Closed-book Question Generation via Contrastive Learning

Xiangjue Dong\textsuperscript{1}, Jiaying Lu\textsuperscript{2}, Jianling Wang\textsuperscript{1,}\textsuperscript{*}, James Caverlee\textsuperscript{1}
\textsuperscript{1} Texas A\&M University, \textsuperscript{2} Emory University
\{xj.dong, jlwang, caverlee\}@tamu.edu, jiaying.lu@emory.edu

Abstract

Question Generation (QG) is a fundamental NLP task for many downstream applications. Recent studies on open-book QG, where supportive answer-context pairs are provided to models, have achieved promising progress. However, generating natural questions under a more practical closed-book setting that lacks these supporting documents still remains a challenge. In this work, we propose a new QG model for this closed-book setting that is designed to better understand the semantics of long-form abstractive answers and store more information in its parameters through contrastive learning and an answer reconstruction module. Through experiments, we validate the proposed QG model on both public datasets and a new WikiCQA dataset. Empirical results show that the proposed QG model outperforms baselines in both automatic evaluation and human evaluation. In addition, we show how to leverage the proposed model to improve existing question-answering systems. These results further indicate the effectiveness of our QG model for enhancing closed-book question-answering tasks.

1 Introduction

Question Generation (QG) has a wide range of applications, such as generating questions for exams (Jia et al., 2021; Lelkes et al., 2021; Dugan et al., 2022) or children’s story books (Zhao et al., 2018; Yao et al., 2022), recommending questions for users in a dialogue system (Shukla et al., 2019; Laban et al., 2020), improving visual (Li et al., 2018; Lu et al., 2022) or textual question-answering tasks (Duan et al., 2017; Lewis et al., 2019a; Zhang and Bansal, 2019; Sultan et al., 2020; Lyu et al., 2021), asking clarification questions (Rao and Daumé III, 2019; Yu et al., 2020; Ren et al., 2021), and generating queries for SQL (Wu et al., 2021) or multimodal documents (Kim et al., 2021).

Previous works on QG are mainly under the open-book setting, which aims to generate questions based on factoid or human-generated short answers under the assumption that there is access to external knowledge like retrieved documents or passages (Du et al., 2017; Zhao et al., 2018; Kim et al., 2019; Fei et al., 2021). After Roberts et al. (2020) demonstrated that feeding a large pre-trained model input questions alone without any external knowledge can lead to competitive results with retrieval-based methods on open-domain question-answering benchmarks, there is an increasing interest in the closed-book setting. This closed-book setting is appealing in practice and can be widely applied, e.g., in question suggestion (Laban et al., 2020; Yin et al., 2021), query recommendation (Kim et al., 2021), and other practical settings where extensive external knowledge is unavailable.

However, generating questions without access to such external knowledge is challenging for two key reasons. First, without access to retrieved documents (or passages), simple open-domain strategies like basing the answers on these documents (or passages) are not possible under the closed-book setting. Instead, models must rely on the answers alone. Second, the data used by most of the closed-book works (Lewis et al., 2021; Wang et al., 2021) are variants of existing open-domain datasets, e.g., SQuAD (Rajpurkar et al., 2018), TriviaQA (Joshi et al., 2017), WebQuestions (Berant et al., 2013) that ignore the answer-related passages. These answers in open-book works are usually short, e.g., entities, and easier to be remembered by the language model and stored in the parameters of the model than long-form answers. Thus, this leads to our motivating research question – How can we empower a QG model to better understand the semantics of long-form abstractive answers and store more information in its parameters?

To tackle the aforementioned challenges existing
in the closed-book setting, this paper proposes a new QG model with two unique characteristics: (i) a contrastive learning loss designed to better understand the semantics of the answers and the semantic relationship between answers and ground-truth questions at a contextual-level; and (ii) an answer reconstruction loss designed to measure the answerability of the generated question. Contrastive learning has shown promising results in many NLP tasks, e.g., (Giorgi et al., 2021; Gao et al., 2021; Yang et al., 2021) and aligns positive pairs better with available supervised signals (Gao et al., 2021); here we show how to learn question representations by distinguishing features of correct question-answer pairs from features of incorrectly linked question-answer pairs. Further, to ensure the generated questions are of good quality and can be answered by the answer that is used for question generation, we frame the model as a generation-reconstruction process (Cao et al., 2019; Zhu et al., 2020), by predicting the original answers given the generated questions by a pre-trained seq2seq model. In addition, we introduce a new closed-book dataset with long-form abstractive answers – WikiCQA – to complement existing datasets like GooAQ (Khashabi et al., 2021) and ELI5 (Fan et al., 2019) and show how to leverage our model to generate synthetic data to improve closed-book question-answering tasks.

Through experiments, we find that the proposed QG model shows improvement through both automatic and human evaluation metrics on WikiCQA and two public datasets. Compared to the baseline, the proposed QG framework shows an improvement of up to 2.0%, 2.7%, and 1.8% on the ROUGE-L score on WikiCQA, GooAQ-S, and ELI5, respectively, and 1.3% and 2.6% in terms of relevance and correctness. Furthermore, we leverage the QG framework to generate synthetic QA data from WikiHow summary data and pre-train a closed-book QA model on it in both an unsupervised and semi-supervised setting. The performance is evaluated on both seen (WikiCQA) and unseen (GooAQ-S, ELI5) datasets. We find consistent improvements across these datasets, indicating the QG model’s effectiveness in enhancing closed-book question-answering tasks.

In conclusion, our contributions can be summarized as follows:

- We propose a contrastive QG model, which to our knowledge is the first work to explore contrastive learning for QG under a closed-book setting.
- The proposed model outperforms baselines on three datasets. The human evaluation also indicates that the questions generated by our model are more informative compared to other baselines.
- We leverage the QG model as a data augmentation strategy to generate large-scale QA pairs. Consistent improvements shown on both seen datasets and unseen datasets indicate the QG model’s effectiveness in enhancing closed-book question-answering tasks.

2 Related Work

Many previous works on QG are under the open-book setting, which takes factoid short answers (Rajpurkar et al., 2016) or human-generated short answers (Kočiský et al., 2018) with the corresponding passages to generate questions (Zhang et al., 2021). Early approaches for question generation rely on rule-based methods (Labutov et al., 2015; Khullar et al., 2018). To bypass handcrafted rules and sophisticated pipelines in QG, Du et al. (2017) introduce a vanilla RNN-based sequence-to-sequence approach with an attention mechanism. The recently proposed pre-trained transformer-based frameworks (Lewis et al., 2020; Raffel et al., 2020) also improve the performance of QG. In addition, Sultan et al. (2020) shows that the lexical and factual diversity of QG provides better QA training. However, their success cannot directly adapt to the closed-book setting, where the model is supposed to generate questions solely relying on answers. In this work, we explore the widely applicable closed-book QG setting, which is still under-explored.

**Contrastive Learning** aims to pull semantically similar neighbors close and push non-neighbors apart. It has achieved great success under both supervised and unsupervised settings. In pioneer works, the contrastive loss function (Hadsell et al., 2006; Chopra et al., 2005) has been proposed as a training objective in deep metric learning considering both similar and dissimilar pairs. Recently, Chen et al. (2020) proposes the SimCLR framework to learn useful visual representations. Viewing contrastive learning as dictionary look-up, He et al. (2020) present Momentum Contrast (MoCo) to build dynamic dictionaries for contrastive learning. Some works apply contrastive learning into
the NLP domain to learn better sentence representations (Giorgi et al., 2021; Gao et al., 2021). In addition, contrastive learning has been applied in multilingual neural machine translation (Pan et al., 2021), abstractive summarization (Liu and Liu, 2021), and multi-document question generation (Cho et al., 2021). The recent most relevant work is (Yang et al., 2021), where they design two contrastive losses for paraphrase generation. In this work, we adopt contrastive learning for improving representation learning in question generation under a closed-book setting.

3 Proposed Approach

To answer our research question – How can we empower a QG model to better understand the semantics of long-form abstractive answers and store more information in its parameters? – we propose a closed-book QG model, which generates questions directly without access to external knowledge. Formally, given an answer sentence \( x \), a closed-book QG engine generates a natural question \( y \). Figure 1 illustrates an overview of the proposed QG framework, which consists of three parts: question generation, contrastive learning, and answer reconstruction. The framework is optimized with the joint losses from these three parts simultaneously.

![Figure 1: An overview of the proposed closed-book QG framework, which consists of three parts: contrastive learning, question generation, and answer construction. \( A_i \): represents answer i; \( Q_i \) represents question i.](image)

3.1 Question Generation

We first focus on question generation through a sequence-to-sequence architecture which consists of an encoder and a decoder (Sutskever et al., 2014; Vaswani et al., 2017). The encoder takes an input sequence of source words \( x = (x_1, x_2, \ldots, x_n) \) and maps it to a sequence of continuous representations \( z = (z_1, z_2, \ldots, z_n) \). Then, the decoder takes \( z \) and generates a sequence of target words \( y = (y_1, y_2, \ldots, y_m) \) at a time. The closed-book QG task is defined as finding \( \hat{y} \):

\[
\hat{y} = \arg \max_y P(y|x),
\]

where \( P(y|x) \) is the conditional likelihood of the predicted question sequence \( y \) given answer \( x \).

Given the answer-question pairs, the training objective of the generation part in the proposed framework is to minimize the Negative Log-Likelihood (NLL) of the training data,

\[
L_{qg} = - \sum_{i=1}^{N} \log p(q_i|A),
\]

where \( q_i \) is the \( i \)-th token in the generated question and \( A \) is the answer.

A naive question generation model will generate questions based on answers but lacks a rich model of the semantics of answers nor can it guarantee the generated questions have a semantic relationship with the answers. Intuitively, an encoded answer should be similar to its question and dissimilar to others. In addition, the generated question should be able to be answered by the answers. Hence, this motivates the following contrastive learning and answer reconstruction modules.

3.2 Contrastive Learning

Contrastive learning aims to pull positive pairs and push apart negative pairs to learn effective representations. Further, the supervised signals can produce better sentence embeddings by improving alignment between positive pairs (Chen et al., 2020). An effective QG model should be able to understand the semantics of the answers and the semantic relationship with the ground-truth questions. Especially, the encoded answer should have semantic similarity with its ground-truth questions but lacks a rich model of the semantics of answers nor can it guarantee the generated questions have a semantic relationship with the answers. In this work, we adopt contrastive learning for improving representation learning in question generation under a closed-book setting.
within a mini-batch are treated as negative examples. The training objective for \((x_i, y_i)\) is:

\[
L_{cl} = - \log \frac{\exp(\text{sim}(z_{x_i}, z_{y_i})/\tau_{cl})}{\sum_{j=1}^{2n} \exp(\text{sim}(z_{x_i}, z_{y_j})/\tau_{cl})},
\]

(4)

where \(z_{x_i}\) and \(z_{y_i}\) is the representation of input \(x_i\) and \(y_i\), \(\text{sim}(z_i, z_j) = z_i^\top z_j / ||z_i|| ||z_j||\) denotes cosine similarity, and \(\tau_{cl}\) is a temperature parameter.

In this work, aiming to learn better answer representations and force the encoder to drive representations of correct question-answer pairs closer than representations of incorrect question-answer pairs, we take the ground-truth question as the positive instance of an answer and fine-tune the model parameters based on the contrastive loss function (Eq. 4), where \(z\) is the embedding of the special token \([CLS]\) from the transformer encoder, representing the meaning of the entire sentence.

### 3.3 Answer Reconstruction

The questions that are generated by the model should be of good quality and should also be able to be answered by the answer that is used for question generation. To measure the answerability of the generated question, we design an answer reconstruction module, which uses a pre-trained seq2seq model to predict the original answer given the generated question. The loss is calculated by a negative log-likelihood loss function:

\[
L_{ar} = - \sum_{i=1}^{N} \log p(a_i | Q)
\]

(5)

where \(a_i\) is the \(i\)-th token in the answer and \(Q\) is the generated question.

A major challenge is that the generated questions from Section 3.1 are not differentiable. Gradients cannot be back-propagated directly. To solve this challenge, we employ the Straight-Through (ST) Gumbel-Softmax for gradient computation (Jang et al., 2017). The ST Gumbel-Softmax is a discrete version of the Gumbel-Softmax and takes different forward and backward paths. In the forward pass, the embedding is discretized by using the argmax, whereas in the backward pass the gradients are computed by the gumbel-softmax (Qader et al., 2019; Lu et al., 2021).

\[
y_i = \frac{\exp((\log(p_i) + g_i)/\tau_{gs})}{\sum_{j=1}^{V} \exp((\log(p_j) + g_j)/\tau_{gs})},
\]

(6)

where \(\tau_{gs}\) is a temperature parameter and \(g_i\) is the Gumbel noise drawn from a uniform distribution \((0, 1)\). In this work, the one-hot embedding from ST Gumbel-Softmax is multiplied with the vocabulary embedding and then fed into the encoder of the pre-trained seq2seq model as the representation of the generated question.

### 3.4 Overall Loss Function

As a result, all three losses are summed together to provide the overall loss function \(\mathcal{L}\) as follows:

\[
\mathcal{L} = \lambda_1 \mathcal{L}_{qg} + \lambda_2 \mathcal{L}_{cl} + \lambda_3 \mathcal{L}_{ar},
\]

(7)

The weights \(\lambda_1, \lambda_2, \text{ and } \lambda_3\) are tuneable hyperparameters to balance losses and the final objective is to minimize the overall loss.

The overall structure of the proposed QG framework is presented in Algorithm 1.

### Algorithm 1: QG framework.

**Input:** Pre-trained language model \(p(q|a)\),
answer reconstruction model \(p(a|q)\)
and answer-question pairs

**Output:** Question generator \(p(q|a)\)

1. for \(i \leftarrow 1 \text{ to } E\) epoch do
   2. \(\hat{q} = p(q|a)\)
   3. Compute \(\mathcal{L}_{qg}\) via Eq. 3
   4. \(a_i = \text{Encoder}([CLS] \oplus a[:0];\)
   5. \(a_i^+ = \text{Encoder}([CLS] \oplus q[:0];\)
   6. Get \(\mathcal{L}_{cl}\) to \((a_i, a_i^+)\) via Eq. 4
   7. \(\hat{a} = p(a|\hat{q})\)
   8. Compute \(\mathcal{L}_{ar}\) via Eq. 5
   9. Calculate total loss \(\mathcal{L}\) via Eq. 7
10. Update generator \(p(q|a)\) with \(\mathcal{L}\)
11. end
12. return question generator \(p(q|a)\)

### 4 Experimental Setup

To evaluate the effectiveness of the proposed QG framework, we aim to answer the following research questions (RQ) via experiments:

**RQ1:** Can this proposed QG framework improve the performance of the closed-book QA tasks?

**RQ2:** Are the generated questions of good quality? That is, are they fluent and relevant to the answer?

**RQ3:** Can this QG framework be leveraged as a good resource to generate synthetic data for the QA task?

**RQ4:** How much does each component in the framework contribute?
4.1 Dataset

We conduct the experiments on two public datasets – GooAQ-S (Khashabi et al., 2021) and ELI5 (Fan et al., 2019) – and a new dataset we curate called WikiCQA.

GooAQ-S is a sub-sampled dataset from GooAQ which contains three different sub-tasks: short, snippet (i.e., multi-sentence description), and collection response questions (Khashabi et al., 2021). Under the long-form closed-book setting, we adopt the snippet part (i.e., questions with snippet answers) for our experiments as in the original paper. Furthermore, from the paper, the model performance on the snippet task does not vary much when supervised with 200K and 2M training instances. Thus, to improve the experimental efficiency, we take 200k instances from the snippet set and split them based on their original scripts.

ELI5 (Fan et al., 2019) is a widely-used large-scale corpus for long-form question-answering with supporting web documents. In this work, we use the question-answer pairs from the dataset and ignore the supporting context to fit the closed-book setting like (Khashabi et al., 2021). We follow the data split from huggingface.

WikiCQA is a new closed-book long-form QA dataset (Section 4.2) introduced here. It contains 20,202 question-answer pairs and is collected from a wiki-style website to complement existing datasets. We shuffle the data and split it to train, dev, and test sets by the ratio of 80%/10%/10%. Table 1 shows detailed data statistics and splits of these three datasets.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Train</th>
<th>Val</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>WikiCQA</td>
<td>16,162</td>
<td>2,020</td>
<td>2,020</td>
</tr>
<tr>
<td>GooAQ-S</td>
<td>200,000</td>
<td>500</td>
<td>498</td>
</tr>
<tr>
<td>ELI5</td>
<td>272,634</td>
<td>9,812</td>
<td>24,512</td>
</tr>
</tbody>
</table>

Table 1: Dataset statistics.

4.2 WikiCQA

WikiCQA contains real user QA pairs collected from WikiHow. WikiHow is a wiki-style website featuring over 200K how-to articles. Different from the existing dataset, ELI5, containing questions/answers from the Reddit forum and leveraging evidence queried from the web to help answer the question (Fan et al., 2019) and GooAQ, containing questions from search auto-complete and answers from answer boxes (Khashabi et al., 2021), WikiCQA involves long-form question-answering grounded on WikiHow articles.

**Dataset Construction** We construct the new dataset by collecting the question-answer pairs from the Q&A section of articles on WikiHow. The questions and answers are related to the specific articles, asked by WikiHow users, and answered by a knowledgeable reader or editor. The questions cannot be answered directly by the content of the article. After removing duplicates and question-answer pairs with meaningless answers, 23,037 QA pairs remain. To keep the same format as other existing QA datasets (e.g., ELI5, GooAQ), we discard questions not starting with a question-type word (e.g., what, how). After the dataset processing steps, we arrive at 20,202 question-answer pairs. More details can be found in Appendix A.

In Table 2, we show an example article from Wikihow, which includes the title and summary of an article and question-answer pairs from the corresponding Q&A section. From the example, we can see that answers are abstractive and long-form, written by real users, and not contained within the context passage. These question-answer pairs from the Q&A section are collected in the newly constructed dataset. Thus, this dataset is different from reading comprehension datasets, where the answers are a short text span in context. More comparisons with ELI5 and GooAQ are shown in Appendix B.

| Title: How to Prepare a Healthy Meal for Your Pet Dog. |
| Summary: To prepare a healthy meal for your dog, choose lean meat with the bones and fat removed, like chicken or beef . . . |
| Q: What can I feed my dog if I have run out of dog food? A: In the short term, any bland human food such as chicken or white fish with rice or pasta is just fine . . . |
| Q: How much homemade dog food do you feed your dog? A: Great question because it highlights one of the problems of feeding home prepared foods . . . |
| Q: What should I not feed my dog? A: There are many human foods that are toxic to dogs. Top of the list of foods NOT to give are . . . |

Table 2: Question-answer pairs from WikiHow Q&A section. Q: question; A: answer.

---

1https://github.com/allenai/gooaq/blob/main/experiments/create_splits.py
2https://huggingface.co/datasets/eli5
3https://www.wikihow.com, under an Attribution-Noncommercial-Share Alike 3.0 Creative Commons License.
4.3 Models & Hyper-parameters

We use BART-base (Lewis et al., 2020), a widely used sequence-to-sequence framework with 12 layers and a hidden size of 1024, as the backbone model. Following previous works (Fan et al., 2019; Khashabi et al., 2021), we finetune it using answers as inputs and questions as outputs as the baselines. To get the answer reconstruction loss (Section 3.3), we use a BART-base model which is fine-tuned on the target QA datasets.

Based on the dataset analysis in Appendix B, we set the maximum sequence length to 128 for the question and 256 for the answer sentence to improve calculation efficiency. We train the models for five epochs with a learning rate of $5 \times 10^{-5}$ and evaluate the checkpoint for each epoch. We select the checkpoint with the highest ROUGE-L score on the validation set and report its corresponding score on the test set. We run each model three times and record the average scores. After performing manual hyper-parameter searching, we set loss parameters $\lambda_1$, $\lambda_2$, and $\lambda_3$ in Equation 7 to 1.0, 0.1, 0.1, respectively, which give the best ROUGE-L score on validation set. The temperature of contrastive loss $\tau_{cl}$ is set to 0.3. The experiments are run on 1 NVIDIA Tesla V100 GPU. The training time takes about 48 hours.

4.4 Evaluation Metrics

To evaluate the performance of the models, we use ROUGE (Lin, 2004) scores, which evaluate the $n$-grams recall of generated sentences with reference sentences, as automatic evaluation metrics (Fan et al., 2019; Khashabi et al., 2021). We report the F1 for ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-Lsum. ROUGE-1, 2, L measures the unigram, bigram, and longest common subsequence between the pair of sentences, respectively. The difference between ROUGE-L and ROUGE-Lsum is that ROUGE-Lsum splits text using “\n”. The higher ROUGE score indicates higher similarity between generated questions and references.

We further perform human evaluations for the quality of generated questions in terms of Fluency: whether the questions are grammatically correct and fluent; Relevance: whether the questions are related to the answers; Correctness: whether the questions can be answered by the answers.

5 Experiment Results

In this section, we answer the four experimental research questions in turn.

5.1 Generation Performance (RQ1)

First, does the proposed QG framework have good performance? Table 3 shows the performance of the proposed closed-book QG model (denoted as $QG_{ours}$) on the new dataset and two public datasets compared with the baseline model trained only with question generation loss (denoted as $QG_b$). We observe that the proposed framework outperforms the baseline on three datasets for all ROUGE scores.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$QG_b$</th>
<th>$QG_{ours}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>WikiCQA</td>
<td>48.39</td>
<td>49.22</td>
</tr>
<tr>
<td>GooAQ-S</td>
<td>44.26</td>
<td>45.26</td>
</tr>
<tr>
<td>ELI5</td>
<td>28.62</td>
<td>29.15</td>
</tr>
</tbody>
</table>

Table 3: QG results on three datasets. $QG_b$: baseline model; $QG_{ours}$: our proposed QG framework.

5.2 Quality of Generated Questions (RQ2)

Next, are the generated questions of good quality? To answer this question, we perform human evaluation to measure the quality of generated questions in terms of fluency, relevance, and correctness. We randomly sample 100 answer-question pairs from the WikiCQA dataset, which contains answers, ground-truth questions, and generated questions from the baseline model and our best model. Then, we ask three annotators to rate the generated question pairs, comparing them with the ground-truth questions. The results demonstrate the effectiveness of our proposed QG framework.
Table 4 shows the average scores and the percentages of preferred questions on the three criteria. Both models are equally good at generating fluent questions. In terms of relevance and correctness, our approach shows 1.3% and 2.6% higher scores than the baseline method. These are consistent with what we expect: the contrastive learning part pulls the answer representation closer to the ground-truth question representation and generates more relevant questions; the reconstruction part can ensure that generated questions are of good quality and answerable. In addition, among 100 answer-question pairs, 24.7% and 23.3% of generated questions from our best model are rated higher while 56.3% and 57.4% of them have the same rating as those from baselines, which indicates that there is still substantial room to improve. To get a sense of the stability of the human evaluation results, we measure the inter-agreement among annotators using the Correlation Coefficient, which is 0.998, 0.944, and 0.976 in terms of these three aspects, showing excellent reliability.

<table>
<thead>
<tr>
<th>Model</th>
<th>Fluency score</th>
<th>Relevance score</th>
<th>Correctness score</th>
</tr>
</thead>
<tbody>
<tr>
<td>QG</td>
<td>4.83</td>
<td>3.84</td>
<td>3.49</td>
</tr>
<tr>
<td>QGsource</td>
<td>4.83</td>
<td>3.80</td>
<td>3.58</td>
</tr>
</tbody>
</table>

Table 4: Results of human evaluation for baseline and our best model. score: is the average score from raters; %: represents the proportion of generated questions from one model that were rated higher than those from its counterpart.

5.3 Synthetic Data Generation (RQ3)

Further, can this QG framework be leveraged to generate effective synthetic data that can improve the closed-book QA task? Data augmentation is one of the main directions that question generation has been used for previously, with several studies finding improvements on the QA task (Lewis et al., 2019b; Alberti et al., 2019). Here, we show how to leverage the proposed QG framework to improve closed-book QA tasks on seen data (WikiCQA) and unseen data (GooAQ and ELI5). Since freely available summary data is a good resource to generate synthetic data (Lyu et al., 2021), we use WikiSum (Cohen et al., 2021), which contains 39,775 coherent-paragraph summaries written by the article’s authors on the WikiHow website. We take each sentence from the article summary as an answer and pass it into the best QG model, described in Section 3.1, to generate a question. Then, we train an unsupervised QA model based on the synthetic QA pairs and optimize it by the following negative log-likelihood loss function:

$$
L_{qa} = - \sum_{i=1}^{N} \log p(a_i|Q),
$$

where \(a_i\) is the i-th token in the generated answer and Q is the question.

In this work, we pre-train a BART-base model on the 200K synthetic QA pairs that are generated through the best QG model (denoted as QA*) and evaluate it on the test set of the seen dataset (WikiCQA) and unseen datasets (GooAQ-S and ELI5). This approach is unsupervised since the model is trained on no labeled question-answer pairs. The results are summarized in Table 5a, showing 22.4%, 12.5%, and 11.4% improvement than the BART-base model without any synthetic pre-training (denoted as QA_i), on WikiCQA, GooAQ-S, and ELI5, respectively. This shows that pre-training on the generated question-answer pairs derived from our QG model leads to significant improvements, echoing findings in previous works that find synthetic data can be helpful (Khashabi et al., 2021; Lewis et al., 2021; Ding et al., 2021). After further fine-tuning on the target training set (denoted as QA_{s+f}), from Table 5b we can see that QA_{s+f} also achieves better results than fine-tuned baselines model QA_{b+f} by 3.6%, 0.4%, and 4.3% on the same three datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>R-1</th>
<th>R-2</th>
<th>R-L</th>
<th>R-Lsum</th>
</tr>
</thead>
<tbody>
<tr>
<td>WikiCQA</td>
<td>QA_i</td>
<td>18.41</td>
<td>5.53</td>
<td>14.72</td>
<td>15.98</td>
</tr>
<tr>
<td></td>
<td>QA_S</td>
<td>24.10</td>
<td>7.07</td>
<td>18.02</td>
<td>20.31</td>
</tr>
<tr>
<td></td>
<td>QA_{s+f}</td>
<td>21.93</td>
<td>6.12</td>
<td>16.62</td>
<td>18.35</td>
</tr>
<tr>
<td>GooAQ-S</td>
<td>QA_i</td>
<td>18.58</td>
<td>5.80</td>
<td>14.77</td>
<td>15.66</td>
</tr>
<tr>
<td></td>
<td>QA_S</td>
<td>21.38</td>
<td>7.07</td>
<td>16.62</td>
<td>18.35</td>
</tr>
<tr>
<td></td>
<td>QA_{s+f}</td>
<td>13.97</td>
<td>2.29</td>
<td>10.00</td>
<td>11.77</td>
</tr>
</tbody>
</table>

(a) Unsupervised QA results.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>R-1</th>
<th>R-2</th>
<th>R-L</th>
<th>R-Lsum</th>
</tr>
</thead>
<tbody>
<tr>
<td>WikiCQA</td>
<td>QA_{b+f}</td>
<td>27.16</td>
<td>7.32</td>
<td>19.85</td>
<td>23.49</td>
</tr>
<tr>
<td></td>
<td>QA_{s+f}</td>
<td>28.44</td>
<td>8.14</td>
<td>20.56</td>
<td>24.77</td>
</tr>
<tr>
<td>GooAQ-S</td>
<td>QA_{b+f}</td>
<td>27.72</td>
<td>7.68</td>
<td>20.08</td>
<td>23.64</td>
</tr>
<tr>
<td></td>
<td>QA_{s+f}</td>
<td>28.44</td>
<td>7.73</td>
<td>20.17</td>
<td>24.12</td>
</tr>
<tr>
<td>ELI5</td>
<td>QA_{b+f}</td>
<td>22.06</td>
<td>3.93</td>
<td>13.93</td>
<td>19.72</td>
</tr>
<tr>
<td></td>
<td>QA_{s+f}</td>
<td>23.37</td>
<td>4.27</td>
<td>14.53</td>
<td>20.88</td>
</tr>
</tbody>
</table>

(b) Semi-supervised QA results.

Table 5: Evaluation on QA tasks. QA_{b+f}: QA model w/o pre-training on synthetic data; QA_{s+f}: QA_{s} fine-tuned on target dataset.

5.4 Ablation Study (RQ4)

Finally, we investigate the contribution of each component in the proposed closed-book QG framework. Table 6 shows the results on the WikiCQA
dataset. As described in Section 3.2, the contrastive learning module takes ground-truth questions as positive pairs to the input answers (denoted as $CL_t$). We also explore its variant – in which we feed the same input (answers) to the encoder with different dropouts to obtain the positive pairs (Gao et al., 2021) (denoted as $CL_s$). We find that the choice of $CL_t$ is slightly better than $CL_s$ in most evaluation metrics. In addition, by adding the answer reconstruction ($AR$) module, the model performance can be further improved. Thus, we can observe the effectiveness of the proposed contrastive learning module and the overall design of the framework.

<table>
<thead>
<tr>
<th>Models</th>
<th>R-1</th>
<th>R-2</th>
<th>R-L</th>
<th>R-LSum</th>
</tr>
</thead>
<tbody>
<tr>
<td>QG_b</td>
<td>48.39</td>
<td>26.84</td>
<td>46.08</td>
<td>46.16</td>
</tr>
<tr>
<td>QG_b+CL_t</td>
<td>48.67</td>
<td>27.34</td>
<td>46.43</td>
<td>46.46</td>
</tr>
<tr>
<td>QG_b+CL_s</td>
<td>48.72</td>
<td>27.26</td>
<td>46.33</td>
<td>46.41</td>
</tr>
<tr>
<td>QG_b+CL_t+AR</td>
<td>49.26</td>
<td>27.77</td>
<td>46.82</td>
<td>46.88</td>
</tr>
<tr>
<td>QG_b+CL_s+AR</td>
<td>49.22</td>
<td>27.79</td>
<td>46.98</td>
<td>47.08</td>
</tr>
</tbody>
</table>

Table 6: Ablation study of our proposed QG framework on WikiCQA. $CL_t$: contrastive learning module; $AR$: answer reconstruction module.

6 Case Study

Finally, we showcase some examples of the generated questions on the WikiCQA dataset. For the first example in Table 7a, we can see our model generates “how many” for the statistics-type answer, which is closer to the reference “what percentage” than the baseline model, generating a “yes/no” question. For the second one, although the reference contains the detailed information “textured surface” which is not in the answer, our model captures the information “without nails”, which is more informative than the baseline model. Thus, the proposed model can produce better questions that are more relevant to the answers and contain more detailed information than questions generated from baseline models. There are also some cases where the proposed model fails to generate good questions shown in Table 7b. Compared with the reference question “what food can be included on a clear liquid diet”, our model generates “what can I eat to lose weight”, which fails in automatic evaluation metrics and is hard for annotators to justify in terms of relevance. For the second one, although the “broken door” generated from our model is relevant to the answer, it’s still far from the reference “door sticking to the weather strip”. The question generated from the baseline model related to “bedroom” is even less relevant. These examples encourage us to explore how to capture more semantics in the answers and generate questions that have more detailed information in the future.

7 Conclusion

In this work, aiming to empower a QG model to better understand the semantics of long-form abstractive answers and store more information in its parameters, we propose a closed-book QG model empowered by a contrastive learning module and an answer reconstruction module. We present a new closed-book long-form QA dataset – WikiCQA involving more than 20K real user QA pairs and show that WikiCQA is a valuable training resource, complementing public datasets. Through the experiments, the proposed QG model shows better performance than baselines through automatic and human evaluations. Moreover, we show how to leverage the proposed model as a data augmentation strategy to improve existing closed-book QA systems. The closed-book QA model, pre-trained on our generated synthetic QA pairs, achieves better performance on the seen dataset (WikiCQA). In addition, it shows strong generalization on unseen datasets (GooAQ and ELI5), which further demonstrates the effectiveness of our QG framework for enhancing closed-book QA performance.
Limitations

While our model shows promising results on English datasets, its efficiency and performance in other languages require further investigation. Moreover, our evaluation is limited to datasets with one-to-one QA pairs, without considering situations where multiple answers correspond to a single question. Before feeding the input into the model, we ignore the tokens that exceed the maximum length we set, which may sometimes bring in an information loss for the input corpus. Additionally, our model is built on transformer architecture and therefore requires huge computation resources, especially for large training data sizes. Our focus in this paper is solely on the closed-book setting and our experiments center around four research questions to evaluate the effectiveness of the proposed framework, which we believe is an essential and emerging area deserving of specialized research investigation. In future work, we aim to explore the feasibility of adapting our framework for other QG scenarios.

Acknowledgements

We thank Zhuoer Wang, Yun He, Ziwei Zhu, anonymous reviewers, and the action editor for providing valuable feedback.

References


Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In Conference ICLR.


Patrick Lewis, Ludovic Denoyer, and Sebastian Riedel. 2019b. Unsupervised question answering by cloze


Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In NIPS.


Bingsheng Yao, Dakuo Wang, Tongshuang Sherry Wu, Zheng Zhang, Toby Jia-Jun Li, Mo Yu, and Ying Xu. 2022. It is a’s turn to ask humans a question: Question-answer pair generation for children’s story books. In ACL.


Appendix A Data Filtering

First, we choose the question-answer pairs where the questions start with a question-word in the list ['how', 'what', 'can', 'is', 'do', 'why', 'are', 'does', 'where', 'when', 'should', 'will', 'did', 'which', 'who', 'would', 'if', 'about', 'for', 'as', 'could', 'in', 'after', 'at', 'while', 'to', 'am', 'has', 'any'] and end with a question mark. Then, we filter out the pairs where the length of answers is less than 8 tokens or the answers are meaningless. For the “meaningless” answers we mean, answers are “please refer to certain website http://xxx” or “please refer to the article xxx”.

Appendix B Dataset Analysis

To better understand the content of WikiCQA in comparison with existing QA datasets, Figure 2 shows the distributions of question length, answer length, and common question word. The questions and answers are tokenized by BART (Lewis et al., 2020). From Figure 2a and 2b we could see more than 80% of the questions in WikiCQA and GooAQ-S have less than 15 tokens and more than 90% of answers of them lie in the range of 16-127 tokens while both questions and answers
in ELI5 have a broader range of lengths. By measuring the proportion of the leading unigram that starts a question (Figure 2c), the dominant pattern is “how” questions in WikiCQA, while ELI5 has many “why” questions. We think WikiCQA has the potential to benefit the community in three ways. Firstly, its source is unique compared to the current two closed-book data sources, as it is collected from the Q&A section of articles on WikiHow. The questions and answers are related to the specific articles, asked by WikiHow users who read the articles and answered by a knowledgeable reader or editor. Secondly, compared to large-scale QA datasets, WikiCQA has longer answers and more open-ended questions with a wider range of question types. Additionally, it has a different data distribution compared to the other two datasets, which is crucial for testing the generalization ability of question-answering models. Finally, the question-answer pairs within relevant articles in WikiCQA can also be used as a valuable resource for conversational question-answering tasks. Our code and data can be found at https://github.com/dongxiangjue/Closed-book-Question-Generation-via-Contrastive-Learning.