Can You Unpack That?
Learning to Rewrite Questions-in-Context
Ahmed Elgohary, Denis Peskov, Jordan Boyd-Graber
Department of Computer Science, UMIACS, iSchool, Language Science Center
University of Maryland, College Park
{elgohary, dpeskov, jbg}@cs.umd.edu

Abstract

Question answering is an AI-complete problem, but existing datasets lack key elements of language understanding such as coreference and ellipsis resolution. We consider sequential question answering: multiple questions are asked one-by-one in a conversation between a questioner and an answerer. Answering these questions is only possible through understanding the conversation history. We introduce the task of question-in-context rewriting: given the context of a conversation’s history, rewrite a context-dependent into a self-contained question with the same answer. We construct, CANARD, a dataset of 40,527 questions based on QUAC (Choi et al., 2018) and train Seq2Seq models for incorporating context into standalone questions.

1 Introduction

Question Answering (QA) is an AI complete problem (Webber, 1992), but existing QA datasets do not rise to the challenge: they lack key NLP problems like anaphora resolution, coreference disambiguation, and ellipsis resolution. The logic needed to answer these types of questions requires deeper NLP understanding that simulates the context in which humans naturally answer questions.

Neural techniques question answering have improved (Devlin et al., 2018) machine reading comprehension (Rajpurkar et al., 2016, MRC): computers can take a single question and extract answers from datasets like Wikipedia. However, QA models struggle to generalize when questions do not look like the standalone questions systems in training data: e.g., new genres, languages, or closely-related tasks (Yogatama et al., 2019).

Conversational question answering (Reddy et al., 2019, CQA) is a generalization that ask multiple questions in an information-seeking dialogs. Unlike MRC, CQA requires models to link questions together to resolve the conversational dependencies between them: each question needs to be understood in the conversation context. For example, the question “What was he like in that episode?” cannot be understood without knowing what “he” and “that episode” refer to, which can be resolved using the conversation context.

We reduce challenging, interconnected CQA examples to independent, stand-alone MRC to create CANARD—Context Abstraction: Necessary Additional Rewritten Discourse—a new dataset that rewrites QUAC (Choi et al., 2018) questions. We crowdsource context-independent paraphrases of QUAC questions and use the paraphrases to train and evaluate question-in-context rewriting.

1http://canard.qanta.org
Section 2 formally defines the task of question de-contextualization. Section 3 constructs CA-NARD, a new dataset of question-in-context with corresponding context-independent paraphrases. Section 5 analyzes our rewrites (and the underlying methodology) to understand the linguistic phenomena that make CQA difficult. We build several baseline rewriting models and compare their BLEU scores to our human rewrites in Section 4.

2 Defining Question-In-Context Rewrites

We formally define the task of question-in-context rewriting (de-contextualization). Given a conversation topic \( t \) and a history \( H \) of \( m - 1 \) turns, each turn \( k \) is a question \( q_k \) and an answer \( a_k \); the task is to generate a rewrite \( q'_m \) for the next question \( q_m \) based on \( H \). Since \( q_m \) is part of the conversation, its meaning often involves references to parts of its preceding history. A valid rewrite \( q'_m \) should be self-contained: a correct answer to \( q'_m \) by itself is a correct answer to \( q_m \) combined with the question’s preceding history \( H \).

Figure 1 shows the assumptions of CQA and how they are made explicit in rewrites. The first question omits the title of the page (Anna Vissi), the second question omits the answer to the first question (replacing both Anna Vissi and her husband with the pronoun “they”), and the last question adds a scalar implicature that must be resolved.

3 Dataset Construction

We elicit paraphrases from human crowdworkers to make previously context-dependent questions unambiguously answerable. Through this process, we resolve difficult coreference linkages and create a pair-wise mapping between ambiguous and context-enriched questions. We derive CANARD from QUAC (Choi et al., 2018), a sequential question answering dataset about specific Wikipedia sections. QUAC uses a pair of workers—a “student” and a “teacher”—to ask and respond to questions. The “student” asks questions about a topic based on only the title of the Wikipedia article and the title of the target section. The “teacher” has access to the full Wikipedia section and provides answers by selecting text that answers the question. With this methodology, QUAC gathers 98k questions across 13,594 conversations. We take their entire dev set and a sample of their train set and create a custom JavaScript task in Mechanical Turk that allows workers to rewrite these questions. JavaScript hints help train the users and provides automated, real-time feedback.

We provide workers with a comprehensive set of instructions and task examples. We ask them to rewrite the questions in natural sounding English while preserving the sentence structure of the original question. We discourage workers from introducing new words that are unmentioned in the previous utterances and ask them to copy phrases when appropriate from the original question. These instructions ensure that the rewrites only resolve conversation-dependent ambiguities. Thus, we encourage workers to create minimal edits; in Section 5.2, we take advantage of this to use BLEU for evaluating model-generated rewrites.

We display the questions in the conversation one at a time, since the rewrites should include only the previous utterance. After a rewrite to the question is submitted, the answer to the question is displayed. The next question is then displayed. This repeats until the end of the conversation. The full set of instructions and the data collection interface are provided in the appendix.

We apply quality control throughout our collection process. During the task, JavaScript checks automatically monitor and warn about common errors: submissions that are abnormally short (e.g., ‘why’), rewrites that still have pronouns (e.g., ‘he wrote this album’), or ambiguous words (e.g., ‘this article’, ‘that’). Many QUAC questions ask about ‘what/who else’ or ask for ‘other’ or ‘another’ entity. For that class of questions, we ask workers to use a phrase such as ‘other than’, ‘in addition to’, ‘side from’, ‘besides’, ‘together with’ or ‘along with’ with the appropriate context in their rewrite.

We gather and review our data in batches to screen potentially compromised data or low quality workers. A post-processing script flags suspicious
Was this an honest mistake by the media?

Was the claim of media regarding Leblanc’s room come to true?

What was a single from their album?

What was a single from horslips’ album?

Did they marry?

Did Hannah Arendt and Heidegger marry?

Table 2: Not all rewrites correctly encode the context required to answer a question. We take two failures to provide examples of the two common issues: Changed Meaning (top) and Needs Context (middle). We provide an example with no issues (bottom) for comparison.

Table 3: BLEU scores of the baseline models on development and test data. Seq2Seq improves up to four points over naive baselines but still well below human accuracy. Human accuracy (*) is computed from a small subset of the validation set.

4 Baselines

We compare three baseline models for the question-in-context rewriting task. In the Copy baseline, the rewrite $q'_m$ is set to be the same as the input question $q_m$ without making any changes.

We also try a Pronoun Substitution baseline in which the first pronoun in $q_m$ is replaced with the topic entity of the conversation. We use the title of the corresponding Wikipedia article as the topic entity. Similar to the Copy baseline, the training data is not used in that baseline.

Unlike the previous baselines which do not use our rewrites as training data, the third baseline is a neural sequence-to-sequence (Seq2Seq) model with attention and a copy mechanism (Bahdanau et al., 2015; See et al., 2017). We construct the input sequence by concatenating all utterances in the history $H$, prepending them to $q_m$, and adding a special separator token between utterances. We use a bidirectional LSTM encoder-decoder model with shared the word embeddings between the encoder and the decoder.

Since questions are written by humans, a human rewrites are the upper-bound for this task. However, annotators (especially crowdworkers) can be inconsistent or disagree. To estimate the human accuracy, we collect 100 pairs of rewritten questions; each pair has two rewrites of the same question (in its given context) by two different workers. We manually verify that all rewrites are valid and then use the pair of rewrites as a hypothesis and a reference.

Table 3 shows the BLEU scores produced by the baselines and humans over both the validation and the test sets.\(^2\) Although a well-trained standard

<table>
<thead>
<tr>
<th></th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy</td>
<td>33.84</td>
<td>36.25</td>
</tr>
<tr>
<td>Pronoun Sub</td>
<td>47.72</td>
<td>47.44</td>
</tr>
<tr>
<td>Seq2Seq</td>
<td>51.37</td>
<td>49.67</td>
</tr>
<tr>
<td>Human Rewrites*</td>
<td>59.92</td>
<td></td>
</tr>
</tbody>
</table>

\(^2\)We initialize the embeddings with GloVE (Pennington et al., 2014) and train with a batch-size of 16 for 200000 steps. We use OpenNMT (Klein et al., 2018) implementation.

\(^3\)We use multi-bleu-detok.perl (Sennrich et al., 2017)
Figure 2: Human rewrites are longer, have fewer pronouns, and have more proper nouns than the original QUAC questions. Rewrites are longer and contain more proper nouns than our Pronoun Sub baseline and trained Seq2Seq model.

neural sequence-to-sequence improves 2–4 BLEU points over naive baselines, it is still 9 BLEU points below human-accuracy. We analyze sources of errors in the following section.

5 Dataset and Model Analysis

We analyze our dataset with automatic metrics after validating the reliability of our data (Section 3). We compare our dataset to the original QUAC questions and to automatically generated questions by our models. Then, we manually inspect the sources of rewriting errors in the seq2seq baseline.

5.1 Anaphora Resolution and Coreference

Our rewrites are longer, contain more nouns and less pronouns, and have more word types than the original data. Machine output lies in between the two human-generated corpora, but quality is difficult to assess. Figure 2 shows these statistics. We motivate our rewrites by exploring linguistic properties of our data. Anaphora resolution and coreference are two core NLP tasks applicable to this dataset, in addition to the downstream tasks evaluated in Section 4.

Pronouns occur in 53.9% of QUAC questions. Questions with pronouns are more likely to be ambiguous than those without any. Only 0.9% of these have pronouns that span more than one category (e.g., ‘she’ and ‘his’). Hence, pronouns within a single sentence are likely unambiguous. However, 75.0% of the aggregate history has pronouns and the percentage of mixed category pronouns increase to 27.8% of our data. Therefore, pronoun disambiguation potentially becomes a problem for a quarter of the original data. An example is provided in Table 4.

Approximately one-third of the questions generated by our pronoun-replacement baseline are within 85% string similarity to our rewritten questions. That leaves two-thirds of our data that cannot be solved with pronoun resolution alone.

5.2 Model Analysis

By manually examining the predictions of the seq2seq model, we notice that the main source of errors is that the model tends to find a short path to completing the rewrites. That often results in under-specified questions as in Example 1 in Table 5, question meaning change as in Example 2 or meaningless questions as in Example 3.

Another source of errors is having related entities mentioned in the context as Example 4 in Table 5, where the model confused “Copa America” with “Argentina”. The model also struggles with listing multiple entities mentioned in different parts of the context. Example 5 in Table 5 show the output and the reference rewrites of the question “Did she have any more works than those 3?”, where two of the three entities—“United States of Banana”, “La Comedia” and “Asalto al tiempo”—are lost in the rewrite.
Table 5: Example erroneous rewrites generated by the Seq2Seq models and their corresponding reference rewrites. The dominant source of error is the model tendency to produce short rewrites (Examples 1–3). Related entities (Copa America and Argentina in Example 4) distract the model. The model struggles with listing multiple entities mentioned in different parts of the context (Example 5).

6 Related Work and Discussion

Recent work in CQA has used simple concatenation (Elgohary et al., 2018), sequential neural models (Huang et al., 2019), and transformers (Qu et al., 2019a) for modeling the interaction between the conversation history, the question and reference documents. Some of the components in those models, such as relevant history turn selection (Qu et al., 2019b), can be adopted in question rewriting models for our task. An interesting avenue for future work is to incorporate deeper context, either from other modalities (Das et al., 2017) or from other dialog comprehension tasks (Sun et al., 2019).

Parallel to our work, Rastogi et al. (2019) and Su et al. (2019) introduce utterance rewriting datasets for dialog state tracking. Rastogi et al. (2019) covers a narrow set of domains and the rewrites of Su et al. (2019) are based on Chinese dialog with two-turn fixed histories. In contrast, CANARD has histories of variable turn lengths, covers wider topics, and is based on CQA.

Training question rewriting using reinforcement learning with the task accuracy as reward signal is explored in retrieval-based QA (Liu et al., 2019) and in MRC (Buck et al., 2018). A natural question is whether reinforcement learning could learn to retain the necessary context to rewrite questions in CQA. However, our dataset could be used to pretrain a question rewriter that can further be refined using reinforcement learning.

More broadly, we hope CANARD can drive human-computer collaboration in QA (Feng and Boyd-Graber, 2019). While questions typically vary in difficulty (Sugawara et al., 2018), existing research either introduces new benchmarks of difficult (adversarial) stand-alone questions (Dua et al., 2019; Wallace et al., 2019, inter alia), or models that simplify hard questions through paraphrasing (Dong et al., 2017) or decomposition (Talmor and Berant, 2018). We aim at studying QA models that can ask for human assistance (feedback) when they struggle to answer a question.

The reading comprehension setup of CQA provides a controlled environment where the main source of difficulty is interpreting a question in its context. The interactive component of CQA also provides a natural mechanism for improving rewriting. When the computer cannot understand (rewrite) a question because of complicated context, missing world knowledge, or upstream errors (Peskov et al., 2019) in the course of a conversation, it should be able to ask its interlocutor, “can you unpack that?” This dataset helps start that conversation; the next steps are developing and evaluating models that efficiently decide when to ask for human assistance, and how to best use this assistance.

Acknowledgments

We thank Jo Shoemaker and Michelle Yuan for crowdsourcing feedback. This work was supported by NSF grant IIS-1822494. Boyd-Graber and Peskov were also supported by IIS-1748663. Any opinions, findings, conclusions, or recommendations expressed here are those of the authors and do not necessarily reflect the view of the sponsors.
References


Appendix A: Data Collection Instructions

<table>
<thead>
<tr>
<th>Rewrite each question in a dialogue so it can be understood by itself without looking at the rest of the dialogue.</th>
</tr>
</thead>
<tbody>
<tr>
<td>• In the given conversation between a student who asks questions about a topic of a Wikipedia article and a teacher who is answering the questions, each question can be understood as part of the ongoing dialogue. Edit each question so that it makes sense by itself without looking at the rest of the dialogue. Think how you would ask the same question to someone who has no idea about what the dialogue is about and has never seen any of the previous utterances and still get the same answer. For example, the question ‘What was he like in that episode?’ cannot be understood without knowing what ‘he’ and ‘that episode’ refer to. Both can be figured out by looking at the dialogue. If we edited the question to ‘What was Daffy Duck like in Porky’s Duck Hunt?’, it would then make sense by itself.</td>
</tr>
<tr>
<td>• Reading and understanding all previous utterances is very important. Do not just look for things like ‘she’ and ‘that’ and replace them. For example, the question Who was the star? is still unclear. We need to specify ‘star in what movie, series, episode, ..etc.?’ as in Who was the star in Porky’s Duck Hunt? To make sure the edits are based on understanding the conversations rather than naive replacements, we will manually review a random group of the edits before accepting the work. We also run pretty mild checks on the edits before allowing the HIT to be submitted. The checks are meant to help you make valid edits, but they still can miss some invalid edits (rather than being too strict). So, do not count only on passing the checks and make sure that each edit makes sense by itself.</td>
</tr>
<tr>
<td>• Questions should stay in a natural-sounding english after editing. Do not use new words other than those in the conversation unless it is necessary. Copy from previous questions, answers and conversation topic as often as possible. Never use any information provided after a question is asked. Only use what was provided up to the question you are editing. When editing a question, keep its original structure whenever possible. For example, do not change ‘Did they arrest him?’ to ‘Was Alex arrested?’, but rather keep the original structure and do ‘Did they arrest Alex?’. Do not come up with a different or better question. Stick to the given wording. It is okay to have pronouns in an edit as long as the edit has enough information about what the pronouns refer to. For example ‘Where was Carroll when she joined the american party?’ is a perfectly fine edit.</td>
</tr>
<tr>
<td>• Pay special attention to questions that ask if there is anything/anyone else or asking for telling more about something. It is important to mention in your edit “anything/anyone else besides what/ whom”. For example, ‘Are there any other interesting aspects about this article?’ should be edited to something like ‘Besides Death of a Ladies Man and End of the Century, are there any other interesting aspects about this article?’ as in example 2 below. Notice that it is okay to keep ‘this article’ without replacing it with something else.</td>
</tr>
<tr>
<td>• If a question can be understood without looking at the rest of the conversation, mark it as Understandable alone. Do not make unnecessary edits. Try to keep the edits short as long as they make sense each by itself. Although typos are quite rare, please fix any typos you encounter.</td>
</tr>
<tr>
<td>• Never look up additional information on the web. Only use the given conversations.</td>
</tr>
<tr>
<td>• We show one question at a time. After editing or marking a question as Understandable alone, we show its following question. After all questions are edited, we allow you to submit the HIT and optionally provide any feedback you have on the HIT.</td>
</tr>
</tbody>
</table>
Appendix B: Data Collection Interface

Figure 1: Data collection interface. The conversation has eight questions in total. We show one question at a time to encourage crowdworkers only use the previous utterances for the rewriting. After the eight questions are rewritten, we enable submitting the HIT. We show question-specific instructions as in question three to remind crowdworkers about relevant instructions. A compact set of the full instructions are shown on the right. Detailed instructions can be displayed by clicking on the “view instructions” button.