Tell Me How to Ask Again: Question Data Augmentation with Controllable Rewriting in Continuous Space

Dayiheng Liu[♠*] **Yeyun Gong**[†] **Jie Fu**[◇] **Yu Yan**[‡]
Jiusheng Chen[‡] **Jiancheng Lv**[♠] **Nan Duan**[†] **Ming Zhou**[†]
[♠]College of Computer Science, Sichuan University [†]Microsoft Research Asia
[◇] Mila [‡]Microsoft
losinuris@gmail.com

Abstract

In this paper, we propose a novel data augmentation method, referred to as Controllable Rewriting based Question Data Augmentation (CRQDA), for machine reading comprehension (MRC), question generation, and question-answering natural language inference tasks. We treat the question data augmentation task as a constrained question rewriting problem to generate context-relevant, high-quality, and diverse question data samples. CRQDA utilizes a Transformer autoencoder to map the original discrete question into a continuous embedding space. It then uses a pre-trained MRC model to revise the question representation iteratively with gradient-based optimization. Finally, the revised question representations are mapped back into the discrete space, which serve as additional question data. Comprehensive experiments on SQuAD 2.0, SQuAD 1.1 question generation, and QNLI tasks demonstrate the effectiveness of CRQDA[1].

1 Introduction

Data augmentation (DA) is commonly used to improve the generalization ability and robustness of models by generating more training examples. Compared with the DA used in the fields of computer vision (Krizhevsky et al., 2012; Szegedy et al., 2015; Cubuk et al., 2019) and speech processing (Ko et al., 2015), how to design effective DA tailored to natural language processing (NLP) tasks remains a challenging problem. Unlike the general image DA techniques such as rotation and cropping, it is more difficult to synthesize new high-quality and diverse text.

Recently, some textual DA techniques have been proposed for NLP, which mainly focus on text classification and machine translation tasks. One way is directly modifying the text data locally with word deleting, word order changing, and word replacement (Fadaee et al., 2017; Kobayashi, 2018; Wei and Zou, 2019; Wu et al., 2019). Another popular way is to utilize the generative model to generate new text data, such as back-translation (Sennrich et al., 2016; Yu et al., 2018), data noising technique (Xie et al., 2017), and utilizing pre-trained language generation model (Kumar et al., 2020; Anaby-Tavor et al., 2020).

Machine reading comprehension (MRC) (Rajpurkar et al., 2018), question generation (QG) (Du et al., 2017; Zhao et al., 2018) and, question-answering natural language inference (QNLI) (Demszky et al., 2018; Wang et al., 2018) are receiving attention in NLP community. MRC requires the model to find the answer given a paragraph[2] and a question, while QG aims to generate the question for a given paragraph with or without a given answer. Given a question and a sentence in the relevant paragraph, QNLI requires the model to infer whether the sentence contains the answer to the question. Because the above tasks require the model to reason about the question-paragraph pair, existing textual DA methods that directly augment question or paragraph data alone may result in irrelevant question-paragraph pairs, which cannot improve the downstream model performance.

Question data augmentation (QDA) aims to automatically generate context-relevant questions to further improve the model performance for the above tasks (Yang et al., 2019; Dong et al., 2019). Existing QDA methods mainly employ the round-trip

[*] Work is done during internship at Microsoft Research Asia.

[1] The source code and dataset will be available at https://github.com/dayihengliu/CRQDA.

[2] It can also be a document span or a passage. For notational simplicity, we use the "paragraph" to refer to it in the rest of this paper.

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 5798–5810,
November 16–20, 2020. ©2020 Association for Computational Linguistics

consistency (Alberti et al., 2019; Dong et al., 2019) to synthesize answerable questions. However, the round-trip consistency method is not able to generate context-relevant unanswerable questions, where MRC with unanswerable questions is a challenging task (Rajpurkar et al., 2018; Kwiatkowski et al., 2019). Zhu et al. (2019) firstly study unanswerable question DA, which relies on annotated plausible answer to constructs a small pseudo parallel corpus of answerable-to-unanswerable questions for unanswerable question generation. Unfortunately, most question answering (QA) and MRC datasets do not provide such annotated plausible answers.

Inspired by the recent progress in controllable text revision and text attribute transfer (Wang et al., 2019; Liu et al., 2020), we propose a new QDA method called **C**ontrollable **R**ewriting based **Q**uestion **D**ata **A**ugmentation (**CRQDA**), which can generate both new context-relevant answerable questions and unanswerable questions. The main idea of CRQDA is to treat the QDA task as a constrained question rewriting problem. Instead of revising *discrete* question directly, CRQDA aims to revise the original questions in a *continuous embedding space* under the guidance of a pre-trained MRC model. There are two components of CRQDA: (i) A Transformer-based autoencoder whose encoder maps the question into a latent representation. Then its decoder reconstructs the question from the latent representation. (ii) A MRC model, which is pre-trained on the original dataset. This MRC model is used to *tell* us how to revise the question representation so that the reconstructed new question is a context-relevant unanswerable or answerable question. The original question is first mapped into a continuous embedding space. Next, the pre-trained MRC model provides the guidance to revise the question representation iteratively with gradient-based optimization. Finally, the revised question representations are mapped back into the discrete space, which act as the additional question data for training.

In summary, our contributions are as follows: (1) We propose a novel controllable rewriting based QDA method, which can generate additional high-quality, context-relevant, and diverse answerable and unanswerable questions. (2) We compare the proposed CRQDA with state-of-the-art textual DA methods on SQuAD 2.0 dataset, and CRQDA outperforms all those strong baselines consistently. (3) In addition to MRC tasks, we further apply CRQDA to question generation and QNLI tasks, and comprehensive experiments demonstrate its effectiveness.

2 Related Works

Recently, textual data augmentation has attracted a lot of attention. One popular class of textual DA methods is confined to locally modifying text in the discrete space to synthesize new data. Wei and Zou (2019) propose a universal DA technique for NLP called easy data augmentation (EDA), which performs synonym replacement, random insertion, random swap, or random deletion operation to modify the original text. Jungiewicz and Smywinski-Pohl (2019) propose a word synonym replacement method with WordNet. Kobayashi (2018) relies on word paradigmatic relations. More recently, CBERT (Wu et al., 2019) retrofits BERT (Devlin et al., 2018) to conditional BERT to predict the masked tokens for word replacement. These DA methods are mainly designed for the text classification tasks.

Unlike modifying a few local words, another commonly used textual DA way is to use a generative model to generate the entire new textual samples, including using variational autoencodes (VAEs) (Kingma and Welling, 2013; Rezende et al., 2014), generative adversarial networks (GANs) (Tanaka and Aranha, 2019), and pre-trained language generation models (Radford et al., 2019; Kumar et al., 2020; Anaby-Tavor et al., 2020). Back-translation (Sennrich et al., 2016; Yu et al., 2018) is also a major way for textual DA, which uses machine translation model to translate English sentences into another language (e.g., French), and back into English. Besides, data noising techniques (Xie et al., 2017; Marivate and Sefara, 2019) and paraphrasing (Kumar et al., 2019) are proposed to generate new textual samples. All the methods mentioned above usually generate individual sentences separately. For QDA of MRC, QG, and QNLI tasks, these DA approaches cannot guarantee the generating question are relevant to the given paragraph. In order to generate context-relevant answerable and unanswerable questions, our CRQDA method utilizes a pre-trained MRC as guidance to revise the question in continuous embedding space, which can be seen as a special *constrained* paraphrasing method for QDA.

Question generation (Heilman and Smith, 2010; Du et al., 2017; Zhao et al., 2018; Zhang and

Bansal, 2019) is attracting attention in the field of natural language generation (NLG). However, most previous works are not designed for QDA. That is, they do not aim to generate context-relevant questions for improving downstream model performance. Compared to QG, QDA is relatively unexplored. Recently, some works (Alberti et al., 2019; Dong et al., 2019) utilize round-trip consistency technique to synthesize answerable questions. They first use a generative model to generate the question with the paragraph and answer as model input, and then use a pre-trained MRC model to filter the synthetic question data. However, they are unable to generate context-relevant unanswerable questions. It should be noted that our method and round-trip consistency are orthogonal. CRQDA can also rewrite the synthetic question data by other methods to obtain new answerable and unanswerable question data. Unanswerable QDA is firstly explored in Zhu et al. (2019), which constructs a small pseudo parallel corpus of paired answerable and unanswerable questions and then generates relevant unanswerable questions in a supervised manner. This method relies on annotated plausible answers for the unanswerable questions, which does not exist in most QA and MRC datasets. Instead, our method rewrites the original answerable question to a relevant unanswerable question in an unsupervised paradigm, which can also rewrite the original answerable question to another new relevant answerable question.

Our method is inspired by the recent progress on controllable text revision and text attribute transfer (Wang et al., 2019; Liu et al., 2020). However, our approach differs in several ways. First, those methods are used to transfer the attribute of the single sentence alone, but our method considers the given paragraph to rewrite the context-relevant question. Second, existing methods jointly train an attribute classifier to revise the sentence representation, while our method unitizes a pre-trained MRC model that shares the embedding space with autoencoder as the guidance to revise the question representation. Finally, the generated questions by our method serve as augmented data can benefit the downstream tasks.

3 Methodology

3.1 Problem Formulation

We consider an extractive MRC dataset $\mathcal{D}$, such as SQuAD 2.0 (Rajpurkar et al., 2018), which has $|\mathcal{D}|$ 5-tuple data: (q, d, s, e, t), where $|\mathcal{D}|$ is the data size, $q = \{q_1, ..., q_n\}$ is a tokenized question with length n, $d = \{d_1, ..., d_m\}$ is a tokenized paragraph with length m, $s, e \in \{0, 1, ..., m-1\}$ are inclusive indices pointing to the start and end of the answer span, and $t \in \{0, 1\}$ represents whether the question q is answerable or unanswerable with d. Given a data tuple (q, d, s, e, t), we aim to rewrite q to a new answerable or unanswerable question q' and obtain a new data tuple (q', d, s, e, t') that fulfills certain requirements: (*i*) The generated answerable question can be answered with the answer span (s, e) with d, while the generated unanswerable question cannot be answered with d. (*ii*) The generated question should be relevant to the original question q and paragraph d. (*iii*) The augmented dataset $\mathcal{D}'$ should be able to further improve the performance of the MRC models.

3.2 Method Overview

Figure 1 shows the overall architecture of CRQDA. The proposed model consists of two components: a pre-trained language model based MRC model as described in § 3.3, and a Transformer-based autoencoder as introduced in § 3.4. Given a question q from the original dataset $\mathcal{D}$, we first map the question q into a continuous embedding space. Then we revise the question embeddings by gradient-based optimization with the guidance of the MRC model (§ 3.5). Finally, the revised question embeddings are inputted to the Transformer-based autoencoder to generate a new question data.

3.3 Pre-trained Language Model based MRC Model

In this paper, we adopt the pre-trained language model (e.g., BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019b)) based MRC models as our MRC baseline model. Without loss of generality, we take the BERT-based MRC model as an example to introduce our method, which is shown in the left part of Figure 1.

Following Devlin et al. (2018), given a data tuple (q, d, s, e, t), we concatenate a “[CLS]” token, the tokenized question q with length n, a “[SEP]” token, the tokenized paragraph d with length m, and a final “[SEP]” token. We feed the resulting sequence into the BERT model. The question q and paragraph d are first mapped into two sequence of embeddings:

$$\mathbf{E}^q, \mathbf{E}^d = \mathrm{BertEmbedding}(q, d), \qquad (1)$$

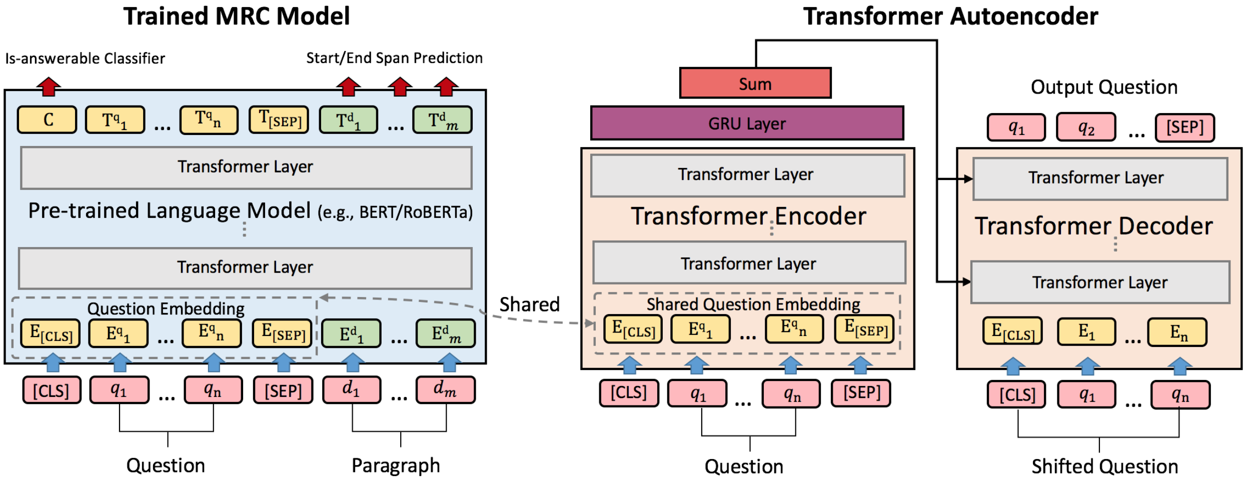


Figure 1: The architecture of CRQDA.

where $\mathrm{BertEmbedding}(\cdot)$ denotes the BERT embedding layer which sums the corresponding token, segment, and position embeddings, $\mathbf{E}^q \in \mathbb{R}^{(n+2)\times h}$ and $\mathbf{E}^d \in \mathbb{R}^{m\times h}$ represent the question embedding and the paragraph embedding.

$\mathbf{E}^q$ and $\mathbf{E}^d$ are further fed into BERT layers which consist of multiple Transformer layers (Vaswani et al., 2017) to obtain the final hidden representations $\{\mathbf{C}, \mathbf{T}^q_1, ..., \mathbf{T}^q_n, \mathbf{T}_{[SEP]}, \mathbf{T}^d_1, ..., \mathbf{T}^d_m\}$ as shown in Figure 1. The representation vector $\mathbf{C} \in \mathbb{R}^h$ corresponding to the first input token ([CLS]) are fed into a binary classification layer to output the probability of whether the question is answerable:

$$P_a(\text{is-answerable}) = \mathrm{Sigmoid}(\mathbf{C}\mathbf{W}_c^T + \mathbf{b}_c), \quad (2)$$

where $\mathbf{W}_c \in \mathbb{R}^{2\times h}$ and $\mathbf{b}_c \in \mathbb{R}^2$ are trainable parameters. The final hidden representations of paragraph $\{\mathbf{T}^d_1, ..., \mathbf{T}^d_m\} \in \mathbb{R}^{m\times h}$ are inputted into two classifier layer to output the probability of the start position and the end position of the answer span:

$$P_s(i =< \text{start} >) = \mathrm{Sigmoid}(\mathbf{T}^d_i \mathbf{W}_s^T + b_s), \quad (3)$$
$$P_e(i =< \text{end} >) = \mathrm{Sigmoid}(\mathbf{T}^d_i \mathbf{W}_e^T + b_e), \quad (4)$$

where $\mathbf{W}_s \in \mathbb{R}^{1\times h}$, $\mathbf{W}_e \in \mathbb{R}^{1\times h}$, $b_s \in \mathbb{R}^1$, and $b_e \in \mathbb{R}^1$ are trainable parameters.

For the data tuple (q, d, s, e, t), the total loss of MRC model can be written as

$$\begin{aligned}\mathcal{L}_{\text{mrc}} &= \lambda\mathcal{L}_a(t) + \mathcal{L}_s(s) + \mathcal{L}_e(e), \quad (5)\\ &= -\lambda \log P_a(t) - \log P_s(s) - \log P_e(e),\end{aligned}$$

where λ is a hyper-parameter.

3.4 Transformer-based Autoencoder

As shown in the right part of Figure 1, the original question q is firstly mapped into question embedding $\mathbf{E}^q$ with the BERT embedding layer. It should be noted that the Transformer encoder and the pre-trained MRC model share[3] the parameters of the embedding layer, which makes the question embedding of the two models in the same continuous embedding space.

We obtain the encoder hidden states $\mathbf{H}_{enc} \in \mathbb{R}^{(n+2)\times h}$ from the Transformer encoder. The objective of the Transformer autoencoder is to reconstruct the input question itself, which is optimized with cross-entropy (Dai and Le, 2015). A trivial solution of the autoencoder would be to simply copy tokens in the decoder side. To avoid this, we do not directly feed the whole $\mathbf{H}_{enc}$ to the decoder, but use an RNN-GRU (Cho et al., 2014) layer with sum pooling to obtain a latent vector $\mathbf{z} \in \mathbb{R}^h$. Then we feed $\mathbf{z}$ to the decoder to reconstruct the question, which follows Wang et al. (2019).

$$\mathbf{H}_{enc} = \mathrm{TransformerEncoder}(q), \quad (6)$$
$$\mathbf{z} = \mathrm{Sum}(\mathrm{GRU}(\mathbf{H}_{enc})), \quad (7)$$
$$\hat{q} = \mathrm{TransformerDecoder}(\mathbf{z}). \quad (8)$$

We can train the autoencoder on the question data of $\mathcal{D}$ or pre-train it on other large-scale corpora, such as BookCorpus (Zhu et al., 2015) and English Wikipedia.

[3] The parameters of the Transformer encoder's embedding layer are copied from the pre-trained MRC, and are fixed during training.

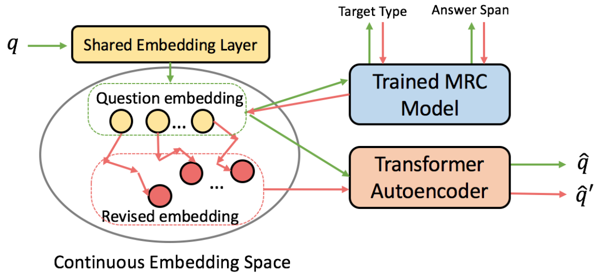


Figure 2: The question rewriting process of CRQDA.

Algorithm 1 Question Rewriting with Gradient-based Optimization.

```
Input: Data tuple $(q, d, s, e, t)$; Original question embedding
    $\mathbf{E}^q$; pre-trained MRC model and Transformer autoencoder; A set of step size $S_\eta = \{\eta_i\}$; Step size decay
    coefficient $\beta_s$; the target answerable or unanswerable label $t'$; Threshold $\beta_t, \beta_a, \beta_b$;
Output: a set of new answerable and unanswerable question
    data tuples $\mathcal{D}' = \{(\hat{q}', d, s, e, t'), .., (\hat{q}', d, s, e, t)\}$;
 1: $\mathcal{D}' = \{\}$;
 2: for each $\eta \in S_\eta$ do
 3:     for max-steps do
 4:         revise $\mathbf{E}^{q'}$ by Eq. (10) or Eq. (9)
 5:         $\hat{q}' = $ TransformerAutoencoder $\left(\mathbf{E}^{q'}\right)$
 6:         if $P_a(t') > \beta_t$ and $\mathcal{J}(q, \hat{q}') \in [\beta_a, \beta_b]$ then
 7:             add $(\hat{q}', d, s, e, t')$ to $\mathcal{D}'$;
 8:         end if
 9:         $\eta = \beta_s \eta$;
10:     end for
11: end for
12: return $\mathcal{D}'$;
```

3.5 Rewriting Question with Gradient-based Optimization

As mentioned above, the question embedding of the Transformer encoder and pre-trained MRC are in the same continuous embedding space, where we can revise the question embedding with the gradient guidance by MRC model. The revised question embedding $\mathbf{E}^{q'}$ is fed into Transformer autoencoder to generate a new question data $\hat{q}'$.

Figure 2 illustrates the process of question rewriting. Specifically, we take the process of rewriting an answerable question to a relevant unanswerable question as an example to present the process. Given an answerable question q, the goals of the rewriting are: (I) the revised question embedding should make the pre-trained MRC model predict the question from answerable to unanswerable with the paragraph d; (II) The modification size of $\mathbf{E}^q$ should be adaptive to prevent the revision of $\mathbf{E}^q$ from falling into local optimum; (III) The revised question $\hat{q}'$ should be similar to the original q, which helps to improve the robustness of the model.

For goal-(I), we take the label $t' = 0$, which denotes the label of question is unanswerable, to calculate the loss $\mathcal{L}_a(t')$ and the gradient of $\mathbf{E}^q$ by the pre-trained MRC model (see the red line in Figure 2). We iteratively revise $\mathbf{E}^q$ with gradients from the pre-trained MRC model until the MRC model predicts the question is unanswerable with the revised $\mathbf{E}^{q'}$ as its input, which means the $P_a(t'|\mathbf{E}^{q'})$ is large than a threshold β_t. Note that here we use the gradient to only modify $\mathbf{E}^q$, and all the model parameters during rewriting process are fixed. The process of each iteration can be written as:

$$\mathbf{E}^{q'} = \mathbf{E}^q - \eta(\nabla_{\mathbf{E}^q} \mathcal{L}_a(t')), \tag{9}$$

where η is the step size. Similarly, we can revise the $\mathbf{E}^q$ of a data tuple (q, d, s, e, t) to generate a new answerable question whose answer is still the original answer span (s, e) as follows:

$$\mathbf{E}^{q'} = \mathbf{E}^q - \eta\left(\nabla_{\mathbf{E}^q}(\lambda \mathcal{L}_a(t) + \mathcal{L}_s(s) + \mathcal{L}_e(e))\right). \tag{10}$$

Rewriting the answerable question into another answerable question can be seen as a special constrained paraphrasing, which requires that the question after the paraphrasing is context-relevant answerable and its answer remains unchanged.

For goal-(II), we follow (Wang et al., 2019) to employ the dynamic-weight-initialization method to allocate a set of step-sizes $S_\eta = \{\eta_i\}$ as initial step-sizes. For each initial step-size, we perform a pre-defined max-step revision with the step size value decay (corresponds to Algorithm 1 line 2-11) to find the target question embedding. For goal-(III), we select the $\hat{q}'$ whose unigram word overlap rate with the original question q is within a threshold range $[\beta_a, \beta_b]$. The unigram word overlap is computed by:

$$\mathcal{J}(q, \hat{q}') = \frac{\text{count}(w_q \cap w_{\hat{q}})}{\text{count}(w_q \cup w_{\hat{q}})}, \tag{11}$$

here w_q is the word in q and $w_{\hat{q}}$ is the word in $\hat{q}'$. The whole question rewriting procedure is summarized in Algorithm 1.

4 Experiments

In this section, we describe the experimental details and results. We first conduct the experiment on the SQuAD 2.0 dataset (Rajpurkar et al., 2018) to compare CRQDA with other strong baselines, which is reported in § 4.1. The ablation study and further analysis are provided in § 4.2. Then we evaluate our method on additional two tasks including question generation on SQuAD 1.1 dataset (Rajpurkar et al., 2016) in § 4.3, and question-answering language inference on QNLI dataset (Wang et al., 2018) in § 4.4.

Methods	EM	F1
BERT$_{\text{large}}$ (Devlin et al., 2018) (original)	78.7	81.9
+ EDA (Wei and Zou, 2019)	78.3	81.6
+ Back-Translation (Yu et al., 2018)	77.9	81.2
+ Text-VAE (Liu et al., 2019a)	75.3	78.6
+ AE with Noise	76.7	79.8
+ 3M synth (Alberti et al., 2019)	80.1	82.8
+ UNANSQ (Zhu et al., 2019)	80.0	83.0
+ CRQDA (ours)	**80.6**	**83.3**

Table 1: Comparison results on SQuAD 2.0.

4.1 SQuAD

The extractive MRC benchmark SQuAD 2.0 dataset contains about 100,000 answerable questions and over 50,000 unanswerable questions. Each question is paired with a Wikipedia paragraph.

Implementation Based on *RobertaForQuestionAnswering*[4] model of Huggingface (Wolf et al., 2019), we train a RoBERTa$_{\text{large}}$ model on SQuAD 2.0 as the pre-trained MRC model for CRQDA. The hyper-parameters are the same as the original paper (Liu et al., 2019b). For training the autoencoder, we copy the word embedding parameters of the pre-trained MRC model to autoencoder and fix them during training. Both of its encoder and decoder consist of 6-layer Transformers, where the inner dimension of feed-forward networks (FFN), hidden state size, and the number of attention head are set to 4096, 1024, and 16.

The autoencoder trains on BookCorpus (Zhu et al., 2015) and English Wikipedia (Devlin et al., 2018). The sequence length, batch size, learning rate, and training steps are set to 64, 256, 5e-5 and 100,000. For each original answerable data, we use CRQDA to generate new unanswerable question data, resulting in about 220K data samples (including the original data samples). The hyper-parameter of β_s, β_t, β_a, β_b, and max-step are set to 0.9, 0.5, 0.5, 0.99, and 5, respectively.

Baselines We compare our CRQDA against the following baselines: (1) **EDA** (Wei and Zou, 2019): it augments question data by performing synonym replacement, random insertion, random swap, or random deletion operation. We implement EDA with their source code[5] to synthesize a new question data for each question of SQuAD 2.0; (2) **Back-Translation** (Yu et al., 2018; Prabhumoye et al., 2018): it uses machine translation model to translate questions into French and back into English. We implement Back-Translation based on the source code[6] to generate a new question data for each original question; (3) **Text-VAE** (Bowman et al., 2016; Liu et al., 2019a): it uses RNN-based VAE to generate a new question data for each question of SQuAD 2.0. The implementation is based on the source code[7]; (4) **AE with Noise**: it uses the same autoencoder of CRQDA for question data rewriting. The only difference is that the autoencoder cannot utilize the MRC gradient but only uses a noise (sampled from Gaussian distribution) to revise the question embedding. This experiment is designed to show necessity of the pre-trained MRC. (5) **3M synth** (Alberti et al., 2019): it employs round-trip consistency technique to synthesize 3M questions on SQuAD 2.0; (6) **UNANSQ** (Zhu et al., 2019): it employs a pair-to-sequence model to generate 69,090 unanswerable questions. Following previous methods (Zhu et al., 2019; Alberti et al., 2019), we use each augmented dataset to fine-tune BERT$_{\text{large}}$ model, where the implementation is also based on Huggingface.

Results For SQuAD 2.0, Exact Match (EM) and F1 score are used as evaluation metrics. The results on SQuAD 2.0 development set are shown in Table 1. The popular textual DA methods (including EDA, Back-Translation, Text-VAE, and AE with Noised), do not improve the performance of the MRC model. One possible reason might be that they introduce detrimental noise to the training process as they augment question data without considering the paragraphs and the associated answers. In sharp contrast, the QDA methods (including 3M synth, UNANSQ, and CRQDA) improve the model performance. Besides, our CRQDA outperforms all the strong baselines, which brings about 1.9 absolute EM score and 1.5 F1 score improvement

[4] https://github.com/huggingface/transformers.

[5] https://github.com/jasonwei20/eda_nlp.

[6] https://github.com/shrimai/Style-Transfer-Through-Back-Translation.

[7] https://github.com/dayihengliu/Mu-Forcing-VRAE.

based on $\text{BERT}_{\text{large}}$. We provide some augmented data samples of each baseline in **Appendix A**.

4.2 Ablation and Analysis

Our ablation study and further analysis are designed for answering the following questions: **Q1**: How useful is the augmented data synthesized by our method if trained by other MRC models? **Q2**: How does the choice of the corpora for autoencoder training influence the performance? **Q3**: How do different CRQDA augmentation strategies influence the model performance?

Methods	EM	F1
$\text{BERT}_{\text{base}}$	73.7	76.3
+ CRQDA	**75.8** (+2.1)	**78.7** (+2.4)
$\text{BERT}_{\text{large}}$	78.7	81.9
+ CRQDA	**80.6** (+1.9)	**83.3** (+1.4)
$\text{RoBERTa}_{\text{base}}$	78.6	81.6
+ CRQDA	**80.2** (+1.6)	**83.1** (+1.5)
$\text{RoBERTa}_{\text{large}}$	86.0	88.9
+ CRQDA	**86.4** (+0.4)	**89.5** (+0.6)

Table 2: Results of different MRC models with CRQDA on SQuAD 2.0.

To answer the first question (**Q1**), we use the augmented SQuAD 2.0 dataset in § 4.1 to train different MRC models ($\text{BERT}_{\text{base}}$, $\text{BERT}_{\text{large}}$, $\text{RoBERTa}_{\text{base}}$, and $\text{RoBERTa}_{\text{large}}$). The hyper-parameters and implementation are based on Huggingface (Wolf et al., 2019). The results are presented in Table 2. We can see that CRQDA can improve the performance of each MRC model, yielding 2.4 absolute F1 improvement with $\text{BERT}_{\text{base}}$ model and 1.5 absolute F1 improvement with $\text{RoBERTa}_{\text{base}}$. Besides, although we use a $\text{RoBERTa}_{\text{large}}$ model to guide the rewriting of question data, the augmented dataset can further improve its performance.

Methods	EM	F1	R-L	B4
$\text{BERT}_{\text{base}}$	73.7	76.3	-	-
+ CRQDA (SQuAD 2)	74.8	77.7	82.9	59.6
+ CRQDA (2M ques)	75.3	78.2	97.8	94.7
+ CRQDA (Wiki)	**75.8**	**78.7**	99.3	98.4
+ CRQDA (Wiki+Mask)	75.4	78.4	**99.7**	**99.4**

Table 3: Results of training autoencoder on different corpora. R-L is short for ROUGE-L, and B4 is short for BLEU-4.

For the second question (**Q2**), we conduct experiments to use the following different corpora to train the autoencoder of CRQDA: (1) **SQuAD 2.0**: we use all the questions from the training set of SQuAD 2.0; (2) **2M questions**: we collect 2,072,133 questions from the training sets of several MRC and QA datasets, including SQuAD2.0, Natural Questions, NewsQA (Trischler et al., 2016), QuAC (Choi et al., 2018), TriviaQA (Joshi et al., 2017), CoQA (Reddy et al., 2019), HotpotQA (Yang et al., 2018), DuoRC (Saha et al., 2018), and MS MARCO (Bajaj et al., 2016); (3) **Wiki**: We use the large-scale corpora English Wikipedia and BookCorpus (Zhu et al., 2015) to train autoencoder; (4) **Wiki+Mask**: We also train autoencoder on English Wikipedia and BookCorpus as **Wiki**. In addition, we randomly mask 15% tokens of the encoder inputs with a special token, which is similar to the mask strategy used in (Devlin et al., 2018; Song et al., 2019).

We firstly measure the reconstruction performance of the autoencoders on the question data of SQuAD 2.0 development set. We use BLEU-4 (Papineni et al., 2002) and ROUGE-L (Lin, 2004) metrics for evaluation. Then we use these autoencoders for the CRQDA question rewriting with the same settings in § 4.1. These augmented SQuAD 2.0 datasets are used to fine-tune $\text{BERT}_{\text{base}}$ model. We report the performance of fine-tuned $\text{BERT}_{\text{base}}$ model in Table 3. It can be observed that with more training data, the reconstruction performance of autoencoder is better. Also, the performance of fine-tuned $\text{BERT}_{\text{base}}$ model is better. When trained with Wiki and Wiki+Mask, the autoencoders can reconstruct almost all questions well. The reconstruction performance of model trained with Wiki+Mask performs the best. However, the fine-tuned $\text{BERT}_{\text{base}}$ model with autoencoder trained on Wiki performs better than that trained on Wiki+Mask. The reason might be that the autoencoder trained with denoising task will be insensitive to the word embedding revision of CRQDA. In other words, some revisions guided by the MRC gradients might be filtered out as noises by the autoencoder, which is trained with a denoising task.

Methods	EM	F1
$\text{RoBERTa}_{\text{large}}$ (Liu et al., 2019b)	86.00	88.94
+ CRQDA (*unans*, $\beta_a = 0.7$)	86.39	89.31
+ CRQDA (*unans*, $\beta_a = 0.5$)	**86.43**	**89.50**
+ CRQDA (*unans*, $\beta_a = 0.3$)	86.26	89.35
+ CRQDA (*ans*)	86.22	89.30
+ CRQDA (*ans+unans*)	86.36	89.38

Table 4: Results of using differnt CRQDA augmented datasets for MRC training.

For the last question **Q3**, we use CRQDA for question data augmentation with different settings. For each answerable original question data sample from the training set of SQuAD 2.0, we use CRQDA to generate both answerable and unanswerable question examples. Then the augmented *unanswerable* question data (*unans*), the augmented *answerable* question data (*ans*), and all of them (*ans* + *unans*) are used to fine-tune RoBERTa$_{\text{large}}$ model. To further analyze the effect of β_a (a larger β_a value means that the generated questions are closer to the original question in the discrete space), we use different $\beta_a = 0.3, 0.5, 0.7$ for question rewriting. The results are reported in Table 4. It can be observed that the MRC achieves the best performance when $\beta_a = 0.5$. Moreover, all of *unans*, *ans*, and *ans* + *unans* augmented datasets can further improve the performance. However, we find that the RoBERTa$_{\text{large}}$ model fine-tuned on *ans* + *unans* performs worse than fine-tuned on *unans* only. The result is mixed in that using more augmented data is not always beneficial.

Method	B4	MTR	R-L
UniLM (Dong et al., 2019)	22.12	25.06	51.07
ProphetNet (Yan et al., 2020)	25.01	26.83	52.57
ProphetNet + CRQDA	**25.95**	**27.40**	**53.15**
UniLM (Dong et al., 2019)	23.75	25.61	52.04
ProphetNet (Yan et al., 2020)	26.72	27.64	53.79
ProphetNet + CRQDA	**27.21**	**27.81**	**54.21**

Table 5: Results on SQuAD 1.1 question generation. B4 is short for BLEU-4, MTR is short for METEOR, and R-L is short for ROUGE-L. The first block follows the data split in Du et al. (2017), while the second block is the same as in Zhao et al. (2018).

4.3 Question Generation

Answer-aware question generation task (Zhou et al., 2017) aims to generate a question for the given answer span with a paragraph. We apply our CRQDA method to SQuAD 1.1 (Rajpurkar et al., 2016) question generation task to further evaluate CRQDA. The settings of CRQDA are the same as in § 4.1 and § 4.2. The augmented answerable question dataset is used to fine-tune the ProphetNet (Yan et al., 2020) model which achieves the best performance on SQuAD 1.1 question generation task. The implementation is based on their source code[8]. We also compare with the previous state-of-the-art model UniLM (Dong et al., 2019). Following Yan et al. (2020), we use BLEU-4, METEOR (Banerjee and Lavie, 2005), and ROUGE-L metrics for evaluation, and we split the SQuAD 1.1 dataset into training, development and test set. We also report the results on the another data split setting as in Yan et al. (2020), which reverses the development set and test set. The results are shown in Table 5. We can see that CRQDA improves ProphetNet on all three metrics and achieves a new state-of-the-art on this task.

Methods	Accuracy
BERT$_{\text{large}}$ (Devlin et al., 2018)	92.3
BERT$_{\text{large}}$ + CRQDA	**93.0**

Table 6: Results on QNLI.

4.4 QNLI

Given a question and a context sentence, question-answering NLI asks the model to infer whether the context sentence contains the answer to the question. QNLI dataset (Wang et al., 2018) contains 105K data samples. We apply CRQDA to QNLI dataset to generate new entailment and non-entailment data samples. Note that this task does not include the MRC model, but uses a text entailment classification model. Similarly, we train a BERT$_{\text{large}}$ model based on the code of *BertForSequenceClassification* in Huggingface to replace the "pre-trained MRC model" of CRQDA to guide the question data rewriting. Following the settings in § 4.1, we use CRQDA to synthesize about 42K new data samples as augmented data. Note that we only rewrite the question but keep the paired context sentence unchanged. Then the augmented data and original dataset are used to fine-tine BERT$_{\text{large}}$ model. Table 6 shows the results. CRQDA increases the accuracy of the BERT$_{\text{large}}$ model by 0.7%, which also demonstrates the effectiveness of CRQDA.

5 Conclusion

In this work, we present a novel question data augmentation method, called CRQDA, for context-relevant answerable and unanswerable question generation. CRQDA treats the question data augmentation task as a constrained question rewriting problem. Under the guidance of a pre-trained MRC model, the original question is revised in a continuous embedding space with gradient-based optimization and then decoded back to the discrete space as a new question data sample. The experimental results demonstrate that CRQDA outperforms

[8] https://github.com/microsoft/ProphetNet.

other strong baselines on SQuAD 2.0. The CRQDA augmented datasets can improve multiple reading comprehension models. Furthermore, CRQDA can be used to improve the model performance on question generation and question-answering language inference tasks, which achieves a new state-of-the-art on the SQuAD 1.1 question generation task.

Acknowledgment

This work is supported in part by the National Natural Science Fund for Distinguished Young Scholar under Grant 61625204, and in part by the State Key Program of the National Science Foundation of China under Grant 61836006.

References

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic qa corpora generation with roundtrip consistency. *arXiv preprint arXiv:1906.05416*.

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. Do not have enough data? deep learning to the rescue! In *AAAI*.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *CoNLL*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. *arXiv preprint arXiv:1808.07036*.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. 2019. Autoaugment: Learning augmentation strategies from data. In *CVPR*.

Andrew M. Dai and Quoc V. Le. 2015. Semi-supervised sequence learning. In *NIPS*.

Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming question answering datasets into natural language inference datasets. *arXiv preprint arXiv:1809.02922*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *NeurIPS*.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. *arXiv preprint arXiv:1705.00106*.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. In *ACL*.

Michael Heilman and Noah A Smith. 2010. Good question! statistical ranking for question generation. In *HLT-NAACL*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Michal Jungiewicz and Aleksander Smywinski-Pohl. 2019. Towards textual data augmentation for neural networks: synonyms and maximum loss. *Computer Science*, 20.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. In *ICLR*.

Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. 2015. Audio augmentation for speech recognition. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *NAACL*.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*.

Ashutosh Kumar, Satwik Bhattamishra, Manik Bhandari, and Partha Talukdar. 2019. Submodular optimization-based diverse paraphrasing and its effectiveness in data augmentation. In *NAACL*.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. *arXiv preprint arXiv:2003.02245*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *TACL*, 7:453–466.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*.

Dayiheng Liu, Jie Fu, Yidan Zhang, Chris Pal, and Jiancheng Lv. 2020. Revision in continuous space: Unsupervised text style transfer without adversarial learning. In *ACL*.

Dayiheng Liu, Yang Xue, Feng He, Yuanyuan Chen, and Jiancheng Lv. 2019a. mu-forcing: Training variational recurrent autoencoders for text generation. *ACM Transactions on Asian and Low-Resource Language Information Processing*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Vukosi Marivate and Tshephisho Sefara. 2019. Improving short text classification through global augmentation methods. *arXiv preprint arXiv:1907.03752*.

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

Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. 2018. Style transfer through back-translation. In *ACL*.

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

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *ACL*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*.

Amrita Saha, Rahul Aralikatte, Mitesh M Khapra, and Karthik Sankaranarayanan. 2018. Duorc: Towards complex language understanding with paraphrased reading comprehension. *arXiv preprint arXiv:1804.07927*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *ACL*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. *arXiv preprint arXiv:1905.02450*.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *CVPR*.

Fabio Henrique Kiyoiti dos Santos Tanaka and Claus Aranha. 2019. Data augmentation using gans. *arXiv preprint arXiv:1904.09135*.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2016. Newsqa: A machine comprehension dataset. *arXiv preprint arXiv:1611.09830*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Ke Wang, Hang Hua, and Xiaojun Wan. 2019. Controllable unsupervised text attribute transfer via editing entangled latent representation. In *NeurIPS*.

Jason W Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *ACL*.

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

Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Conditional bert contextual augmentation. In *International Conference on Computational Science*.

Ziang Xie, Sida I Wang, Jiwei Li, Daniel Lévy, Aiming Nie, Dan Jurafsky, and Andrew Y Ng. 2017. Data noising as smoothing in neural network language models. In *ICLR*.

Yu Yan, Weizhen Qi, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. *arXiv preprint arXiv:2001.04063*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*.

Shiyue Zhang and Mohit Bansal. 2019. Addressing semantic drift in question generation for semi-supervised question answering. *arXiv preprint arXiv:1909.06356*.

Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. 2018. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *EMNLP*.

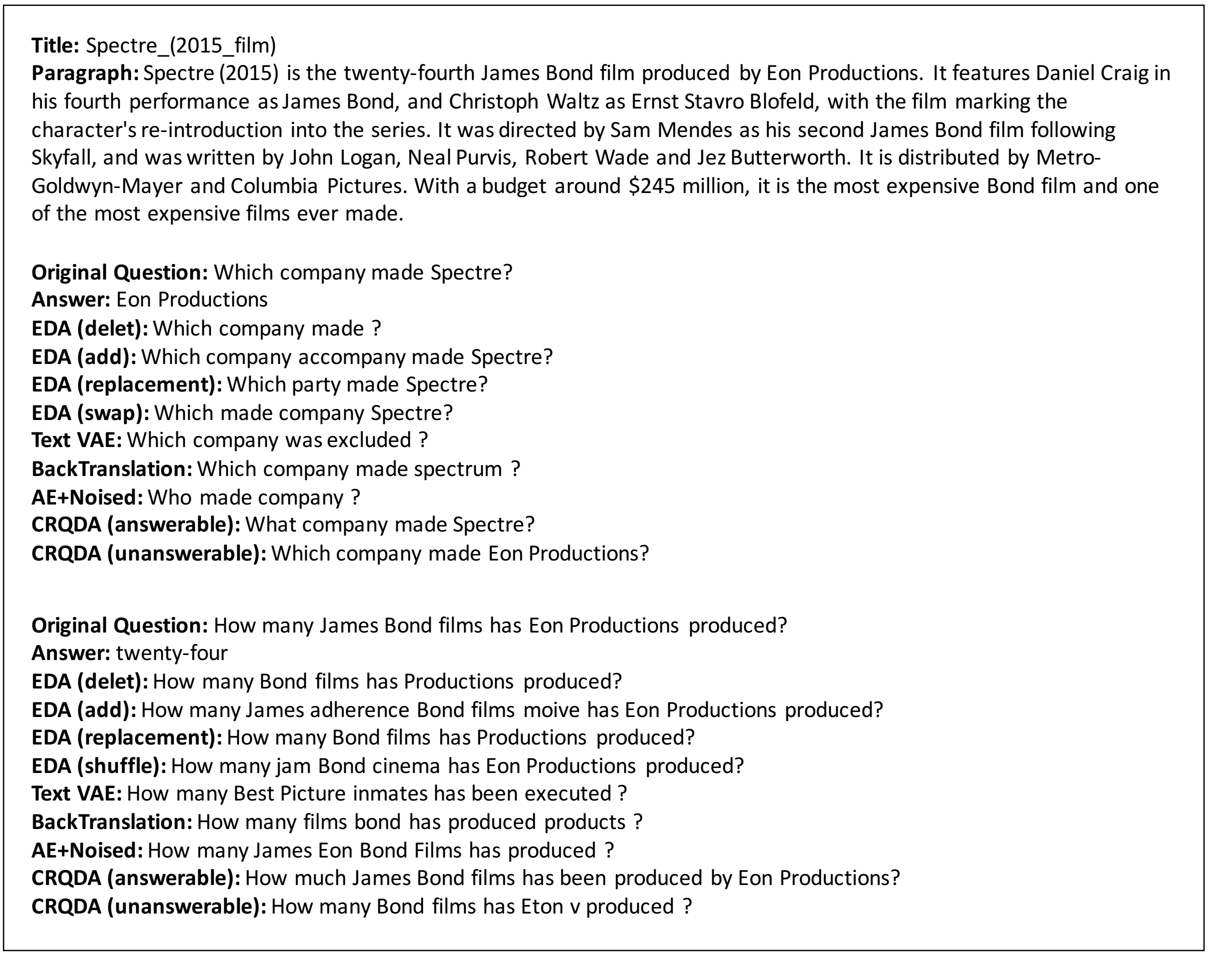
Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. 2017. Neural question generation from text: A preliminary study. In *National CCF Conference on Natural Language Processing and Chinese Computing*, pages 662–671.

Haichao Zhu, Li Dong, Furu Wei, Wenhui Wang, Bing Qin, and Ting Liu. 2019. Learning to ask unanswerable questions for machine reading comprehension. In *ACL*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

A Augmented Dataset

Figure 3 and Figure 4 provide some augmented data samples of each baseline on SQuAD 2.0. We can see that the baseline **EDA** tends to introduce noise which destroys the original sentence structure. The baselines of **Text VAE**, **BackTranslation** and **AE+Noised** often change some important words of the original question. This can cause the augmented question to miss the original key information and not to able to infer the original answer. In contrast, it can be observed that the generated answerable questions of **CRQDA** still maintain the key information for the original answer inference. Its generated unanswerable questions tend to introduce some context-relevant words to convert an original answerable question into an unanswerable one.

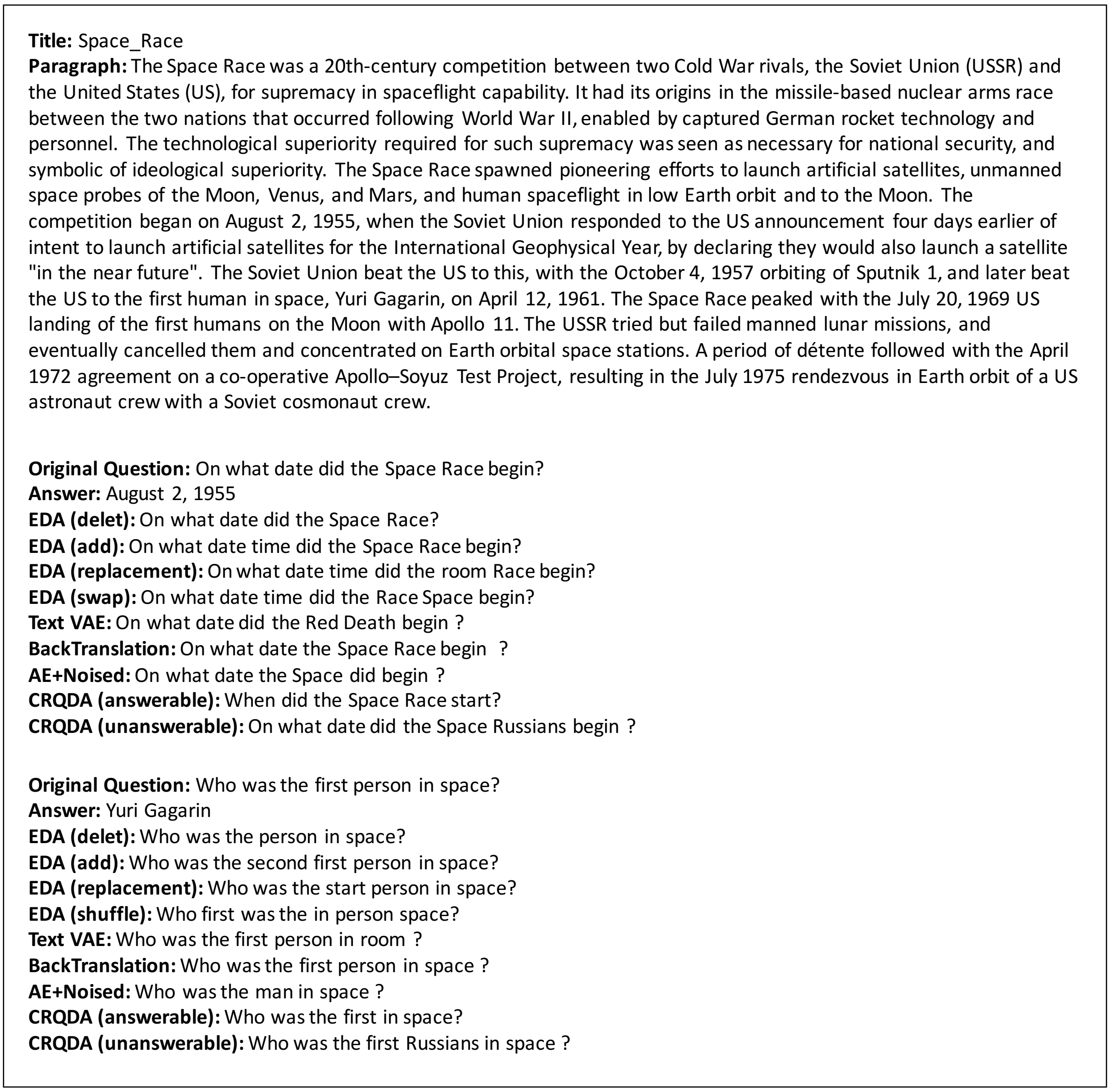


Title: Spectre_(2015_film)
Paragraph: Spectre (2015) is the twenty-fourth James Bond film produced by Eon Productions. It features Daniel Craig in his fourth performance as James Bond, and Christoph Waltz as Ernst Stavro Blofeld, with the film marking the character's re-introduction into the series. It was directed by Sam Mendes as his second James Bond film following Skyfall, and was written by John Logan, Neal Purvis, Robert Wade and Jez Butterworth. It is distributed by Metro-Goldwyn-Mayer and Columbia Pictures. With a budget around $245 million, it is the most expensive Bond film and one of the most expensive films ever made.

Original Question: Which company made Spectre?
Answer: Eon Productions
EDA (delet): Which company made ?
EDA (add): Which company accompany made Spectre?
EDA (replacement): Which party made Spectre?
EDA (swap): Which made company Spectre?
Text VAE: Which company was excluded ?
BackTranslation: Which company made spectrum ?
AE+Noised: Who made company ?
CRQDA (answerable): What company made Spectre?
CRQDA (unanswerable): Which company made Eon Productions?

Original Question: How many James Bond films has Eon Productions produced?
Answer: twenty-four
EDA (delet): How many Bond films has Productions produced?
EDA (add): How many James adherence Bond films moive has Eon Productions produced?
EDA (replacement): How many Bond films has Productions produced?
EDA (shuffle): How many jam Bond cinema has Eon Productions produced?
Text VAE: How many Best Picture inmates has been executed ?
BackTranslation: How many films bond has produced products ?
AE+Noised: How many James Eon Bond Films has produced ?
CRQDA (answerable): How much James Bond films has been produced by Eon Productions?
CRQDA (unanswerable): How many Bond films has Eton v produced ?

Figure 3: Augmented data samples on SQuAD 2.0.



Title: Space_Race
Paragraph: The Space Race was a 20th-century competition between two Cold War rivals, the Soviet Union (USSR) and the United States (US), for supremacy in spaceflight capability. It had its origins in the missile-based nuclear arms race between the two nations that occurred following World War II, enabled by captured German rocket technology and personnel. The technological superiority required for such supremacy was seen as necessary for national security, and symbolic of ideological superiority. The Space Race spawned pioneering efforts to launch artificial satellites, unmanned space probes of the Moon, Venus, and Mars, and human spaceflight in low Earth orbit and to the Moon. The competition began on August 2, 1955, when the Soviet Union responded to the US announcement four days earlier of intent to launch artificial satellites for the International Geophysical Year, by declaring they would also launch a satellite "in the near future". The Soviet Union beat the US to this, with the October 4, 1957 orbiting of Sputnik 1, and later beat the US to the first human in space, Yuri Gagarin, on April 12, 1961. The Space Race peaked with the July 20, 1969 US landing of the first humans on the Moon with Apollo 11. The USSR tried but failed manned lunar missions, and eventually cancelled them and concentrated on Earth orbital space stations. A period of détente followed with the April 1972 agreement on a co-operative Apollo–Soyuz Test Project, resulting in the July 1975 rendezvous in Earth orbit of a US astronaut crew with a Soviet cosmonaut crew.

Original Question: On what date did the Space Race begin?
Answer: August 2, 1955
EDA (delet): On what date did the Space Race?
EDA (add): On what date time did the Space Race begin?
EDA (replacement): On what date time did the room Race begin?
EDA (swap): On what date time did the Race Space begin?
Text VAE: On what date did the Red Death begin ?
BackTranslation: On what date the Space Race begin ?
AE+Noised: On what date the Space did begin ?
CRQDA (answerable): When did the Space Race start?
CRQDA (unanswerable): On what date did the Space Russians begin ?

Original Question: Who was the first person in space?
Answer: Yuri Gagarin
EDA (delet): Who was the person in space?
EDA (add): Who was the second first person in space?
EDA (replacement): Who was the start person in space?
EDA (shuffle): Who first was the in person space?
Text VAE: Who was the first person in room ?
BackTranslation: Who was the first person in space ?
AE+Noised: Who was the man in space ?
CRQDA (answerable): Who was the first in space?
CRQDA (unanswerable): Who was the first Russians in space ?

Figure 4: Augmented data samples on SQuAD 2.0.