Improving Reading Comprehension Question Generation with Data Augmentation and Overgenerate-and-rank

Nischal Ashok Kumar1, Nigel Fernandez1, Zichao Wang2, Andrew Lan1
University of Massachusetts Amherst1, Rice University2
{nashokkumar, nigel, andrewlan}@cs.umass.edu, zw16@rice.edu

Abstract

Reading comprehension is a crucial skill in many aspects of education, including language learning, cognitive development, and fostering early literacy skills in children. Automated answer-aware reading comprehension question generation has significant potential to scale up learner support in educational activities. One key technical challenge in this setting is that there can be multiple questions, sometimes very different from each other, with the same answer; a trained question generation method may not necessarily know which question human educators would prefer. To address this challenge, we propose 1) a data augmentation method that enriches the training dataset with diverse questions given the same context and answer and 2) an overgenerate-and-rank method to select the best question from a pool of candidates. We evaluate our method on the FairytaleQA dataset, showing a 5% absolute improvement in ROUGE-L over the best existing method. We also demonstrate the effectiveness of our method in generating harder, "implicit" questions, where the answers are not contained in the context as text spans.

1 Introduction

Reading comprehension is crucial in assessing students’ language learning ability and complex reasoning skills. Comprehending and interpreting stories such as fairy tales, with specific emphasis on narratives, foster early intellectual and literacy development in children (Sim and Berthelsen, 2014; Lynch et al., 2008). Asking suitable educational-focused questions can help students understand the context of the fairy tales better and inspire their interests (Ganotice Jr et al., 2017; Zevenbergen and Whitehurst, 2003; Xu et al., 2021). However, constructing suitable questions at scale is hard since it is both time intensive and cognitively challenging (Golinkoff et al., 2019). Researchers have developed models that can automatically generate questions or question-answer pairs to meet the demand for a large pool of relevant questions (Kurdi et al., 2020; Yao et al., 2022). These advances can potentially facilitate the development of artificial intelligence (AI)-supported learning platforms to help students develop reading comprehension skills (Zhang et al., 2022).

Prior work on question generation in educational applications can be broadly classified into two categories: answer-aware, which is the focus of our current work, and answer-unaware (see Dugan et al. (2022) for a feasibility study), depending on whether the desired answer is given or not. For answer-aware question generation, the goal is to build an AI-based system to generate a question given both the context and the answer (Wang et al., 2018). The context can be any text segment, from a few sentences to a possibly long document, that provides background information on which the question is grounded in. The answer is a short span of text that is either part of the context (explicit) or not part of the context but can be inferred from the context (implicit). More specifically, in answer-aware question generation, the question generation system is trained using the context-answer pairs as input and the question as the output (Yao et al., 2022). See Section 2 for a detailed discussion on related work.

A key challenge in answer-aware question generation is that there are often multiple relevant questions for a given context-answer pair. Existing question generation systems are limited in identifying which questions human educators would prefer from multiple relevant ones. Table 1 shows an example context-answer pair from the FairytaleQA dataset (Xu et al., 2022b) with four relevant questions that can be answered by “a lovely dinner”, the given answer. The first and second questions focus on describing the setting of the context framed using the object (table) and the subject (Tom and Hunca), respectively. The third question adds a
causal element inquiring about the cause of Tom and Hunca’s emotion. The fourth question is predictive in nature, asking about an event which can be inferred from the context.

Selecting the top question from multiple relevant and diverse question candidates is challenging. For a question generation system to perform this challenging task, it needs to be able to both generate diverse and valid question candidates and also accurately rank and select the top question. To generate diverse question candidates, a question generation system needs to be trained on multiple different relevant questions for a given context-answer pair. To accurately select the top question, a question generation system needs to learn to rank the question candidates by matching the preferences of human educators. We incorporate both of these ideas in our proposed methods in this work.

1.1 Contributions

In this paper, we detail two novel methods to improve the robustness of automated answer-aware reading comprehension question generation. We validate their effectiveness through both quantitative and qualitative experiments on the FairytaleQA dataset (Xu et al., 2022b); we make our implementation publicly available.¹ Built on top of a Flan-T5 (Chung et al., 2022) fine-tuning backbone, our contributions are summarized as follows:

- We propose a **data augmentation** method to augment the training set with synthetically generated diverse and relevant questions. Specifically, we prompt a larger language model, OpenAI Codex (Chen et al., 2021), to first generate a diverse question pool and then filter out questions that are inconsistent with the given context-answer pair using a question-answering model.

- We propose an **overgenerate-and-rank** method to rank multiple generated question candidates for the given context-answer pair. Specifically, we fine-tune a separate BERT-based model by optimizing a distribution matching objective to learn which questions are more preferable to human educators and use the model to rank them.

<table>
<thead>
<tr>
<th>Context</th>
<th>Tom Thumb and Hunca Munca went up-stairs and peeped into the dining room. Then they squeaked with joy. Such a lovely dinner was laid out upon the table . . .</th>
</tr>
</thead>
<tbody>
<tr>
<td>Answer</td>
<td>a lovely dinner</td>
</tr>
</tbody>
</table>
| Questions | 1. What was laid upon the table?  
2. What did Tom and Hunca see in the dining room?  
3. What made Tom and Hunca squeak with joy?  
4. What will Tom and Hunca enjoy eating in the dining room? |

Table 1: Example context-answer pair from the FairytaleQA dataset with multiple valid questions.

- We conduct extensive experiments to validate the effectiveness of our methods. Our best method achieves a 5% absolute increase in the ROUGE-L score over the best existing baseline (Xu et al., 2022b). We also observe that 1) the data augmentation method can be used to balance questions of different types in the training data and 2) the overgenerate-and-rank method is particularly effective at generating harder questions, i.e., those with answers not explicitly present in the text as text spans.

2 Related Work

2.1 QA Datasets on Narratives

There have been several works proposing QA and QG datasets of educational importance. NarrativeQA (Kočiský et al., 2018) requires students to answer questions written by crowd workers based on books or movie scripts. TellMeWhy (Lal et al., 2021) is another dataset that contains only “why” based questions that need additional information not directly present in the text to be answered. A recent and popular dataset to facilitate assessment and training of students’ narrative comprehension skills is the FairytaleQA (Xu et al., 2022b) dataset. FairytaleQA contains question-answer pairs written by education experts on fairy tale stories obtained from Project Gutenberg.² FairytaleQA is composed of questions focusing on several narrative elements. We validate the effectiveness of our question generation methods with extensive experiments on FairytaleQA.

---

¹The code for the paper can be found at: [https://github.com/umass-ml4ed/question-gen-aug-ranking](https://github.com/umass-ml4ed/question-gen-aug-ranking)

²[https://www.gutenberg.org](https://www.gutenberg.org)
2.2 Question Generation

There are several works on question generation for reading comprehension. Stasaski et al. (2021) and Zou et al. (2022) propose question generation methods based on causal relations and unsupervised learning, respectively. However, their methods are focused on very specific questions and are thus not generalizable. In contrast, our work focuses on a broad variety of questions covering different narrative elements in reading comprehension. Rathod et al. (2022) propose to generate multiple semantically similar but lexically diverse questions for a given answer. However, their work is limited to generating only two questions per answer. In contrast, our approach is capable of generating multiple diverse and relevant questions, along with a ranking method to select the best question aligned with human educator preferences. Recent work on the FairytaleQA dataset develops event-based question generation methods (Zhao et al., 2022; Xu et al., 2022a). However, their results are reported on only a small subset of attributes: action, causal relationship, and outcome resolution. In contrast, we report our results over all attributes on the complete FairytaleQA dataset and compare with the current state-of-the-art baseline. Yuan et al. (2022) propose a prompt-based question generation method that leverages large language models (LM) like GPT-3. However, these black-box LMs have limited API only access. In contrast, our method uses open-source language models to achieve competitive results. The FairytaleQA dataset paper (Xu et al., 2022b) proposes the current state-of-the-art question generation method by fine-tuning the BART (Lewis et al., 2020) LM to generate the ground truth question given the input context-answer pair. Improving upon LM fine-tuning, we propose two question generation methods for increased robustness, data augmentation and overgenerate-and-rank, which are able to both generate diverse and valid question candidates and also accurately rank and select the top question aligned with human educator preference.

3 Methodology

In this section, we first introduce the problem setup for question generation on FairytaleQA (Xu et al., 2022b). We then detail our question generation approach, building upon the baseline of fine-tuning a language model, by adding our data augmentation method to augment the training set with diverse questions, followed by our over-generate-and-rank method to select the top question from the diverse question candidates generated.

3.1 Problem Formulation and Dataset Details

FairytaleQA (Xu et al., 2022b) is a popular dataset for both question answering and question generation in the education community supporting narrative comprehension, targeting students from kindergarten to eighth grade. Written by education experts, FairytaleQA contains 10,580 question-answer pairs \((q_i, a_i)\), indexed by \(i\), from 278 classical fairytale stories. Each question-answer pair is sourced from a section of a story referred to as the context \(c_i\). The goal for a trained question generation model is to generate the ground truth question \(q_i\) conditioned on the input context-answer pair \((c_i, a_i)\).

Question-answer pairs in FairytaleQA can be categorized in two major ways: 1) by attributes and 2) by the source of answers. In attribute categorization, question-answer pairs capture seven different narrative elements or relations, referred to as attributes, which are character, setting, action, feeling, causal relationship, outcome resolution, and prediction. Orthogonal to the previous categories, questions can also be categorized by whether the answer span is explicitly contained within the context or is implicit and need to be inferred from the context. Explicit questions capture specific story facts while implicit questions require summarization and inference skills. FairytaleQA is imbalanced with respect to question attributes, with action and causal relationship questions accounting for 60% of the dataset. Our data augmentation method helps balance questions of different attributes.

3.2 Language Model Fine-tuning

We first describe our LM fine-tuning approach for question generation. We use a pre-trained Flan-T5 (Chung et al., 2022) model as our base LM for question generation. We also tried using vanilla T5 (Raffel et al., 2020) and GPT-2 (Radford et al., 2019) as our base LM which gave a comparable but lower performance, possibly because Flan-T5 is instruction fine-tuned on a large number of tasks relevant to both QA and QG. Therefore, for simplicity of exposition, we detail our question generation methods using Flan-T5 as the base LM. We construct the input using a combination of the context \(c_i\) and answer \(a_i\) with the following...
template: Generate question given context and answer: Context: \( c \)  
Answer: \( a \).
Let \( \theta \) represent the LM parameters to be learned. We fine-tune our LM over all context-answer pairs \((c_i, a_i)\) to generate the corresponding ground truth question \(q_i\) using a language modeling objective. The language modeling objective is the negative log-likelihood of generating the ground truth question calculated at the token level. The objective \( L_i(\theta) \) for the \( i \)th training sample is given by:

\[
L_i(\theta) = -\sum_t \log P(q_{i,t}|c_i, a_i, q_{i,<t})
\]

where \( q_{i,t} \) is the \( t \)th token of question \( q_i \) and \( q_{i,<t} \) refers to all tokens preceding the \( t \)th token. Our finetuning objective is the sum of this loss across all training questions.

3.3 Data Augmentation
For a question generation system to be robust in selecting the best question for context-answer pairs with multiple relevant questions, it must first be able to generate diverse and suitable question candidates for a given context-answer pair. Moreover, education experts who created the FairytaleQA dataset followed the pattern of first reading the context, then writing a question, and finally writing the answer. This process implies that there could often be multiple valid questions associated with the same context-answer pair in addition to the ground-truth question, which can be used to augment the dataset (as seen in Table 1). Therefore, we propose an automated data augmentation method to enrich the training set with diverse and relevant questions for each context-answer-question triplet. We prompt a larger LM, OpenAI Codex (Chen et al., 2021), in an in-context prompting fashion (Brown et al., 2020) to first generate diverse questions for each context-answer pair and then filter out unsuitable questions with consistency matching; we detail both steps below.

Synthetic Data Generation. We first generate synthetic data, i.e., \( M = 4 \) diverse question candidates \( \{\hat{q}_{i,1}, \ldots, \hat{q}_{i,M}\} \) for each context-answer-question triplet \((c_i, a_i, q_i)\) using the OpenAI Codex LM (Chen et al., 2021) in an in-context prompting fashion. We construct the in-context prompt by randomly selecting five context-answer-question triplets from the training set with the same attribute as the target context-answer-question triplet to augment. We then append the target triplet followed by the prompt: “Another question with the same answer is”. These examples help Codex to adapt to the style of questions written by education experts. We use nucleus sampling (Holtzman et al., 2020) to generate synthetic questions with a \( p \) value of 0.9 and temperature of 0.8 to ensure diversity.

Consistency Matching. Since there is no guarantee that the generated questions are faithful and match the context-answer pair, we filter out inconsistent questions using a consistency matching criterion inspired from Wang et al. (2021). A generated question is consistent with respect to its input context-answer-question triplet if the answer of the generated question is the same as (or similar to) the input ground-truth answer. This consistency criterion enables us to include diverse yet consistent synthetic questions to augment the ground-truth questions during training.

To obtain the answer of a generated question, we again use Codex in an in-context prompting fashion with a subtle change in the prompt. We use the same five in-context examples of context-answer-question triplets taken from the same attribute as the target context-answer-question being augmented. However, we change the earlier context-answer-question pattern suitable for question generation and reformulate in the order of context-question-answer appropriate for question answering. We denote the answer to the generated question \( \hat{q}_{i,j} \) as \( \bar{a}_{i,j} \). We use greedy decoding since we need the single best answer. We observe that comparing the similarity of this obtained answer generated by Codex to the ground truth answer \( a_i \) written by human education experts can sometimes exclude consistent synthetic questions incorrectly. We alleviate this issue by obtaining another reference answer to compare with; we prompt Codex in an in-context fashion to obtain the answer to the ground truth question \( q_i \), which we denote as \( \tilde{a}_i \). Note that \( \tilde{a}_i \) could be different from the ground truth answer \( a_i \) as shown in an example in Table 6 in the Supplementary Material.

To check consistency, we measure the similarity between \( \bar{a}_{i,j} \) and both \( a_i \) and \( \tilde{a}_i \) using the ROUGE-1 F1 score (Lin, 2004). If either similarity is greater than a threshold of 0.5, we include the context-answer-synthetic question triplet \((c_i, a_i, \hat{q}_{i,j})\) in our augmented training set. We outline our method in Figure 1 and also in Algorithm 1 in the Supplementary Material.
3.4 Overgenerate-and-Rank

Selecting the top question preferable to human educators from multiple relevant and diverse question candidates for the given context-answer pair is hard. We propose an overgenerate-and-rank method which first overgenerates several question candidates for each context-answer pair using the fine-tuned model (as described in Section 3.2). We use various decoding strategies, including nucleus sampling (Holtzman et al., 2020) and contrastive search (Su et al., 2022) to ensure diversity. We then rank these generated questions based on a criterion. We use two kinds of ranking methods, perplexity-based ranking and distribution matching-based ranking, which we detail below.

**Perplexity-based Ranking.** In this ranking method, we use perplexity as a metric to rank the generated questions. The perplexity of a language model given a question measures the uncertainty of generating the question under that language model. The lower the perplexity of a question, the more probable is the question according to the language model. We first overgenerate $K = 10$ questions for the given context-answer pair using nucleus sampling or contrastive search. We then compute the perplexity of these questions given the fine-tuned language model. We then select the question with the lowest perplexity as the best question for the given context-answer pair.

**Distribution Matching-based Ranking.** In this ranking method, we fine-tune a separate language model to rank the overgenerated question candidates by predicting scores over these generated questions with a similar distribution to the ROUGE-L scores between the generated questions and the ground truth question. This distribution matching objective encourages the ranking language model to associate higher scores with questions similar to the ground truth question written by human educators. We select the question with the highest score predicted by the ranking model as the best question for the given context-answer pair. Our method inspired from (Shi et al., 2023) trains a ranking language model to minimize the KL divergence (Joyce, 2011) between the distribution of the model-predicted scores over the generated questions and the distribution of ROUGE-L scores computing similarity of the generated questions to the human educator-written ground truth question. We outline the training process of the ranking model in Figure 2.

More specifically, we use a pre-trained ConvBERT (Jiang et al., 2020) model as our ranking language model. We use a combination of the given context-answer pair and the generated question to rank as input to the model. We feed the [CLS] embedding vector to a learnable linear layer during fine-tuning. For the $i$th training question, $P_{\theta}(\hat{q}_i) \in [0, 1]^K$ denotes the probability distribution of the model-predicted scores for generated questions and $R(\hat{q}_i, q_i) \in [0, 1]^K$ denotes the probability distribution of the ROUGE-L scores.
between the generated questions and the ground-truth question. Equation 2 shows the fine-tuning objective of the ranking language model to minimize the KL divergence between the model-predicted score distribution and the ROUGE-L score distribution. The softmax in equation 3 computes the distribution of the model-predicted scores where \( \phi(\hat{q}_{i,j}, c_i, a_i) \) denotes the score predicted by the ranking language model for the \( j \)th generated question \( \hat{q}_{i,j} \) corresponding to the \( i \)th context-answer pair \((c_i, a_i)\). The softmax in equation 4 computes the probability distribution of the ROUGE-L scores where \( r(\hat{q}_{i,j}, q_i) \) denotes the ROUGE-L score between the \( j \)th generated question \( \hat{q}_{i,j} \) and the ground-truth question \( q_i \). The hyperparameters \( \alpha_P \) and \( \alpha_R \) control the temperature of the softmax over the model-predicted scores and the ROUGE-L scores, respectively. The optimization problem is formally written as:

\[
\text{minimize}_{\phi} \frac{1}{N} \sum_{i} KL(P_{\phi}(\hat{q}_i)||R(\hat{q}_i, q_i)), \tag{2}
\]

where

\[
[P_{\phi}(\hat{q}_i)]_{j} = \frac{\exp \alpha_P \phi(\hat{q}_{i,j}, c_i, a_i)}{\sum_{j} \exp \alpha_P \phi(\hat{q}_{i,j}, c_i, a_i)}, \tag{3}
\]

\[
[R(\hat{q}_i, q_i)]_{j} = \frac{\exp \alpha_R r(\hat{q}_{i,j}, q_i)}{\sum_{j} \exp \alpha_R r(\hat{q}_{i,j}, q_i)}. \tag{4}
\]

4 Experimental Evaluation

In this section, we describe the experimental setup to validate the effectiveness of our question generation methods.

4.1 Metrics and Baselines

To compare with prior work (Xu et al., 2022b), we use the ROUGE-L F1 score (Lin, 2004) (referred to as ROUGE-L) to evaluate the quality of generated questions. We compare our question generation methods to the existing state-of-the-art baseline (Xu et al., 2022b) which fine-tunes a BART LM (Lewis et al., 2020) to generate the ground-truth question conditioned on the given context-answer pair.

4.2 Implementation Details

We use a pre-trained Flan-T5-Large model (Chung et al., 2022) with 770M parameters as our base LM for question generation; all implementation was done using the HuggingFace (Wolf et al., 2020) transformers library. We fine-tune the base LM for 10 epochs with early stopping on the validation loss using the AdamW (Loshchilov and Hutter, 2017) optimizer with a learning rate of \( 3e^{-4} \) and a batch size of 8. Each epoch takes 20 minutes on a single NVIDIA A100 GPU.

FairytaleQA is imbalanced with respect to question attributes, with action and causal relationship accounting for 60% of the dataset. Our data augmentation method generates around 2500 synthetic questions over only the minority attributes: character, setting, feeling, outcome resolution, and prediction, to balance the training set. We fine-tune our base LM with the same setup described before on the augmented training set using a weight \( \lambda \) for the loss objective (see Equation 1) with original human educator-written questions and a different weight \( 1 - \lambda \) for synthetic questions. Through a grid search, we find that setting \( \lambda = 0.8 \) results in the best performance.

Our overgenerate-and-rank method generates question candidates using contrastive search (Su et al., 2022) (top-k of 4, \( \alpha \) penalty of 0.6) and nucleus sampling (Holtzman et al., 2020) (top-p of 0.9, temperature of 1) for perplexity-based ranking and distribution matching-based ranking, respectively. Through a grid search, we find that setting the softmax temperature hyperparameters as \( \alpha_P = 1 - 3 \) and \( \alpha_R = 1 - 2 \) results in the best performance.

5 Results and Discussion

Overall Performance. We report the average ROUGE-L across all test questions in the FairytaleQA dataset for all question generation methods in Table 2. The choice of the base language model is key when fine-tuning language models for question generation; fine-tuning Flan-T5 provides a significant improvement of 3.7% over the current state-of-the-art baseline of fine-tuning BART (Xu et al., 2022b), possibly because Flan-T5 is instruction fine-tuned on a large number of tasks relevant to both question answering and question generation. Our data augmentation method, which enriches the training set with diverse questions, further improves performance by 0.25% over fine-tuning Flan-T5 on the original training set. Among our overgenerate-and-rank methods, perplexity-based ranking and distribution matching-based ranking provide a 0.5% and 1.4% improvement over fine-tuning Flan-T5, respectively. Overall, our best method, distribution matching-based ranking method, provides a 5% absolute improvement over
### Method Questions

<table>
<thead>
<tr>
<th>Method</th>
<th>Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>All</td>
</tr>
<tr>
<td>BART (Xu et al., 2022b)</td>
<td>0.5270</td>
</tr>
<tr>
<td>Flan-T5</td>
<td>0.5639</td>
</tr>
<tr>
<td>Data Augmentation</td>
<td>0.5664</td>
</tr>
<tr>
<td>Perplexity-based Ranking</td>
<td>0.5689</td>
</tr>
<tr>
<td>Distribution Matching-based Ranking</td>
<td><strong>0.5778</strong></td>
</tr>
</tbody>
</table>

Table 2: Experimental results on the FairytaleQA dataset in ROUGE-L (higher is better). Our methods significantly outperform existing baselines.

The current state-of-the-art BART baseline. This significant improvement shows that our data augmentation and overgenerate-and-rank methods are effective at making question-generation systems more robust, which results in better questions being generated. We also experiment with combining our data augmentation and overgenerate-and-rank methods. However, perhaps surprisingly, this combination does not lead to significant improvement in performance. We think that this result is possibly due to synthetic questions being too diverse in many cases with respect to the ground truth question. Therefore, controlling the diversity of synthetic questions for better alignment with those written by human educators is an important direction for future work.

### Performance Stratified by Question Category

To gain more insight into the performance of our question generation methods, we also report the average ROUGE-L over test questions in the explicit and implicit categories. For the harder implicit questions with answers not explicitly included in the context as text spans, our data augmentation and distribution matching-based ranking methods improve performance by 1.2% and 2.3% over fine-tuning Flan-T5, respectively. This significant performance improvement shows that our data augmentation and overgenerate-and-rank methods are well-suited for harder question generation tasks, especially when given an answer that needs to be inferred from the context, for which the ground-truth questions are already highly diverse.

### Data Augmentation Variants

We report ROUGE-L scores for several variants of our data augmentation method in Table 4 in the Supplementary Material. FairytaleQA is imbalanced with respect to question attributes, with action and causal relationship accounting for 60% of the dataset. Augmenting all questions across all attributes results in a drop in performance. This observation validates our best data augmentation method, which is to generate synthetic questions for only the minority attributes: character, setting, feeling, outcome resolution, and prediction, to balance the training set. Moreover, fine-tuning Flan-T5 by weighting the human educator-written questions and synthetically-generated questions differently further improves performance.

### Different Decoding Strategies

We report ROUGE-L scores for our overgenerate-and-rank methods combined with different choices of decoding strategy for overgeneration: greedy, nucleus sampling, and contrastive search, in Table 5 in the Supplementary Material. We compare perplexity-based ranking and two variants of distribution matching-based ranking trained on questions generated by nucleus sampling and contrastive search, respectively. We see that there is no single best decoding strategy that works across all ranking methods. We also observe that using the same decoding strategy for overgenerating candidate questions for both training and testing of the ranking method might not provide the best performance. For example, the distribution matching-based ranking method trained on questions generated by contrastive search works best at test time by ranking questions generated by nucleus sampling.

### 5.1 Qualitative Analysis

**Analysis of Questions Generated.** We provide a qualitative analysis of our question generation methods on an example context-answer-question triplet from the test set of FairytaleQA in Table 3. We observe that there are multiple relevant questions with different linguistic style and structure for the example context-answer pair; among them, our question generation methods need to generate the human educator-written ground truth question, “What did the man tell dullhead to do?”. Our fine-tuned Flan-T5 model generates a plausible but vague question excluding the subject of the context, “the old man”, that is not very similar to the ground truth question, possibly due to limitations.
… and when they had finished the little grey old man said to the dullhead: “Now I will bring you luck, because you have a kind heart and are willing to share what you have with others. there stands an old tree; cut it down, and amongst its roots, you’ll find something.” …

| Context | … and when they had finished the little grey old man said to the dullhead: “Now I will bring you luck, because you have a kind heart and are willing to share what you have with others. there stands an old tree; cut it down, and amongst its roots, you’ll find something.” … |
| Answer | cut down an old tree. |
| Ground truth question | What did the man tell dullhead to do? |
| Flan-T5 | What did dullhead need to do to find something? |
| Data Augmentation | What did the little grey old man tell dullhead to do? |
| Perplexity-based Ranking | 1. What did the little man tell dullhead to do because he was willing to share what he had? 2. What did the little man tell dullhead to do because he wanted to find something? 3. What will dullhead do after he has eaten and drank the cake and beer? 4. What will dullhead do to find something? 5. What will dullhead do when he meets the grey old man? |
| Distribution Matching-based Ranking | 1. What did the grey old man ask dullhead to do? 2. What did the little grey old man say he wanted dullhead to do? 3. What did the little man tell dullhead to do because he was willing to share what he had? 4. What did the little man tell dullhead to do because he wanted to find something? 5. What will dullhead need to do? |

Table 3: Qualitative analysis with an example input context-answer-question from the FairytaleQA dataset and question generated by our methods. Both data augmentation and overgenerate-and-rank improve diversity among the generated questions, which makes question generation more robust.

of greedy decoding. Our data augmentation method generates a much better question that is similar in structure and style to the ground truth question, which suggests that training on diverse questions is effective.

We also show the top five questions among the candidates, ranked by our overgenerate-and-rank methods. Our perplexity-based ranking method improves upon the fine-tuned Flan-T5 model by matching the structure of the ground truth question, “What did the man tell dullhead …”, but favors longer questions with more context information than the human educator-written question. Our distribution matching-based ranking method performs best by matching both the structure and style of the ground truth question. This example demonstrates that ranking methods trained on actual human preference information can be effective at identifying human-like questions among diverse candidates.

**Error Analysis.** We randomly select 30 context-answer pairs from the FairytaleQA test set with low ROGUE-L scores (less than 0.2) and investigate the questions generated by our best method, distribution matching-based ranking, and analyze why it does not perform well in these cases. We identify three main error types and list them in Table 7 in the Supplementary Material, with corresponding examples containing the input context-answer pair, the ground truth question, and the best generated question. The three main error types are: 1) character coreference resolution, 2) out-of-context ground-truth questions, and 3) multiple evidence angles in the context.

The first two error types are beyond our control but the third type suggests that our methods have plenty of room for improvement. Errors of type character coreference resolution can occur when an input context has multiple characters and coreferences. In the first example, “self” is used as a complex coreference and confuses the question generation method. Errors of type out-of-context ground-truth questions can occur for ground-truth questions using information present outside the context the model sees as input. These ground-truth questions are human errors often referring to named entities present in other sections of the same story but not included in the input context. In the second example, the ground truth question refers to the character “Ian” who is not present in the context; the generated question uses the reference of “fisher’s son” that is has access to in the given context. Errors of type multiple evidence angles can occur when the input context discusses different aspects of an answer. In the third example, the event of “Norseman invasion” in the answer could have questions related to either its cause, “people being wicked”, or its timeline, “happening after the two Countesses fled to Scotland” As a result, among the top decoder output questions, there are none that discusses the latter, which is contained in the ground-truth question. Therefore, it is important to develop methods that can take all possible question
angles into account during decoding.

6 Conclusions and Future Work

In this paper, we proposed methods for improving automated answer-aware reading comprehension question generation by generating diverse question candidates and ranking them to align with human educator preferences. First, we proposed a data augmentation method that augments the training dataset with diverse questions obtained from a larger language model. Second, we proposed an overgenerate-and-rank method with two choices of ranking criterion, perplexity-based ranking and distribution matching-based ranking. The latter learns to rank the generated candidate questions to select ones that are closer to human-written questions. We conducted extensive experiments on the FairytaleQA dataset to validate the effectiveness of our methods showing that our best method provides an absolute improvement of 5% in ROUGE-L over the current state-of-the-art on this dataset. We also showed that our methods are significantly better than baselines in generating harder questions whose answers are not directly present in the context as text spans and have to be inferred.

There are several directions for future work. First, we can experiment with other data augmentation methods, e.g., by fine-tuning the base language model by weighting synthetically-generated questions according to their ROUGE-L scores with respect to the ground truth question. Second, we can explore the use of chain-of-thought (Wei et al., 2022) or self-ask (Press et al., 2022) to prompt the large language model in our data augmentation method. Third, we can experiment with other ranking objectives, such as ones using the Bradley-Terry model (Bradley and Terry, 1952) or ones using reinforcement learning with human feedback framework (Ziegler et al., 2019), to select the best questions that are aligned with human preference. Fourth, we can apply our methods to other question generation scenarios that require reasoning, such as logistical questions in online course discussion forums (Zyllich et al., 2020), to help instructors anticipate common student questions.

Acknowledgements

We thank the anonymous reviewers for their helpful comments. We thank the Learning Agency Lab for organizing the Quest for Quality Questions challenge\(^3\) which inspired our initial work. We also thank Alexander Scarlatos and Naiming Liu for helpful discussions around this work. The authors also thank the NSF (under grants 1917713, 2118706, 2202506, 2215193, 2237676) for partially supporting this work.

References


\(^3\)https://www.thequestchallenge.org


Supplementary Material

T ← \{(c_1, a_1, q_1), \ldots, (c_N, a_N, q_N)\};
for i ← 1 to N do
\{\hat{q}_{i,1}, \ldots, \hat{q}_{i,M}\} ← 
GenQuesCodex((c_i, a_i, q_i));
\tilde{a}_i ← GenAnsCodex((c_i, q_i));
for j ← 1 to M do
\hat{a}_{i,j} ← GenAnsCodex((c_i, \hat{q}_{i,j}));
if ROUGE(\hat{a}_{i,j}, a_i) > 0.5 or ROUGE(\hat{a}_{i,j}, \tilde{a}_i) > 0.5 then
T ← T \cup \{(c_i, a_i, \hat{q}_{i,j})\};
end
end

Algorithm 1: Our automated data augmentation method which first generates question candidates and then filters them using consistency matching.

<table>
<thead>
<tr>
<th>Decoding Type</th>
<th>Perplexity-based Ranking</th>
<th>Distribution Matching-based Ranking with Nucleus Sampling (0.95, 1, 10)</th>
<th>Distribution Matching-based Ranking with Contrastive Search (4, 0.6, 10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greedy (No ranking)</td>
<td>0.5639</td>
<td>0.5639</td>
<td>0.5639</td>
</tr>
<tr>
<td>Nucleus Sampling (0.9, 1, 10)</td>
<td>0.5664</td>
<td><strong>0.5778</strong></td>
<td>0.5657</td>
</tr>
<tr>
<td>Nucleus Sampling (0.95, 1, 10)</td>
<td>0.5618</td>
<td>0.5717</td>
<td><strong>0.5678</strong></td>
</tr>
<tr>
<td>Nucleus Sampling (0.95, 1, 75)</td>
<td>0.5671</td>
<td>0.5766</td>
<td>0.5638</td>
</tr>
<tr>
<td>Contrastive Search (4, 0.6, 10)</td>
<td><strong>0.5689</strong></td>
<td>0.5719</td>
<td>0.5647</td>
</tr>
</tbody>
</table>

Table 5: Experimental results on the FairytaleQA dataset in ROUGE-L (higher is better) comparing different decoding strategies across our overgenerate-and-rank methods. We denote Nucleus Sampling N questions with a p value of P and temperature of T as Nucleus Sampling (P, T, N) and Contrastive Search of N questions with a top-k of K and alpha penalty of A as Contrastive Search (K, A, N).

<table>
<thead>
<tr>
<th>Data Augmentation Method Variant</th>
<th>ROUGE-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Augmentation</td>
<td>0.5639</td>
</tr>
<tr>
<td>All Questions</td>
<td>0.5499</td>
</tr>
<tr>
<td>Minority Questions</td>
<td>0.5607</td>
</tr>
<tr>
<td>Minority Questions + \lambda Weighting</td>
<td><strong>0.5664</strong></td>
</tr>
</tbody>
</table>

Table 4: Experimental results on the FairytaleQA dataset in ROUGE-L (higher is better) comparing different variants of our data augmentation method.
. . . and with that the rat laid a linen thread in the youth’s hand. “Heaven be praised!”, said the youth when he was up above once more. “I’ll not go down there again in a hurry.” But he held the thread in his hand and danced and sang as usual . . .

Table 6: Our data augmentation method on an example context-answer pair from FairytaleQA. We use two reference answers for consistency matching. In this example, although the generated answer of generated question (happy) does not match the reference ground truth answer (excited), the generated question is still consistent and included in the augmented training set since the generated answer matches the alternate reference of generated answer of the ground truth question (happy).

Table 7: Qualitative error analysis of our best method, distribution matching-based ranking, showing error examples from the FairytaleQA for each error type.