making available all the pieces of relevant information emerged in the conversation, which can be selected and exploited by the model, without the need to produce a long natural sentence. Note that besides being more effective, this solution is also more human-like: Speakers do not *repeat* all the contextual information as they

[4] Note that we use $conv_{n-1}$, and not $conv_{[0:n-1]}$, due to the max length limit of the reader of BERTserini.

[5] The details of the Rewrite and Summarization models are in Appendix C.

[6] We use the code by QReCC authors: `github.com/apple/ml-qrecc/tree/main/utils`.

[7] We provide in Appendix D the whole conversation, plus additional examples of (nearly) impossible rewrites.

	Original Question	Question Rewriting	Common Ground
1	What's the average starting salary in the UK?	What's the average starting salary for a physician assistant in the UK?	{*the average starting salary*, *the UK*, *a physician assistant* }
2	What about in the US?	What about in the US?	{*the average starting salary*, *the US*, *a physician assistant* }
3	Are flows bidirectional?	<u>Are flows bidirectional?</u>	{*data network architectures*, *edge switches*, *bidirectional flows*, *FAT tree topology*, *upstream packet*, *routes*, *core*, *aggregator*}

Table 2: Examples of rewrites and CG. Predicted rewrites are in plain text, gold rewrites underlined.

make a question, but, rather, they *remember* the key points of the conversation.

CG vs. Summary Summaries convey all contextual information, which makes them suitable for the retriever, but not for the reader. CG is superior because, as said above, is an *optimized* summary conditioned on the current question. In fact, when we create the CG without considering the current question, the model cannot identify the relevant information, and the results are comparable to those of summary (F1=12.6). For example, for the question 'where did he come from?', the CG predicted in the normal scenario is {*Rick Barry*}, while, without the current question, is {*the ABA*, *free-throw percentage*, *the 1968–69 season*, *Rick Barry*}.

Conv* vs. *Doc We measure the performance for the best setup (CG-full/CG) when the CG is created considering either *doc* or *conv*: with the former, the F1 is 13.38, with the latter 13.65. The decrease in performance of *doc* and *conv* compared to *doc*+*conv* indicates that considering multiple source of information is beneficial for the overall performance of the model. Also, the fact that *conv* yields better results than *doc* is expected: in QReCC, the information from *doc* is mostly leveraged at the first turn, while the information from *conv* is relevant throughout the full conversation.

5 Related Work

Approaches to modelling conversational information have used either sparse or dense representation (Qu et al., 2019a,b, 2020). This work focuses on the former. In this group, concatenation was proposed as an initial approach (Christmann et al., 2019; Ju et al., 2019; Qu et al., 2019b), followed by Question Rewriting (Elgohary et al., 2019). The main models for QR are either generative (Vakulenko et al., 2021a; Yu et al., 2020) or extractive one (Voskarides et al., 2020) - i.e., the relevant tokens in the context are appended to the question. When a single model is used for both retriever and reader, generative model overperform extractive ones (Vakulenko et al., 2021b); however, mixing the two approaches further improves the performance (Del Tredici et al., 2021). Our work is related to (Voskarides et al., 2020), as we also aim at extracting the relevant contextual information. However, instead of appending this information to the question, we stack it in the CG, and enable the model to pick the relevant information at each turn.

6 Conclusions

We introduced the Common Ground, a novel approach for leveraging contextual information. We show that CG outperforms the main existing approaches in the ODCQA task, due to its ability to select and maintain the relevant information in a more effective and human-like way.

We see two main directions for future research on CG. First, we will exploit the ability of CG to include several kinds of information to make it more informative. For example, to answer the question 'how many Covid cases today?', a QA system needs to be aware of the *time* and *location* of the person asking it (Zhang and Choi, 2021). We want to include these and other information in the CG. Second, we want to use CG to make QA models more transparent. Currently, virtual assistants (such as Alexa, Siri and Google Assistant) are black boxes, i.e, the user does not know which information they extract from the input question, and which one they leverage to provide answers. This can make the interaction with them frustrating. CG offers a solution to the problem, as it allows to see what the assistant *has in mind* at each conversational turn. We will conduct experiments in which the CG is shared with the user, and see how this can make the interaction with the assistant more engaging and successful.

References

Raviteja Anantha, Svitlana Vakulenko, Zhucheng Tu, Shayne Longpre, Stephen Pulman, and Srinivas Chappidi. 2020. Open-domain question answering goes conversational via question rewriting. *arXiv preprint arXiv:2010.04898*.

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*, pages 729–738.

Herbert H Clark. 1996. *Using language*. Cambridge university press.

Marco Del Tredici, Gianni Barlacchi, Xiaoyu Shen, Weiwei Cheng, and Adrià de Gispert. 2021. Question rewriting for open-domain conversational qa: Best practices and limitations. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21)*.

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*, pages 5918–5924, Hong Kong, China. ACL.

Ying Ju, Fubang Zhao, Shijie Chen, Bowen Zheng, Xuefeng Yang, and Yunfeng Liu. 2019. Technical report on conversational question answering. *arXiv preprint arXiv:1909.10772*.

Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W.B̃ruce Croft, and Mohit Iyyer. 2020. Open-retrieval conversational question answering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 539–548, Xi'an, China. ACM.

Chen Qu, Liu Yang, Minghui Qiu, W. Bruce Croft, Yongfeng Zhang, and Mohit Iyyer. 2019a. 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*, page 1133–1136, Paris, France. ACM.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Svitlana Vakulenko, Shayne Longpre, Zhucheng Tu, and Raviteja Anantha. 2021a. Question rewriting for conversational question answering. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 355–363.

Svitlana Vakulenko, Nikos Voskarides, Zhucheng Tu, and Shayne Longpre. 2021b. A comparison of question rewriting methods for conversational passage retrieval. *CoRR*, arXiv:2101.07382. ECIR short paper.

Nikos Voskarides, Dan Li, Pengjie Ren, Evangelos Kanoulas, and Maarten de Rijke. 2020. Query resolution for conversational search with limited supervision. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 921–930, Xi'an, China. ACM.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with bertserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 72–77.

Shi Yu, Jiahua Liu, Jingqin Yang, Chenyan Xiong, Paul Bennett, Jianfeng Gao, and Zhiyuan Liu. 2020. Few-shot generative conversational query rewriting. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1933–1936, Xi'an, China. ACM.

Michael Zhang and Eunsol Choi. 2021. Situatedqa: Incorporating extra-linguistic contexts into qa. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7371–7387.

A Enriching QReCC

Approx. 78% of the conversations in QReCC are derived from the QuAC dataset (https://quac.ai/). In QuAC, dialogues are created by showing to the student (i.e., the person making questions) the title of the section of a Wikipedia page and the first sentence of first paragraph in the page. We retrieve this information from the QuAC dataset, and add it to the QReCC dataset. As mentioned in the main paper, we add the information from Wikipedia to all the turns in a conversations. As a results, 76.5% of the datapoints in the train split and 71.8% of those in the test split have additional information. We will release the code for enriching QReCC with CG and Wikipedia information upon publication.

B Model for CG prediction

Generator In order to generate the CG, we use the T5-base model available at: https://huggingface.co/transformers/model_doc/t5.html.

We fine-tuned the model on the task of generating the CG with the following parameters: max source length= 512; max target length= 64; val max target length= 64; evaluation strategy= steps; num train epochs= 5; per device train batch size= 4; per device eval batch size= 8; eval steps= 82; seed= 42; warmup steps= 500; eval beams= 5; learning rate= 5e-5.

Selector In order to select the relevant propositions in the CG, we use the DistilBert model available at: https://huggingface.co/transformers/model_doc/distilbert.html.

We fine-tuned the model with the following parameters: max source length= 512; evaluation strategy= steps; num train epochs= 5; per device train batch size= 16; per device eval batch size= 64; eval steps= 82; seed= 42; warmup steps= 0; learning rate= 5e-5.

C Generative models for QR and Summarization

QR model In order to generate the rewrites, we use the same T5-base model used to implement the Generator. We fine-tuned the model on the QR task using the QReCC train split, with the same parameters reported in Appendix B.

Summarization model In order to generate the summaries, we use again the same T5-base model used for the Generator and the QR model. In this case, however, we do not need to fine-tuned the model, since it was already optimized for the task: to generate the summaries, we simply provide to the model as input the string 'summarize: ' followed by the contextual information.

D Example of conversation

We report below the full conversation up to the question used as an example in Table 2, row 3.

q_1: What are scalable data center network architectures?
a_1: DCNs need to be scalable and efficient to connect tens or even hundreds of thousands of servers to handle the growing demands of Cloud computing.
q_2: What are some examples of scalable data center network architectures?
a_2: 1 Three-tier DCN 2 Fat tree DCN 3 DCell
q_3: Describe the characteristics of FAT tree topology
a_3: In a fat tree, branches nearer the top of the hierarchy are fatter (thicker) than branches further down the hierarchy. In a telecommunications network, the branches are data links; the varied thickness (bandwidth) of the data links allows for more efficient and technology-specific use.
q_4: What routes can be taken by an upstream packet?
a_4: The router is upstream of the computer, connecting the computer to the whole internet. ... Each router does not need to know the whole route to the destination;
q_5: Describe core, aggregator and edge switches.
a_5: In small networks of a few hundred users, edge switches can be connected redundantly directly to core switch/router devices. However, for larger networks, , an additional layer of switching, called the distribution layer, aggregates the edge switches.

In Table 3, we report examples for which the gold rewrite provided in the QReCC dataset is equal to the original question, despite the fact that the question needs contextual information to be correctly understood. For each example, we provide the information in the CG, and a comment about why creating a rewrite is not possible, or very unnatural. Due to space reasons, we do not report the full conversation. However, we report the conversation and turns IDs, which can be used to look up for the full conversation in the QReCC dataset available at https://github.com/apple/ml-qrecc/tree/main/dataset.

17-10	**Question**: What form of energy is used in eating? **Common Ground**: *energy, light energy, heat energy, gravitational energy, form, type, motion, mechanical energy, examples, potential energy, electrical energy, sound energy, chemical energy, nuclear energy, atomic energy, kinetic energy* **Comment**: the question comes at the end of a long conversation, and refers to the previously mentioned forms of energy. The hypothetical QR should include them all: *What form of energy, among light energy, heat energy, [..] is used in eating?*
22-9	**Question**: What is the oldest spice? **Common Ground**: *spices, cumin, world, coriander, cilantro, herb, garlic, oregano, root, stem, seed, fruit, flower, bark, tree, plant, Indian, pepper, Nutmeg, mace, Mustard, seeds, Fenugreek, Turmeric, Saffron* **Comment**: similarly to the previous example, the question comes at the end of a long conversation, and refers to all previous information. The hypothetical QR should be: *What is the oldest spice among cumin, coriander [...]?*
28-4	**Question**: What can I do as an individual level? **Common Ground**: *global warming, long-term rise, average temperature, Earth's climate system, climate change, temperature measurements, dangers, scientists, sea ice, sea level rise, heat waves, methods, Carbon dioxide, oil, coal, fossil fuels, energy, homes, cars, smartphones* **Comment**: again, the user's question encompasses all previous conversation, in which several problems related to global warming were mentioned. A (tentative) rewrite which captures the information up to this point should therefore be of the kind: *What can I do in order to better use energy for my home, car, smartphone, thus reducing the emission of carbon dioxide and reduce impact on global warming?*
583-6	**Question**: Was there anyone opposed to him in this? **Common Ground**: *Ira Hayes, World War II, civilian life, war, family, 1946, Gila River Indian Community, Edward Harlon Block, Hank Hansen, flag-raiser controversy, Marine Corps* **Comment**: in this dialogue, many facts about Ira Hayes are explained. The original question refers to several of them, and a (very tentative) rewrite should be like: *Was there anyone opposed to Ira Hayes in revealing the truth that Harlon Block was still being misrepresented publicly as Hank Hansen?*
590-6	**Question**: What was the impact of this column? **Common Ground**: *Israel, Krauthammer, Oslo accords, 2006 Lebanon War, column, Let Israel Win the War* **Comment**: also in this case, the conversation touches upon several related facts, and in order to correctly interpret the question in the light of such facts, it should be rewritten like: *What was the impact of the column 'Let Israel Win the War' written by Krauthammer during the 2006 Lebanon War, in which he opposes the Oslo accords?*

Table 3: Examples in which the rewrite is nearly impossible or very unnatural. In the left column we report the conversation-turn IDs.