On the Robustness of Question Rewriting Systems to Questions of Varying Hardness

Hai Ye[1] **Hwee Tou Ng**[1] **Wenjuan Han**[2]
[1]Department of Computer Science, National University of Singapore
[2]Beijing Institute for General Artificial Intelligence (BIGAI), Beijing, China
{yeh,nght}@comp.nus.edu.sg
hanwenjuan@bigai.ai

Abstract

In conversational question answering (CQA), the task of question rewriting (QR) in context aims to rewrite a context-dependent question into an equivalent self-contained question that gives the same answer. In this paper, we are interested in the robustness of a QR system to questions varying in rewriting hardness or difficulty. Since there is a lack of questions classified based on their rewriting hardness, we first propose a heuristic method to automatically classify questions into subsets of varying hardness, by measuring the discrepancy between a question and its rewrite. To find out what makes questions hard or easy for rewriting, we then conduct a human evaluation to annotate the rewriting hardness of questions. Finally, to enhance the robustness of QR systems to questions of varying hardness, we propose a novel learning framework for QR that first trains a QR model independently on each subset of questions of a certain level of hardness, then combines these QR models as one joint model for inference. Experimental results on two datasets show that our framework improves the overall performance compared to the baselines[1].

1 Introduction

In conversational question answering (CQA) (Choi et al., 2018; Reddy et al., 2019), several sequential questions need to be answered one by one given a relevant article. To answer a question in CQA, we need to understand the historical context of the question. For example, to answer the question "*When did he begin writing these pieces?*", we need to know what *he* refers to in the conversation context. In our work, we address question-in-context rewriting (QR), which aims to rewrite a context-dependent question into an equivalent self-contained question in CQA, e.g., replacing *he* in the above example with its referent from the context. The task is formulated as a text generation task that generates the rewrite of a question given the original question and its conversation context (Elgohary et al., 2019).

Topic words: Benigno Aquino III; Senate (2007 - 10)
$\mathbf{q_1}$: What changes did he make while in the Senate?
$\mathbf{a_1}$: I don't know.
$\mathbf{q_2}$: When was *he* elected?
$\rightarrow \mathbf{q_2'}$: When was *Benigno Aquino III* elected to Senate?
$\mathbf{a_2}$: May 15, 2007
$\mathbf{q_3}$: Was he a republican or democrat?
$\mathbf{a_3}$: Genuine Opposition (GO), a coalition comprising a number of parties, including Aquino's own Liberal Party, ...
$\mathbf{q_4}$: Are there any other interesting aspects about this article?
$\rightarrow \mathbf{q_4'}$: Are there any other interesting aspects about Benigno Aquino III article *aside from political affiliation or when Benigno was elected?*
$\mathbf{a_4}$: Aquino was endorsed by the pentecostal Jesus Is Lord Church.

Table 1: One dialogue example from Elgohary et al. (2019) including questions ($\mathbf{q_i}$) and answers ($\mathbf{a_i}$) and certain rewrites ($\mathbf{q_i'}$) of the questions.

We are interested in how robust a QR system is to questions with different rewriting hardness (or difficulty). As we can see from the examples in Table 1, rewriting the question $\mathbf{q_2}$ requires only replacing the pronoun *he* by its referent, which usually appears in the conversation context, and the model can identify the referent by *attention* (Luong et al., 2015). However, for the question $\mathbf{q_4}$, to find the missing *aside from* clause, the model needs to understand the entire conversation since the question asks about other interesting aspects about the article related to the topic of the entire conversation. Understanding the whole context will be challenging for the model. Can a QR model still work well when rewriting the hard questions?

In section 6.3, our first study is on evaluating the performance of a QR model under questions varying in hardness. One issue in this process is

[1]Our source code is available at https://github.com/nusnlp/DiffQRe. This work was done while Wenjuan Han was a research fellow at the National University of Singapore.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 2100 - 2113

that there is a lack of classified questions in different rewriting hardness. Though we can rely on human labor to annotate the questions, it is expensive and not scalable. Instead, we propose a simple yet effective heuristic method to classify the questions automatically. We measure the *discrepancy* between a question and its rewrite, where the larger the discrepancy, the more difficult to rewrite the question. The intuition is that if a question is very dissimilar to its rewrite, more information has to be filled into the rewrite, which means the question is harder to rewrite. We specifically use the BLEU score to measure the discrepancy, and lower scores mean larger discrepancies. Using this method, we then split the questions into three subsets: hard, medium, and easy, and evaluate the baseline systems using these subsets.

In order to verify the classified subsets and find out what makes questions different in rewriting difficulty, in section 6.3.2, we further evaluate the question characteristics in hard, medium, and easy subsets through human evaluation. We first manually summarize the commonly used rules for rewriting questions from the training set, and then annotate the questions using the labels of summarized rewriting rules, followed by counting the number of these rewriting rules used in these subsets.

Finally, to enhance the robustness of a QR model to questions varying in difficulties, we propose a novel learning framework in section 5, where we first separately train a QR model on each hard, medium, and easy subset, and then combine these models into a joint model for inference. Training one sole model on each subset is to let the model better learn domain-specific information to deal with one specific type of questions (hard/medium/easy). By combining the models together, we have a joint model capable of rewriting questions differing in rewriting hardness. Specially, we introduce adapters (Houlsby et al., 2019) to reduce parameters when building private models and we present sequence-level adapter fusion and distillation (SLAF and SLAD) to effectively combine the private models into a joint model.

Our contributions in this paper include:

- We are the *first* to study the robustness of a QR system to questions with varying levels of rewriting hardness;
- We propose an effective method to identify questions of different rewriting hardness;
- We manually annotate questions sampled from the subsets with summarized rewriting rules for validity and address what makes questions hard or easy for rewriting;
- We propose a novel QR framework by taking into account the rewriting hardness.

We have the following observations in our paper:

- The baseline systems perform much worse on the hard subset but perform well on the easy subset;
- We find that easy questions usually only require *replacing pronouns* but hard questions involve more complex operations like *expanding special Wh* questions*;
- Experiments show that our QR learning framework enhances the rewriting performance compared to the baselines.

2 Related Work

Elgohary et al. (2019) created the QR dataset which rewrites a subset of the questions from QuAC (Choi et al., 2018). Based on this dataset, some recent work has studied this task and formulates QR as a text generation task with an encoder-decoder architecture (Elgohary et al., 2019; Kumar and Joshi, 2016; Vakulenko et al., 2020; Li et al., 2019; Lin et al., 2020a).

The difficulty of answering a question given a relevant document has been studied in the question answering community (Dua et al., 2019; Wolfson et al., 2020). Sugawara et al. (2018) examine 12 reading comprehension datasets and determine what makes a question more easily answered. Perez et al. (2020); Min et al. (2019); Talmor and Berant (2018); Dong et al. (2017) study how to make a hard question more easily answered. However, there is no work to date that studies whether rewriting difficulties exist in QR and how to measure the difficulties. Some other work is similar to QR but focuses on other tasks such as dialogue tracking (Rastogi et al., 2019; Su et al., 2019; Liu et al., 2020) and information retrieval (Voskarides et al., 2020; Lin et al., 2020b; Liu et al., 2019).

Varying rewriting difficulties can result in multiple underlying data distributions in the QR training data. The shared-private framework has been studied to learn from training data with multiple distributions (Zhang et al., 2018; Liu et al., 2017). One issue of the shared-private framework is parameter inefficiency when building private models We use adapter tuning (Rebuffi et al., 2018, 2017)

to build the private models. Adapter tuning was recently proposed for adapting a pre-trained language model, e.g., BERT (Devlin et al., 2019), to downstream tasks (Pfeiffer et al., 2020a,c; Houlsby et al., 2019), and its effectiveness has been verified by previous work (Bapna and Firat, 2019; Pfeiffer et al., 2020b; Wang et al., 2020; He et al., 2021). We are the first to apply it to reduce model parameters in the shared-private framework. How to combine the knowledge stored in multiple adapters is also important. Pfeiffer et al. (2020a) propose adapter fusion to build an ensemble of adapters in multi-task learning. We propose sequence-level adapter fusion in our work.

3 Question-in-Context Rewriting

Question-in-context rewriting (QR) aims to generate a self-contained rewrite from a context-dependent question in CQA. Given a conversational dialogue $\mathcal{H}$ with sequential question and answer pairs $\{\mathbf{q}_1, \mathbf{a}_1, \cdots, \mathbf{q}_n, \mathbf{a}_n\}$, for a question $\mathbf{q}_i$ from $\mathcal{H}$ with its history $\mathbf{h}_i = \{\mathbf{q}_1, \mathbf{a}_1, \cdots, \mathbf{q}_{i-1}, \mathbf{a}_{i-1}\}$, we generate its rewrite $\mathbf{q}'_i$. We define the labeled dataset $\mathcal{D} = \{\mathbf{q}_i, \mathbf{h}_i, \mathbf{q}'_i\}_{i=1}^{|\mathcal{D}|}$ which is a set of tuples of question $\mathbf{q}$, history $\mathbf{h}$, and rewrite $\mathbf{q}'$. Following previous work (Elgohary et al., 2019), we model QR in an encoder-decoder framework, by estimating the parameterized conditional distribution for the output $\mathbf{q}'$ given the input question $\mathbf{q}$ and history $\mathbf{h}$. For $(\mathbf{q}, \mathbf{h}, \mathbf{q}') \in \mathcal{D}$, we minimize the following loss function parameterized by θ:

$$\begin{aligned}\mathcal{L}^{\theta}_{NLL} &= -\log P(\mathbf{q}'|\mathbf{q}, \mathbf{h}; \theta) \\ &= -\sum_{t=1}^{T_{\mathbf{q}'}} \sum_{k=1}^{|V|} \mathbb{1}\{q'_t = k\} \log P(q'_t = k|\mathbf{q}'_{<t}, \mathbf{q}, \mathbf{h}; \theta)\end{aligned} \tag{1}$$

in which $T_{\mathbf{q}'}$ is the length of $\mathbf{q}'$ and $|V|$ is the vocabulary size. Following Elgohary et al. (2019), $\mathbf{q}$ and $\mathbf{h}$ are concatenated into one sequence as the input. All previous turns of the history information are combined for learning. The choice of the encoder-decoder framework can be LSTM (Elgohary et al., 2019), transformer (Vakulenko et al., 2020), or pre-trained language models (Lin et al., 2020a). In our work, we build our model based on the pre-trained language model BART (Lewis et al., 2020).

4 Difficulty of Question Rewriting

The difficulty of rewriting a question varies across questions. We propose a simple yet effective heuristic to formulate rewriting difficulty as the discrepancy between a question and its rewrite. To generate a self-contained rewrite, we need to identify relevant information from the conversation context to incorporate it into the original question. We observe that if the discrepancy is large, we need to identify more missing information from the conversation context which makes the rewriting task more difficult.

In this work, we use BLEU score to measure the discrepancy. BLEU has been widely used to measure how similar two sentences are (Papineni et al., 2002). Given a question $\mathbf{q}$ and its rewrite $\mathbf{q}'$, we define the difficulty score z for rewriting $\mathbf{q}$ as:

$$\mathrm{z} = BLEU(\mathbf{q}, \mathbf{q}') \tag{2}$$

where the rewrite $\mathbf{q}'$ is the reference and $\mathrm{z} \in [0, 1]$. A low z score indicates a larger discrepancy between $\mathbf{q}$ and $\mathbf{q}'$, making it more difficult to rewrite $\mathbf{q}$ into $\mathbf{q}'$. Besides BLEU, we also study the effectiveness of ROUGE, lengths of $\mathbf{q}$ and $\mathbf{q}'$, and $|\mathbf{q}|/|\mathbf{q}'|$ in §6.5 to measure rewriting difficulty.

5 Difficulty-Aware QR with Adapters

Previous work on QR learns to rewrite questions with only one shared model (Elgohary et al., 2019), which cannot adequately model all questions with different rewriting difficulties. Instead of using only one shared model, we propose a novel method to classify a question into several classes by measuring its rewriting difficulty (§5.1), learn a private model for each class (§5.2), and finally combine the private models for inference (§5.3). Different questions with varying rewriting difficulties result in multiple data distributions in the training set. By dividing the training data into several classes with varying rewriting difficulties, we can better learn the data distributions with the help of private models (Zhang et al., 2018).

5.1 Question Classification

We compute the difficulty score z of each question in the dataset. We set score intervals and group the questions with difficulty scores within the same interval together. Specifically, we divide the original dataset $\mathcal{D}$ into m classes: $\{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_m\}$. Setting m to a large number (e.g., the number of training samples) can more accurately model the

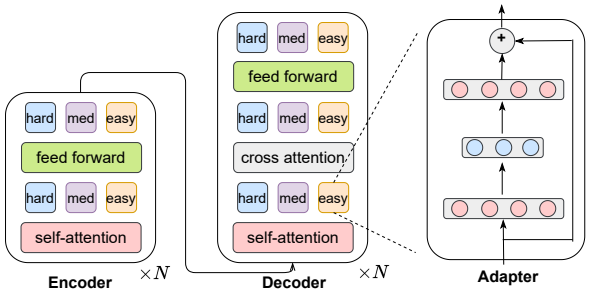

Figure 1: Illustration of our model architecture. Class-private adapters are added into the transformer. The original PLM weights are shared across all private models. N is the number of transformers in the encoder and decoder.

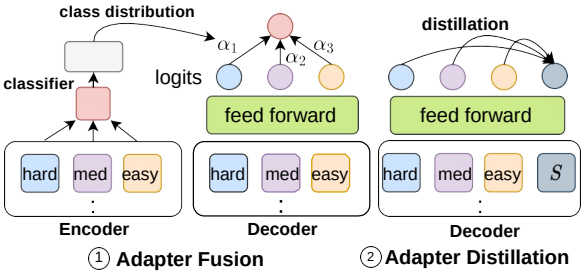

Figure 2: Illustration of our sequence-level adapter fusion and distillation.

data distribution of the training data, but at the expense of data sparsity in each class such that a private model cannot be adequately trained.

5.2 Learning Private Models

After dividing the questions into m classes, we learn a private model for each class. By training on each class of data, the private model can better learn the domain-specific information. The common way to use a pre-trained language model (PLM) such as BART is to fine-tune the model on the downstream task. However, doing so will require m times the number of PLM parameters to build all private models, where m is the number of classes. This results in a large number of parameters, leading to inefficiency.

To reduce the number of model parameters in learning the private models, we introduce adapters into the PLM. Adapters are light-weight neural networks and are plugged into the PLM. When adapting the PLM to downstream tasks, we only need to update the parameters of the adapters but keep the original parameters of the PLM frozen and shared among all private models. Where to place the adapters in the neural architecture will affect the efficacy of adapters. As shown in Figure 1, for each transformer layer in the encoder, we add the adapters after the self-attention layer and feed-forward layer. We further add the adapters after the cross-attention layer in the decoder. Though our model is built on BART, our proposed placement of adapters can also be used in other PLMs, such as T5 (Raffel et al., 2020).

In Figure 1, the adapter is a module with a stack of two linear layers following Houlsby et al. (2019). Formally, given an input hidden vector $\mathbf{x}$ from the previous layer, we compute the output hidden vector $\mathbf{x}'$ of the adapter as:

$$\mathbf{x}' = f_2(\tanh(f_1(\mathbf{x}))) + \mathbf{x} \quad (3)$$

where $f_1(\cdot)$ is the down-scale linear layer and $f_2(\cdot)$ is the up-scale linear layer. The hidden vector size is smaller than the dimension of the input vector. Learning a private model for one class only introduces $5 \times N$ adapters, where N is the number of layers in the encoder and decoder. The original parameters of the PLM are shared by all adapters, so the number of parameters required when building the private models can be much reduced.

5.3 Model Ensemble

After learning the private models for all classes, at test time, we present the question to the corresponding private model to generate its rewrite if we know which class this question belongs to. However, it is not possible to determine the difficulty score by calculating the BLEU score between the question and its rewrite since there is no gold-standard rewrite for the question at test time. As such, we need to combine the private models into one model for inference. In this work, we propose two methods to combine the private models, as explained below.

Sequence-level Adapter Fusion (SLAF). After dividing the training set into m classes based on the difficulty scores, we assign a difficulty label to each class to obtain a set of class labels $\{l_1, l_2, \cdots, l_m\}$. We introduce a classifier to learn to predict the difficulty label l, given a question $\mathbf{q}$ and its conversation history $\mathbf{h}$. As shown in Figure 2, during inference, we obtain the logistic output from each private model. The classifier generates the class distribution to combine the logistic outputs for sequence generation.

By assigning a difficulty label to each question, we obtain the dataset $\mathcal{D}' = \{\mathbf{q}_i, \mathbf{h}_i, \mathbf{q}'_i, l_i\}_{i=1}^{|\mathcal{D}'|}$. For each training sample $(\mathbf{q}, \mathbf{h}, \mathbf{q}', l) \in \mathcal{D}'$, we mini-

mize the following loss function:

$$\mathcal{L}_{NLL}^{\theta_c} = -\log \text{softmax}\Big(\sum_{i=1}^{m} \alpha_i f_i(\mathbf{q}, \mathbf{h}; \theta_i)\Big) - \log P(l|\mathbf{q}, \mathbf{h}; \theta_c) \quad (4)$$

where f_i is the ith private model, α_i is the class weight of the ith private model, and θ_c is the parameter of the classifier. We jointly estimate the conditional distribution for sequence generation and the distribution for classification. In this process, the private models are frozen and not updated. We combine the vectors out of the private models to calculate the vector f_c as the input for the classifier:

$$f_c = \frac{1}{m} \sum_{i=1}^{m} f_{encoder}^{i}(\mathbf{q}, \mathbf{h}; \theta_i) \quad (5)$$

where $f_{encoder}^{i}$ is the encoder of the ith private model. For each private model, we average the token embeddings from the last layer of the encoder.

Sequence-level Adapter Distillation (SLAD). SLAF provides a way to combine the private models, but it is time-consuming during inference since it requires each private model to compute its logistic output before combination. Another drawback is that the domain classifier in SLAF cannot generate the best class distributions at test time, causing non-optimal rewriting results by SLAF. As shown in Figure 2, to speed up inference and better combine the private models, we distill the private models into one shared model. We expect the student model S (modeled by adapters) to be able to generate questions with different rewriting difficulties. For each training sample $(\mathbf{q}, \mathbf{h}, \mathbf{q}', l) \in \mathcal{D}'$, we define the knowledge distillation loss function as follows:

$$\mathcal{L}_{KD}^{\theta_S} = -\sum_{t=1}^{T_{\mathbf{q}'}} \sum_{k=1}^{|V|} P^{(l)}\{q'_t = k|\mathbf{q}'_{<t}, \mathbf{q}, \mathbf{h}; \theta^{(l)}\} \times \log P(q'_t = k|\mathbf{q}'_{<t}, \mathbf{q}, \mathbf{h}; \theta_S) \quad (6)$$

in which we approximate the output distribution of the teacher private model l parameterized by $\theta^{(l)}$ with the student model parameterized by θ_S. We learn the student model with the following function:

$$\mathcal{L}_{distill}^{\theta_S} = (1-\gamma) \cdot \mathcal{L}_{KD}^{\theta_S} + \gamma \cdot \mathcal{L}_{NLL}^{\theta_S} \quad (7)$$

where $\mathcal{L}_{NLL}^{\theta_S}$ is the same loss function in Eq. 1, and γ is a hyper-parameter. The private models are fixed in the distillation process. Since we directly distill the knowledge of the private models into a shared model without the soft weights generated by the domain classifier from SLAF, SLAD can better combine the private models and achieve better rewriting performance.

	Train	Valid	Test	All
CANARD	31,526	3,430	5,571	40,527
QRECC	57,150	6,351	16,451	79,952

Table 2: Data splits of CANARD and QRECC.

CANARD	Hard	Medium	Easy	All
Ratio (%)	32.36	33.45	34.20	-
BLEU score	[0, 0.2)	[0.2, 0.5)	[0.5, 1]	-
Avg. # tokens in q + h	111.98	103.53	90.23	101.72
Avg. # tokens in q′	14.46	11.46	9.95	11.60
QRECC	**Hard**	**Medium**	**Easy**	**All**
Ratio (%)	29.53	41.20	29.27	-
BLEU score	[0, 0.2)	[0.2, 0.4)	[0.4, 1]	-
Avg. # tokens in q + h	126.07	106.92	94.53	108.95
Avg. # tokens in q′	14.72	10.36	10.07	11.56

Table 3: Statistics of each class for the training set of CANARD and QRECC.

6 Experiments

6.1 Dataset

We conduct our experiments on CANARD (Elgohary et al., 2019) and QRECC (Anantha et al., 2021), which are designed for the task of question rewriting in CQA. CANARD was created from QuAC (Choi et al., 2018), by rewriting a subset of the questions by humans. The dataset consists of tuples of question, conversation history, and rewrite. QRECC answers conversational questions within large-scale web pages. Detailed data splits for the two datasets are shown in Table 2. We divide the questions into hard, medium, and easy classes, and the statistics are presented in Table 3.

6.2 Setup

Model Settings. We build our models on the pre-trained language model of BART (Lewis et al., 2020). Specifically, we use BART-base to initialize our models. There are 6 transformer layers for the encoder and decoder in BART-base. For our

Model ╲ $\mathcal{D}$	Hard	Medium	Easy	Mean
LSTM-S	26.29	50.79	79.41	49.81
Fine-tune-S	39.38	53.70	66.33	53.14
Adapter-S	39.20	53.14	65.97	52.77

Table 4: BLEU scores (in %) on hard, medium, and easy classes from CANARD, based on the shared model. Fine-tune-S and Adapter-S are based on BART-base.

	Rewriting Rules	Examples
1	replace pronoun, e.g., he/his/she/her/they/their/it/its...	when was ***he*** born ? → when was ***Corbin Bleu*** born ?
2	add prepositional phrase	what happened in 1998 ? → what happened ***to Debra Marshall, Manager of Jeff Jarrett*** in 1998 ?
3	explain *else* context for questions with the forms, e.g., else/other/as well	Was there ***any other*** views he had in regards to them ? → ***Other than Peter Tatchell condemned the Soviet Union's invasions of Czechoslovakia***, was there any other views Peter Tatchell had in regards to Soviet Union ?
4	extend the nominal phrase, e.g., name/entity	Who wrote the ***song*** ? → Who wrote the ***'03 Bonnie & Clyde song*** ?
5	expand the special Wh* questions, e.g., why?/what happened/which	***Which*** of the show is the biggest ? → ***Which espisode of The Oprah Winfrey Show*** is the biggest?
6	add completed sentences after that/this	What was the aftermath of ***that*** ? → What was the aftermath of ***Robert Kennedy was chosen by McCarthy as a counsel for ...*** ?
7	other options	

Table 5: The commonly used rewriting rules for QR in CANARD.

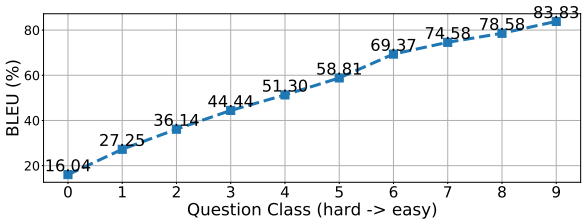

Figure 3: 10-class BLEU scores on CANARD with LSTM-S.

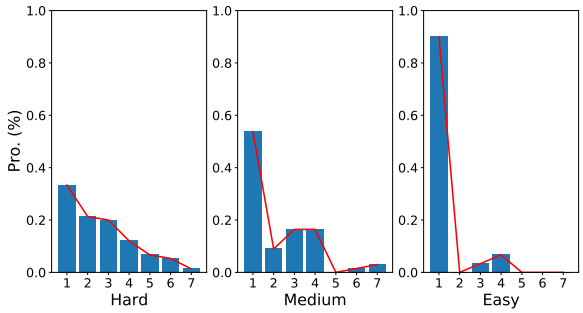

Figure 4: The distributions of rewriting rules on hard, medium, and easy subsets in CANARD.

adapter, we map the dimension of the input hidden vector from 768 to 384 which is re-mapped to 768 for the output vector. The hidden vector size for adapter tuning is the default value of 384. Based on BART-base, we need a total of $6 \times 2 + 6 \times 3 = 30$ adapters for each private model. We set γ to 0.5 in Eq. 7 for CANARD and 0.9 for QRECC. α from Eq. 4 is set to 2 for both CANARD and QRECC. When fine-tuning BART, we set the learning rate to 1e-5, and for adapter tuning, the learning rate is 1e-4 (both values are tuned from {1e-4, 1e-5}). We use the validation set to keep the best model based on the BLEU score. We implement our models with HuggingFace (Wolf et al., 2019) and keep the other default training settings. In CANARD, about 20% of the questions can be rewritten by replacing pronouns with their referents, so we carry out pronoun replacement first for the questions (if any) before using BLEU scores to measure rewriting difficulties. More details are given in Appendix A.

Baselines. We compare to the following baselines. **S** denotes training only one shared model with all the training data, which is commonly used in previous work (Elgohary et al., 2019; Lin et al., 2020a). By adapting BART, **P-hard**, **P-medium**, and **P-easy** are the baselines that train private models on the hard, medium, and easy classes respectively, using fine-tuning or adapter-tuning. Assuming that rewriting difficulty labels are accessible for questions at test time (i.e., the oracle setting), **Mix-gold** processes a question by the corresponding private model using the difficulty label. **SLAF** and **SLAD** denote sequence-level adapter fusion and adapter distillation respectively for combining the private models of P-hard, P-medium, and P-easy. **SLAF-uni.** combines the private models with uniform distributions. **SLAF-pred** predicts the class label for the input and then chooses the corresponding private model for generation. **LSTM-S** trains one model using an LSTM-based Seq2Seq model with copy mechanism (See et al., 2017) which was used in Elgohary et al. (2019).

Evaluation Metric. Following Elgohary et al. (2019), we use BLEU[2] to obtain the results on hard, medium, and easy classes, and the three results are

[2]https://github.com/mjpost/sacrebleu

averaged to obtain the mean result.

6.3 Robustness Evaluation

6.3.1 Rewriting Difficulty

We first study rewriting difficulties across different questions. Table 4 shows the results on hard, medium, and easy classes on CANARD. **Each class vs. Overall**: Comparing to the overall results, the rewriting performances of hard questions drop substantially, but are much higher on the easy class. **LSTM-S vs. BART-S**: By comparing LSTM-S to tuning on BART, LSTM-S achieves higher performance on the easy class but much worse performance on hard and medium classes. This is probably because for easy questions, the model only needs to copy some words from the context and LSTM-S has an explicit copy mechanism to achieve this goal but not BART. Since BART learns a more complex model than LSTM-S, it can better deal with harder questions.

We further divide the test set into ten classes in Figure 3, where the interval $[0, 1]$ is equally divided into ten sub-intervals of size 0.1. We find that when z gets smaller, rewriting performance degrades, indicating an increase in rewriting difficulty.

6.3.2 Human Evaluation

The above evaluation results show that our method can effectively divide the questions into subsets with different rewriting difficulties. Here, we conduct a human evaluation to evaluate the question characteristics on these subsets for validity and see what makes the questions hard or easy to rewrite.

Question Annotation. To find out what makes the questions different, we first summarize the commonly used rewriting rules, which describe the operations of translating a question into its rewrite. 6 rules are summarized from the training set of CANARD and presented in Table 5. Different rules account for different rewriting hardness for QR systems. For example, the rule of *replace pronoun* is very simple since it only requires the model to determine the pronoun to replace. However, rules 5 and 6 shown in the table will be much harder because the model needs to understand the conversational history well, and the information to be filled in is substantial.

Then we randomly select 50 examples from each subset (hard, medium, and easy) from the test set and annotate what rules in Table 5 are used for each example. One question may have multiple rewriting rules. More details are in Appendix B.

Model $\diagdown$ $\mathcal{D}$	**Hard**	**Medium**	**Easy**	**Mean**
LSTM based				
S	26.29	50.79	79.41	52.16
Mix-gold	27.79	51.91	86.53	55.41
Fine-tune BART-base				
S	39.38	53.7	66.33	53.14
Mix-gold	40.91	56.15	74.00	57.02
Tuning BART-base with adapters				
S	$39.20_{0.52}$	$53.14_{0.11}$	$65.97_{0.12}$	$52.77_{0.16}$
P-hard	$41.33_{0.27}$	$46.39_{0.46}$	$55.24_{0.93}$	$47.66_{0.51}$
P-medium	$34.41_{0.19}$	$54.68_{0.31}$	$62.98_{0.14}$	$50.69_{0.11}$
P-easy	$27.42_{0.26}$	$55.55_{0.16}$	$73.63_{0.18}$	$52.20_{0.12}$
SLAF-uni.	$34.05_{0.09}$	$55.88_{0.65}$	$67.27_{0.09}$	$52.40_{0.23}$
SLAF-pred	$32.96_{0.26}$	$55.62_{0.38}$	$70.83_{0.21}$	$53.14_{0.12}$
SLAF	$34.55_{0.05}$	$56.05_{0.32}$	$69.05_{0.15}$	$^{*}53.22_{0.17}$
SLAD	$38.26_{0.39}$	$54.22_{0.10}$	$67.57_{0.30}$	$^{*}53.35_{0.17}$
Mix-gold	$41.33_{0.27}$	$54.68_{0.31}$	$73.63_{0.18}$	$56.55_{0.12}$

Table 6: The test results (mean and standard deviation) on CANARD. We run 3 times for adapter tuning. $*$ indicates statistically significant improvement over S and SLAF-uni. ($p < 0.05$).

Results. We sum the number of each rewriting rule in each subset and show the distributions of rewriting rules for each subset in Figure 4. The three distributions are quite different. We find that:

- the easy subset mainly uses rule 1 for rewriting questions;
- for medium and hard subsets, other rules are used, such as rules 2, 3, and 4 which are more complex than rule 1;
- the hard class uses more rules 2, 3, 5, and 6 compared to the medium class, which demonstrates that the hard class is more difficult than the medium class.

Discussion. By knowing the characteristics of each class of questions, we can optimize the model architecture of private models accordingly. For hard questions, we can add some rules to deal with *Wh** questions. For easy questions, LSTM-based models seem to be good enough as Table 4 indicates. In this work, we have shown that the questions vary in rewriting difficulties and to improve the overall rewriting performance, we focus on the ensemble method to combine the private models. We leave optimizing the model architecture to future work.

6.4 Question Rewriting

We report our results on question rewriting based on CANARD and QReCC. From the results in Tables 6 and 7, we first show the results of each class, then the mean performances are displayed. **Mix-gold, SLAF, SLAD vs. S**: (**a**) Mix-gold, SLAF,

Model ╲ $\mathcal{D}$	Hard	Medium	Easy	Mean
Tuning BART-base with adapters				
S	$45.43_{0.27}$	$60.60_{0.21}$	$78.47_{0.17}$	$61.50_{0.02}$
P-hard	$49.48_{0.16}$	$53.13_{0.09}$	$67.32_{0.19}$	$56.65_{0.10}$
P-medium	$43.17_{0.28}$	$61.83_{0.26}$	$76.63_{1.19}$	$60.54_{0.50}$
P-easy	$37.56_{0.80}$	$63.17_{0.19}$	$82.79_{0.40}$	$61.17_{0.22}$
SLAF-uni.	$43.28_{0.39}$	$62.21_{0.23}$	$78.89_{0.17}$	$61.46_{0.17}$
SLAF-pred	$43.60_{0.72}$	$61.69_{0.64}$	$79.05_{0.92}$	$61.45_{0.28}$
SLAF	$43.76_{0.53}$	$62.13_{0.19}$	$79.71_{0.24}$	$^{*}61.87_{0.17}$
SLAD	$44.99_{0.25}$	$61.35_{0.21}$	$79.93_{0.08}$	$^{*}62.09_{0.05}$
Mix-gold	$49.48_{0.16}$	$61.83_{0.26}$	$82.79_{0.40}$	$64.70_{0.09}$

Table 7: The test results (mean and standard deviation) on QRECC. We run 3 times for adapter tuning. $*$ indicates statistically significant improvement over S, SLAF-uni., and SLAF-pred ($p < 0.05$).

Method ╲ $\mathcal{D}$	$\mathcal{D}_1$	$\mathcal{D}_2$	$\mathcal{D}_3$	Trend	Std.
$\|q\|$	5.27	7.22	10.04	↗	–
BLEU	38.90	52.78	63.29	↗	12.2
$\|q'\|$	6.82	10.14	17.07	↗	–
BLEU	39.46	61.51	50.08	↗, ↘	11.0
$\|q\|/\|q'\|$	0.47	0.75	0.97	↗	–
BLEU	43.18	55.93	56.69	↗	7.6
ROUGE-L (%)	56.25	76.04	94.25	↗	–
BLEU	40.79	50.39	74.20	↗	17.2
BLEU (%)	16.13	44.59	90.27	↗	–
BLEU	39.58	53.80	65.60	↗	13.0

Table 8: Results of measuring rewriting difficulty on CANARD.

and SLAD are consistently better than S, which demonstrates the effectiveness of learning private models to model multiple underlying distributions. (**b**) From the results on each class, SLAF and SLAD can substantially enhance the performance on medium and easy classes compared to S. (**c**) SLAD is more effective than SLAF and SLAD is more efficient during inference. (**d**) We find Mix-gold to be better than SLAF and SLAD, since Mix-gold is an oracle model that uses the correct difficulty label to select the private model for inference.

We find that by learning a private model for each class, the performance on the corresponding class can be consistently improved, which explains why Mix-gold, SLAF, and SLAD can outperform S. We also find that the sole private model cannot improve the overall rewriting performance of the three classes, but SLAF and SLAD can outperform S after model ensemble, which demonstrates the necessity of combining the private models.

Model Ensemble. One question is whether the improvements of SLAF and SLAD simply come from combing multiple models and whether applying only one private model selected by the predicted class label is better. As shown in Tables 6 and 7, we find SLAF-uni. performs worse than SLAF and SLAD, which demonstrates that the benefits of SLAF and SLAD are not simply because of the model ensemble, but class estimation also helps (In SLAD, class estimation lies in using gold class labels of questions for knowledge distillation during training). SLAF-pred can be regarded as an ensemble method since it uses multiple private models during inference. Compared to SLAF, SLAF-pred uses one-hot class weights to combine the private models. However, SLAF-pred performs worse than SLAF, and the reason could be that classifying the question into the corresponding class is nontrivial, wrong predictions will have much worse rewriting results as the results of P-hard, -medium, -easy on other classes indicate.

6.5 Further Analysis

Analysis of Rewriting Difficulty Measures. In our work, we use BLEU to measure the discrepancy between a question and its rewrite. We further experiment with other methods to assess their effectiveness for difficulty measurement. CANARD is evaluated here. As shown in Table 8, we first use the length of a question ($|q|$), its rewrite ($|q'|$), and their ratio ($|q|/|q'|$) to calculate a difficulty score. After re-ranking the questions with a difficulty score, we divide the ranked questions equally into three classes. Interestingly, we find that $|q|$ works well. After analysis, we find that rewriting short questions requires finding much missing information, which makes short questions hard questions. The $|q|/|q'|$ metric is not very useful, since $|q|/|q'|$ can only measure the discrepancy in question lengths, but does not necessarily measure their semantic difference. $|q'|$ does not work for difficulty measurement. Not surprisingly, the ROUGE score is also useful in measuring discrepancy just like BLEU.

Analysis of Learning Data Distribution. Tables 6 and 7 show that learning private models can enhance performance on each class. We further divide the data into eleven classes ($z \in [0, 0.1]$, $(0.1, 0.2], \cdots, (0.9, 1), 1$) and learn a private model for each class. We build the private models using LSTM-S, in which we first train a shared model on the full training data, then fine-tune the shared model on each class to obtain the private models. Table 9 shows the BLEU scores where the score in the (i, j) entry is obtained by training on class

	0	1	2	3	4	5	6	7	8	9	10
0	19.2	28.3	34.7	39.9	44.2	50.3	57.9	64.6	71.6	80.3	71.9
1	17.7	28.1	36.1	43.3	48.5	53.6	61.4	66.5	74.5	75.1	74.5
2	16.0	28.6	36.2	44.0	49.3	55.9	64.7	70.3	79.6	86.2	78.6
3	15.0	26.8	35.7	45.3	51.3	57.5	66.9	70.8	80.0	88.4	81.2
4	12.8	26.0	35.9	44.8	52.1	60.1	68.9	73.5	78.5	95.7	81.8
5	12.5	25.3	35.1	44.9	50.3	61.1	70.4	75.9	79.9	94.0	84.4
6	11.8	25.0	34.9	44.4	51.7	61.7	71.0	77.4	81.9	89.4	86.7
7	11.9	24.4	34.5	44.2	51.5	61.8	71.7	80.2	84.9	91.1	87.9
8	9.4	20.8	31.3	41.7	49.4	58.6	68.1	76.0	85.6	97.6	92.0
9	15.8	27.3	35.3	44.7	50.9	60.2	69.5	75.6	83.7	89.4	85.9
10	13.5	24.7	34.8	44.4	51.9	60.2	69.7	75.4	82.0	98.4	92.2

Table 9: BLEU scores for different classes on CANARD. The rows are the private models and the columns are the classes.

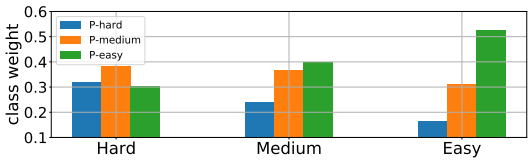

Figure 5: Class weights for different classes on CANARD.

i and testing on class j. On the whole, learning private models can enhance the performance of the corresponding class. With these private models, we can better model the data distributions, but how to combine a large number of private models is a challenge, since it is hard to train a classifier to correctly predict so many class labels, which will have some negative effects on the model ensemble.

Analysis of SLAF & SLAD. We plot the class distributions of hard, medium, and easy classes in Figure 5. We find that in the hard class, the class weights are almost equally distributed among the private models, which means that the hard questions are difficult for classification. This result explains why SLAF performs worse than S for hard questions in Tables 6 and 7. We further study the contribution of distillation in SLAD. In Figure 6, on the whole, when γ increases, the contribution of distillation decreases, and the performance drops, indicating that distillation is important for SLAD.

Case Study. We further show generated rewriting samples of various methods on CANARD in Appendix C.

7 Conclusion

In this work, we study the robustness of a QR system to questions varying in rewriting hardness. We use a simple yet effective heuristic to measure the rewriting difficulty. We further propose a novel method to deal with varying rewriting difficulties. Tested on CANARD and QReCC, we show the effectiveness of our methods.

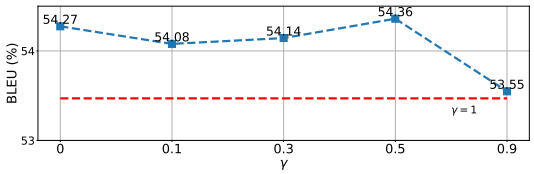

Figure 6: BLEU socres for different γ values on CANARD.

Acknowledgments

This research is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-RP-2018-007 and AISG2-PhD-2021-08-016[T]). The computational work for this article was partially performed on resources of the National Supercomputing Centre, Singapore (https://www.nscc.sg).

References

Raviteja Anantha, Svitlana Vakulenko, Zhucheng Tu, Shayne Longpre, Stephen Pulman, and Srinivas Chappidi. 2021. Open-domain question answering goes conversational via question rewriting. In *Proceedings of NAACL-HLT*.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of EMNLP-IJCNLP*.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. In *Proceedings of EMNLP*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*.

Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. 2017. Learning to paraphrase for question answering. In *Proceedings of EMNLP*.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of NAACL-HLT*.

Ahmed Elgohary, Denis Peskov, and Jordan L. Boyd-Graber. 2019. Can you unpack that? learning to rewrite questions-in-context. In *Proceedings of EMNLP-IJCNLP*.

Ruidan He, Linlin Liu, Hai Ye, Qingyu Tan, Bosheng Ding, Liying Cheng, Jia-Wei Low, Lidong Bing, and

Luo Si. 2021. On the effectiveness of adapter-based tuning for pretrained language model adaptation. In *Proceedings of ACL/IJCNLP*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of ICML*.

Vineet Kumar and Sachindra Joshi. 2016. Non-sentential question resolution using sequence to sequence learning. In *Proceedings of COLING*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of ACL*.

Qian Li, Hui Su, Cheng Niu, Daling Wang, Zekang Li, Shi Feng, and Yifei Zhang. 2019. Answer-supervised question reformulation for enhancing conversational machine comprehension. In *Proceedings of the 2nd Workshop on MRQA@EMNLP*.

Sheng-Chieh Lin, Jheng-Hong Yang, Rodrigo Nogueira, Ming-Feng Tsai, Chuan-Ju Wang, and Jimmy Lin. 2020a. Conversational question reformulation via sequence-to-sequence architectures and pretrained language models. *CoRR*.

Sheng-Chieh Lin, Jheng-Hong Yang, Rodrigo Nogueira, Ming-Feng Tsai, Chuan-Ju Wang, and Jimmy Lin. 2020b. Query reformulation using query history for passage retrieval in conversational search. *CoRR*.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-task learning for text classification. In *Proceedings of ACL*.

Qian Liu, Bei Chen, Jian-Guang Lou, Bin Zhou, and Dongmei Zhang. 2020. Incomplete utterance rewriting as semantic segmentation. In *Proceedings of EMNLP*.

Ye Liu, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip S. Yu. 2019. Generative question refinement with deep reinforcement learning in retrieval-based QA system. In *Proceedings of CIKM*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of EMNLP*.

Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019. Multi-hop reading comprehension through question decomposition and rescoring. In *Proceedings of ACL*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL*.

Ethan Perez, Patrick S. H. Lewis, Wen-tau Yih, Kyunghyun Cho, and Douwe Kiela. 2020. Unsupervised question decomposition for question answering. In *Proceedings of EMNLP*.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2020a. Adapterfusion: Non-destructive task composition for transfer learning. *CoRR*.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulic, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020b. Adapterhub: A framework for adapting transformers. In *Proceedings of EMNLP*.

Jonas Pfeiffer, Ivan Vulic, Iryna Gurevych, and Sebastian Ruder. 2020c. MAD-X: an adapter-based framework for multi-task cross-lingual transfer. In *Proceedings of EMNLP*.

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. *J. Mach. Learn. Res.*

Pushpendre Rastogi, Arpit Gupta, Tongfei Chen, and Lambert Mathias. 2019. Scaling multi-domain dialogue state tracking via query reformulation. In *Proceedings of NAACL-HLT*.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning multiple visual domains with residual adapters. In *Proceedings of NeurIPS*.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2018. Efficient parametrization of multi-domain deep neural networks. In *Proceedings of CVPR*.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. Coqa: A conversational question answering challenge. *Trans. Assoc. Comput. Linguistics*.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of ACL*.

Hui Su, Xiaoyu Shen, Rongzhi Zhang, Fei Sun, Pengwei Hu, Cheng Niu, and Jie Zhou. 2019. Improving multi-turn dialogue modelling with utterance rewriter. In *Proceedings of ACL*.

Saku Sugawara, Kentaro Inui, Satoshi Sekine, and Akiko Aizawa. 2018. What makes reading comprehension questions easier? In *Proceedings of EMNLP*.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proceedings of NAACL-HLT*.

Svitlana Vakulenko, Shayne Longpre, Zhucheng Tu, and Raviteja Anantha. 2020. Question rewriting for conversational question answering. *CoRR*.

Nikos Voskarides, Dan Li, Pengjie Ren, Evangelos Kanoulas, and Maarten de Rijke. 2020. Query resolution for conversational search with limited supervision. In *Proceedings of SIGIR*.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2020. K-adapter: Infusing knowledge into pre-trained models with adapters. *CoRR*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Tomer Wolfson, Mor Geva, Ankit Gupta, Yoav Goldberg, Matt Gardner, Daniel Deutch, and Jonathan Berant. 2020. Break it down: A question understanding benchmark. *Trans. Assoc. Comput. Linguistics*.

Ye Zhang, Nan Ding, and Radu Soricut. 2018. SHAPED: shared-private encoder-decoder for text style adaptation. In *Proceedings of NAACL-HLT*.

A Experimental Setup

We use HuggingFace (Wolf et al., 2019) to implement our model. We follow the training script from https://github.com/huggingface/transformers/tree/master/examples/seq2seq to train the model. Models are trained for 10 epochs. Batch size is selected from {10, 16, 32}. Learning rate is selected from {1e-5, 1e-4}. We train 10 epochs for CANARD and 8 epochs for QReCC. The best model based on the BLEU score on the validation set is kept. The beam width for beam search is the default value of 4.

For our QR framework, we first train a private model for each class. For model ensemble, the weights of the private models are frozen without updating. On QReCC, to build the private models, on each class of data, we fine-tune the shared model which is trained on all the training data, since we find that this can enhance the final performance, but on CANARD, we do not see the improvement. The learning rate of fine-tuning in this process is 1e-5.

To pre-process the dataset, we only tokenize the sentences. And we append the question and its history context with "|||".

In CANARD, about 20% of the questions can be rewritten by replacing pronouns with their referents, so we carry out pronoun replacement first for the questions (if any) before using BLEU scores to measure rewriting difficulties.

B Human Assessment for Rewriting Rules

We first ask one annotator to summarize some common rewriting rules by looking at the training set of CANARD (Elgohary et al., 2019). When accessing the rewriting rules used for each question, the second annotator will rely on the summarized rewriting rules for annotation. For each class, we randomly select 50 questions from the test set for annotation.

Case Study. Table 10 shows some annotated results from the hard, medium, and easy classes.

C Case Study of Generated Rewrites

We further show some cases of generated rewrites from various methods (S, Mix-gold, SLAF, and SLAD). We use adapter tuning to build these models. Tables 11, 12, and 13 show the generated rewrites on hard, medium, and easy classes respectively.

		1	2	3	4	5	6	7
Question	Rewrite	replace pronoun, e.g., he/ his/ she/ her/ they/ their/ it/ its...	add prepositional phrase	explain *else* context for questions with the forms, e.g., else/other/as well	extend the nominal phrase, e.g., name/entity	expand the special Wh* questions, e.g., why?/what happened/which	add completed sentences after that/this	other options
easy								
How does he try to take over the world ?	How does Brain try to take over the world ?	1						
what were some of his careers that involved flying ?	what were some of Luis Walter Alvarez 's careers that involved flying ?	1						
What was so special about the song ?	What was so special about the song Purple Haze ?				1			
medium								
When did she start to campaign ?	When did Jeanine Pirro start to campaign for the lieutenant governor role ?	1	1					
Did this lead to another choreographing job ?	Did Etude in D Minor lead to another choreographing job ?	2						
What did they do after the album ?	What did Fat Freddy 's Drop do after the album , " Live at the Matterhorn " ?	1			1			
hard								
Was there anything else noteworthy about the autobiography ?	Aside from Chester 's friends being uncomfortable with his writing , was there anything else noteworthy about Chester Brown 's autobiography ?	1		1				
was the tour domestic or international ?	was the Sanctus Real 's Fight the Tide Tour domestic or international ?				1			
Was there any other views he had in regards to them ?	Other than Peter Tatchell condemned the Soviet Union 's invasions of Czechoslovakia , Afghanistan and its internal repression , was there any other views Peter Tatchell had in regards to Soviet Union ?	2		1	1			

Table 10: The case study of human annotation

Hard	
Models	**Generated Rewrites**
Reference	In addition to Ezio Pinza 's role in La Vestale and his performance of Don Giovanni , are there any other interesting aspects about this article ?
S	Besides Ezio Pinza 's operas , are there any other interesting aspects about this article ?
Mix-gold	Besides Ezio Pinza singing Don Giovanni in Spontini 's La vestale , are there any other interesting aspects about this article ?
SLAF	Are there any other interesting aspects about this article besides Ezio Pinza 's operas ?
SLAD	Are there any other interesting aspects about this article besides Ezio Pinza 's operas ?
Reference	did the scathing review by Saibal Chatterjee have a bad effect on Kapoor 's future work ?
S	did Phata Poster Nikhla Hero have a bad effect on Shahid Kapoor 's future work ?
Mix-gold	Did the negative review of Phata Poster Nikhla Hero have a bad effect on Shahid Kapoor 's future work ?
SLAF	did the comedy Phata Poster Nikhla Hero have a bad effect on Shahid Kapoor 's future work ?
SLAD	Did Phata Poster Nikhla Hero have a bad effect on Shahid Kapoor 's future work ?
Reference	Besides his college honors , what other awards did Victor Davis Hanson win ?
S	What other awards did Victor Davis Hanson win besides his BA and PhD ?
Mix-gold	Besides being a Senior Fellow at the Hoover Institution and professor emeritus at California State University , Fresno , what other awards did Victor Davis Hanson win ?
SLAF	Besides the awards , what other awards did Victor Davis Hanson win ?
SLAD	What other awards did Victor Davis Hanson win other than being a Senior Fellow at the Hoover Institution and professor emeritus at California State University ?

Table 11: Generated rewrites on the hard class.

Medium	
Models	**Generated Rewrites**
Reference	Besides Do or Die any other chart toppers ?
S	Besides Do or Die , did Super Furry Animals have any other chart toppers ?
Mix-gold	Besides Do or Die , any other chart toppers ?
SLAF	Besides Guerrilla , any other chart toppers ?
SLAD	Besides " Do or Die " , any other chart toppers ?
Reference	What did Jeanine Pirro do after running for lieutenant governor ?
S	What did Jeanine Pirro do next after being Assistant District Attorney of Westchester County ?
Mix-gold	What did Jeanine Pirro do after writing appeals ?
SLAF	What did Jeanine Pirro do next after serving as Assistant District Attorney ?
SLAD	What did Jeanine Pirro do next after being appointed Assistant District Attorney of Westchester County ?
Reference	Besides trouble adapting to the faster pace of the Premiership , what else was bad about Juan Sebastián Verón 's time at Old Trafford ?
S	Besides having trouble adapting to the faster pace of the Premiership and being not allowed the same space and time on the ball , what else was bad about Juan Sebastián Verón 's time at Old Trafforr
Mix-gold	What else was bad about Juan Sebastián Verón 's time at Old Trafford besides adapting to the faster pace of the Premiership ?
SLAF	What else was bad about Juan Sebastián Verón 's time at Old Trafford other than the faster pace of the Premiership ?
SLAD	What else was bad about Juan Sebastián Verón 's time at Old Trafford other than being not allowed the same space and time on the ball ?

Table 12: Generated rewrites on the medium class.

Easy	
Models	**Generated Rewrites**
Reference	Did Robert Fripp win any awards ?
S	Did Robert Fripp win any awards for his music ?
Mix-gold	Did Robert Fripp win any awards ?
SLAF	Did Robert Fripp win any awards for Biography ?
SLAD	Did Robert Fripp win any awards ?
Reference	Are there any other interesting aspects about this article aside from Brown collaborating ?
S	Besides Chester Brown bringing Ed to an abrupt end in Yummy Fur # 18 , are there any other interesting aspects about this article ?
Mix-gold	Besides Ed , Are there any other interesting aspects about this article ?
SLAF	Besides Chester Brown bringing Ed to an abrupt end in Yummy Fur # 18 , are there any other interesting aspects about this article ?
SLAD	Besides Chester Brown bringing Ed to an abrupt end in Yummy Fur # 18 , are there any other interesting aspects about this article ?
Reference	What are some of the Green Day related works ?
S	What are some of Jason White 's Green Day related works ?
Mix-gold	What are some of the Green Day related works ?
SLAF	What are some of the Green Day related works ?
SLAD	What are some of Jason White 's Green Day related works ?

Table 13: Generated rewrites on the easy class.