Towards Diverse and Effective Question-Answer Pair Generation from Children Storybooks

Sugyeong Eo[1*], **Hyeonseok Moon**[1*], **Jinsung Kim**[1*], **Yuna Hur**[1*], **Jeongwook Kim**[1*]
Songeun Lee[2], **Changwoo Chun**[2], **Sungsoo Park**[2], **Heuiseok Lim**[1†]
[1]Dept. of Computer Science and Engineering, Korea University
[2]Hyundai Motor Group
{djtnrud,limhseok}@korea.ac.kr songeun.lee@hyundai.com

Abstract

Recent advances in QA pair generation (QAG) have raised interest in applying this technique to the educational field. However, the diversity of QA types remains a challenge despite its contributions to comprehensive learning and assessment of children. In this paper, we propose a QAG framework that enhances QA type diversity by producing different interrogative sentences and implicit/explicit answers. Our framework comprises a QFS-based answer generator, an iterative QA generator, and a relevancy-aware ranker. The two generators aim to expand the number of candidates while covering various types. The ranker trained on the in-context negative samples clarifies the top-N outputs based on the ranking score. Extensive evaluations and detailed analyses demonstrate that our approach outperforms previous state-of-the-art results by significant margins, achieving improved diversity and quality. Our task-oriented processes are consistent with real-world demand, which highlights our system's high applicability. Our code is available at https://github.com/sugyeonge/Towards-diverse-QAG.git.

1 Introduction

Pedagogical studies over the years have demonstrated that asking questions about a given storybook nurtures insight and expands knowledge (Janusheva and Pejchinovska, 2009; Etemadzadeh et al., 2013; Shanmugavelu et al., 2020). Hence, posing questions becomes a fundamental part of education to engage children and promote literacy (Cotton, 1988; Ellis, 1993; Dillon, 2006). Along with the remarkable strides in natural language processing, recent studies have actively explored question-answer pair generation (QAG) systems that target education (Xu et al., 2022; Yao et al., 2022; Zhao et al., 2022). As QAG is a labor-intensive manual process, it benefits from automated production methods. Furthermore, sustainable system update and utilization emphasize their high applicability (Le et al., 2014; Jerome et al., 2021).

A challenge in educational QAG is the diversity of generated QA pairs as well as their quality (Lee et al., 2020; Zhang et al., 2021). Exploiting various QA types facilitates comprehensive learning, as each question inquires information specific to its type and stimulates different brain activities in the answering process (Guszak, 1967; Dillon, 2006). Controlling difficulty by adopting different types of questions or answers also enables a balanced assessment of the reading comprehension skills of children (Xu et al., 2022). Consequently, actively using questions with various interrogative words and answers reflecting both implicitness and explicitness is important. Yet, existing educational QAG studies have rarely considered diversity. Generated questions of existing models are extremely biased to the 'What' and 'Who' type questions. Answer extraction focuses on detecting spans within passages, resulting in an inability to create implicit answers that do not directly appear in the passage.

To address the limitation, we propose an effective QAG framework that enhances diversity and quality. Our framework consists of a QFS-based answer generator, an iterative QA generator, and a relevancy-aware ranker. Specifically, **QFS-based answer generator** adopts query-focused summarization (QFS)-based (Vig et al., 2022) answer generation model (AGM), with the aim of obtaining diverse and proper answer candidates. **Iterative QA generator** is designed to increase question type variety by exploiting the interrogative word-indicated question generation model (QGM). We jointly execute this QGM with the question-answering model (QAM) to adjust the final answers. **Relevancy-aware ranker** inspects quality to determine the final top-N outputs among the generated candidates.

[*] Equal Contribution
[†] Corresponding Author

Findings of the Association for Computational Linguistics: ACL 2023, pages 6100–6115
July 9-14, 2023

To grasp better pairs with high relevancy, the ranker is trained using in-context negative samples.

The experimental results indicate that our framework outperforms the existing state-of-the-art method by a large margin, with a gain of up to 0.435→0.503 on MAP@N with Rouge-L f1 and 0.9077→0.9178 on MAP@N with BERTScore (Yao et al., 2022). Additional statistical and human evaluations with detailed analyses consistently show higher QA type diversity and quality compared to previous studies, which demonstrates the superiority of the proposed approach. The three modules of our framework are process-oriented, providing outputs from each, which is in line with the real-world demand investigated by Wang et al. (2022). This highlights the high applicability to the education field in terms of Human-AI collaboration. We summarize our contributions as follows:

(i) We propose a novel QAG framework that enhances the diversity of question and answer types while increasing quality.

(ii) Extensive experiments show that our framework remarkably outperforms previous state-of-the-art results with high diversity and relevance.

(iii) The task-oriented process is consistent with real-world demand, emphasizing the applicability of our framework in the education field.

2 Related Works

The question-answer pair generation (QAG) task aims to automatically generate QA pairs based on the input text. In the early days, rule-based QAG systems are dominant (Lindberg et al., 2013; Labutov et al., 2015). With the advent of a deep learning-based paradigm, it was demonstrated for the first time by Du et al. (2017) that a fully end-to-end QAG system generates exceptionally good questions. Accordingly, diverse studies have been conducted and developed (Shakeri et al., 2020; Li et al., 2022; Zhou et al., 2019). Kang et al. (2019) adopt an interrogative words-based approach to clarify the semantics of words from a passage, resulting in the generation of questions containing key information of the context. Scialom et al. (2019) attempt to generate questions in an answer-agnostic manner by adapting the self-attention mechanism of Transformers with a copying mechanism, placeholders, and contextual word embeddings. Dong et al. (2022) propose a QAG model for closed-book setting without access to external knowledge by modeling the semantic relationships between questions and answers at a contextual level and measuring the answerability of the generated questions.

In recent times, several attempts have been made to automatically generate valid QA pairs for educational purposes. FairytaleQA proposed by Xu et al. (2022) is a representative dataset in educational QAG. Education experts manually generated QA pairs suitable for learning and assessing children's reading comprehension skills. With the dataset, Yao et al. (2022) present an educational QAG system through a combination of three-step modules. Zhao et al. (2022) deal with the high-cognitive demand question generation based on three out of seven narrative elements in the FairytaleQA. Dugan et al. (2022) summarize QA pairs to given book chapters and provide them to the fine-tuned T5 (Raffel et al., 2020) models.

However, these studies rarely consider diversity when performing QAG. Leveraging diverse QA types are important aspect in QAG since using various interrogative words promotes different parts of the brain, which facilitates children's comprehensive learning (Guszak, 1967; Dillon, 2006). Varying answer types is also a factor that contributes to a balanced assessment, as the difficulty can be controlled by adjusting whether the answer is revealed in the passage (Xu et al., 2022). The evidence emphasizes the importance of considering a variety of QA pairs from a broad perspective for the effectiveness of reading comprehension (Kim, 2017).

3 Method

Our QAG framework comprises three task-oriented processes: a QFS-based answer generator, an iterative QA generator, and a relevancy-aware ranker. The main goal of the two generators is to expand QA pair candidates containing diverse question and answer types. The ranker aims to determine the final output by scoring QA pair candidates. The overall QAG architecture of our framework is depicted in Figure 1.

3.1 QFS-based Answer Generator

In the initial answer generation process, we employ query-focused summarization (QFS) to capture salient information related to a given sentence. After the QFS model generates a query-focused

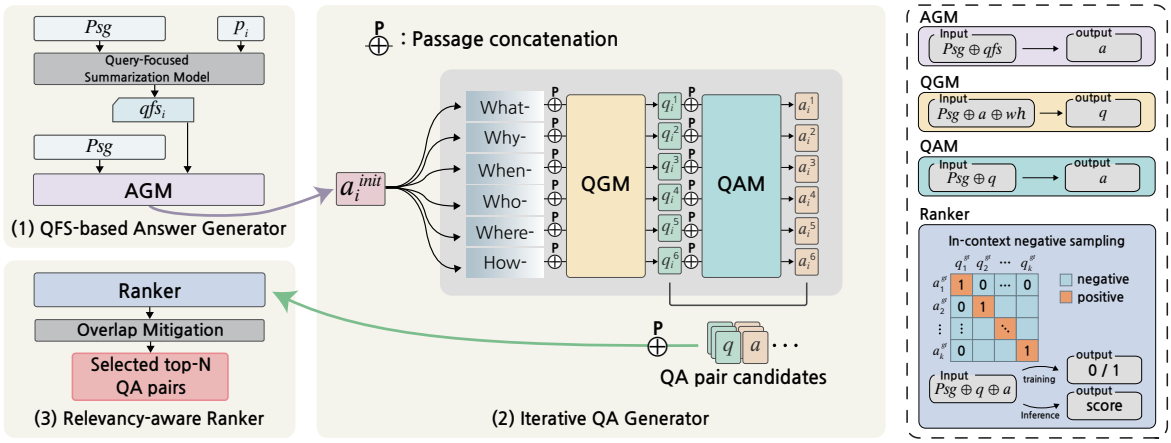

Figure 1: Overall architecture of our QAG framework. The rightmost side on the figure describes the training process of each model.

summary of a given passage by referring to the relevant key, the summary is fed into the generative answer generation model (AGM) to output implicit or explicit answers.

Let *Psg* denote a passage consisting of n sentences, $p_1, ..., p_n$, and the corresponding ground-truth (GT) QA pair be $(Q^{\text{gt}}, A^{\text{gt}}) = \{(q_j^{\text{gt}}, a_j^{\text{gt}})\}_{j=1}^{m}$. First, we generate a query q_j^{gt} focused summary $qfs_j^{\text{gt}} = QFS(Psg, q_j^{\text{gt}})$ of *Psg* using the pre-trained QFS model, *QFS*. We then train AGM, termed as θ_{AGM}, with the concatenated input of *Psg* and qfs_j^{gt} in a sequence-to-sequence manner. The loss function for each *Psg* is estimated as shown in Equation (1).

$$L_{\text{AGM}} = - \sum_{(q_j^{\text{gt}}, a_j^{\text{gt}}) \in (Q^{\text{gt}}, A^{\text{gt}})} \mathbf{E}_{\theta_{\text{AGM}}}(a_j^{\text{gt}} \mid Psg, qfs_j^{\text{gt}}) \quad (1)$$

In the inference phase, for each sentence p_i in *Psg*, we generate $qfs_i = QFS(Psg, p_i)$. Then AGM produces a single initial answer a_i^{init} for corresponding qfs_i. The resulting answer set A^{init} has n answers since answers are generated for every sentence in the passage. A^{init} is expressed as follows:

$$A^{init} = \{\theta_{\text{AGM}}(Psg, qfs_i) \mid p_i \in Psg\} \quad (2)$$

3.2 Iterative QA Generator

After the initial answer set A^{init} is generated, the next step is to expand the QA pair candidates to reflect the question type diversity. To achieve this, we propose an interrogative word-indicated question generation model (QGM), denoted by θ_{QGM}, and a generative question-answering model (QAM), denoted by θ_{QAM}. The QGM and QAM are sequentially executed based on the initial answer to generate a set of QA pair candidates. The following paragraphs describe the training and inference processes of each model.

Interrogative word-indicated QGM We train QGM with GT QA pair set to generate questions by referring to the answers and their passages. Including interrogative words in the training phase allows controllable question generation to follow the desired interrogative type during inference.

We denote the interrogative word of each q_j^{gt} in a GT QA pair set as wh_j^{gt}. In our setting, wh is an element of the interrogative word set *WH*={Who, When, What, Where, Why, How}. θ_{QGM} is trained to generate question q_j^{gt} by feeding the concatenated input of Psg, a_j^{gt}, and wh_j^{gt}. Training is performed in a sequence-to-sequence manner and is optimized using the following loss function:

$$L_{\text{QGM}} = - \sum_{(q_j^{\text{gt}}, a_j^{\text{gt}}) \in (Q^{\text{gt}}, A^{\text{gt}})} \mathbf{E}_{\theta_{\text{QGM}}}(q_j^{\text{gt}} \mid Psg, a_j^{\text{gt}}, wh_j^{\text{gt}}) \quad (3)$$

In the inference phase, we prioritize diversity and generate questions by considering each interrogative word in *WH* as an indicator. For each $a_i^{init} \in A^{init}$ generated in the first step and its corresponding passage *Psg*, θ_{QGM} configures QA pair set QA^1 which can be expressed as follows:

$$QA^1 = \{(\theta_{\text{QGM}}(Psg, a_i^{init}, wh), a_i^{init}) \mid wh \in WH, \; a_i^{init} \in A^{init}\} \quad (4)$$

In this way, QA pair candidates with high relevance to the passage can be generated. Note that

this process encourages the expansion of question types, not all questions generated are related to the initial answers.

Answer Adjustment To consider relevancy between QA pairs, we reconstruct answers through θ_{QAM} trained with a set of GT QA pairs. This process helps avoid linking inappropriate questions to a given initial answer, such as asking a 'How' question for an answer aimed at a specific person. Training of θ_{QAM} is proceeded by optimizing the following loss function.

$$L_{\text{QAM}} = - \sum_{(q_j^{\text{gt}}, a_j^{\text{gt}}) \in (Q^{\text{gt}}, A^{\text{gt}})} \mathbf{E}_{\theta_{\text{QAM}}}(a_j^{\text{gt}} \mid Psg, q_j^{\text{gt}}) \quad (5)$$

In the subsequent inference phase, we adjust the answers to all questions in QA^1 through θ_{QAM}. The reconstructed QA pair set, denoted by QA^2, is expressed as Equation (6).

$$QA^2 = \{(q_i^j, \theta_{\text{QAM}}(Psg, q_i^j)) \mid (q_i^j, a_i^{init}) \in QA^1\} \quad (6)$$

QA^2 is a final QA pair candidate set in which the relevance between the pairs is supervised through the QAM while maintaining the diversity of question types.

3.3 Relevancy-aware Ranker

With the relevancy-aware ranker model, we select top-N ranked QA pairs that exhibit high relevance between passages and QA pairs.

The ranking model denoted by θ_{Rank} produces the relevance score for each QA pair. To train the ranking model θ_{Rank}, we compose a contrastive training dataset by collecting in-context negative samples in GT QA pair set. In the training data, the GT QA pairs are considered a positive samples, and the other QA pairs within the same passage are considered negative samples. For a given passage Psg and the corresponding GT QA pair set $(Q^{\text{gt}}, A^{\text{gt}})$, we construct positive sample set $POS = \{(q_i^{\text{gt}}, a_j^{\text{gt}}) \mid q_i^{\text{gt}} \in Q^{\text{gt}}, a_j^{\text{gt}} \in A^{\text{gt}}, i = j\}$ and negative sample set $NEG = \{(q_i^{\text{gt}}, a_j^{\text{gt}}) \mid q_i^{\text{gt}} \in Q^{\text{gt}}, a_j^{\text{gt}} \in A^{\text{gt}}, i \neq j\}$[1].

Then the QA pairs and their corresponding passages are concatenated to construct the input sequences for training θ_{Rank}. By feeding this input sequence, θ_{Rank} is trained to classify binary labels representing negative and positive.

In the inference phase, θ_{Rank} returns the scores of the input QA pair to be classified as positive and negative, respectively. We further rank each QA pair by referring both scores.

Through this process, the ranker is trained to prioritize the selection of data that exhibits a high correlation between QA pairs and high relevance to the corresponding passages.

Overlap Mitigation While the ranker model enhances the relevance of QA pairs, the issue of duplication exists where the top-ranked pairs constitute similar forms. To alleviate this issue, we compute a re-scaled ranking score to diminish the lexical overlap of answers in the QA pair candidates.

We sequentially select QA pairs in the order of high scores computed using the ranking model. To consider lexical overlap in each selection process, we measure the Rouge-L score between the selecting pair and the previously selected QA pairs. The score s of each pair measured by the ranking model is re-scaled as $s - Rouge * abs(s)$. Through this process, we down-scale the scores of the QA pairs that exhibit high lexical overlap with previously selected QA pairs. This allows the selection of various types of QA while reflecting the scores calculated by the ranking model. The detailed procedure of the overlap mitigation algorithm is presented in Algorithm 1 in Appendix A.

4 Experiments

4.1 Experimental Setup

Dataset In our experiments, we leverage the FairytaleQA dataset (Xu et al., 2022). FairytaleQA is specifically designed for children's storybook learning and assessment, which corresponds to our purpose of education. In the data construction process, educational experts manually created QA pairs to ensure reliability and validity. The training, validation, and test sets contain 8,548 QA pairs from 232 books, 1,025 pairs from 23 books, and 1,007 pairs from 23 books, respectively. Instead of using narrative elements (*i.e.* character, setting, action, etc.) presented in the dataset, we diversify the questions based on interrogative words to induce expanded types of questions beyond these elements. We use the existing answer types, as they are mutually exclusive.

[1]We consider QA pairs in a different passages as easy negative cases and do not include them as negative samples in the ranker training.

Method	MAP@N (Rouge-L F1)				MAP@N (BERTScore F1)			
	Top 10	Top 5	Top 3	Top 1	Top 10	Top 5	Top 3	Top 1
FQAG(Yao et al., 2022)	0.440 / 0.435	0.375 / 0.374	0.333 / 0.324	0.238 / 0.228	0.9077 / 0.9077	0.8990 / 0.8997	0.8929 / 0.8922	**0.8768** / 0.8776
SQG(Dugan et al., 2022)	0.460 / 0.455	0.392 / 0.388	0.344 / 0.337	0.234 / 0.242	0.9056 / 0.9062	0.8953 / 0.8955	0.8876 / 0.8878	0.8707 / 0.8723
Ours	**0.500 / 0.503**	**0.426 / 0.429**	**0.369 / 0.372**	**0.247/ 0.254**	**0.9156 / 0.9178**	**0.9046 / 0.9068**	**0.8956 / 0.8977**	0.8752 / **0.8783**

Table 1: The main experimental results for our QAgen framework. We report Map@N score with Rouge-L F1 and BERTScore F1 for each model. The result for the validation split is on the left side, and the right side is for the test split.

Method	*global*			*local*				
	Diversity-Q ↓	Diversity-A ↓	Quality-E↓	Relevancy ↑	Acceptability ↑	Usability ↑	Readability ↑	Difficulty ↑
FQAG(Yao et al., 2022)	3.03	3.06	2.66	2.65	2.14	1.74	**2.64**	1.11
SQG(Dugan et al., 2022)	2.96	3.03	3.30	2.44	1.87	1.34	2.55	1.36
Ours	**2.35**	**2.18**	**2.35**	**2.69**	**2.22**	**1.9**	2.35	**1.98**
GT	1.65	1.71	1.68	2.97	2.65	2.50	2.80	1.95

Table 2: Human evaluation results for the QA pairs generated by the QAG systems on eight criteria. *global* represents the human ranking results for the three QAG systems and GT. *local* indicates the human scoring results for each QAG system and GT, on a 0-3 scale. Note that the scores between the two settings are completely different.

Models All models comprising our framework are trained with the FairytaleQA dataset. In the case of the QFS model, we produce a summary using model checkpoints provided by Vig et al. (2021). In training AGM, QGM, and QAM, we exploit the pre-trained BART-large (Lewis et al., 2020) model and framework provided by Fairseq[2]. For the hyper-parameters, 2048 max tokens, early stopping 10, and polynomial decay scheduler are adopted. For the learning rate and dropout, we set 3e-05 and 0.1 in AGM and QGM, 2e-05 and 0.2 in QAM, respectively. All models are trained on 2 RTX8000 GPUs. We used the RoBERTa-base (Liu et al., 2019) model and Huggingface[3] framework for our ranking model. We train it for five epochs with a fixed learning rate of 5e-07 and a single GPU is used for training ranker.

4.2 Evaluation Metrics

For the evaluation metric, we adopt the MAP@N score as a primary metric utilized by Yao et al. (2022). MAP@N with Rouge-L refers to the averaged value of the maximum score set added by computing Rouge-L between each GT pair and the top-N generated QA pairs. Each question and answer in the QA pair is concatenated in the process. However, when MAP@N is measured by the Rouge-L precision score as in Yao et al. (2022), short results are advantageous. This is because it measures the longest overlap over the number of candidates. Instead of using precision, we select the F1 score for accurate measurement. Since the metrics based on the n-gram overlap do not guarantee quality (Zhang and Bansal, 2019), we additionally adopt BERTScore for MAP@N to evaluate semantic equivalence based on similarity scores (Zhang et al., 2019).

4.3 Baselines

We adopt two educational QAG systems as a baseline.

FQAG *FQAG* (Yao et al., 2022) is a state-of-the-art study of FairytaleQA. They perform QAG through a three-step pipeline comprising answer generation, question generation, and ranking modules. For re-implementation, we load the provided checkpoints to generate QA pairs for the validation and test sets of FairytaleQA.

SQG *SQG* (Dugan et al., 2022) is a recently published paper in educational QAG, which utilizes summaries of the given passages. QA pairs are generated leveraging three models: answer generation, question generation, and question answering models. In this case, to match the number of top-N, we select QA pairs based on the generated order or increase outputs by adjusting the beam size.

5 Results and Analysis

5.1 Automated Evaluation

Result on MAP@N with Rouge-L Table 1 shows the main result of MAP@N with Rouge-L F1 scores according to the QAG systems. As

[2] https://github.com/facebookresearch/fairseq.git

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

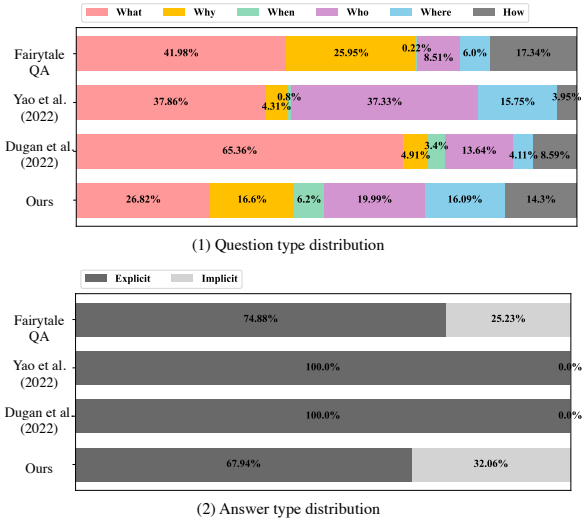

Figure 2: Distribution of the question and answer types on the test set. It represents the portion of what, why, when, who, where, and how from left to right.

a result, our system significantly outperforms the baseline model in all splits and top-N outcomes. Especially in the test set, we outperform *FQAG* by +0.068 in the top 10, +0.055 in the MAP@5, and +0.048 in the MAP@3, which is a significant gain. *SQG* achieves better results than *FQAG* but still does not outperform ours. Compared to the *SQG*, our system shows improvement in all top-N results mainly from 0.455 to 0.503 (+0.048). The result implies that generating various QA pair candidates and properly establishing plausible pairs serve as one contributing factor to performance improvement.

Result on MAP@N with BERTScore We measure MAP@N by employing BERTScore to evaluate the semantic equivalence between GT and generated QA pairs. Namely, we use the F1 value of BERTScore instead of Rouge-L F1 score when measuring MAP@N. As a result, our system achieves higher performance in all settings except for the MAP@1 validation result. In the best case from the test result (MAP@10), *FQAG* and *SQG* showed 0.9077 and 0.9062 respectively and we recorded 0.9178, outperforming by +0.0101 and +0.0116. The tendency for our performance to be the highest is consistent with the MAP@N with Rouge-L F1 result. However, we observe that *FQAG* reports higher performance in BERTScore than *SQG*. Although the performance gap is marginal, this outcome suggests that the generated QA pairs of *FQAG* are semantically better than *SQG*.

5.2 Statistical Evaluation

To evaluate the question and answer type diversity of generated QA pairs, we perform a statistical evaluation. The distribution according to interrogative type and answer type is presented in Figure 2. As a result, the reported question types of ours are more balanced than others. Unlike other models that usually create 'what' and 'who', our QAG system is well balanced with questions of 'why' and 'how' that require reasoning. This suggests the potential of children to think from various perspectives by being asked different types of questions.

For answer types, our system contains 32.06% of implicit answers, indicating that implicit answers are also well generated, which allows our model to help balance assessments of children. Conversely, other models use the answer span extraction method, resulting in a 0% of implicit answers.

5.3 Human Evaluation

We further conduct a human evaluation for a detailed inspection. For each paragraph, three human evaluators with degree holders or experts in education rate each of the three QA pairs generated by the GT and three QAG systems. Human evaluation is performed on a total of 20 passages, and we select three QA pairs sequentially for unscored GT and *SQG*. Due to the brevity of space, we further describe human evaluation details in the Appendix C. The following criteria are used for human evaluation. For the *global* setting, we instruct the evaluator to rank the entire system, and in the *local* cases to select how many of the three QA pairs generated by each system correspond to property items.

(*global* setting) **Diversity-Q**: This ranks the generation results of GT and three QAG systems in terms of question diversity. **Diversity-A**: This ranks the generation results of GT and the three QAG systems in terms of answer diversity. **Quality-E**: This ranks the entire system quality from an overall perspective.

(*local* setting) **Relevancy**: This evaluates the relevance between a passage and a QA pair. If either question or answer is not relevant, it is irrelevant. **Acceptability**: This evaluates whether a question and its corresponding answer are correctly generated. Relevance with the passage is not considered, and if either of them is awkward, it is considered incorrect. **Usability**: This evaluates whether the generated QA pairs can be used for education purposes. **Readability**: This evaluates whether the

generated QA pairs are grammatically right. **Difficulty**: This evaluates whether the generated QA pairs are excessively easy.

Table 2 presents the results of the human evaluation. Our approach achieves remarkable performance in terms of the question and answer diversity with an average ranking of 2.35 and 2.18 respectively. In the global setup, we observe that the Quality-E is 2.66 in *FQAG* and 3.30 in *SQG*, while our system outperforms them with a score of 2.35. These results demonstrate that our QAG is both quantitatively and qualitatively superior in direct comparison with other systems through ranking while enhancing diversity. The results of the local setting indicate that we outperform both *FQAG* and *SQG* except for the readability. As an evaluation result of the QA pairs we generated, the relevance of the passages to the generated QA (2.69), the acceptance of the questions to the answers (2.22), and the usability for educational purposes (1.9) show the highest result compared to other systems. We even observe a slight performance gain over GT in case of difficulty. However, in readability, our result showed 2.35, which is lower than the 2.64 and 2.55 of the existing model. We speculate that the average length of our generated QA pairs is longer, resulting in a small trade-off with difficulty. From the results, we conclude that our generated QA pairs are truly effective in not only ensuring quality but also diversity.

5.4 Ablation Study

We perform ablation studies to further analyze the contribution of each process in our framework to the overall performance. The results are shown in Table 3.

Impact of Query-focused Summarization Instead of the AGM model, we generate answers using noun phrase and noun entity extraction method performed in *FQAG*. When the AGM model is changed, the performance decreased by -0.031 in MAP@10. This indicates that introducing a summary containing intensive information benefits from generating more plausible answers. We also analyze that introducing the AGM model in a generative manner yields higher performance because it is capable to generate implicit answers.

Impact of Iterative QA Pair Generation For investigating the most effective iteration, we reduce the number of iterations in the iterative QA generator. Namely, we eliminate the QAM step and execute only QGM. The experimental results show a marginal change in most of the cases, except for the top 10 results. We attribute this performance drop in the top 10 to the process of additionally adjusting interrogative word-indicated questions to correct answers through QAM. An experiment on increasing iterations is presented in Section 6.

	MAP@N (Rouge-L F1)			
Method	**Top 10**	**Top 5**	**Top 3**	**Top 1**
Ours	0.503	0.429	0.372	0.254
w/o QFS	0.472	0.401	0.348	0.248
w/o Iteration	0.463	0.427	0.378	0.253
w/o Contrastive learning	0.438	0.375	0.326	0.261

Table 3: Ablation results on the test set to claim the necessity of each module. Every module functions in *Ours*.

Impact of Contrastive Learning To observe the role of our relevancy-aware ranker, we eliminate our ranker and utilize the DistilBERT (Sanh et al., 2019) ranking model of *FQAG*. As a result of the experiment, the overall performance is degraded largely, such as 0.503→0.438 in the MAP@10 and 0.429→0.375 in the MAP@5. We interpret that constituting the training examples for contrastive learning through in-context negative samples boosts the overall performance gain.

However, our performance of changing the ranking model to that of *FQAG* can also be compared with the *FQAG* test result of Table 1 (MAP@10: 0.435, MAP@5: 0.374, MAP@3: 0.324, MAP@1: 0.228). This case is a comparison in which the ranking model is unified and only varies the QA pair generation part. Results show an insignificant difference, with *FQAG* test results in Table 1 performing lower than ablation results of contrastive learning. We analyze that our method generates more QA pair candidates with the goal of increasing diversity, but the DistilBERT ranking model does not rank them well.

6 Case Study

Performance of Multiple AGM We investigate various methods to add a clue that can be a key element when constructing AGM inputs. The clue is then fed into the BART large-based AGM input along with the given passages, and the answer is predicted. *A* is the baseline where this learns to directly generate answers for given passages. In *DS*, one to three sentences in the passage closest to

the question are retrieved. In *Ext-Ret*, a phrase or sentence closest to the question is retrieved from the external resource NarrativeQA (Kočiskỳ et al., 2018).

Method	Rouge-L	BLEU	BERTScore
A	0.216	8.31	0.875
DS1	0.232	10.21	0.879
DS2	0.244	9.65	0.874
DS3	0.256	10.55	0.878
Ext-Ret (Sent)	0.283	14.86	0.89
Ext-Ret (Phrase)	0.304	16.74	0.896
QFS	**0.362**	**23.21**	**0.903**

Table 4: FairytaleQA test set evaluation results according to the answer generation model

Table 4 is the experimental result, and the performance of QFS outperforms all other methodologies. For this result, we judge that the summary, in which the information is compressed and regenerated, contributes more to the final answer generation.

Performance on Adding Iteration We observe the performance fluctuation when increasing iteration on the iterative QA generator. We create QA pairs by recursively executing QGM and QAM on the QA pairs generated by our main framework. Experimental results in Table 5 show that the performance degrades as the iteration increases. We judged that no additional performance improvement would be obtained even if iterations were repeated more than this.

	Map@N (Rouge-L F1)			
Method	**Top 10**	**Top 5**	**Top 3**	**Top 1**
Ours	0.503	**0.429**	**0.372**	**0.254**
Ours +1 iteration	**0.506**	0.423	0.361	0.246
Ours +2 iteration	0.502	0.419	0.362	0.243

Table 5: Result on iteration. +1 iteration refers to additionally attaching a QGM model. +2 refers to successively applying the QAM model.

Performance on Overlap Mitigation Methods In this section, we investigate the effect of overlap mitigation techniques. *EM* is a baseline, which remains the highest-scored QA pair for each unique *Criterion*.

The experiment is designed to modify two factors: In *Criterion*, we divide the criterion of overlap measurement into two parts of question or answer. *Overlap Metric* divides the overlap measurement metric into BLEU and Rouge.

Criterion	Overlap Metric	MAP@N (Rouge-L F1)			
		Top 10	**Top 5**	**Top 3**	**Top 1**
Answer	EM	0.491	0.414	0.357	0.254
	BLEU	0.497	**0.431**	0.369	0.254
	Rouge-L	**0.503**	0.429	**0.372**	0.254
Question	EM	0.483	0.404	0.354	0.254
	BLEU	0.491	0.421	0.365	0.254
	Rouge-L	0.**493**	**0.431**	0.**366**	0.254

Table 6: Experimental results for various overlap mitigation methods. The Top 1 score for the overall models is the same since the QA pair with the highest score is always selected first.

The experimental results are presented in Table 6. The results represent that re-scaling the scores of the ranking model by using overlap mitigation methods yields higher performance than the method of simply removing overlap based on exact matching. Also, overall performance shows better when the overlap metric is set to Rouge than BLEU. This demonstrates that the output of the ranking model can be utilized more effectively by applying the proposed overlap mitigation method. Notably, the overlap mitigation method based on the answer record higher performance when the question is used as the criterion.

7 Conclusion

In this paper, we proposed a QAG framework for educational purposes featuring diverse and valid question and answer types. Our framework is structured with three task-oriented processes, with a particular emphasis on expanding diverse and valid types of QA pair candidates in the generator, and selecting high-quality QA pairs in the ranker. We conducted extensive evaluations of generated QA pairs, including quantitative, qualitative, and statistical evaluations with detailed analyses, and observed that our system achieved remarkable performance. Our framework has the potential to promote various cognitive activities in children learning by providing diverse and effective QA pairs for educational purposes. As our modularized task-oriented frameworks are tailored to real-world demand, we further expect the collaborative use of humans and AI.

Limitations

We used only the pre-trained BART-large model when training each model within the QAG framework. We assume that comparative experiments using several sequence-to-sequence language models will be good future works. Also, we only used six interrogative words, and did not consider 'whose' and 'whom' in the process. We considered these as originating from 'who', but generating eight interrogative words including 'whose' and 'whom' would be a good approach. At last, in order to create a robust ranker, it is best to have a dataset that contains positive and negative samples. Since the manual data generation process required a time-consuming process, we utilize in-context negative samples as an alternative. If there is a dataset for the ranker learning purpose, much better performance can be achieved.

Ethics Statement

Deployment Our approach exploits parametric knowledge in the pre-trained model for language generation, which runs the risk of reflecting the bias of the training data. Undoubtedly, it is a well-known threat in tasks using a pre-trained model, but we must be more careful about social impact when using this method since our model aims to create educational QAs. Therefore, we plan to request model users to necessarily include a human review process of the generated QA pairs when used for educational purposes.

Human evaluation We paid human workers more than the legal minimum wage. We also guided them to work remotely at any time they wanted and to rest when they are in a state of fatigue during work. Their B.A. degree certificate was discarded immediately upon confirmation to prevent personal information leakage. We made a task force to quickly respond to them if they have any questions or concerns by contacting them directly.

Acknowledgments

We thank the anonymous reviewers for their valuable feedback and constructive suggestions. This work was supported by Hyundai Motor Company and Kia. This research was supported by the Ministry of Science and ICT (MSIT), Korea, under the Information Technology Research Center (ITRC) support program (IITP-2023-2018-0-01405) supervised by the Institute for Information & Communications Technology Planning & Evaluation (IITP) and this work was supported by IITP grant funded by MSIT (No. 2020-0-00368, A Neural-Symbolic Model for Knowledge Acquisition and Inference Techniques) and this research was supported by MSIT, under the ICT Creative Consilience program(IITP-2023-2020-0-01819) supervised by the IITP.

References

Kathleen Cotton. 1988. Classroom questioning. *School improvement research series*, 5:1–22.

James T Dillon. 2006. Effect of questions in education and other enterprises. In *Rethinking schooling*, pages 145–174. Routledge.

Xiangjue Dong, Jiaying Lu, Jianling Wang, and James Caverlee. 2022. Closed-book question generation via contrastive learning. *arXiv preprint arXiv:2210.06781*.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1342–1352.

Liam Dugan, Eleni Miltsakaki, Shriyash Upadhyay, Etan Ginsberg, Hannah Gonzalez, DaHyeon Choi, Chuning Yuan, and Chris Callison-Burch. 2022. A feasibility study of answer-agnostic question generation for education. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1919–1926, Dublin, Ireland. Association for Computational Linguistics.

Kathleen Ellis. 1993. Teacher questioning behavior and student learning: What research says to teachers.

Atika Etemadzadeh, Samira Seifi, and Hamid Roohbakhsh Far. 2013. The role of questioning technique in developing thinking skills: The ongoing effect on writing skill. *Procedia-Social and Behavioral Sciences*, 70:1024–1031.

Frank J Guszak. 1967. Teacher questioning and reading. *The reading teacher*, 21(3):227–234.

Violeta Janusheva and Milena Pejchinovska. 2009. Questions posing importance and role in the teaching process.

Bill Jerome, Rachel Van Campenhout, and Benny G Johnson. 2021. Automatic question generation and the smartstart application. In *Proceedings of the Eighth ACM Conference on Learning@ Scale*, pages 365–366.

Junmo Kang, Haritz Puerto San Roman, and Sung-Hyon Myaeng. 2019. Let me know what to ask: Interrogative-word-aware question generation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 163–171.

Young-Suk Grace Kim. 2017. Why the simple view of reading is not simplistic: Unpacking component skills of reading using a direct and indirect effect model of reading (dier). *Scientific Studies of Reading*, 21(4):310–333.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Klaus Krippendorff. 2011. Computing krippendorff's alpha-reliability.

Igor Labutov, Sumit Basu, and Lucy Vanderwende. 2015. Deep questions without deep understanding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 889–898.

Nguyen-Thinh Le, Tomoko Kojiri, and Niels Pinkwart. 2014. Automatic question generation for educational applications–the state of art. *Advanced computational methods for knowledge engineering*, pages 325–338.

Dong Bok Lee, Seanie Lee, Woo Tae Jeong, Donghwan Kim, and Sung Ju Hwang. 2020. Generating diverse and consistent QA pairs from contexts with information-maximizing hierarchical conditional VAEs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 208–224, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Yunji Li, Sujian Li, and Xing Shi. 2022. Consecutive question generation via dynamic multitask learning. *arXiv preprint arXiv:2211.08850*.

David Lindberg, Fred Popowich, John Nesbit, and Phil Winne. 2013. Generating natural language questions to support learning on-line. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 105–114.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Thomas Scialom, Benjamin Piwowarski, and Jacopo Staiano. 2019. Self-attention architectures for answer-agnostic neural question generation. In *Proceedings of the 57th annual meeting of the Association for Computational Linguistics*, pages 6027–6032.

Siamak Shakeri, Cicero Nogueira dos Santos, Henghui Zhu, Patrick Ng, Feng Nan, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2020. End-to-end synthetic data generation for domain adaptation of question answering systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5445–5460, Online. Association for Computational Linguistics.

Ganesan Shanmugavelu, Khairi Ariffin, Manimaran Vadivelu, Zulkufli Mahayudin, and Malar Arasi RK Sundaram. 2020. Questioning techniques and teachers' role in the classroom. *Shanlax International Journal of Education*, 8(4):45–49.

Jesse Vig, Alexander Fabbri, Wojciech Kryscinski, Chien-Sheng Wu, and Wenhao Liu. 2022. Exploring neural models for query-focused summarization. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1455–1468, Seattle, United States. Association for Computational Linguistics.

Jesse Vig, Alexander R Fabbri, Wojciech Kryściński, Chien-Sheng Wu, and Wenhao Liu. 2021. Exploring neural models for query-focused summarization. *arXiv preprint arXiv:2112.07637*.

Xu Wang, Simin Fan, Jessica Houghton, and Lu Wang. 2022. Towards process-oriented, modular, and versatile question generation that meets educational needs. *arXiv preprint arXiv:2205.00355*.

Ying Xu, Dakuo Wang, Mo Yu, Daniel Ritchie, Bingsheng Yao, Tongshuang Wu, Zheng Zhang, Toby Li, Nora Bradford, Branda Sun, Tran Hoang, Yisi Sang, Yufang Hou, Xiaojuan Ma, Diyi Yang, Nanyun Peng, Zhou Yu, and Mark Warschauer. 2022. Fantastic questions and where to find them: FairytaleQA – an authentic dataset for narrative comprehension. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 447–460, Dublin, Ireland. Association for Computational Linguistics.

Bingsheng Yao, Dakuo Wang, Tongshuang Wu, Zheng Zhang, Toby Li, Mo Yu, and Ying Xu. 2022. It is AI's turn to ask humans a question: Question-answer pair generation for children's story books. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 731–744, Dublin, Ireland. Association for Computational Linguistics.

Ruqing Zhang, Jiafeng Guo, Lu Chen, Yixing Fan, and Xueqi Cheng. 2021. A review on question generation from natural language text. *ACM Transactions on Information Systems (TOIS)*, 40(1):1–43.

Shiyue Zhang and Mohit Bansal. 2019. Addressing semantic drift in question generation for semi-supervised question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2495–2509.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Zhenjie Zhao, Yufang Hou, Dakuo Wang, Mo Yu, Chengzhong Liu, and Xiaojuan Ma. 2022. Educational question generation of children storybooks via question type distribution learning and event-centric summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5073–5085, Dublin, Ireland. Association for Computational Linguistics.

Wenjie Zhou, Minghua Zhang, and Yunfang Wu. 2019. Multi-task learning with language modeling for question generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3394–3399, Hong Kong, China. Association for Computational Linguistics.

A Overlap Mitigation Details

The detailed process for overlap mitigation is as follows. We first define **Criterion** and **Metric**. **Criterion** means a sentence to be subject to overlap checking among questions or answers, and **Metric** means an evaluation metric to measure overlap. In our paper, we suggest using Rouge-L, or BLEU, as Metric. In our main experiments, we choose criterion$_i$ as a$_i$ (*i.e.* answer in QA pair), and Metric as ROUGE-L. In this process, Metric returns overlap score between 0 and 1. In estimating overlap, we lemmatize all the sentences and remove all the stop words in every QA pair.

Algorithm 1 Overlapping based reranking

Given: Passage *Psg*,
Input: Generated QA pair $QA^{gen} = \{(q_i, a_i)\}_{i=1}^{N}$
Parameter: int k
Define: score$_i$ ← RankingModule(q_i, a_i, Psg)
Choose: criterion $_i$ ← q_i or a_i
Choose: Metric ← ROUGE-L or BLEU

output ← [], comparing ← []
while len(output) $\leq$ k **do**
  for (q_j, a_j) in QA^{gen} **do**
    if comparing is not EMPTY **then**
      overlaps$_j$ = [Metric(criterion$_j$, item) for item in comparing]
      overlap$_j$ = max(overlaps$_j$)
      Define: score$^*_j$ ← score$_j$ - overlap$_j$ * |score$_j$|
    else
      Define: score$^*_j$ ← score$_j$
    end if
  end for
  (q_i, a_i) ← **Pick from** QA^{gen} with highest score$^*_j$
  output ← **Append** (q_i, a_i)
  comparing ← **Append** criterion$_i$
  QA^{gen} ← **Pop** (q_i, a_i)
end while
return output

B Implementation Performance on BART QGM and QAM

	QGM			QAM		
	Rouge-L	BLEU	BERTScore	Rouge-L	BLEU	BERTScore
Ours	0.600	28.50	0.934	0.542	43.29	0.936

Table 7: Performances on the BART QGM and QAM model.

In the iterative QAGen process, the QGM model and QAM model generate QA pairs, thereby obtaining a variety of QA pair candidates. We leverage FairytaleQA dataset for the model training, and results are shown in Table 7.

Our model utilizes a BART-large model identical to Yao et al. (2022). Although the apples-to-apples comparison between the QGM model is impossible since ours are trained with interrogative word indicator, our QAM model performs slightly better than the result of *FQAG* (0.536 Rouge-L).

C Human Evaluation Details

In our human evaluation process, all evaluators are degree holders in education or educational domain experts. We provide an evaluation sheet in the form of an API, and evaluators check the part corresponding to each question or write a rank order.

Figure 3 describes the human evaluation script. In the local setting, we instruct evaluators to select how many of the three QA pairs produced by each

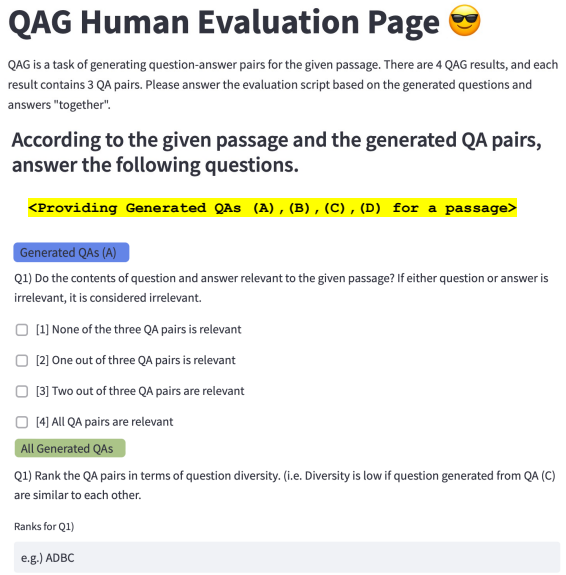

Figure 3: Human evaluation script. We provide three QA pairs, each generated by the three systems and GT, and instruct the evaluators to score or rank them for a total of eight evaluation categories.

system correspond to the given property. Rater responses are further converted to a 0-3 scale. We also inform raters to rank four systems according to each of the three attributes per paragraph in the global setting. Both evaluations are performed for a total of 20 paragraphs in common. The intercoder-reliability scores (Krippendorff's alpha) (Krippendorff, 2011) among the evaluators are between 0.59 and 0.61. This is because each attribute of the criterion requires highly subjective assessment.

D QAGen Results

To examine the practical applicability, we analyze the QA pairs generated by our framework. We compare our results to other QAgen systems, including *FQAG* and *SQG*, as well as the ground-truth QA pairs. We present the QAGen results for four passages, each containing three QA pairs. From a qualitative perspective, our framework generates QA pairs of higher quality compared to other systems.

Passage Long, long ago japan was governed by hohodemi, the fourth mikoto (or augustness) in descent from the illustrious amaterasu, the sun goddess. He was not only as handsome as his ancestress was beautiful, but he was also very strong and brave, and was famous for being the greatest hunter in the land. Because of his matchless skill as a hunter, he was called "yama - sachi - hiko" or "the happy hunter of the mountains."

Ours
Q) What was yama-sachi-hiko called?
A) The happy hunter of the mountains.

Q) Why was he called "the happy hunter of the mountains"?
A) He was matchless in his skill as a hunter.

Q) What was special about hohodemi?
A) He was not only as handsome as his ancestress was beautiful, but he was also very strong and brave.

FQAG
Q) What was yama-sachi?
A) The happy hunter of the mountains.

Q) What was hohodemi called?
A) Yama-sachi-hiko.

Q) Who was the greatest hunter in japan?
A) The fourth mikoto.

SQG
Q) What was the name of the fourth mikoto?
A) Hohodemi

Q) Hohodemi was a descendant of what goddess?
A) Amaterasu

Q) Hohodemi was the fourth mikoto from what goddess?
A) Sun

GT
Q) Who governed japan long ago?
A) Hohodemi.

Q) What was special about hohodemi?
A) Handsome.

Q) Why was hohodemi called yama-sachi-hiko?
A) His matchless skill as a hunter.

Passage Then the dragon king interviewed the doctor and blamed him for not curing the queen. The doctor was alarmed at rin jin's evident displeasure, and excused his want of skill by saying that although he knew the right kind of medicine to give the invalid, it was impossible to find it in the sea. "Do you mean to tell me that you can't get the medicine here?" asked the dragon king. "It is just as you say!" said the doctor. "Tell me what it is you want for the queen?" demanded rin jin. "I want the liver of a live monkey!" answered the doctor. "The liver of a live monkey! Of course that will be most difficult to get," said the king. "If we could only get that for the queen, her majesty would soon recover," said the doctor. "Very well, that decides it; we must get it somehow or other. But where are we most likely to find a monkey?" asked the king.

Ours
Q) Where did the doctor say it was impossible to find the right kind of medicine to give the invalid?
A) In the sea.

Q) What happened after the dragon king interviewed the doctor and blamed him for not curing the queen?
A) The doctor was alarmed at rin jin's evident displeasure, and excused his want of skill by saying that although he knew the right kind

Q) Who did the doctor think would recover from the liver of a live monkey?

A) Her majesty.

FQAG

Q) Who blamed the doctor for not curing the queen?
A) The dragon king.

Q) What did rin jinn want for the queen?
A) A live monkey.

Q) Who did the dragon king ask where they would find a monkey's liver?
A) The king.

SQG

Q) Who interviewed the doctor and blamed him for not curing the queen?
A) Dragon king

Q) Who blamed rin jin for not curing the queen?
A) The dragon king

Q) Who did the dragon king blame for not curing the queen?
A) Rin jin

GT

Q) Who was blamed for not curing the queen?
A) The doctor.

Q) Why was the liver of a live monkey difficult to get?
A) They are under water.

Q) How did the doctor feel when he was blamed by the king?
A) Alarmed.

Passage After a time they all came to a town where a king reigned whose daughter was so serious and solemn that no one could ever manage to make her laugh. So the king had decreed that whoever should succeed in making her laugh should marry her. When dullhead heard this he marched before the princess with his goose and its appendages, and as soon as she saw these seven people continually running after each other she burst out laughing, and could not stop herself. Then dullhead claimed her as his bride, but the king, who did not much fancy him as a son-in-law, made all sorts of objections, and told him he must first find a man who could drink up a whole cellarful of wine. Dullhead bethought him of the little grey man, who could, he felt sure, help him; so he went off to the forest, and on the very spot where he had cut down the tree he saw a man sitting with a most dismal expression of face.

Ours

Q) Where did dullhead see the man sitting with a dismal expression of face?
A) The very spot where he cut down the tree.

Q) What happened when the princess saw the seven people continually running after each other?
A) She burst out laughing and could not stop herself.

Q) Who did the king want dullhead to find before he could marry the princess?
A) A man who could drink up a whole cellarful of wine.

FQAG

Q) Who reigned in a town where no one could ever manage to make the daughter laugh?
A) A king.

Q) Who could help dullhead?
A) The little grey man.

Q) Where did dullhead go to find the little grey man?
A) The forest.

SQG

Q) What did the king decree that whoever succeeded in making her laugh should do?
A) Marry her

Q) How many people were running after each other?
A) Seven

Q) Where did dullhead go to find a man who could help him?
A) The forest

GT

Q) Who did the king decree should marry his daughter?
A) Whoever should succeed in making her laugh.

Q) How will the little grey man help dullhead?
A) Drink up a whole cellarful of wine.

Q) How did the king feel about dullhead as a son-in-law?
A) Unhappy.

Passage Many, many years ago there lived a good old man who had a wen like a tennis-ball growing out of his right cheek. This lump was a great disfigurement to the old man, and so annoyed him that for many years he spent all his time and money in trying to get rid of it. He tried everything he could think of. He consulted many doctors far and near, and took all kinds of medicines both internally and externally. But it was all of no use. The lump only grew bigger and bigger till it was nearly as big as his face, and in despair he gave up all hopes of ever losing it, and resigned himself to the thought of having to carry the lump on his face all his life.

Ours

Q) What did the good old man have?
A) A wen like a tennis-ball growing out of his right cheek.

Q) How long did the old man have the wen like a tennis-ball growing out of his right cheek?
A) Many, many years.

Q) Where did the lump grow out of?
A) His right cheek.

FQAG

Q) Who had a wen like atennis-ball growing out of his right cheek?
A) The old man.

Q) Where did the lump grow?
A) His right cheek.

Q) What did the old man do to get rid of his lump?
A) He consulted many doctors far and near.

SQG

Q) What type of ball did the old man have a wen like?

A) Tennis

Q) What was the wen like a tennis - ball growing out of his right cheek to the old man?
A) Great disfigurement

Q) What did the old man try to get rid of the lump?
A) Everything

GT

Q) Why was the man not able to get rid of his wen?
A) The doctors did not know how to get rid of it.

Q) How did the man feel about his wen?
A) Annoyed.

Q) What did the good old man have growing in his right cheek?
A) A wen.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations

☑ A2. Did you discuss any potential risks of your work?
Limitations

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract

☑ A4. Have you used AI writing assistants when working on this paper?
ChatGPT. We use it for checking grammar and finding synonyms for some words. This was done for specific sentences, but it appears throughout the section.

B ☑ Did you use or create scientific artifacts?

3 Method

☑ B1. Did you cite the creators of artifacts you used?
4.1 Experimental Setup - Models

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
We will discuss terms for use/distribution in the README on github.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
4.3 Baselines

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
We use existing data that is already anonymized.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
1 Introduction

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
4.1. Experimental Setup - Dataset

C ☑ Did you run computational experiments?

4.1. Experimental Setup - Models

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
4.1 Experimental Setup - Models

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
4.1 Experimental Setup - Models

☐ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
No response.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
4.2 Evaluation Metrics

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?

We conduct a human evaluation. 5.3 Human Evaluation

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
5.3 Human Evaluation, Appendix C Human Evaluation Details

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Ethics Statement, Appendix C Human Evaluation Details

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.