Exploring Data Acquisition Strategies for the Domain Adaptation of QA Models

Maurice Falk **Adrian Ulges** **Dirk Krechel**
RheinMain University of Applied Sciences, Germany
{firstname.lastname}@hs-rm.de

Abstract

Domain adaptation in Question-Answering (QA) is of importance when deploying models in new target domains where specific terminology and information needs exist. Adaptation commonly relies on a supervised fine-tuning using datasets composed of contexts, questions, and answers from the new domain. However, the annotation of such datasets is known to demand significant time and resources. In this work, a semi-automatic approach is investigated, where – instead of a fully manual acquisition – only answer spans (or questions, respectively) are selectively labeled, and a generative model provides a corresponding question (or answer). The efficacy of the proposed approach is compared against LLM-based autogenerative methods. Through experiments on diverse domain-specific QA datasets, both from the research community and industry practice, the superiority of the semi-automatic approach in obtaining higher QA performance is demonstrated.

1 Introduction

Question answering (QA) is one of natural language processing's most prominent tasks, targeted at identifying answers to questions from a given text corpus. At its core sits a reading comprehension (short, *reader*) model, which derives the answer given the question and a candidate context (or passage). Readers either *extract* the answer as a subspan of the candidate context, or *generate* new answers altogether. While the latter approach has recently gained popularity as *retrieval-augmented generation (RAG)* in the context of large language models (LLMs), extractive approaches offer benefits in terms of interpretability, speed, and – most importantly – in the fact that their answers are always grounded in source material.

In this work, we focus on extractive readers, and specifically on the issue of domain adaptation. This is of relevance when QA systems are deployed in new target domains and have to cope with specific terminology, but also with specific information needs of their users, as depending on the domain, different aspects of a text may be of relevance.

A common approach towards domain adaptation would be a supervised fine-tuning of readers, given target-domain triples of questions, candidate context and answers. This, however, would require extensive annotation effort, which raises the questions how to collect training triples more efficiently. To do so, several approaches have recently proposed generative (L)LMs as an option to synthesize questions and answers from contexts. In this paper, we investigate a **semi-automatic approach**, where a human annotator only labels interesting (answer) spans (or questions), instead of both. We argue that it might still be difficult for an LLM to identify question-worthy answer spans or generate questions if only given a context. In contrast, given a context and an answer, formulating a corresponding question is relatively easy and could, hence, be automated. This would lead to a domain adaptation procedure in which users label potentially relevant answers (or questions) in contexts, and a language model generates a corresponding questions (or answer), completing triples on which the reader is fine-tuned. In this paper, we compare the above semi-automatic approach to a fully-synthetic one, where both questions and answers are generated. Our findings (on three common research benchmarks and a closed-domain dataset from an industry partner) are:

- Manually labeling a limited amount of answers leads to strong performance improvements, compared both to labeling questions and to fully automated data generation.
- To achieve this improvement, even medium-sized LMs as question generators suffice, which suggests that localizing interesting answers is key to a successful reader adaptation.

- Given a small number of semi-automatic QA pairs, we examine how bootstrapping the auto-generative models impacts their performance.

2 Related Work

The domain adaptation of readers was examined using various approaches. While Hazen et al. (2019) have shown that transfer learning, i.e., fine-tuning the reader on a common large-scale QA dataset, can lead to good performance of the reader on a new domain. But they also report that further supervised fine-tuning using QA pairs of the target domain further improves performance. Therefore, further work focused on obtaining good QA pairs for training while using the same reader architecture (Devlin et al., 2018) for evaluation. Due to the costs of manual annotation of QA pairs, other works have explored ways to automatically obtain QA pairs of the target domain without human annotators. One differentiates between answer-first and question-first approaches. The answer-first approach starts by selecting candidate answer spans from the context directly and then uses the context and candidate answers to generate questions. The answer span selection can be done either in an extractive way using an answer span detector (Alberti et al., 2019; Puri et al., 2020; Bartolo et al., 2021; Luo et al., 2021), or in a generative way, where an (encoder-)decoder language model generates answer tokens from the context (Shakeri et al., 2020; Bartolo et al., 2021). In the question-first approach, possible questions for a given context are generated, which are then used to generate the answers (Shakeri et al., 2020).

3 Approach

Extractive QA is targeted at localizing an answer to a given question in a context. For example, given the context "Dune is a science fiction epos produced by Denis Villeneuve, [...]", the answer to the question "Who is the producer of Dune?" would be the last two words, "Denis Villeneuve." Following the reader architecture proposed by Devlin et al. (2018), given a context **c** and question **q**, both are tokenized into token sequences, concatenated, and processed by a transformer encoder to obtain contextualized embeddings. Finally, these embeddings are fed through a head model, which returns two probabilities indicating every token's likelihood to be the start or end token of the answer. The answer is then estimated to be the span between the most probable start and end token.

Following Hazen et al. (2019), the training of domain-specific readers happens in two phases: (1) a base reader model is obtained by fine-tuning a pretrained LM on a large-scale QA corpus such as SQuAD (Rajpurkar et al., 2016) (Engl.) or GermanQuAD (Möller et al., 2021) (German), and (2) performance on the target domain is improved by further fine-tuning the base model on some domain-specific QA pairs.

3.1 Domain Adaptation Data

While a manual annotation of domain-specific QA pairs yields high-quality data, it is also quite expensive. We, therefore, investigate other labeling approaches that require only partial or no manual annotation.

Generating questions and answers This setup tries to overcome the need for manual labeling altogether by estimating both question $\hat{q}$ and answer $\hat{a}$ from each given context c, using a model η:

$$\hat{q}, \hat{a} = \eta(c)$$

Note that η is a generative model, and that – to form training data for an extractive model – the generated answer has to be matched within the context. If the answer does not exist in the context, $\hat{a}$ is undefined and no training triple is generated. We compare two different generators:

QAGen2S: The model proposed by Shakeri et al. (2020) is an encoder-decoder model that generates questions and answers in two steps. First, the model generates a candidate question for a given context. The generated question is then included in the second step to generate a corresponding answer.

LLaMA-QAGen: Following the above approach of applying larger-scale LLMs, LLaMA 2 is used to generate both question and answer. Because we observed that many generated answers could not be located in the context, we fine-tuned the non-instruction model for question- and answer generation.

Generating Questions Only (GQO) Given a context c, a human annotator labels an interesting (answer) span a, but does not continue to formulate a question (which drastically reduces the costs of labeling). Instead, an answer-aware Question Generation (AA-QG) model ϕ is used to estimate a corresponding question $\hat{q}$, given context and answer:

$$\hat{q} = \phi(c, a)$$

We test two different question generators ϕ:

QGen: Chan and Fan (2019) propose a transformer-based encoder-decoder model, which is pointed at the answer span by inserting special tokens into the context. In the above example, the model input would become "Dune is a science fiction epos produced by **<hl>Denis Villeneuve<hl>**." We start from a pretrained LM and fine-tune the model specifically for question generation.

LLaMA-QGen: Inspired by the recent success of instruction-tuned large-scale LMs as task-agnostic problem solvers (Zhao et al., 2023), we use the instruction-tuned variant of LLaMA 2 (Touvron et al., 2023) as an answer-aware question generator. The prompt template is shared in A.3.

Generating Answers Only (GAO) In this setup, questions are assumed to be manually created, and an answer detection model ψ localizes the answer:

$$\hat{a} = \psi(c, q).$$

We test this setup with the QAGen2S encoder-decoder model, feeding manually acquired questions and generating only the answer.

Any fine-tuning of the aforementioned models was conducted on a generalist QA dataset.

3.2 Data Gathering and Bootstrapping

Given the above models, the following labeling procedures for gathering a domain adaptation dataset are examined:

- **Generation-Only (GO)**: No manual annotation is carried out, but QA pairs for domain adaptation are fully generated by applying the generator η on all available domain contexts.
- **Semi-Automatic (SA)**: A fixed number n of answer spans only **or** questions only are annotated by human experts, which limits the annotation effort. The corresponding answer span / question is generated by ψ / ϕ.
- **Bootstrapping (BS)**: The QA dataset obtained by *SA* is used to further fine-tune a generative model η, obtaining a domain-specific generator η'. By applying η' to all domain contexts, a larger-scale domain adaptation set is bootstrapped.

4 Experiments

We examine the effectiveness of different datasets obtained through the scenarios and models described in the previous section. For evaluation, we use four different domain-specific datasets: **BioASQ** (Tsatsaronis et al., 2015), containing QAs from the biomedical domain; **CovidQA** (Möller et al., 2020), containing QAs about Covid-19 from biomedical articles; **TextbookQA** (Kembhavi et al., 2017), which contains QAs from Life-, Earth-, and Physical Science textbooks; and a manually annotated German QA dataset, referred to as **BankQA**, from handbooks from an industry partner in the German banking domain. For BioASQ and TextbookQA, we use the datasets from the MRQA 2019 Shared Task (Fisch et al., 2019), which unifies the pre-processing of the datasets. We randomly sample 80 percent of contexts as a training corpus and remove all QA pairs for the domain adaptation task. The QA pairs of the remaining contexts are used as a test set. More details about the datasets is given in A.1.

4.1 Setup

For the evaluation of a dataset, a new reader is fine-tuned on the dataset's QA samples. The resulting model is then applied to the test set, and F1 (word-level) and exact match (EM) scores are reported. We use *electa-base* (Clark et al., 2020) as the encoder of our reader and fine-tune a model on SQuAD / GermanQuAD as our base model for all our runs. Details about hyperparameters and fine-tuning for the reader and all other models can be found in A.2. At the core of our QAGen2S model, we use *bart-base* and fine-tune the model for QA generation on the training split of the SQuAD (GermanQuAD) dataset, following the hyperparameters reported in the original paper. The checkpoints with the lowest Cross-Entropy loss on the dev set are used as our final models. Finally, for *LLaMA-QAGen*, we fine-tune the base-version of LLaMA 7B for QA generation using *QLoRA* (Dettmers et al., 2023), following the same procedure described by QAGen2S.

4.2 Manual Labeling of Questions versus Answers

In this experiment, we compare how effective labeling only questions / answers would be for domain adaptation. To obtain the **GQO** datasets, we simulate the manual labeling of answer spans by using

	BankQA		BioASQ		CovidQA		TextbookQA	
	F1	EM	F1	EM	F1	EM	F1	EM
No domain adaptation *	49.22	21.52	60.30	46.15	56.20	32.70	41.95	30.50
Manually annotated QAs	**63.99 ±1.02**	**39.55 ±0.74**	**89.84 ±1.11**	**86.82 ±1.81**	66.33 ±0.81	43.02 ±1.30	**57.41 ±1.44**	**50.06 ±1.25**
Generating Questions/Answers Only (GQO / GAO)								
Ann. Answers + ϕ (T5)	<u>59.81 ±1.13</u>	<u>33.99 ±2.58</u>	79.57 ±1.34	75.92 ±1.38	<u>**67.28 ±0.93**</u>	<u>**43.90 ±1.13**</u>	41.78 ±3.33	36.35 ±3.17
Ann. Answers + ϕ (LLaMA2)	53.06 ±2.27	30.13 ±2.60	<u>84.83 ±3.02</u>	<u>82.47 ±3.18</u>	51.00 ±2.89	28.93 ±2.81	27.56 ±3.04	22.26 ±2.25
Ann. Questions + ψ (QAGen2S)	38.62 ±0.85	12.83 ±1.33	62.68 ±2.59	46.35 ±2.76	11.61 ±1.40	2.01 ±0.53	33.43 ±3.79	24.97 ±3.07
Semi-Automatic (SA) (*n* annotated answers + ϕ T5)								
n = 10	51.11 ±1.71	24.39 ±1.84	59.04 ±2.03	46.56 ±1.94	58.54 ±3.39	29.69 ±4.74	38.97 ±5.76	31.07 ±5.05
n = 25	54.10 ±2.58	27.89 ±2.58	59.33 ±3.45	47.22 ±3.11	62.04 ±1.54	34.72 ±1.21	42.65 ±2.26	34.40 ±2.02
n = 50	54.12 ±1.19	29.06 ±1.24	58.89 ±1.52	46.02 ±1.45	63.31 ±2.07	35.97 ±2.29	43.37 ±1.61	34.21 ±2.32
n = 100	57.28 ±2.11	33.09 ±2.48	61.88 ±4.53	50.84 ±2.96	63.48 ±2.49	37.11 ±2.22	41.86 ±2.95	33.71 ±2.71
Generation Only (GO) (η)								
QAGen2S (BART-base)	47.38 ±0.66	19.01 ±1.33	51.43 ±3.48	35.18 ±3.85	18.12 ±1.85	7.42 ±1.86	38.49 ±1.42	27.36 ±1.86
QAGen (LLaMA2)	51.44 ±1.58	22.42 ±3.86	61.96 ±3.21	48.76 ±3.24	59.83 ±0.56	34.21 ±1.92	<u>44.31 ±2.68</u>	<u>37.23 ±2.86</u>
Bootstrap (BS) η **with** ***n* = *100***								
QAGen2S (Bootstrapped)	48.91 ±1.23	21.79 ±1.64	55.40 ±2.06	45.48 ±1.75	21.36 ±10.09	8.05 ±5.61	38.72 ±2.54	32.33 ±2.01
QAGen (Bootstrapped)	49.52 ±1.53	21.44 ±1.96	60.11 ±2.32	52.31 ±2.03	34.81 ±4.23	22.52 ±1.63	39.52 ±3.81	33.77 ±3.99

Table 1: F1 and EM scores of a reader on the test splits when the reader is fine-tuned on the obtained datasets. The best scores for each domain dataset are indicated by **bold** cells, the best scores where no fully-labeled domain dataset is used are indicated by <u>underlined</u> cells. For experiment, the mean and standard deviation of 5 runs are reported. (*): The base reader was not further fine-tuned on a domain dataset.

the annotated ones from the original training sets, and generate corresponding questions with ϕ. For every annotated answer span from the training set, at most one question is generated. The procedure is analogous for **GAO** with ψ. The results reported in Table 1 show significant improvements compared to the baseline for the **GQO** approach using the T5-based ϕ. For *CovidQA*, even better scores can be achieved than when using the original training set. Only for the *TextbookQA* dataset almost no change in F1 is reported. This might be due to the format of the manually labeled questions, which vastly differs from the questions in the dataset used to train ϕ. A comparison of *TextbookQA* questions, as well as QA examples obtained by the different models can be found in B.2.

Due to the strong performance of the **GQO** approach, we further investigate how the number of manually annotated answer spans impacts the performance. We randomly sample $n = 10, 25, 50, 100$ answer spans and use ϕ (T5) to obtain related questions. To prevent overfitting of the reader, the model is fine-tuned for 5 epochs (instead of 20). The results in Table 1 suggest that, while a performance increase for *BankQA* and *CovidQA* with only 10 annotated answer spans can be observed, having more annotated answer spans also lead to better results. For *BioASQ*, the performance even slightly decreases for $n = 10, 25, 50$, but 100 answer spans account for less than 10 percent of the manually labeled answer spans in the training set.

4.3 Evaluation of Generation-Only and Generator Bootstrapping

Here, we use η to generate QA pairs from all contexts (see A.3 for details). The results in Table 1 shows that the QA pairs generated by *QAGen* slightly increase the reader's performance, do no catch up with the semi-automatic approach. On the other hand, the QA pairs generated by *QAGen2S* decrease the reader's performance on all domains. Differences to Shakeri et al. (2020) are given in C.

Finally, we examine if η can improved by being *bootstrapped* on the new domain. For this, we further fine-tune η for two epochs on 100 QA pairs obtained with ϕ (T5). Compared to the non-bootstrapped variant, bootstrapping show improvements for *QAGen2S*, but lowers the performance of *QAGen*. Even with bootstrapping, *GO* lags behind the *SA* approach.

5 Conclusion

We have investigated semi-automatic methods for acquiring domain-specific QA datasets, and have shown that utilizing annotated answer spans alongside an answer-aware question generator surpasses other methods in performance, whereas bootstrapping domain-specific LLM generators with a limited number of annotated samples remains an open challenge. Our results suggest future research should prioritize identifying potential answer spans for further advancements in QA dataset acquisition.

Ethical Considerations

The proposed methods aim to support the annotation process of QA datasets, and our results indicate that human annotations continue to be indispensable to achieve the best possible quality.

For the BankQA dataset, we can assure that appropriate working conditions were guaranteed for all persons involved in the annotation of the samples.

Limitations

We are unable to share the confidential data from the BankQA dataset, which prevents others from replicating our results or conducting further research with this dataset. It is important to emphasize that all our experiments were conducted to the best of our knowledge and belief.

It is important to note that this work focuses explicitly on extractive QA, where answers are located in a known context. While this eliminates the risk of falsely generated answers in a productive QA system, it does not guarantee the correctness of the generated questions and answers. This could lead to falsely predicted answers, highlighting the need to question an answer and consider the surrounding context in real-world applications, as is standard in any QA system.

Furthermore, the diverse nature of language, data, and domains may yield varied results. Additionally, obtaining basic requirements like a large-scale QA dataset for fine-tuning base models is not readily available in every language. This limitation also applies to LLMs such as LLaMA2, which was fine-tuned on documents from a limited number of languages.

Moreover, utilizing LLMs to generate synthetic data incurs significant computational expenses. Due to these costs and time constraints, we could not utilize larger LMs that might offer even better performance.

Acknowledgments

This work was funded by the German Federal Ministry of Education and Research (BMBF), Program "FH-Kooperativ", Project "SCENT" (ID:13FH003KX0).

References

Llm prompting guide. `https://huggingface.co/docs/transformers/main/tasks/prompting`. Accessed: 2024-05-02.

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic qa corpora generation with roundtrip consistency. *arXiv preprint arXiv:1906.05416*.

Max Bartolo, Tristan Thrush, Robin Jia, Sebastian Riedel, Pontus Stenetorp, and Douwe Kiela. 2021. Improving question answering model robustness with synthetic adversarial data generation. *arXiv preprint arXiv:2104.08678*.

Ying-Hong Chan and Yao-Chung Fan. 2019. A recurrent bert-based model for question generation. In *Proceedings of the 2nd workshop on machine reading for question answering*, pages 154–162.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *Preprint*, arXiv:2003.10555.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Preprint*, arXiv:2305.14314.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. Mrqa 2019 shared task: Evaluating generalization in reading comprehension. *arXiv preprint arXiv:1910.09753*.

Timothy J Hazen, Shehzaad Dhuliawala, and Daniel Boies. 2019. Towards domain adaptation from limited data for question answering using deep neural networks. *arXiv preprint arXiv:1911.02655*.

Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern recognition*, pages 4999–5007.

Hongyin Luo, Shang-Wen Li, Mingye Gao, Seunghak Yu, and James Glass. 2021. Cooperative self-training of machine reading comprehension. *arXiv preprint arXiv:2103.07449*.

Timo Möller, Anthony Reina, Raghavan Jayakumar, and Malte Pietsch. 2020. COVID-QA: A question answering dataset for COVID-19. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*, Online. Association for Computational Linguistics.

Timo Möller, Julian Risch, and Malte Pietsch. 2021. Germanquad and germandpr: Improving non-english question answering and passage retrieval. *arXiv preprint arXiv:2104.12741*.

Raul Puri, Ryan Spring, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2020. Training question answering models from synthetic data. *arXiv preprint arXiv:2002.09599*.

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

Siamak Shakeri, Cicero 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.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, et al. 2015. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*, 16:1–28.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing. *Preprint*, arXiv:1910.03771.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

A Appendix

A.1 Dataset stats

We share details about the QA datasets obtained by the different approaches in Table 2. Table 3 contains stats about the test splits for each domain dataset.

A.2 Fine-tuning and Hyperparameters

In the following, we explain the fine-tuning and hyperparameters used for each model in more detail.

A.2.1 Reader

We used the already fine-tuned and publicly available models *deepset/electra-base-squad2* and *deepset/gelectra-base-germanquad* from Huggingface (Wolf et al., 2020) as our base models. During fine-tuning on the domain datasets, we use the *AdamW optimizer* with a *learning rate* of 5×10^{-5}, a *weight decay* of 0.01, and a *learning rate warm-up* of 10 percent. A *batch size* of 16 is used. We performed experiments with and without gradient clipping and report the best results. We fine-tune the reader for 20 epochs and keep the checkpoint after the last epoch. Due to the small number of annotated QA pairs in each dataset, we decided against further sampling a validation split from the training data and perform no early-stopping. During fine-tuning and inference, a *maximum sequence length* of 384 and a *stride* of 128 is used.

A.2.2 Answer-Aware Question Generator (T5)

For the T5-based AA-QG, we use the already pre-trained and publicly available models *valhalla/t5-base-qg-hl* and *dehio/german-qg-t5-quad* from Huggingface. These models were not further fine-tuned in our experiments.

A.2.3 QAGen2S

We fine-tune a BART encoder-decoder model as described by Shakeri et al. (2020). Due to hardware limitations, we use *base* variant of BART (*facebook/bart-base* for English / *Shahm/bart-german* for German) as our base models. The base model is fine-tuned on SQuAD / GermanQuAD for 5 epochs with a *batch size* of 8. A *gradient accumulation size* of 3 is used. The *AdamW optimizer* with a *learning rate* of 3×10^{-5} with a *warm-up* of 10 percent is used. The model epoch with the lowest Cross Entropy loss on the dev / test split is used as final model.

A.2.4 QAGen

We used the 7B variant of LLaMA 2 as our base model and fine-tuned it for question and answer generation on *SQuAD* for English / *GermanQuAD* for German for 5 epochs. For memory-efficient fine-tuning, we used QLoRA (Dettmers et al., 2023), with an alpha of 16 and 10 percent dropout. A batch size of 8 and a gradient accumulation step

Dataset	# Contexts	# QAs	Avg. Context Length	Avg. Question Length	Avg. Answer Length
BankQA					
Original	310	776	438.66	43.25	106.04
Ann. Answers + ϕ (T5)	310	751	438.66	63.07	104.79
Ann. Answers + ϕ (LLaMA2)	310	776	438.66	73.65	106.04
Ann. Questions + ψ (QAGen2S)	310	645	438.66	43.51	34.73
QAGen2S (BART-base)	310	788	438.66	58.66	51.63
QAGen (LLaMA2)	308	1303	440.93	63.63	74.89
BioASQ					
Original	1192	1205	1436.94	64.28	13.99
Ann. Answers + ϕ (T5)	1192	4070	1436.94	66.43	9.05
Ann. Answers + ϕ (LLaMA2)	1192	1275	1436.94	95.41	13.99
Ann. Questions + ψ (QAGen2S)	1192	1096	1436.94	64.42	16.20
QAGen2S (BART-base)	1192	2993	1436.94	58.26	22.04
QAGen (LLaMA2)	1192	5811	1436.94	57.59	25.94
CovidQA					
Original	117	614	4356.21	55.57	70.83
Ann. Answers + ϕ (T5)	117	611	4356.21	60.04	70.99
Ann. Answers + ϕ (LLaMA2)	108	614	4351.13	97.3	70.83
Ann. Questions + ψ (QAGen2S)	117	72	4356.21	54.07	92.64
QAGen2S (BART-base)	117	11	4356.21	60.00	64.27
QAGen (LLaMA2)	117	571	4356.18	57.95	27.51
TextbookQA					
Original	311	1185	2919.46	57.09	12.79
Ann. Answers + ϕ (T5)	311	3893	2919.46	58.08	9.95
Ann. Answers + ϕ (LLaMA2)	311	1185	2919.46	64.65	12.79
Ann. Questions + ψ (QAGen2S)	311	859	2919.46	57.58	30.08
QAGen2S (BART-base)	311	512	2919.46	52.96	23.92
QAGen (LLaMA2)	311	1483	2919.46	54.23	21.42

Table 2: Details about the datasets obtained from different labeling approaches. The lengths refer to the average number of characters.

size of 2 is used. We used AdamW as an optimizer with a learning rate of 2×10^{-4} and a warm-up of 10 percent. The following format was used for fine-tuning and inference:

```
Context: {context}
Question: {question}
Answer: {answer}
```

For German data, we translated the format into German.

A.3 Decoding

For the decoding, i.e., the generation of questions and answers, the following parameters were used for all models:

- **Question Generation**: We follow the generation parameters reported by Shakeri et al. (2020), namely, *Top K+Nucleus sampling*. We set $k = 20$ and the token probability mass to $p = 0.95$. For the QAGen2S model, we sample up to 10 unique questions for each context and keep the ones with the highest LM scores during answer generation (*LM Filtering*, also proposed by Shakeri et al. (2020)). For QAGen, up to 5 unique questions are generated for each context. No filtering is applied.
- **Answer Generation**: We use greedy decoding to generate one answer span for every (context, question)-pair. If the generated answer span is not included in the context, the (context, question)-pair is discarded.

Following known prompting guidelines (pro), we came up with the following template for prompting LLaMA2 for answer generation:

```
Generate a question for the given
context and answer, so that the
question can be answered by the
given answer. Only output the question.
Context: {context}
Answer: {answer}
Question:
```

Dataset	# Contexts	# QAs	Avg. Context Length	Avg. Question Length	Avg. Answer Lenght
BankQA	78	223	400.42	44.64	98.39
BioASQ	298	319	1450.12	63.59	12.9
CovidQA	30	159	4389.73	55.75	66.84
TextbookQA	78	318	2997.72	52.19	12.29

Table 3: Details about the test splits. The lengths refer to the average number of characters.

We translated the prompt for German data.

B Questions and Answers

B.1 Examples

For comparison, examples of questions and answers obtained by the different approaches are given for *BioASQ* in Table 4, and *TextbookQA* in Tables 5 and 6. Due to the high context length of samples in *CovidQA*, no examples are given for the dataset.

B.2 TextbookQA Questions

The format of the annotated questions in the *TextbookQA* dataset differ from those in the *SQuAD* dataset on which the QA generators are fine-tuned on. In the following, we give some examples of questions:

TextbookQA:

- *this much of the municipal groundwater supplies in the united states are polluted.*
- *crude oil is a mixture of many different*
- *which of these substances has the highest freezing point?*
- *in hyperopia, the eyeball is*
- *when an earthquake happens, we say that its ___________ was located 100 miles northwest of san francisco.*

SQuAD1.1:

- *To whom did the Virgin Mary allegedly appear in 1858 in Lourdes France?*
- *"The Closer I get to You" was recorded with which artist?*
- *In therapy, what does the antibacterial interact with?*
- *At what age did Chopin leave Poland?*
- *What does SDK stand for?*

The questions presented in *SQuAD* (and the other datasets *GermanQuAD*, *BioASQ*, *CovidQA* and *BankQA*) are mostly well structured, i.e., end with a question mark and contain w-words, while the questions in *TextbookQA* are more diversely structured and do not always follow the syntax of a question.

C QAGen2S Setup Differences

We identified two main differences between our setup and the setup used by Shakeri et al. (2020), which might explain the differences in performance:

1. **The number of contexts the QAs were generated on**: Due to limited compute- and time resources, we did not craw additional domain contexts to generate QA pairs on. Thus, the number of samples generated by Shakeri et al. (2020) is a multiple of ours.
2. **Smaller generator**: Due to limited compute- and time resources, we used the smaller *bart-base* variant, compared to *bart-large*.

Passage: A mutation in the alpha-synuclein gene has recently been linked to some cases of familial Parkinson's disease (PD). We characterized the expression of this presynaptic protein in themidbrain, striatum, and temporal cortex of control, PD, and dementia with Lewy bodies (DLB) brain. Control brain showed punctate pericellular immunostaining. PD brain demonstrated alpha-synucleinimmunoreactivity in nigral Lewy bodies, pale bodies and abnormal neurites. Rare neuronal soma in PD brain were immunoreactive for alpha-synuclein. DLB cases demonstrated these findings as well asalpha-synuclein immunoreactivity in cortical Lewy bodies and CA2-3 neurites. These results suggest that, even in sporadic cases, there is an early and direct role for alpha-synuclein in the pathogenesis of PD and the neuropathologically related disorder DLB.	
Original:	
Q: Against which protein is the antibody used for immonostaining of Lewy bodies raised?	A: alpha-Synuclein
Annotated Answer + ϕ (T5):	
Q: What protein is associated with the pathogenesis of Parkinson's disease?	A: alpha-Synuclein
Annotated Answer + ϕ (LLaMA):	
Q: What is the relationship between alpha-synuclein and Parkinson's disease (PD)?	A: alpha-Synuclein
Annotated Question + ψ (QA2S):	
Q: Against which protein is the antibody used for immonostaining of Lewy bodies raised?	A: punctate pericellular immunostaining
QAGen2S (BART):	
Q1: What gene has been linked to some cases of familial Parkinson's disease?	A1: alpha-synuclein gene
Q2: What gene has recently been linked to some cases of familial Parkinson's disease?	A2: alpha-synuclein gene
Q3: What does DLB stand for?	A3: Lewy bodies
QAGen (LLaMA):	
Q1: Punctate pericellular immunostaining was shown in what part of the brain?	A1: Control
Q2: What gene is associated with familial Parkinson's?	A2: alpha-synuclein
Q3: In familial PD, alpha-synuclein is linked to which gene?	A3: alpha-synuclein
Q4: What does PD stand for?	A4: Parkinson's disease
Q5: What gene has recently been linked to some cases of familial Parkinson's disease?	A5: alpha-synuclein

Table 4: Example QA pairs for a context of the *BioASQ* dataset.

Passage: The Paleozoic is the furthest back era of the Phanerozoic and it lasted the longest. But the Paleozoic was relatively recent, beginning only 570 million years ago. [...] The Paleozoic begins and ends with a supercontinent. At the beginning of the Paleozoic, the supercontinent Rodinia began to split up. At the end, Pangaea came together. A mountain-building event is called an orogeny. Orogenies take place over tens or hundreds of millions of years. [...] Geologists find evidence for the orogenies that took place while Pangaea was forming in many locations. For example, Laurentia collided with the Taconic Island Arc during the Taconic Orogeny. The remnants of this mountain range make up the Taconic Mountains in New York. The Taconic Orogeny is an example of a collision between a continent and a volcanic island arc. Laurentia experienced other orogenies as it merged with the northern continents. The southern continents came together to form Gondwana. When Laurentia and Gondwana collided to create Pangaea, the Appalachians rose. Geologists think they may once have been higher than the Himalayas are now. Pangaea was the last supercontinent on Earth. Evidence for the existence of Pangaea was what Alfred Wegener used to create his continental drift hypothesis, which was described in the chapter Plate Tectonics. As the continents move and the land masses change shape, the shape of the oceans changes too. During the time of Pangaea, about 250 million years ago, most of Earths water was collected in a huge ocean called Panthalassa.	
Original:	
Q1: this mountain range grew much higher when gondwana and laurentia collided to create pangaea.	A1: the appalachians
Q2: the remnants of the taconic mountain range are found in ________________.	A2: new york
Annotated Answer + ϕ (T5):	
Q1: When Laurentia and Gondwana collided, what mountain range rose?	A1: the appalachians
Q2: Where do the Taconic Mountains lie?	A2: new york
Annotated Answer + ϕ (LLaMA):	
Q1: What mountain range in North America is believed to have formed during the collision between Laurentia and the Taconic Island Arc during the Taconic Orogeny?	A1: the appalachians
Q2: What was the name of the mountain range that formed during the orogeny that occurred when Laurentia collided with the Taconic Island Arc?	A2: new york
Annotated Question + ψ (QA2S):	
Q1: this mountain range grew much higher when gondwana and laurentia collided to create pangaea.	A1: the Appalachians rose
Q2: the remnants of the taconic mountain range are found in ________________.	A2: Taconic Mountains in New York
QAGen2S (BART):	
Q1: Pangaea was the last supercontinent on Earth	A1: Pangaea came together
Q2: Pangaea was the last supercontinent on Earth.	A2: Pangaea came together
QAGen (LLaMA):	
Q1: How many years ago did most of Earth's water collect in a huge ocean called Panthalassa?	A1: 250 million years ago
Q2: The Paleozoic is the furthest back era of what?	A2: Phanerozoic
Q3: What are the Paleozoic and Phanerozoic eras?	A3: era of the Phanerozoic
Q4: When was the Paleozoic?	A4: 570 million years ago
Q5: How long did the Paleozoic last?	570 million years

Table 5: Example QA pairs for a context of the *TextbookQA* dataset. We observed that the ϕ (LLaMA) sometimes fails to formulate questions that are answered by the provided span.

Passage: Most fossils are preserved by one of five processes outlined below (Figure 1.1): Most uncommon is the preservation of soft-tissue original material. Insects have been preserved perfectly in amber, which is ancient tree sap. [...] Scientists collect DNA from these remains and compare the DNA sequences to those of modern counterparts. The most common method of fossilization is permineralization. After a bone, wood fragment, or shell is buried in sediment, mineral-rich water moves through the sediment. This water deposits minerals into empty spaces and Five types of fossils: (a) insect preserved in amber, (b) petrified wood (permineralization), (c) cast and mold of a clam shell, (d) pyritized ammonite, and (e) compression fossil of a fern. produces a fossil. Fossil dinosaur bones, petrified wood, and many marine fossils were formed by permineralization. When the original bone or shell dissolves and leaves behind an empty space in the shape of the material, the depression is called a mold. The space is later filled with other sediments to form a matching cast within the mold that is the shape of the original organism or part. Many mollusks (clams, snails, octopi, and squid) are found as molds and casts because their shells dissolve easily. The original shell or bone dissolves and is replaced by a different mineral. For example, calcite shells may be replaced by dolomite, quartz, or pyrite. If a fossil that has been replace by quartz is surrounded by a calcite matrix, mildly acidic water may dissolve the calcite and leave behind an exquisitely preserved quartz fossil. Some fossils form when their remains are compressed by high pressure, leaving behind a dark imprint. Compression is most common for fossils of leaves and ferns, but can occur with other organisms. [...]	
Original:	
Q1: when insects are fossilized, it is usually in	A1: amber
Q2: when water deposits minerals into empty spaces to produce a fossil, it is called	A2: permineralization.
Q3: the most common way that fossil leaves are made is by	A3: compression
Annotated Answer + ϕ (T5):	
Q1: What is the oldest tree sap that was used as a foundation to contain fossils?	A1: amber
Q2: What is the process by which dinosaur bones are made?	A2: permineralization.
Q3: What process is most common for fossils of leaves and ferns?	A3: compression
Annotated Answer + ϕ (LLaMA):	
Q1: Which of the following processes is the most common method of fossilization?	A1: amber
Q2: Which process of fossilization most commonly results in the formation of fossil dinosaur bones?	A2: permineralization.
Q3: What is the most common method of fossilization?	A3: compression
Annotated Question + ψ (QA2S):	
Q1: when insects are fossilized, it is usually in	A1: amber
Q2: when water deposits minerals into empty spaces to produce a fossil, it is called	A2: mold
Q3: the most common way that fossil leaves are made is by	A3: compression
QAGen2S (BART):	
Q: What is the most common method of fossilization? A: permineralization	
QAGen (LLaMA):	
Q1: What are most fossil preserved by?	A1: mineral-rich water
Q2: When a clam's shell is replaced, what is its shape?	A2: mold
Q3: How are most fossils preserved?	A3: permineralization
Q4: What is the original bone or shell replaced with?	A4: different mineral

Table 6: Second example of QA pairs obtained for a context of the *TextbookQA* dataset.