It is AI's Turn to Ask Human a Question: Question and Answer Pair Generation for Children Storybooks in FairytaleQA Dataset

Anonymous ACL submission

Abstract

Existing question answering (QA) techniques are created mainly to answer questions asked by humans. But in educational applications, teachers and parents sometimes may not know what questions they should ask best help children develop their narrative understanding abilities. We design an automated question-answer generation (QAG) system for education purposes: given a storybook at the kindergarten to eighth-grade level, our system can automatically produce QA pairs that are capable of testing a variety of student comprehension skills. Using a new QA dataset FairytaleQA that has 278 child-friendly storybooks with 10,580 QA pairs labeled by experts, we design a novel QAG system architecture to generate QA pairs. Automatic and human evaluations show that our model outperforms state-of-the-art QAG systems. On top of our QAG system, we also build an interactive story-telling application for future real-world deployment.

1 Introduction

There has been substantial progress in the development of state-of-the-art (SOTA) question-answering (QA) models in the natural language processing community in recent years. Training models for exceptional performances depend on the availability of high-quality, large-scale reading comprehension (RC) datasets. Such datasets should contain questions that focus on a well-defined construct (e.g., narrative comprehension) and measure a full coverage of sub-skills within this construct (e.g., reasoning causal relationship and understanding emotion within narrative comprehension) using items of varying difficulty levels (e.g., inference making and information retrieval). However, many of the existing datasets are either collected via crowd-sourcing (Rajpurkar et al., 2016; Kočiský et al., 2018; Reddy et al., 2019), or using automated question-answer pair (QA-pair) retrievers (Nguyen et al., 2016; Joshi et al., 2017; Dunn et al., 2017; Kwiatkowski et al., 2019), thus risking the quality and validity of labeled QA-pairs.

FairytaleQA Dataset Source (Section)

Maie sighed. she knew well that her husband was right, but she could not give up the idea of a cow. the buttermilk no longer tasted as good as usual in the coffee;

... ...

they were students, on a boating excursion, and wanted to get something to eat.'bring us a junket, good mother,' cried they to Maie.'ah! if only i had such a thing!' sighed Maie.

Ground-Truth

- **Q**: What did the three young men ask for?
- **A**: A junket.

`2-Step Baseline` (Shakeri et al., 2020)

- **Q**: Why no more buttermilk for her husband to make?
- **A**: She could not give up the idea of a cow.

PAQ Baseline (Lewis et al., 2021)

- **Q**: What did maie think of when she thought of buttermilk?
- **A**: Sweet cream and fresh butter.

Our System

- **Q**: Why did the three young men want a junket?
- **A**: They wanted to get something to eat.

Table 1: A sample of FairytaleQA story as input and the QA pairs generated by human education experts, `2-step baseline` model, `PAQ` baseline, and our QAG System.

This becomes especially problematic when applying QA models in the education domains. While existing QA models perform well in generating factually correct QA pairs, they fall short in generating *useful* QA pairs for educational purposes. As RC a complex skill vital for children's achievement and later success (Snyder et al., 2005), the community desperately needs a large-scale dataset that can support RC for education purposes. Furthermore, automated QA-pairs Generation (QAG) has been considered a promising approach to cost-efficiently build large-scale learning and assessment systems. Yet, existing RC datasets are not suitable for this task due to the aforementioned limitations (Das et al., 2021).

In this work, we target the lack of high-quality RC datasets in the educational domain. We aim to develop a QAG system to generate high-quality QA-pairs, similar to a teacher or parent would ask children when reading stories to them (Xu et al., 2021). Our system is built on a novel dataset we constructed in parallel, FairytaleQA. This dataset focuses on narrative comprehension for elementary to middle school students and contains 10,580 QA-pairs from 278 narrative text passages of classic fairytales. FairytaleQA is annotated by education experts and includes well-defined and validated narrative elements laid out in the education research (Paris and Paris, 2003), making it an appealing dataset for RC research in the education domain.

Our QAG system consists of a three-step pipeline: (1) to extract candidate answers from the given storybook passages through carefully designed heuristics based on a pedagogical framework; (2) to generate appropriate questions corresponding to each of the extracted answers using a state-of-the-art (SOTA) language model; and (3) to rank top QA-pairs with a specific threshold for the maximum amount of QA-pairs for each section.

We compare our QAG system with two existing SOTA QAG systems: a `2-step baseline` system (Shakeri et al., 2020) fine-tuned on FairytaleQA, and the other is an end-to-end generation system trained on a large-scale automatically generated RC dataset (`PAQ`) (Lewis et al., 2021). We evaluate the generated QA-pairs in terms of similarity by Rouge-L precision score with different thresholds on candidate QA-pair amounts and semantic as well as syntactic correctness by human evaluation. We demonstrate that our QAG system performs better in both the automated evaluation and the human evaluation.

We conclude the paper by demoing an interactive story-telling application that built upon our QAG system to exemplify the applicability of our system in a real-world educational setting.

2 Related Work

2.1 QA Datasets

There exists a large number of datasets available for narrative comprehension tasks. These datasets were built upon different knowledge resources and went through various QA-pair creating approaches. For instance, some focus on informational texts such as Wikipedia and website articles(Rajpurkar et al. (2016), Nguyen et al. (2016), Dunn et al. (2017), Kwiatkowski et al. (2019), Reddy et al. (2019)). Prevalent QA-pair generating approaches include crowd-sourcing (Rajpurkar et al., 2016; Kočiskỳ et al., 2018; Reddy et al., 2019), using automated QA-pair retriever (Nguyen et al., 2016; Joshi et al., 2017; Dunn et al., 2017; Kwiatkowski et al., 2019), and etc. Datasets created by the approaches mentioned above are at risk of not consistently controlling the quality and validity of QA pairs due to the lack of well-defined annotation protocols specifically for the targeting audience and scenarios. Despite many of these datasets involving large-scale QA pairs, recent research (Kočiskỳ et al., 2018) found that the QA pairs in many RC datasets do not require models to understand the underlying narrative aspects. Instead, models that rely on shallow pattern matching or salience can already perform very well.

NarrativeQA, for instance, (Kočiskỳ et al., 2018) is a large dataset with more than 46,000 human-generated QA-pairs based on abstractive summaries. Differing from most other RC datasets that can be answerable by shallow heuristics, the NarrativeQA dataset requires the readers to integrate information about events and relations expressed throughout the story content. Indeed, NarrativeQA includes a significant amount of questions that focus on narrative events and the relationship among events (Mou et al., 2021). One may expect that NarrativeQA could also be used for QAG tasks. In fact, a couple of recent works use this dataset and train a network by combining a QG module and a QA module with a reinforcement learning approach(Tang et al., 2017). For example, Wang et al. (2017) use the QA result to reward the QG module then jointly train the two sub-systems. In addition, Nema and Khapra (2018) also explore better evaluation metrics for the QG system. However, the NarrativeQA dataset is in a different domain than the educational context of our focus. Thus the domain adaptation difficulty is unknown.

2.2 QAG Task

A few years back, rule-based QAG systems (Heilman and Smith, 2009; Mostow and Chen, 2009; Yao and Zhang, 2010; Lindberg et al., 2013; Labutov et al., 2015) were prevalent, but the generated QA suffered from the lack of variety. Neural-based models for question generation tasks (Du et al., 2017; Zhou et al., 2017; Dong et al., 2019; Scialom et al., 2019) have been an emerging research theme

FairytaleQA Dataset	Train				Validation				Test			
	232 Books with 8548 QA-pairs				23 Books with 1025 QA-pairs				23 Books with 1007 QA-pairs			
	Mean	**S.D.**	**Min**	**Max**	**Mean**	**S.D.**	**Min**	**Max**	**Mean**	**S.D.**	**Min**	**Max**
# section per story	14.4	8.8	2	60	16.5	10.0	4	43	15.8	10.8	2	55
# tokens per story	2073.9	1320.1	208	7035	2365.8	1646.5	406	5762	2228.6	1340.7	310	6287
# tokens per section	143.6	61.7	12	434	143.1	54.5	31	298	140.4	55.6	24	285
# questions per story	36.8	28.9	5	161	44.5	29.5	13	100	43.7	28.8	12	107
# questions per section	2.8	2.440	0	18	2.9	2.3	0	16	3.0	2.4	0	15
# tokens per question	10.2	3.2	3	27	10.9	3.2	4	24	10.5	3.1	3	25
# tokens per answer	7.1	6.0	1	69	7.7	6.3	1	70	6.8	5.2	1	44

Table 2: Core statistics of the FairytaleQA dataset, which has 278 books and 10580 QA-pairs.

in recent years.

In this paper, we use a recent work Shakeri et al. (2020) as our baseline. They proposed a two-step and two-pass QAG method that firstly generate questions (QG), then concatenate the questions to the passage and generate the answers in a second pass (QA). In addition, we include the recently-published Probably-Asked Questions (PAQ) (Lewis et al., 2021) work as a second baseline. The PAQ system is an end-to-end QAG system trained on the PAQ dataset, a very large-scale QA dataset containing 65M automatically generated QA-pairs from Wikipedia. The primary issue with deep-learning-based models in the targeted children education application is that existing datasets and models do not consider the specific audience's language preference and the educational purposes (Hill et al., 2015; Yao et al., 2012).

Because both rule-based and neural-network-based approaches have their limitations inherently, in our work, we combine these two approaches to balance both the controllability of what types of QA pairs should be generated and the diversity of the generated QA sequences.

3 FairytaleQA Dataset

As previously mentioned, the general-purpose QA datasets (*e.g.,* SQuAD (Rajpurkar et al., 2016), MS MARCO (Nguyen et al., 2016)) are unsuitable for children education context, as they impose little structure on what comprehension skills are tested and heavily rely on crowd workers typically with limited education domain knowledge. To solve those issues and complement the lack of a high-quality dataset resource for the education domain, we developed a new RC dataset targeting students from kindergarten to eighth grade.

We developed the annotation schema based on an established framework for assessing reading comprehension (Paris and Paris, 2003), together with three experts in literacy education in our team. This schema is designed to comprehensively capture seven aspects contributing to reading comprehension (see Appendix for example in each aspect):

Character ask test takers to identify the character of the story or describe characteristics of characters

Setting ask about a place or time where/when story events take place and typically start with "Where" or "When."

Feeling ask about the character's emotional status or reaction to certain events and are typically worded as "How did/does/do . . . feel"

Action ask characters' behaviors or additional information about that behavior

Casual Relationship focus on two events that are causally related where the prior events have to causally lead to the latter event in the question. This type of questions usually begins with "Why" or "What made/makes."

Outcome Resolution ask for identifying outcome events that are causally led to by the prior event in the question. This type of questions is usually worded as "What happened/happens/has happened. . . after..."

Prediction ask for the unknown outcome of a focal event. This outcome is predictable based on the existing information in the text

The QA-pair generation process on FairytaleQA was accomplished by five annotators, all of whom have a B.A. in Education, Psychology, or Cognitive Science and have substantial experience teaching and assessing students' reading skills. These annotators were supervised by the education experts.

The annotators were instructed to develop QA pairs that address all of the seven narrative elements described above. They were also asked to create questions as if they were to assess elementary school students who have already read the complete story, and all questions should be open-ended

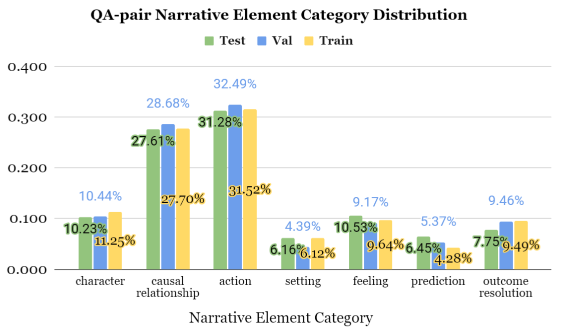


Figure 1: Distribution of the QA-pairs belongs to each of the seven narrative element categories in the FairytaleQA dataset.

"wh-" questions instead of "yes" or "no" questions.

A rigorous quality control protocol was implemented. All annotators received training lasting for weeks. In the training stage, they used the coding template to generate questions for the same stories. They then discussed their coding with their fellow annotators and the expert supervisors. After the formal coding began, weekly meetings were held to provide sufficient opportunities for review and discussion. All QA-pairs were reviewed by another annotator, and one-tenth were additionally reviewed by the expert supervisors. This process is to ensure that 1) the annotation guideline was followed, 2) the style of questions generated by coders was consistent, and 3) the questions focused on key information to the narrative and the answers to the questions were correct.

When the annotation task is accomplished, we obtain a dataset containing 10,580 high-quality QA-pairs from 278 books. We split the dataset into train/validation/test splits with 232/23/23 books and 8,548/1,025/1,007 QA pairs. The split is random, but the statistical distributions in each split are consistent. Table 2 shows core statistics of the FairytaleQA dataset in each split, and Figure 1 shows the distribution of seven types of annotations for the QA pairs across the three splits.

4 Question Answer Generation System Architecture

There are three sub-modules in our QA generation (QAG) pipeline: a heuristics-based answer generation module (AG), followed by a BART-based (Lewis et al., 2019) question generation module (QG) module fine-tuned on FairytaleQA dataset, and a DistilBERT-based(Sanh et al., 2019) ranking module fine-tuned on FairytaleQA dataset to rank and select top N QA-pairs for each input section. The complete QAG pipeline of our system is shown in Figure 2.

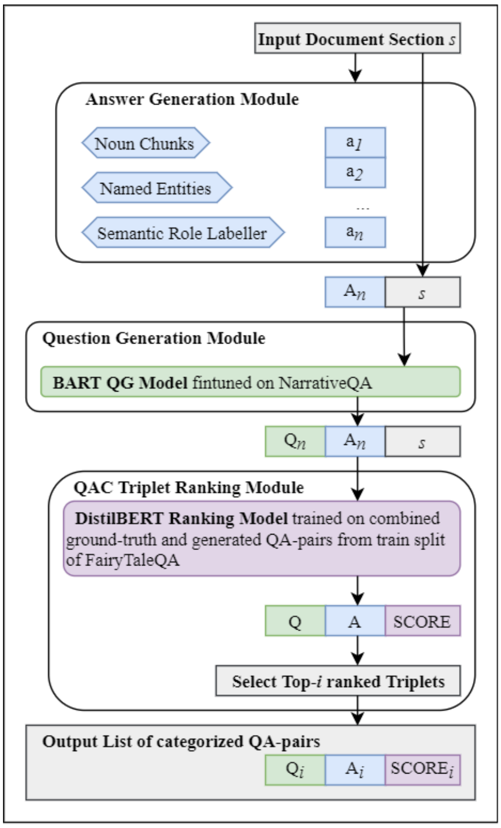

Input Document Section s
Answer Generation Module
Noun Chunks
Named Entities
Semantic Role Labeller
a_1
a_2
...
a_n
A_n
s
Question Generation Module
BART QG Model fintuned on NarrativeQA
Q_n
A_n
s
QAC Triplet Ranking Module
DistilBERT Ranking Model trained on combined ground-truth and generated QA-pairs from train split of FairyTaleQA
Q
A
SCORE
Select Top-i ranked Triplets
Output List of categorized QA-pairs
Q_i
A_i
$SCORE_i$

Figure 2: QAG system design with four steps: rule-based answer extraction, NN-based question generation, NN-based ranking, and QA classification.

4.1 Heuristics-based AG Module

Based on our observation of the FairytaleQA dataset, educational domain experts seem to have uniform preferences over certain types of question and answer pairs. This may be because these experts take the young children's learning objectives into consideration – children's learning ability should be oriented toward specific types of answers to maximize their learning outcome. That is why educational experts rarely ask yes/no questions in developing or assessing children's reading comprehension. For automated QAG systems, we can design the system to mimic human behaviors either by defining heuristics rules for the answer extraction module or leaving the filtering step to the end after the QA pairs are generated. However, the latter approach may have inherent risks that the training data could influence the types of answers generated.

We decided to develop and apply the heuristic rules to the answer extraction module. We observed that some narrative elements such as characters, setting, and feelings are mostly made up of name entities and noun chunks, for instance, the character name in a story, a particular place where the story takes place, or a specific emotional feeling. We then leverage the Spacy English model for Part-of-speech tagging on the input content to extract named entities and noun chunks as candidate answers to cover these three types of narrative elements.

We further observed that the QA pairs created by education experts around the action, causal relationship, prediction, and outcome resolution categories are all related to a particular *action event* in the story. Thus, the answers to these four types of questions are generally the description of the action event. We realize that Propbank's semantic roles labeler toolkit is constructive for extracting the action itself and the event description related to the action. We then leverage this toolkit to extract the trigger verb as well as other dependency nodes in the text content that can be put together as a combination of subject, verb, and object and use these as candidate answers for the latter four categories.

Our answer extraction module can generate candidate answers that cover all 7 narrative elements with the carefully designed heuristics.

4.2 BART-based QG Module

Following the answer extraction module that yields candidate answers, we design a QG module which takes a story passage and an answer as input, and generates the corresponding question as output. The QG task is basically a reversed QA task. Such a QG model could be either transfer-learned from another large QA dataset or fine-tuned on our FairytaleQA dataset. Mainstream QA datasets do cover various types of questions in order to comprehensively evaluate QA model's reading comprehension ability; for instance, NarrativeQA (Kočiskỳ et al., 2018) is a large-scale QA corpus with questions that examine high-level abstractions to test the model's narrative understanding.

We choose NarrativeQA dataset as an alternative option for fine-tuning our QG model because this dataset requires human annotators to provide a diverse set of questions about characters, events, etc., which is similar to the types of questions that education experts created for our FairytaleQA dataset. In addition, we leverage BART(Lewis et al., 2019) as the backbone model because of its superior performance on NarrativeQA according to the study in (Mou et al., 2021).

QG Models Comparison for Our QAG System	Rouge-L	
	Validation	Test
BART fine-tuned on NarrativeQA	0.424	0.442
BART fine-tuned on FairytaleQA	**0.527**	**0.527**
BART fine-tuned on NarrativeQA + FairytaleQA	0.508	0.519

Table 3: Comparison on FairytaleQA dataset among QG models fine-tuned with different settings for the QG module of our QAG system.

We perform a QG task comparison to examine the quality of questions generated for FairytaleQA dataset by one model fine-tuned on NarrativeQA, one on FairytaleQA, and the other on both the NarrativeQA and FairytaleQA. We fine-tune each model with different parameters and acquire the one with the best performance on the validation and test splits of FairytaleQA dataset. Results are shown in Table 3. We notice that the model fine-tuned on FairytaleQA alone outperforms the other methods. We attribute this to the domain and distribution differences between the two datasets. That is why the model fine-tuned on both NarrativeQA and FairytaleQA may be polluted by the NarrativeQA training. The best-performing model is selected for our QG module in the QAG pipeline.

4.3 DistilBERT-based Ranking Module

Our QAG system has generated all candidate QA-pairs through the first two modules. However, we do not know the quality between generated QA-pairs by far, and it is unrealistic to send back all the candidate QA-pairs to users in a real-world scenario. Consequently, a ranking module is added to rank and select the top candidate QA-pairs, where the user is able to determine the upper limit of generated QA-pairs for each input text content. Here, the ranking task can be viewed as a classification task between the ground-truth QA-pairs created by education experts and the generated QA-pairs generated by our systems.

We put together QA-pairs generated with the first two modules of our QAG system as well as ground-truth QA-pairs from the train/validation/test splits of FairytaleQA dataset, forming new splits for the ranking model, and fine-tuned on a pre-trained DistilBERT model. We test different input settings for the ranking module, including the concatenation of text content and answer only, as well as the concatenation of text content, question, and answer in various orders. Both input settings can achieve over 80% accuracy on the test split, while the input setting of the concatenation of text content, question, and answer can achieve $F1 = 86.7\%$ with a leading more than 5% over other settings. Thus, we acquire the best performing ranking model for the ranking module in our QAG system and allow users to determine the amount of top N generated QA-pairs to be outputted.

5 Evaluation

We provide one automated evaluation and one human evaluation for the QAG task. The input of the QAG task is a section of the story (may have multiple paragraphs), and the outputs are generated QA pairs. Unlike QA or QG tasks that each input corresponds to a single generated output no matter what model is used, the QAG task does not have a fixed number of QA-pairs to be generated for each section. Besides, various QAG systems will generate different amounts of QA-pairs for the same input content. Therefore, we carefully define an evaluation metric that is able to examine the quality of generated QA-pairs over a different amount of candidate QA-pairs. The comparison is on the validation and test splits of FairytaleQA.

5.1 Automated Evaluation of QAG Task

5.1.1 Baseline QAG Systems

We select a SOTA QAG system that uses a two-step generation approach (Shakeri et al., 2020) as one baseline system (referred as `2-Step Baseline`). In the first step, it feeds a story content to a QG model to generate questions; then, it concatenates each question to the content passage and generates a corresponding answer through a QA model in the second pass. The quality of generated questions not only relies on the quality of the training data for the QG and QA models but also is not guaranteed to be semantically or syntactically correct because of the nature of neural-based models.

We replicate this work by fine-tuning a QG model and a QA model on FairytaleQA dataset with the same procedures that help us select the best model for our QG module. We use pre-trained BART just like ours as the backbone model to ensure different model architectures do not influence the evaluation results. Unlike our QG module that takes both an answer and text content as the input, their QG model only takes the text content as input. Thus, we are not able to evaluate the QG model solely for this baseline. We replicate the fine-tuning parameters for our QG module to fine-tune the baseline QG model. For the selection of QA model used in the `2-Step Baseline`, similar to the QG experiments we present in Table 3, we fine-tune a pre-trained BART on each of the three settings: NarrativeQA only, FairytaleQA only, and both datasets. According to Table 4, the model that fine-tuned on both NarrativeQA and FairytaleQA datasets performs much better than the other settings and outperforms the model that fine-tuned on FairytaleQA only by at least 6%. We leverage the best performing QA model for the `2-Step Baseline` system.

QA Models for `2-Step Baseline`	Rouge-L	
	Validation	Test
BART fine-tuned on NarrativeQA	0.475	0.492
BART fine-tuned on FairytaleQA	0.533	0.536
BART fine-tuned on NarrativeQA + FairytaleQA	**0.584**	**0.601**

Table 4: Comparison on FairytaleQA dataset among QA models fine-tuned with different settings for the `2-Step Baseline` system.

In addition, we also include the recently published Probably-Asked Questions (`PAQ`) work as a second baseline system (Lewis et al., 2021). PAQ dataset is a semi-structured, very large scale Knowledge Base of 65M QA-pairs. `PAQ` system is an end-to-end QA-pair generation system that is made up of four modules: Passage Scoring, Answer Extraction, Question Generation and Filtering Generated QA-pairs. The `PAQ` system is trained on the PAQ dataset. It is worth pointing out that during the end-to-end generation process, their filtering module requires loading the complete PAQ corpus into memory for passage retrieval, which leads us to an out-of-memory issue even with more than 50G RAM. [1] In comparison, our QAG system requires less than half of RAM in the fine-tuning process.

5.1.2 Evaluation Metrics

Since the target of QAG task is to generate QA-pairs that are most similar to the ground-truth QA-pairs given the same text content, we concatenate the question and answer to calculate the Rouge-L

[1] we do not use the filtering module for `PAQ` system in the evaluation because of unable to solve the memory issue with their provided code.

precision score for every single QA-pair evaluation. However, the amount of QA-pairs generated by various systems is different. It is unfair and inappropriate to directly compare all the generated QA-pairs from different systems. Moreover, we would like to see how QAG systems perform with different thresholds on candidate QA-pair amounts. In other words, we are looking at ranking metrics that given an upper bound N as the maximum number of QA-pairs can be generated per section, how similar the generated QA-pairs are to the ground-truth QA-pairs.

Generally, there are three different ranking metrics: Mean Reciprocal Rank (MRR), Mean Average Precision (MAP), and Normalized Discounted Cumulative Gain (NDCG). While MRR is only good to evaluate a single best item from the candidate list and NDCG requires complete rank ratings for each item, neither metric is appropriate in our case. As a result, We decide to use $MAP@N$, where $N \in [1, 3, 5, 10]$, as our evaluation metric for the QAG generation task. Furthermore, since the average amount of ground-truth answers is close to 3 per section in FairytaleQA dataset (Table 2), we expect the $MAP@3$ is the most similar to the actual use case, and we provide four N to describe the comparison results and trends for QAG systems on the FairytaleQA.

Here is the detailed evaluation process on $MAP@N$: for each ground-truth QA-pair, we find the highest Rouge-L precision score on the concatenation of generated question and answer, among top N generated QA-pairs from the same story section. Then we average over all ground-truth QA-pairs to get the $MAP@N$ score. This evaluation metric evaluates the QAG system's performance on different candidate levels and is achievable even there is no ranking module in the system. For our QAG system, we just need to filter top N QA-pairs from our ranking module; for the `2-Step Baseline` and the `PAQ` baseline system, we simply adjust a *topN* parameter in the configuration.

5.1.3 Evaluation Results

Table 5 presents the evaluation results of our system and two SOTA baseline systems in terms of $MAP@N, N \in [1, 3, 5, 10]$. We observe our system outperforms both the `2-Step baseline` system and `PAQ` system in all settings with significantly better Rouge-L precision performance on both the validation and test splits of FairytaleQA dataset. According to the evaluation results, the `2-Step baseline` system suffers from the inherent lack of quality control of neural models over both generated answers and questions. We notice that the ranking module in our QAG system is an essential component of the system in locating the best candidate QA-pairs across different limits of candidate QA-pair amounts. The more candidate QA-pairs allowed being selected for each section, the better our system performs compared to the other two baseline systems. Still, the Rouge-L score lacks the ability to evaluate the syntactic and semantic quality of generated QA-pairs. As a result, we further conduct a human evaluation to provide qualitative interpretations.

QAG Systems	MAP@N with Rouge-L Precision on Q+A			
	$N = 10$	$N = 5$	$N = 3$	$N = 1$
Ours	**0.620** **0.596**	**0.543** **0.523**	**0.485** **0.452**	**0.340** **0.310**
`2-Step Baseline`	0.443 0.422	0.370 0.353	0.322 0.305	0.225 0.216
`PAQ`	0.504 0.485	0.436 0.424	0.387 0.378	0.288 0.273

Table 5: Results of QAG task by our system and two baseline systems. Top numbers are for validation split and bottom numbers are for test split.

5.2 Human Evaluation of QA Generation

We recruited five human participants ($N = 5$) to conduct a human evaluation to evaluate further our model generated QA quality against the ground-truth and the baseline (only against `PAQ` system as it outperforms the `2-Step Baseline`).

In each trial, participants read a storybook section and multiple candidate QA pairs for the same section: three generated by the baseline `PAQ` system, three generated by our system (top-3), and the others were the ground-truth. Participants did not know which model each QA pair was from. The participant was asked to rate the QA pairs along three dimensions using a five-point Likert-scale.

- *Readability*: The generated QA pair is in readable English grammar and words.
- *Question Relevancy*: The generated question is relevant to the storybook section.
- *Answer Relevancy*: The generated answer is relevant to the question.

We first randomly selected 7 books and further randomly selected 10 sections out of these 7 books (70 QA pairs). Each participant was asked to rate these same 70 QA pairs to establish coding consistency. The intercoder reliability score (Krippendoff's alpha (Krippendorff, 2011)) among five participants along the four dimensions are between 0.73 and 0.79, which indicates an acceptable level of consistency.

Then, we randomly selected 10 books (5 from test and 5 from validation splits), and for each book, we randomly selected 4 sections. Each section, on average, has 9 QA-pairs (3 from each model). We assigned each section randomly to two coders. In sum, each coder coded 4 books (i.e. 16 sections and roughly 140 QA-pairs), and in total 722 QA-pairs were rated.

We conducted *t-tests* to compare each model's performance. The result shows that for the *Readability* dimension, our model (avg=4.71, s.d.=0.70) performed significantly better than the PAQ model (avg=4.08, s.d.=1.13, $t(477) = 7.33, p < .01$), but was not as good as the ground-truth (avg=4.95, s.d.=0.28, $t(479) = -4.85, p < .01$).

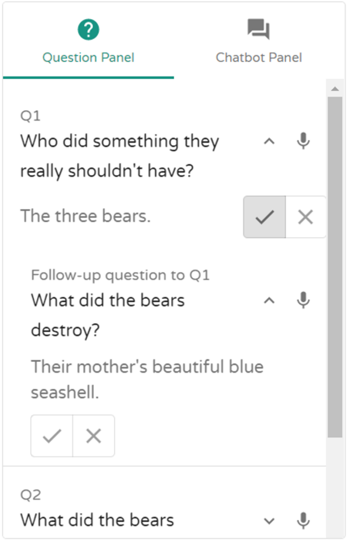
For the *Question Relevancy* dimension, ground-truth also has the best rating (avg=4.92, s.d.=0.33), which was significantly better than the other two models. Our model (avg=4.39, s.d.=1.15) comes in second and outperforms baseline (avg=4.18, s.d.=1.22, $t(477) = 1.98, p < .05$). The result suggests that questions generated by our model can generate more relevant to the story plot than those generated by the baseline model.

For the *Answer Relevancy* dimension, in which we consider how well the generated answer can answer the generated question, the ground-truth (avg=4.83,s.d.=0.57) significant outperformed two models again. Our model (avg=3.99, s.d.=1.51) outperformed PAQ baseline model (avg=3.90, s.d.=1.62, $t(477) = 0.58, p = .56$), but the result is not significant.

All results show our model has above-average (>3) ratings, which suggests it reaches an acceptable user satisfaction along all three dimensions.

5.3 Question Answer Generation in an Interactive Storytelling Application

To exemplify the real-world application of our QAG system, we developed an interactive storytelling application built upon our QAG system. This system is designed to facilitate the language and cognition development of pre-school children via interactive QA activities during a storybook reading session. For example, as children move on to a new storybook page, the back-end QAG system will generate questions for the current section. Furthermore, to optimize child engagement in the QA session, the QAG system also generates follow-up questions for each answered question. A conversational chatbot interacts with children, reads the story, facilitates questioning-and-answering via speech. The system can also keep track of child performance for the parents.



Q1

Who did something they really shouldn't have?

The three bears.

Follow-up question to Q1

What did the bears destroy?

Their mother's beautiful blue seashell.

Q2

What did the bears

Figure 3: The QA panel of our interactive storytelling application built upon our QAG system. The full user interface is shown in Appendix B.

A preliminary user study with 12 pairs of parents and children between the ages of 3-8 suggests that this application powered by our QAG system can successfully maintain engaging conversations with children about the story content. In addition, both parents and children found the system useful, enjoyable, and easy to use. Further evaluation and deployment of this interactive storytelling system are underway.

6 Conclusion and Future Work

In this work, We explore the question-answer pair generation task (QAG) in an education context for pre-school children. With a newly-constructed expert-annotated QA dataset with children-oriented fairytale storybooks, we further implement a QA generation pipeline which, as observed in human and automated evaluation, effectively supports our objective of automatically generating high-quality questions and answers at scale. To examine the model's applicability in the real world, we further build an interactive conversational storybook reading system that can surface the QAG results to children via speech-based interaction. Our work lays a solid foundation for the promising future of using AI to automate educational question answering tasks.

References

Bidyut Das, Mukta Majumder, Santanu Phadikar, and Arif Ahmed Sekh. 2021. Automatic question generation and answer assessment: a survey. *Research and Practice in Technology Enhanced Learning*, 16(1):1–15.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *arXiv preprint arXiv:1905.03197*.

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

Matthew Dunn, Levent Sagun, Mike Higgins, V Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. Searchqa: A new q&a dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179*.

Michael Heilman and Noah A Smith. 2009. Question generation via overgenerating transformations and ranking. Technical report, Carnegie-Mellon Univ Pittsburgh pa language technologies insT.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2015. The goldilocks principle: Reading children's books with explicit memory representations. *arXiv preprint arXiv:1511.02301*.

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

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.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

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.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. Paq: 65 million probably-asked questions and what you can do with them. *arXiv preprint arXiv:2102.07033*.

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.

Jack Mostow and Wei Chen. 2009. Generating instruction automatically for the reading strategy of self-questioning. In *AIED*, pages 465–472.

Xiangyang Mou, Chenghao Yang, Mo Yu, Bingsheng Yao, Xiaoxiao Guo, Saloni Potdar, and Hui Su. 2021. Narrative question answering with cutting-edge open-domain qa techniques: A comprehensive study. *arXiv preprint arXiv:2106.03826*.

Preksha Nema and Mitesh M Khapra. 2018. Towards a better metric for evaluating question generation systems. *arXiv preprint arXiv:1808.10192*.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*.

Alison H Paris and Scott G Paris. 2003. Assessing narrative comprehension in young children. *Reading Research Quarterly*, 38(1):36–76.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

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

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, Henry 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. *arXiv preprint arXiv:2010.06028*.

Lynn Snyder, Donna Caccamise, and Barbara Wise. 2005. The assessment of reading comprehension: Considerations and cautions. *Topics in Language Disorders*, 25(1):33–50.

Duyu Tang, Nan Duan, Tao Qin, Zhao Yan, and Ming Zhou. 2017. Question answering and question generation as dual tasks. *arXiv preprint arXiv:1706.02027*.

Tong Wang, Xingdi Yuan, and Adam Trischler. 2017. A joint model for question answering and question generation. *arXiv preprint arXiv:1706.01450*.

Ying Xu, Dakuo Wang, Penelope Collins, Hyelim Lee, and Mark Warschauer. 2021. Same benefits, different communication patterns: Comparing children's reading with a conversational agent vs. a human partner. *Computers & Education*, 161:104059.

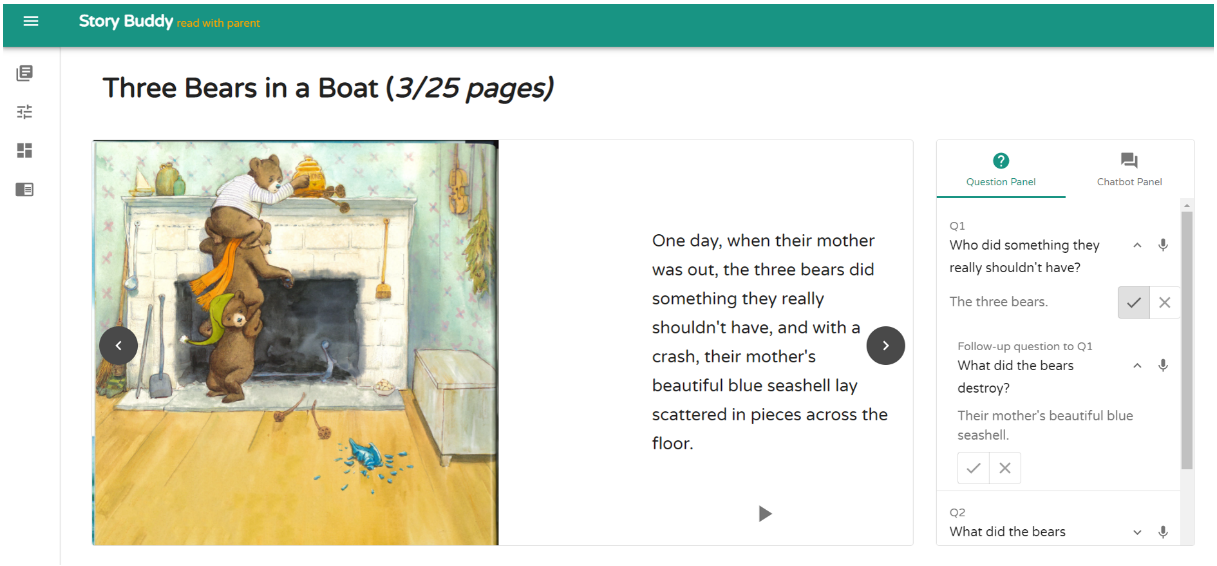
Xuchen Yao, Gosse Bouma, and Yi Zhang. 2012. Semantics-based question generation and implementation. *Dialogue & Discourse*, 3(2):11–42.

Xuchen Yao and Yi Zhang. 2010. Question generation with minimal recursion semantics. In *Proceedings of QG2010: The Third Workshop on Question Generation*, pages 68–75. Citeseer.

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

Category	Train		Validation		Test	
	Count	Percentage	Count	Percentage	Count	Percentage
Character	962	0.112	107	0.104	103	0.102
Causal Relationship	2368	0.277	294	0.286	278	0.276
Action	2694	0.315	333	0.324	315	0.312
Setting	523	0.061	45	0.043	62	0.061
Feeling	824	0.096	94	0.091	106	0.105
Prediction	366	0.0428	55	0.053	65	0.064
Outcome Resolution	811	0.094	97	0.094	78	0.077

Table 6: The number of QA-pairs belongs to each of the seven narrative element categories in the FairytaleQA dataset, inspired by (Paris and Paris, 2003).



Three Bears in a Boat (*3/25 pages*)

One day, when their mother was out, the three bears did something they really shouldn't have, and with a crash, their mother's beautiful blue seashell lay scattered in pieces across the floor.

Q1
Who did something they really shouldn't have?

The three bears.

Follow-up question to Q1
What did the bears destroy?

Their mother's beautiful blue seashell.

Q2
What did the bears

Figure 4: The user interface of our down-streaming interactive storytelling system.

A Distribution of FairytaleQA annotations on 7 narrative elements

Table 6 shows the distribution of QA-pair annotations on 7 essential narrative elements that are defined in (Paris and Paris, 2003) of FairytaleQA dataset. The distribution of narrative elements is consistent across train/validation/test splits.

B User Interface of down-streaming application

Figure 4 is a screenshot of the interactive storytelling system interface for the down-streaming task of our QAG system in a real-world use scenario. Children can listen to the automatic story reading and try to answer the plot-relevant questions generated by the QAG system. They can answer the question via a microphone, and the system will judge the correctness of their answer. After answering a 'parent' question, children can go further to answer a follow-up question or try out other 'parent' questions.