StoryQA: Story Grounded Question Answering Dataset

Sanqiang Zhao, Seokhwan Kim, Yang Liu, Robinson Piramuthu, Dilek Hakkani-Tur

{sanqiang, seokhwk, yangliud, robinpir, hakkanit}@amazon.com
Amazon Alexa AI

Abstract

The abundance of benchmark datasets supports the recent trend of increased attention given to Question Answering (QA) tasks. However, most of them lack a diverse selection of QA types and more challenging questions. In this work, we present StoryQA, a new task and dataset addressing diverse QA problems for both in-context and out-of-context questions. Additionally, we developed QA models based on large pretrained language models. Our experiments on the new dataset show that performance of our developed model is comparable to that by humans. The resources in this work will be released to foster future research.

Introduction

Recent years have seen a lot of attention paid to QA systems. This trend is further supported by an abundance of benchmark datasets that are specifically designed to encourage research in this field. As Khashabi et al. (2020) summarized, current QA datasets can be categorized into four common types: Extractive QA (Rajpurkar et al. 2016; Kočiskỳ et al. 2018), Abstractive QA (Kočiskỳ et al. 2018; Nguyen et al. 2016), Yes/No QA (Clark et al. 2019) and Multiple-Choice QA (Lai et al. 2017). In this paper, we address the first three QA problems since they occur more frequently in real world use cases such as human conversations to ask and answer questions. In Extractive QA, the answer is always a span in the given document context; in Yes/No QA, the answer is always either "yes" or "no"; and in Abstractive QA the response is based on a given context but not restricted to the exact substrings of the given context.

The majority of existing datasets were collected specifically for a single research problem, therefore most of them only contain a single QA type. In addition, their data collection approach limits the scope to in-context questions only, and thus the datasets do not contain any out-of-context questions, which occur in realistic QA use cases.

To address these weaknesses, we introduce a new dataset called StoryQA that includes multiple types of QAs on the *same* context, including Extractive QA, Yes/No QA and Abstractive QA. Our work addresses also the out-of-context questions that are still related to the context, but not directly answerable just by the given context. FairytaleQA (Xu et al. 2022) is probably the most relevant dataset for us, with a focus on stories. We expect models trained on FairytaleQA to be able to address out-of-context questions, however, the answer remains unreasonable. Note that SQuAD2.0 (Rajpurkar, Jia, and Liang 2018) contains out-of-context unanswerable questions, but their goal is to just identify and filter out those, rather than answering them. During the creation of our dataset, we observed that many of the out-of-context questions, especially those asking for non-factual information in a fictional story, can still be answered by humans. One example is shown in Table 1 where the question is about what was in the boy's mind. Although the story does not have an explicit answer for this, humans can still provide a reasonable answer after reading the story. Most existing models are unable to answer such questions reasonably, due to the in-context limitation of the training datasets.

We summarize our contributions as follows:

1) We publish a new dataset called StoryQA that contains multiple types of in-context and out-of-context questions. It is collected based on Aesop's Fables[1], because we found that compared with questions in non-fictional contexts such as Wikipedia or news articles, fictional stories are better to collect more diverse questions. This dataset aims to tackle the following three QA problems: Extractive QA, Yes/No QA, and Abstractive QA. Among them, Abstractive QA is the most challenging problem with out-of-context questions that most existing models cannot answer properly.

2) We propose a unified QA model that handles all three QA types and demonstrate via both automatic and human evaluation that it performs consistently better than the fine-tuned models on just a single QA type. The results also show that our unified model achieves comparable performances to the human references.

Related Work

Most existing datasets were collected by asking crowd workers to provide questions and answers following specific guidelines designed for a particular research problem. As representatives of Extractive QA datasets, SQuAD 1.1 (Rajpurkar et al. 2016), SQuAD 2.0 (Rajpurkar, Jia, and Liang 2018) and NaturalQuestion (Kwiatkowski et al. 2019) were

[1] http://read.gov/aesop/index.html

Story (given context):
A Boy was given permission to put his hand into a pitcher to get some filberts. But he took such a great fistful that he could not draw his hand out again. There he stood, unwilling to give up a single filbert and yet unable to get them all out at once. Vexed and disappointed he began to cry. "My boy," said his mother, "be satisfied with half the nuts you have taken and you will easily get your hand out. Then perhaps you may have some more filberts some other time."

Question: Why was the Boy so greedy?

Answer:	
Human	The boy was greedy because he really liked filberts
SQuAD2.0	he took such a great fistful
NaturalQuestion	unwilling to give up a single filbert
DROP	vexed and disappointed he began to cry
UnifiedQA	he was greedy
FairytaleQA	He was greedy.
FairytaleQA + UnifiedQA	He was a bad person.
`StoryQA` (ours)	The boy was greedy because he wanted to get as many nuts as possible.

Table 1: Sample responses from models trained on various datasets (left column) for an out-of-context question for the popular fable "The Boy and the Filberts". FairytaleQA + UnifiedQA indicates that we finetuned UnifiedQA-11B using FairytaleQA dataset.

collected by asking each worker to write down a pair of question and answer together. Since every answer is always restricted to a span in the context, the datasets contain the in-context questions only.

This limitation also applies to the Multiple-choice[2] and Yes/No QA datasets. For the Multiple-choice QAs, workers need to provide a list of answer candidates including a correct answer and other distractors. Since each correct answer needs to be explicitly validated by the given context, the collected data covers in-context questions only. In addition, workers are asked to provide both questions and answers, as in the RACE (Lai et al. 2017) data collection. As a result, workers are likely to provide questions that are easy to identify the correct answers. Similarly, the Yes/No QA datasets (such as BoolQ (Clark et al. 2019)) include the in-context questions that can be clearly answered by either "yes" or "no" based on a given context.

Abstractive QA datasets place less constraint, but still have narrow scopes due to the specific data collection requirements to guide the collected data towards particular research problems. For example, as part of the NarrativeQA (Kočiskỳ et al. 2018) data collection process, workers are instructed to avoid copying from the context, but provide specific and diverse QA pairs. In DROP (Dua et al. 2019), workers are encouraged to provide questions that need to be answered through discrete reasoning. MS MARCO (Nguyen et al. 2016) is based on search logs. As mentioned in Kočiskỳ et al. (2018), many answers are in fact verbatim copies of short spans from the context.

To the best of our knowledge, SQuAD 2.0 is the only dataset including out-of-context questions. However, SQuAD 2.0 aims to filter out out-of-context questions rather than providing the answer to them. We suppose that the lack of out-of-context QA data is because (1) most existing datasets require workers to provide paired questions and answers together, thus discouraging them from asking out-of-context questions, and (2) workers are instructed to ask questions mainly to test the reading comprehension skills rather than pretending to be inquisitive about the given context. We found that out-of-context QAs occurred more frequently in fictional stories. Fairytale QA focuses on similar stories as ours, but it also lacks out-of-context QAs. To collect more realistic and challenging QAs, our data was collected in an alternative way where each question and its answer were collected by different workers from each other. We believe this results in more diverse data, because the workers can provide the questions with no consideration about how to answer them at the same time.

[2]Multiple-choice QA is not covered in this work.

`StoryQA` Dataset

In this section, we introduce our new dataset, `StoryQA`, which addresses many of the above-mentioned limitations of the existing QA datasets.

Desiderata

From the limitations of other datasets discussed above, we define our desiderata as follows. First, we construct a dataset containing a large number of QA pairs collected by two groups of crowd workers for questions and answers separately, where the questioners can ask diverse questions regardless of whether and how they can be answered. Second, we set as few restrictions as possible to make the collected data plausible in real-world use cases. We took fictional children stories from Aesop's Fables and asked crowd workers to pretend they were 5-8 years old, which aims to collect more flexible and creative questions. These are expected to be more challenging and beneficial for the QA research.

Data Collection Method

We collected three subsets, each of which addresses Extractive, Yes/No, and Abstractive QA types.

Extractive QA Subset: Here every answer must be a span in a given story context. We first automatically generated the answer candidates from each story in Aesop's Fables. We revised the Extractive QA model (see description later), where the story context and questions are fed into a base model and two pointers are learned to locate a single answer, and only the context and question is fed in order to locate multiple spans for answer candidates. We used AlBERT-xxLarge (Lan et al. 2019) as the base model and trained it on the SQuAD 2.0 dataset. For each story-answer pair, we asked crowd workers to provide a question that can be answered by the span.

Yes/No QA Subset: The generated answer spans above were used also for collecting Yes/No QA subset. Here we provided each span as the additional information to guide annotators with their questions. Similar to Extractive QA, the crowd workers were shown the full story with highlighted span and asked to submit a Yes/No question for the given Yes/No answer.

Subset	#QAs	Priming	Data	
			Q	A
Extractive QA	12,148	story span	✓	
Yes/No QA	11,779	story span	✓	✓
Abstractive QA	14,776	1. none	✓	
		2. Q from step 1		✓

Table 2: **Collection of `StoryQA` subsets.** For Extractive QA and Yes/No QA datasets, crowd workers were shown a story span extracted by an Extractive QA model along with the full story. Abstractive QA dataset was collected in 2 steps where a free-form question was collected in step 1 from a given story, and later showed to an independent set of workers to get their answers.

Abstractive QA Subset: Different from the first two categories, the abstract QAs were collected by two crowdsourcing tasks, first collecting questions, followed by obtaining the answers. To collect diverse questions, we asked the crowd workers to provide free-form questions. We only require that the questions should be relevant to the given story context. Then, we had a subsequent task to collect the answer for each question. To categorize the answer sources for the questions, we first asked the crowd workers to specify whether it can be answered only with the given story context or requires any external knowledge beyond the story content. Then for the question they provided the answer in their own words. Such an answer is grounded on either the story context or their background knowledge.

To ensure the answer quality, a pilot task was conducted first with a small amount of data followed by a manual evaluation task. Then, the full data collection was done only with the highly-scored workers in the pilot task. Table 2 shows the statistics of the collected data. These are collected using 148 Aesop's Fables as the story context.

Analysis of Abstractive QA Subset

We analyze the Abstractive QA subset in more detail since it differs from most existing datasets and introduces new research challenges.

Question Format: Table 3 shows a breakdown of the Abstractive QA subset by question format. We observed that 39.27% of the questions in the Abstractive QA subset can be answered by *Yes/No*, which is the most common category followed by *What* and *Why* questions. In addition, we notice that some questions belong to multiple question formats, which introduces more challenges to QA models.

Knowledge Source for Answer: As we mentioned earlier, there are many out-of-context questions in the Abstractive QA subset and thus it is important to understand the properties of such questions and how to develop models to answer them. During data collection, we explicitly asked crowd workers to identify the category of answer sources. As shown in Table 4, only 43.38% of the questions have the explicit answers within the story content. Among the rest out-of-context questions, only a small percentage (14.49%) requires external factual knowledge, while the other 42.13% of the questions can be answered by common sense.

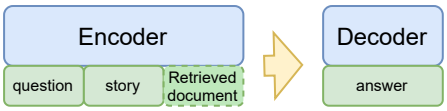

Figure 1: Model Architecture for Abstractive QA. Question and story are separated by "\n". Out-of-context questions are handled by concatenating the most relevant retrieved external content as shown in dashed box.

Comparison with Other Datasets Table 5 shows a comparison of `StoryQA` to relevant existing datasets. The only dataset that contains out-of-context questions is SQuAD2.0, but their task is only to filter them out, while our dataset includes more challenging out-of-context questions and we also provide their ground truth answers. Furthermore, our dataset contains multiple QA types. FairytaleQA is the most similar dataset to StoryQA, since it also focuses on stories. But it includes a smaller number of questions than ours and does not address out-of-context (OOC) questions, which is a focus in our work.

Model Development

We present baseline models for each QA type as well as a unified model to address all QA types.

Model for Abstractive QA

As discussed earlier, Abstractive QA poses the most challenges due to its diversity and hence we elaborate more.

Analysis of Existing Models Table 1 shows a typical out-of-context question in `StoryQA`. Although ExtractiveQA models fail for these questions, we observed in general that the UnifiedQA model (Khashabi et al. 2020) can generate the most reasonable answers.[3] We will initially focus on fine tuning pretrained language models and adapting the knowledge in these large language models to generate reasonable answers on `StoryQA`.

Model Architecture In this section, we show our model architecture for Abstractive QA. We followed the UnifiedQA architecture and employed Transformer-based Encoder-Decoder framework (Vaswani et al. 2017). As in Figure 1, we concatenate the question and story into a single packed sequence. These are separated by the new line character "\n" and fed into Transformer Encoder to obtain the hidden states $\mathbf{T}_{enc}$. Khashabi et al. (2020) explains how this ensures a human-like encoding while not making it overly-specific to a certain format. The Transformer Decoder models the probabilities of each word w_i in the answer as $p(w_i|w_{i-1}, w_{i-2}....\mathbf{T}_{enc})$ in an auto-regressive manner. The sum of log-likelihoods of w_i is used as the training objective.

Handling Out-of-Context Questions Considering that our dataset includes many out-of-context questions that require external knowledge sources, we attempt to retrieve

[3]We tried several models trained on popular datasets and pretrained base models, including SQuAD 2.0 (AlBERT (Lan et al. 2019)), Natural Question (RoBERTa (Liu et al. 2019)), DROP (BERT (Devlin et al. 2018), RoBERTa).

Format	%Samples	Examples	
Yes/No	39.27%	So their dad kind of tricked them, huh?	Did the poor miller gain anything?
What	20.34%	What does a jackdaw look like?	What is a hare? is that a rabbit?
Why	23.07%	Why were they so mean to the stag?	Why didn't the lamb try to get away?
How	8.19%	How did the fox get caught in the trap?	How did the mice answer?
Where	2.94%	Where were the travelers from?	Where did the fox first see the lion?
Who	4.11%	Who is going to bell the Cat?	But who took that gold?
When	2.08%	When does owl sleep?	When did he run away? that is so sad

Table 3: **Breakdown by Question Format:** Questions are statistically analyzed by their "question" word. Questions such as "Ohh, why did the fox hurry? what were they sick with?", "Who is the shepherd and why would he care?" will be counted twice, once for *Why*-based questions and once for *What*-based questions.

Knowledge Source	%Samples	Description	Examples
In-context	43.38%	Can be answered within story content only.	Why did the Camel envy the Monkey?
Common Sense	42.13%	Not related to any facts but can provide an answer based on commonsense.	Do you think it was mean for the other animals to kill and eat the Camel for being foolish?
Factual Knowledge	14.49%	Need to look up external sources to find relevant facts.	How much does a Camel weigh?

Table 4: **Breakdown by Knowledge Source.** Only questions grouped under "In-context" can be answered just from the given story context; the rest "out-of-context" questions require factual or commonsense knowledge outside the story contents.

Dataset	# QAs	QA Type			OOC
		EX	YN	AB	
SQuAD2.0	∼142.2k	✓			33.38% no ans.
FairytaleQA	∼10.5k			✓	-
StoryQA (ours)	∼36k	✓	✓	✓	56.62%

Table 5: **Dataset Comparison:** EX = Extractive, YN = Yes/No, AB = Abstractive, OOC = Out-of-Context. StoryQA contains diverse QA types and more challenging out-of-context questions. For *out-of-context* questions, SQuAD2.0 only needs to detect without answering them, while StoryQA provides all answers.

additional relevant contexts and incorporate them into answering the questions. We investigate two retrieval methods widely used in QA research communities, namely DPR (Karpukhin et al. 2020) on Wikipedia passages and ColBERT (Khattab and Zaharia 2020; Lin, Yang, and Lin 2021) on MARCO (web pages). Both models are trained by minimizing the distance between the question and the relevant document in an information retrieval fashion. We used ColBERT and DPR to retrieve Wikipedia and web pages respectively, and appended the retrieved document to the end of the story context, again using the new line character "\n" as a separator (Figure 1).

Model for Extractive QA

Extractive QA requires the answer to be a span in the story and has been widely studied in the QA research community. We followed the standard procedure to extract answers, where a pretrained language model is used as the basis to predict the start and end positions of the span for the answer in the given context. Specifically, we concatenate the input question and story, and fed them into a pretrained language model to obtain hidden states $\mathbf{T}$. The probability of word w_i being the start of the answer span is computed as the dot product between $\mathbf{T}_i$ and $\mathbf{T}_s$ where s is the start position of the answer. The same thing is done for the end of the answer span. The training objective is the sum of the log-likelihoods of the correct start and end positions.

Model for Yes/No QA

All of the answers in this subset are either "yes" or "no", and can be answered from the given context. Due to limited resources in Yes/No QA research, we followed the UnifiedQA model (Khashabi et al. (2020)) as described for Abstractive QA, but with the constrained decoding to generate tokens of either "yes" or "no".

Our Unified StoryQA Model

In addition to above the QA problem-specific models, we propose a unified StoryQA model for all the question types. Inspired by the prefix constraint idea (Takeno, Nagata, and Yamamoto 2017; Liu, Luo, and Zhu 2018; Zhao et al. 2019; Martin et al. 2019), we add the question-type prefix before the question, as shown in Figure 2. Specifically, we fine-tuned the UnifiedQA model on the entire StoryQA dataset containing all three QA types (Abstractive, Extractive, Yes/No) by using prefix tokens "abstractive", "extractive" and "yesno" respectively. With such a design, we hope that different QA datasets will complement each other and improve the performances across all the sub-

Split	Description	Extractive	Yes/No	Abstractive
Training	Training dataset with QAs sampled from 128 stories	8,397	8,652	10,772
Dev-seen	Sample 1000 QAs share the same stories in training set	1,000	1,000	1,000
Dev-unseen	Sample 10 stories not in Training or Test sets	797	552	995
Test-seen	Sample 1000 QAs share the same stories in training set	1,000	1,000	1,000
Test-unseen	Sample 10 stories not in Training or Dev sets	954	575	999

Table 6: **Data Splits.** `StoryQA` contains five splits. Both Dev and Test splits contain seen and unseen story versions, which indicate whether the splits share the same stories with training or not, respectively.

Model Config	#Params	Test-seen				Test-unseen			
		BLEU	Rouge1	Rouge2	RougeL	BLEU	Rouge1	Rouge2	RougeL
UnifiedQA-**BART-Large**	406M	0.10	11.30	2.05	11.13	0.07	12.49	2.17	12.24
UnifiedQA-**T5-Base**	220M	1.05	15.07	5.05	14.20	1.09	14.28	4.58	13.25
UnifiedQA-**T5-Large**	770M	1.07	15.56	5.33	14.57	1.01	15.26	4.62	14.28
UnifiedQA-**T5-3B**	3B	1.15	17.02	5.87	16.10	0.96	15.65	5.00	14.88
UnifiedQA-**T5-11B**	11B	3.81	24.02	9.66	22.10	3.29	19.69	7.02	18.25
UnifiedQA-**BART-Large**-FT	406M	10.64	33.59	17.73	31.44	8.97	31.01	15.23	28.71
UnifiedQA-**T5-Base**-FT	220M	10.29	34.89	17.66	32.35	8.84	32.98	15.79	30.22
UnifiedQA-**T5-Large**-FT	770M	10.94	35.20	18.42	32.93	9.29	33.58	16.32	30.99
UnifiedQA-**T5-3B**-FT	3B	11.19	36.40	19.05	33.59	9.71	34.49	16.77	31.68
UnifiedQA-**T5-11B**-FT	11B	**11.38**	**36.81**	**19.45**	**34.22**	**10.33**	**35.20**	**17.69**	**32.42**

Table 7: **Abstractive QA subset**: "-FT" indicates fine-tuned version on the *Abstractive QA subset*. Base models are shown in bold in column 1. Best values are shown bold faced.

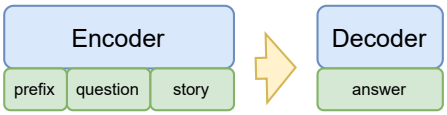

Figure 2: Model Architecture for our Single Unified `StoryQA` Model. Prefix identifies QA type.

sets.

Experiments and Result Analysis

Experiment Setup

Our data contains five splits as shown in Table 6. For all of our experiments, we picked the best models based on the merged Dev split (Dev-seen + Dev-unseen) and reported the performance separately for Test-seen and Test-unseen. All models were trained on an 8 A100 GPU machine with LAMB optimizer (Khashabi et al. 2020) and learning rate warm-up technique. For larger models, such as T5-3B and T5-11B, we used ZeRO (Ren et al. 2021; Rajbhandari et al. 2021) to train our model.

Experiments on Abstractive QA Subset

In Abstractive QA, we compare transformer-based encoder-decoder frameworks, with a particular focus on different size of T5 (Raffel et al. 2019) and BART models (Lewis et al. 2019) as well as their fine-tuned versions, all based on the UnifiedQA model (Khashabi et al. 2020). In this section, we discuss the automatic evaluation results with the reference-based metrics including BLEU (Papineni et al. 2002) and Rouge (Lin 2004). Human evaluation results will be presented later.

Effect of Model Size and Fine-tuning: Table 7 shows performance of UnifiedQA on the Abstractive QA subset, with base models of different sizes. Note that T5-base performs significantly better than BART-Large (about twice the size) in all the metrics while it improves as we increase the number of parameters up to 11B. This is due to the increased distilled common sense and general knowledge as we use the larger base models. In addition, fine-tuning on our Abstractive QA subset consistently yields significant improvements.

Stratifed Performance Analysis: Tables 8 and 9 show how "UnifiedQA-T5-11B-FT" (the best model from Table 7) performs across the breakdowns presented in Tables 3 and 4. We can see that the model performed worse for the *How* and *Why* questions, as expected, due to high diversity in free-form answers for such question types. Surprisingly, the model achieved the worst performances for the when-based questions. We speculate that this is caused by the difficulties in answering the questions about any out-of-context temporal events. Table 9 shows that it was much harder for the model to answer the questions that require common sense knowledge compared to the other knowledge sources. On the other hand, the results on the questions that require external factual knowledge were relatively good due to the distilled knowledge from the pretrained language model.

Handling Out-of-Context Questions: As mentioned earlier, we hypothesize that the additional context retrieved from external knowledge helps to improve the model performances for the out-of-context questions. Table 10 compares the performance of "UnifiedQA-T5-11B-FT" when augmented with web pages (MS MARCO dataset) or Wikipedia passages (Wang et al. 2019), both of which show small im-

Question Format	Test-seen				Test-unseen			
(% samples)	BLEU	Rouge1	Rouge2	RougeL	BLEU	Rouge1	Rouge2	RougeL
Yes/No (39.3%)	9.61	34.96	17.94	32.31	8.38	31.52	15.12	29.22
What (20.3%)	15.34	40.76	23.43	38.31	15.66	42.26	24.18	39.14
Why (23.07%)	8.41	33.38	15.2	29.93	8.03	31.5	13.86	28.53
How (8.2%)	9.20	30.3	13.39	28.3	6.15	30.29	11.83	27.39
Where (2.9%)	21.59	50.35	32.24	48.39	14.82	45.53	25.81	44.61
Who (4.1%)	22.39	53.75	34.95	50.96	15.10	46.74	27.28	43.52
When (2.1%)	7.92	26.69	12.55	23.92	4.20	22.35	9.01	21.17

Table 8: Performance of UnifiedQA-T5-11B-FT on *Abstractive QA subset* based on the breakdown as in Table 3.

Knowledge Source	Test-seen				Test-unseen			
	BLEU	Rouge1	Rouge2	RougeL	BLEU	Rouge1	Rouge2	RougeL
In-context	15.31	42.45	24.08	39.53	14.26	42.45	24.36	39.43
Common Sense	6.76	29.48	12.94	26.87	6.62	27.59	11.47	25.26
Factual Knowledge	11.97	36.96	20.16	35.11	10.80	38.01	17.73	34.92

Table 9: Performance of UnifiedQA-T5-11B-FT on *Abstractive QA subset* based on the breakdown as in Table 4.

Model Config	Test-seen				Test-unseen			
	BLEU	Rouge1	Rouge2	RougeL	BLEU	Rouge1	Rouge2	RougeL
UnifiedQA-T5-11B-FT	11.38	36.81	19.45	34.22	10.33	35.20	17.69	32.42
UnifiedQA-T5-11B-FT + MARCO	11.59	37.27	20.11	**34.87**	10.51	34.96	17.63	32.13
UnifiedQA-T5-11B-FT + Wiki	**11.83**	**37.32**	**20.24**	34.77	**10.80**	**35.79**	**18.29**	**32.97**

Table 10: Performance of UnifiedQA-T5-11B-FT (the best model from Table 7) on *Abstractive QA subset*, when augmented with relevant retrieved web pages from MS Marco ("+MARCO") or Wikipedia passages ("+Wiki"). Best values are shown bold faced.

Model Config	#Params	Test-seen		Test-unseen	
		EM	F1	EM	F1
AlBERT-Base	12M	51.1	69.20	53.46	72.15
AlBERT-Large	18M	54.3	73.17	53.77	74.47
AlBERT-xLarge	60M	56.5	75.14	54.4	76.46
AlBERT-xxLarge	235M	57.6	76.05	55.56	76.10
DeBERTa-Base	139M	54.3	72.79	52.73	73.48
DeBERTa-Large	405M	56.7	75.80	54.4	76.52
AlBERT-Base-FT	12M	51.8	71.88	52.73	72.08
AlBERT-Large-FT	18M	54.4	73.63	55.35	75.38
AlBERT-xLarge-FT	60M	58.0	75.94	57.13	77.64
AlBERT-xxLarge-FT	235M	55.7	74.89	57.23	77.71
DeBERTa-Base-FT	139M	58.4	76.15	56.60	75.69
DeBERTa-Large-FT	405M	**59.5**	**78.79**	**58.39**	**78.95**

Table 11: **Extractive QA subset**: "-FT" indicates fine-tuned version on the *Extractive QA subset*. EM = Exact Match. Best values are shown bold faced. All models were pre-trained on SQuAD2.0.

provements by incorporating external knowledge.

Experiments on Extractive QA Subset

For the extractive QAs, we compared the performances of SQuAD (Rajpurkar et al. 2016) model variations with AlBERT (Lan et al. 2019) and DeBERTa (He et al. 2020) as base models, and also when fine-tuning them on our Extractive QA dataset. Following the SQuAD evaluation set-ups, we used the Exact Match and F1 as the evalution metrics. Table 11 shows that the larger models perform the better in general (except AlBERT-xxLarge-FT); and the fine-tuning helps to improve the performances significantly, especially when using DeBERTa.

Model Config	Accuracy	
	Test-seen	Test-unseen
UnifiedQA-T5-Base	68.6	70.61
UnifiedQA-T5-Large	77.8	76.87
UnifiedQA-T5-3B	86.1	85.57
UnifiedQA-T5-11B	54.6	53.57
UnifiedQA-T5-Base-FT	88.0	86.78
UnifiedQA-T5-Large-FT	85.6	86.26
UnifiedQA-T5-3B-FT	91.1	90.43
UnifiedQA-T5-11B-FT	**92.4**	**91.13**

Table 12: **Yes/No QA subset**: "-FT" indicates fine-tuned version on the *Yes/No QA subset*. Best values are shown bold faced.

Experiments on Yes/No QA Subset

For the Yes/No QAs, we experiment with the Unified QA variations by changing the base models. All of these models were fine-tuned on our Yes/No QA subset and evaluated on accuracy for the binary predictions as in (Clark et al. 2019). Table 12 indicates that UnifiedQA models do not perform well, but when fine-tuned they improve significantly. This may be due to the limited amount of Yes/No QA datasets in training the UnifiedQA models and our dataset greatly expands such resources.

Model Config	Test-seen				Test-unseen			
	BLEU	Rouge1	Rouge2	RougeL	BLEU	Rouge1	Rouge2	RougeL
UnifiedQA-T5-11B-FT	11.38	36.81	19.45	34.22	10.33	35.20	17.69	32.42
Unified StoryQA (ours)	**11.95**	**37.86**	**20.71**	**35.30**	**10.88**	**35.86**	**18.60**	**33.27**

Table 13: Abstractive QA Model Performance: "-FT" indicates fine-tuned version. Unified StoryQA (ours) indicates fine-tuned unified model.

Model Config	Test-seen		Test-unseen	
	EM	F1	EM	F1
DeBERTa-Large-FT	59.5	78.79	58.39	78.95
Unified StoryQA (ours)	**59.7**	**79.77**	**60.27**	**82.67**

Table 14: Extractive QA Model Performance: "-FT" indicates fine-tuned version. Unified StoryQA (ours) indicates the single unified model. EM = Exact Match.

Model Config	Accuracy	
	Test-seen	Test-unseen
UnifiedQA-T5-11B-FT	**92.4**	91.13
Unified StoryQA (ours)	92.1	**92.35**

Table 15: Yes/No QA Model Performance: "-FT" indicates fine-tuned version. Unified StoryQA (ours) indicates the single unified model.

Our Unified StoryQA Model

As described earlier, our Unified StoryQA model is based on the UnifiedQA model with adaptations as in Figure 2. Our model is based on the best configuration on Abstractive QA subset, namely "UnifiedQA-T5-11B" (see Table 7), and fine-tuned on the entire StoryQA dataset (not one subset). We call this single model as "Unified StoryQA Model". We compare this single model against the best models we presented for each of the three subsets. Note again that these competing models were fine-tuned on only the relevant subsets and not the entire StoryQA dataset. Comparisons are shown in Tables 13, 14 and 15 for Abstractive QA, Extractive QA and Yes/No QA subsets of StoryQA dataset, respectively. Our single *Unified StoryQA Model* achieves the best performance for all three subsets (except the Yes/No QA subset on Test-seen where it is still close), including the challenging Abstractive QA subset that has out-of-context questions. This also shows that the different subsets specialized in different QA types complement each other and can further improve performance when we combine them together.

Human Evaluation on Abstractive QA Subset

Since automatic metrics are known to be limited in capturing comprehensive model performances beyond overlaps with the references, we conducted human evaluation to further analyze the models. We used *all* questions in the Test dataset and shuffled the predicted answers and ground truth to determine the qualitative gap between model predictions and human provided answers. The crowd workers from Amazon Mechanical Turk were asked to read a story and a question, and then rate each answer on a 5-point scale (1-5, where 5 is the best) for appropriateness.

Table 16 compares the average ratings for each *fine-tuned* model. Results are consistent with earlier findings from automatic evaluations, indicating that larger models have superior performance. We can also see that our best model performs very closely to ground truth answers. Note that answers are rated for how accurate they are for the given question, rather than how natural they are. The average number of whitespace-delimited tokens per answer from UnifiedQA-T5-11B, UnifiedQA-T5-11B-FT, Unified StoryQA and Ground Truth (Human) is 4.30, 10.28, 10.08 and 12.16 respectively. For reference, it is 4.19 for the NarrativeQA dataset. Therefore answers produced by models fine-tuned on our dataset seem to be more expressive.

We also conducted a human evaluation study to analyze the effect of Retrieving Relevant Context. We followed a similar setup as above, but sampled 300 questions from the test dataset and shuffled the model predictions of all models in Table 10 for evaluation. Table 17 shows consistent results.

Model Config	Test-seen	Test-unseen
UnifiedQA-**BART-Large**-FT	3.30	3.07
UnifiedQA-**T5-Base**-FT	3.26	2.85
UnifiedQA-**T5-Large**-FT	3.52	3.15
UnifiedQA-**T5-3B**-FT	3.77	3.55
UnifiedQA-**T5-11B**-FT	3.99	3.81
Unified StoryQA (ours)	**4.02**	3.82
Ground Truth (human)	**4.02**	**3.95**

Table 16: Human Evaluation For Abstractive QA Model Performance based on a 5-point scale (1-5, where 5 is the best) for appropriateness of the answer.

Model Config	Test-seen	Test-unseen
UnifiedQA-T5-11B-FT	3.97	3.91
UnifiedQA-T5-11B-FT + MARCO	4.03	**3.99**
UnifiedQA-T5-11B-FT + Wiki	**4.12**	3.88

Table 17: Human Evaluation For Handling Out-of-Context Questions based on a 5-point scale (1-5, where 5 is the best) for appropriateness of the answer.

Conclusion

We introduced a new task and dataset, named StoryQA. Our dataset covers three types of QA problems: Extractive QA, Yes/No QA and Abstractive QA. In addition, it includes many challenging questions, especially those that are out-of-context. We conducted extensive experiments showing insights related to the size of the models, fine-tuning, sources of knowledge and types of questions. We also proposed a Unified StoryQA Model and showed it performs

better than the equivalent models fine-tuned on a single specific subset. We hope that our proposed `StoryQA` dataset, baseline models and experimental findings will inspire moving towards a QA system that addresses more open-ended and diverse questions. More contextual QA across multiple turns is also a natural future extension from the current settings.

References

Clark, C.; Lee, K.; Chang, M.-W.; Kwiatkowski, T.; Collins, M.; and Toutanova, K. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dua, D.; Wang, Y.; Dasigi, P.; Stanovsky, G.; Singh, S.; and Gardner, M. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. *arXiv preprint arXiv:1903.00161*.

He, P.; Liu, X.; Gao, J.; and Chen, W. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Karpukhin, V.; Oğuz, B.; Min, S.; Lewis, P.; Wu, L.; Edunov, S.; Chen, D.; and Yih, W.-t. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Khashabi, D.; Min, S.; Khot, T.; Sabharwal, A.; Tafjord, O.; Clark, P.; and Hajishirzi, H. 2020. Unifiedqa: Crossing format boundaries with a single qa system. *arXiv preprint arXiv:2005.00700*.

Khattab, O.; and Zaharia, M. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, 39–48.

Kočiskỳ, T.; Schwarz, J.; Blunsom, P.; Dyer, C.; Hermann, K. M.; Melis, G.; and Grefenstette, E. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6: 317–328.

Kwiatkowski, T.; Palomaki, J.; Redfield, O.; Collins, M.; Parikh, A.; Alberti, C.; Epstein, D.; Polosukhin, I.; Devlin, J.; Lee, K.; et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7: 453–466.

Lai, G.; Xie, Q.; Liu, H.; Yang, Y.; and Hovy, E. 2017. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*.

Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; and Soricut, R. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, 74–81.

Lin, S.-C.; Yang, J.-H.; and Lin, J. 2021. Contextualized Query Embeddings for Conversational Search. *arXiv preprint arXiv:2104.08707*.

Liu, Y.; Luo, Z.; and Zhu, K. 2018. Controlling length in abstractive summarization using a convolutional neural network. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 4110–4119.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Martin, L.; Sagot, B.; de la Clergerie, E.; and Bordes, A. 2019. Controllable sentence simplification. *arXiv preprint arXiv:1910.02677*.

Nguyen, T.; Rosenberg, M.; Song, X.; Gao, J.; Tiwary, S.; Majumder, R.; and Deng, L. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311–318.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Rajbhandari, S.; Ruwase, O.; Rasley, J.; Smith, S.; and He, Y. 2021. ZeRO-Infinity: Breaking the GPU Memory Wall for Extreme Scale Deep Learning. *arXiv preprint arXiv:2104.07857*.

Rajpurkar, P.; Jia, R.; and Liang, P. 2018. Know what you don't know: Unanswerable questions for SQuAD. *arXiv preprint arXiv:1806.03822*.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Ren, J.; Rajbhandari, S.; Aminabadi, R. Y.; Ruwase, O.; Yang, S.; Zhang, M.; Li, D.; and He, Y. 2021. Zero-offload: Democratizing billion-scale model training. *arXiv preprint arXiv:2101.06840*.

Takeno, S.; Nagata, M.; and Yamamoto, K. 2017. Controlling target features in neural machine translation via prefix constraints. In *Proceedings of the 4th Workshop on Asian Translation (WAT2017)*, 55–63.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wang, Z.; Ng, P.; Ma, X.; Nallapati, R.; and Xiang, B. 2019. Multi-passage bert: A globally normalized bert model for open-domain question answering. *arXiv preprint arXiv:1908.08167*.

Xu, Y.; Wang, D.; Yu, M.; Ritchie, D.; Yao, B.; Wu, T.; Zhang, Z.; Li, T.; Bradford, N.; Sun, B.; et al. 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)*, 447–460.

Zhao, S.; Sharma, P.; Levinboim, T.; and Soricut, R. 2019. Informative image captioning with external sources of information. *arXiv preprint arXiv:1906.08876*.