CCQA: A New Web-Scale Question Answering Dataset for Model Pre-Training

Anonymous ACL submission

Abstract

We propose a novel open-domain question-answering dataset based on the Common Crawl project. With a previously unseen number of around 130 million multilingual question-answer pairs (including about 60 million English data-points), we use our large-scale, natural, diverse and high-quality corpus to in-domain pre-train popular language models for the task of question-answering. In our experiments, we find that our Common Crawl Question Answering dataset (CCQA) achieves promising results in zero-shot, low resource and fine-tuned settings across multiple tasks, models and benchmarks[1].

1 Introduction

Open-domain question-answering (ODQA) has evolved into a core problem in Natural Language Processing (NLP), receiving growing interest from the research community (Raffel et al., 2020; Roberts et al., 2020). Despite the notoriously difficult challenge to correctly answer open-domain questions from arbitrary domains, recent advances of pre-trained language models (such as BERT (Devlin et al., 2018), BART (Lewis et al., 2019) and T5 (Raffel et al., 2020)) have stimulated new research into additional, task-dependent pre-training steps. Specifically, recent publications show that in-domain pre-training regimes can improve models for several downstream tasks (Gururangan et al., 2020). For open-domain question-answering, newly proposed pre-training tasks such as the Inverse Cloze Task (ICT) (Lee et al., 2019), Body First Selection (BFS), Wiki Link Prediction (WLP) (Chang et al., 2020) and Question Answering Infused Pre-training (QUIP) (Jia et al., 2021) show consistent improvements over baselines. However, most of these approaches still rely on either unlabeled text, or synthetically generated question-answer (QA) pairs. In this paper, we explore a second, somewhat orthogonal dimension to these lines of work, examining if a web-scale collection of natural QA pairs can support ODQA through in-domain pre-training.

Per definition, an ODQA system should be able to answer any question from an arbitrary domain. We believe that to approach this ability with in-domain pre-training, a suitable dataset should address the following 5 challenges: (1) Size; ODQA requires knowledge of a wide variety of topics. The underlying dataset used for in-domain pre-training hence needs to cover this abundance of domains, requiring a web-scale dataset. (2) Naturalness; While synthetic corpora can potentially capture a wide variety of language phenomena, to understand and generate truly natural language in all facets, synthetic datasets are not sufficient. (3) Quality; Given the requirement for a diverse, large-scale dataset, high data quality in terms of cleanliness and sensibility becomes a major challenge. Given that web-scale data sources require highly automated approaches operating on noisy data, assuring data quality is non-trivial. (4) Diversity; Besides size, another challenge for any ODQA in-domain pre-training dataset is the generality of the corpus. The dataset needs to support answering many diverse questions to allow models to learn general concepts. (5) Evaluation Fairness; A web-scale question-answering dataset potentially overlaps with existing benchmark corpora, leading to inflated performance measures and impeding the evaluation fairness (Lewis et al., 2021a).

To overcome these challenges, we propose a new large-scale dataset for open-domain question-answering called the **C**ommon **C**rawl **Q**uestion **A**nswering (CCQA) dataset. Similar to popular datasets, such as C4 (Raffel et al., 2020), CCNet (Wenzek et al., 2020), CC-100 (Conneau et al., 2019) and HTLM (Aghajanyan et al., 2021b), we

[1] We will publish our dataset generation script and CCQA pre-training checkpoints for the evaluated models in the camera-ready version.

generate a large-scale, diverse and high-quality question-answering dataset from Common Crawl.

More specifically, Common Crawl allows us to obtain a large number of truly natural question-answer pairs, asked and answered by real humans on the web, rather than inferred through computational methods. Using the abundantly available *schema.org* question annotation[2], we generate question-answer pairs from explicit annotations, instead of heuristic rules, leading to high-quality data points, as shown in this paper. To evaluate the diversity and evaluation fairness of our dataset, we compute topic distributions and train-test overlaps with benchmark datasets.

In a large set of evaluations, we show that in-domain pre-training on our CCQA dataset achieves promising results across different settings, models and benchmarks. Using the rich information available on the web, we augment our dataset with additional data attributes beyond just question-answer pairs, such as votes, multiple (competing) answers, question summaries and intra-textual HTML markup, which can be used for a variety of tasks beyond question-answering in future work.

To summarize, our main contributions in this paper are as follows:

- We generate the first truly large-scale, natural question-answering dataset, containing around 130 million unfiltered question-answer pairs (55M unique), including about 60 million English data points (24M unique).
- We present key dataset statistics, confirming the high quality of our question-answer pairs, the wide range of diverse topics and a low overlap with existing benchmarks.
- We show the effectiveness of the dataset for in-domain pre-training by evaluating the performance of the unfiltered English subset on two question-answering tasks, three different settings, four models and five benchmarks.

2 Related Work

This work is inspired by a range of previous approaches using Common Crawl web-data, such as the Colossal Clean Crawled Corpus (C4) for language model pre-training (Raffel et al., 2020), the word/sentence representation generation corpus CCNet (Wenzek et al., 2020), the CC-100

[2] https://schema.org/Question

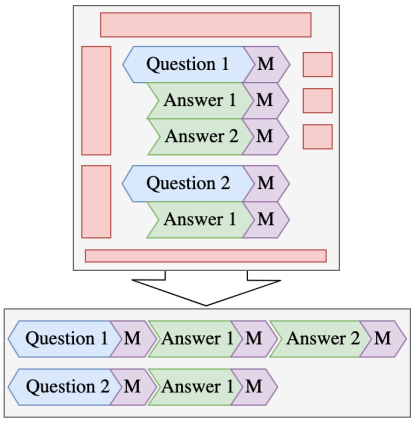

Figure 1: Dataset generation overview from the initial raw HTML file (top) to general purpose, webpage aggregated question-answer pairs (bottom). M = Additional question/answer metadata. Red boxes = Non-question-answer related webpage components.

dataset for translation (Conneau et al., 2019) and the markup-style language modelling HTLM corpus for zero-shot summarization (Aghajanyan et al., 2021b). Despite all previously mentioned applications directly relying on large-scale web data from Common Crawl, their scope and application vary significantly. Compared to previously proposed datasets based on Common Crawl, we are the first to extract well-structured question-answer pairs with additional meta-data, making our corpus a valuable resource for ODQA research, and a multitude of related tasks, such as question summarization, answer rating, and answer ranking.

Further web-based datasets outside the Common Crawl domain are the TriviaQA (Joshi et al., 2017) and ELI5 corpora (Fan et al., 2019), extracting small-scale question-answer datasets from Trivia websites and Reddit threads respectively. The large-scale GooAQ dataset (Khashabi et al., 2021) is similarly based on web data, however exploits the Google auto-complete feature and related answer boxes to generate semi-synthetic question-answer pairs. As a large-scale, completely synthetic dataset, the PAQ corpus (Lewis et al., 2021b) automatically generates a large set of *Probably Asked Questions* from Wikipedia articles. In contrast to these previously proposed datasets, our CCQA corpus presents a large-scale, natural and diverse question-answering resource in the same order of magnitude as the largest synthetic datasets.

Besides the generation of the CCQA dataset, we evaluate its potential as an in-domain pre-training corpus for open-domain question-answering. Our work is thereby aligned with previous in-domain

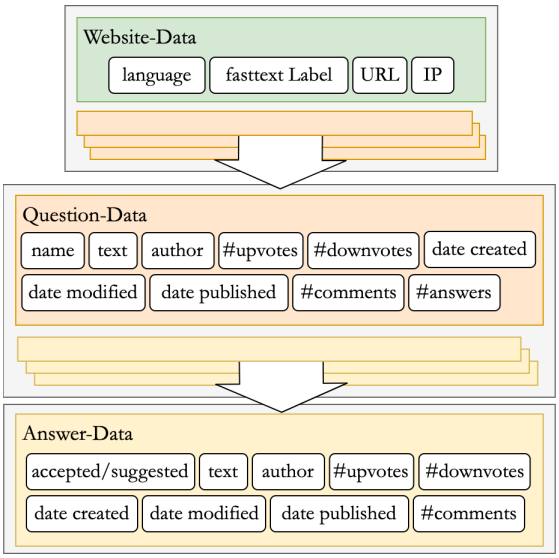

Figure 2: JSON data structure following the schema.org annotation. Fasttext language labels (Joulin et al., 2016a,b) added for language distinction.

pre-training approaches, which have shown to improve a variety of downstream tasks (Gururangan et al., 2020). Similar to in-domain pre-training, multiple domain-dependent pre-training tasks have been proposed for open-domain question-answering. For example, Lee et al. (2019) propose the Inverse Cloze Task (ICT), Chang et al. (2020) introduce Body First Selection (BFS) and Wiki Link Prediction (WLP) and Jia et al. (2021) describe a novel Question Answering Infused Pre-training (QUIP) task. Along similar lines, Aghajanyan et al. (2021a) propose pre-finetuning, an alternative to in-domain pre-training, using around 50 domain-dependent datasets, showing that their MUPPET approach generalizes well to many tasks. Khashabi et al. (2020) introduce a similar concept for question-answering in their UnifiedQA framework. While we propose a somewhat orthogonal dimension to most of these works, they nevertheless present us with strong intuition regarding the effectiveness of domain-dependent pre-training.

3 The Common Crawl Question Answering (CCQA) Dataset

3.1 Dataset Collection

Our Common Crawl Question Answering (CCQA) dataset contains around 130 million question-answer pairs (55M unique), extracted from 13 Common Crawl snapshots[3] between May 2020 and May 2021. A high-level overview of the dataset generation process is depicted in Figure 1. Starting from a set of raw HTML webpages, we make use of the schema.org definition to find relevant tags, such as the question, answer, author and votes (for the full set of tags see Figure 2). Using the explicit schema.org annotation (commonly used for search-engine optimization), instead of simple heuristics (e.g. question marks), we optimize the resulting corpus for high-quality data points. Specifically, due to the added efforts for website creators to define schema.org conforming meta-data, we believe that annotated question-answer pairs are likely to be relevant to the general public, mostly exclude rhetorical and contextual questions, and as a result constitute high quality QA data, despite the noisy nature of webpages.

During the dataset processing steps, we remove all HTML elements that do not contain valid schema.org markers (red in Figure 1) and subsequently clean every question on the webpage to only conserve markup related to the textual content of schema.org tags[4]. We further remove any unrelated markup attributes (e.g., CSS and JavaScript classes), before converting the content into a well-defined JSON object shown in Figure 2.

Using the 13 consecutive Common Crawl snapshots, we generate an initial dataset containing 130 million question-answer pairs. Within this initial corpus, there are two types of potential duplicates: (1) Same-URL duplicates; where a webpage is updated between any two Common Crawl snapshots and (2) Content duplicates; where webpages from any Common Crawl snapshot contain same questions with potentially similar answers. Here, we use the original, un-cleaned version of the dataset. This way, we present a lower-bound for the performance, which can be further improved through additional filtering steps in future work[5].

Our dataset generation procedure is further outlined in Algorithm 1, found in Appendix A. We also provide a detailed description of the dataset format in Appendix B. For qualitative examples of our generated dataset format, we refer readers to Appendix H.

3.2 Dataset Dimensions

To gain better insights into this massive amount of data, we present a mix of automatically obtained

[3] https://commoncrawl.org/

[4] Set of textual tags taken from developer.mozilla.org/en-US/docs/Web/HTML/Element

[5] We will provide a de-duplication script for same-URL duplicates, removing duplicates due to snapshot overlap.

Q-Sens^H	Q-Ans^H	QA-Sens^H	Markup	Q-Summ
96.5%	86%	82.25%	47.5%	11.7%
No A	Avg #A*	Mean Q	Mean A	Lang Tags
5.9%	1.41	43	57	77.9%

Table 1: Key CCQA dataset dimensions. Q=Question, A=Answer, QA=Question-answer pair, Sens=Sensibility, Ans=Answerability, Lang=Language, Summ=Summarization, Mean=Average number of words, ^{H}Human pilot study, *Excluding questions without answers.

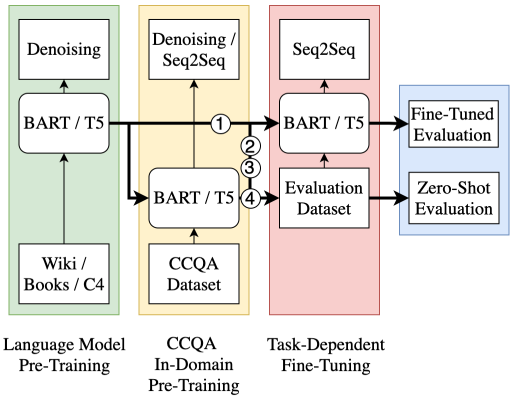

Figure 3: High-level overview of the closed-book CCQA in-domain pre-training step (yellow) as part of the training pipeline for BART and T5. Language model pre-training shown in green. Task-dependent fine-tuning presented in red. Evaluation in blue. (1) Baseline pre-training/fine-tuning pipeline, (2) In-domain pre-training/fine-tuning pipeline, (3) zero-shot baseline setting and (4) zero-shot in-domain pre-training setting.

dataset dimensions and a small-scale human pilot study as well as a set of key dataset distributions.

Regarding the small-scale human pilot study, we analyze a random subset of 400 individual question-answer pairs and evaluate their sensibility and answerability. We define *question sensibility* as to whether the annotator understands the questions itself, while *question answerability* refers to whether the question provides enough context for a perfect question-answering system to correctly answer the question. Furthermore, *QA-sensibility* denotes if the question-answer pair makes sense[6]. We refer interested readers to Table 9 in Appendix F for further explanations on sensibility/answerability.

As shown in Table 1, our CCQA corpus contains nearly exclusively sensible questions, with the vast majority of them also answerable and sensible as a pair. To complement our small-scale human annotation, we further explore key dataset dimension, including the fraction of samples with advanced markup, questions containing both, question name and question text (as defined by the schema.org annotation), the number of questions without gold-answers, average question and answer length and the number of webpages with a valid language label, indicating that the schema.org annotation highly correlates with carefully curated webpages.

Besides the key corpus-level statistics, we take a closer look at important dataset distributions in Table 2. Specifically, we present the top 5 domains at the top of Table 2, showing the largest number of webpages originating from the *stackexchange* domain, accounting for about 8% of data points. Regarding the topical distribution of our dataset, we use the DMOZ/Curlie taxonomy, automatically extracting hierarchical topic information[7]. We randomly sample $1,000$ question webpages and show the top 5 topics in the second row of Table 2. A more detailed topic distribution, also considering second-level assignments, can be found in Table 6 in Appendix C. Regarding the question-word distribution in our CCQA dataset, we observe that the majority of 36% of question words are *what* questions, followed by *how*, *when*, *which* and *where*. A full list of all 8 questions words and their relative appearance in our corpus can be found in Table 7 in Appendix D. Lastly, expanding on the number of non-trivial markup tags presented in Table 1, we explore the frequency of HTML markup tags in our dataset in the last row in Table 2. For a list of the top-25 tags found in our corpus, we point interested readers to Table 8 in Appendix E.

4 Evaluation

In this section, we showcase the value of our CCQA dataset with experiments on closed-book question-answering (section 4.1) as well as passage retrieval for open-book QA (section 4.2).

4.1 Closed-Book Question-Answering

4.1.1 Task

The closed-book question-answering task challenges systems to answer questions without any additional information sources, such as knowledge bases or evidence documents. As a result, models are solely relying on the question text and the information stored inside the model weights during training. Here, we evaluate our new CCQA dataset

[6]We do not check the answer for factual correctness but merely evaluate if it *could* be the answer for the given question.

[7]https://www.curlie.org

Metric	Top 5 Appearances in CCQA				
Domains	stackexchange (07.78%)	hotels (03.46%)	viamichelin (02.51%)	ccm (01.86%)	vrbo (01.74%)
Topics	Regional (38.90%)	Society (21.10%)	Business (08.30%)	Sports (07.00%)	Rec (06.20%)
Q-words	What (36.20%)	How (29.80%)	When (09.68%)	Which (09.64%)	Where (06.04%)
Markup	p (28.48%)	a (14.89%)	br (14.86%)	li (10.04%)	span (05.77%)

Table 2: CCQA dataset distribution for top 5 domains, topics according to the DMOZ/Curlie annotation, question words (Q-words, only computed on the English subset) and most common markup tags. % for q-words and markup tags presents portion of all q-word/markup appearances. ccm=commentcamarche, Rec=Recreational.

as an in-domain pre-training corpus for this highly challenging task by converting the JSON representation into plain question-answer pairs, removing markup tags and additional metadata.

4.1.2 Models & Training

Using the question-answer pairs from the CCQA dataset, we in-domain pre-train large language models for question-answering. We start with vanilla BART and T5 transformer models, shown on the left side (green) in Figure 3. We then further in-domain pre-train the models using a denoising or sequence-to-sequence (seq2seq) setup (yellow box in Figure 3). For the denoising task, we follow the vanilla BART approach (Lewis et al., 2019), using a concatenation of *Q:*||<*question*>||*A:*||<*answer*> as the model input. For the seq2seq task, we train the model to predict the gold answer given a question as input. With the additional in-domain pre-training step, a variety of training-flows emerge, shown as numbered circles in Figure 3:

(1) Using a vanilla pre-trained language model to fine-tune on the benchmark dataset.

(2) Using the CCQA dataset for in-domain pre-training and subsequently fine-tune on the benchmark dataset.

(3) Using a pre-trained language model to directly infer answers on the benchmark dataset.

(4) Using the CCQA in-domain pre-trained model to directly infer answers on the benchmark dataset.

4.1.3 Datasets

We evaluate our CCQA dataset on 5 common benchmarks, based on 4 publicly available datasets in the closed-book setting:

TriviaQA (TQA) is a short-form, factoid-style question-answering dataset (Joshi et al., 2017). For the closed-book task, we ignore the available contexts and focus exclusively on question-answer pairs. Since the official test-split of the dataset is not publicly available, we use the official validation set as our test split and randomly sample a validation set from the training data, as commonly done in previous work (Roberts et al., 2020).

Natural Questions (NQ) (Kwiatkowski et al., 2019) represents a popular corpus for question-answering research. Despite most recent work focusing on the **short-form** answers (NQ-Short), the NQ corpus also provides additional **long-form** answers (NQ-Long) for a large subset of questions. In this work, we use both, short, factoid answers and long-form responses.

ELI5, introduced by Fan et al. (2019), constitutes the first large-scale long-form dataset for open-ended question-answering. We again do not take available evidence documents into account, but focus on the question-answer pairs only.

GooAQ (Khashabi et al., 2021) contains semi-automatically extracted question-answer pairs from the Google question auto-complete feature.

4.1.4 Metrics

For datasets with short-form answers, we use the Exact Match (EM) metric for fine-tuned systems, in line with previous work by Roberts et al. (2020) and Lewis et al. (2021b). While the EM metric works well for systems that are aware of the task-specific format, it punishes potentially correct answers with additional context, which we believe is overly harsh in a zero-shot setting, where the specific output format is not known. Therefore, we propose using the Answer-level Recall (AR) metric for our zero-shot experiments, while limiting the answer length with the *max-length* and *length-penalty* inference parameters. As such, the AR metric requires the correct answer to be a continuous sub-sequence of the predicted tokens, while allowing for additional context. Since AR operates on token-level, the prediction of super/sub-words, e.g., ***fun****damental* instead of ***fun***, is considered incorrect.

For long-form question-answer datasets, we choose the Rouge-L (RL) score as our evaluation metric, which has shown strong correlation with Rouge-1 and Rouge-2 scores, and is commonly used in previous work (Khashabi et al., 2021).

Model	Zero-Shot TQA AR	NQ-Short AR	NQ-Long R-L	ELI5 R-L	GooAQ R-L	Fine-Tuned TQA EM	NQ-Short EM	NQ-Long R-L	ELI5 R-L	GooAQ R-L
BART Large						BART Large				
Rand. Init.	0.04	0.11	0.10	0.26	0.16	0.71	0.75	16.04	14.37	16.21
Vanilla	†4.91	†1.93	10.39	11.88	14.67	**28.67**	23.79	23.47	16.96	35.67
Vanilla[a]							26.50			
CCQA	†**5.14**	†**2.16**	**12.18**	†**15.21**	†**17.5**	25.82	22.91	21.25	17.23	32.53
CCQA-d	4.80	2.13	10.33	11.91	14.88	27.84	**23.96**	**24.56**	**17.27**	**35.92**
T5 Small						T5 Small				
Rand. Init.	0.05	0.11	1.13	1.49	0.80	0.44	0.54	10.86	13.06	8.71
Vanilla	†5.06	†1.74	9.16	7.55	†8.92	**21.02**	**21.16**	**22.09**	16.28	24.70
Vanilla[b]								19.00	23.00	
CCQA	†**5.13**	†**1.86**	†**13.63**	†**15.28**	†**15.46**	17.55	19.50	22.05	**16.33**	**25.35**
T5 Base						T5 Base				
Rand. Init.	0.04	0.11	0.00	0.00	0.00	0.32	0.38	13.58	12.72	7.93
Vanilla	†5.49	†2.02	†14.39	12.27	†14.99	**26.25**	**23.04**	**25.36**	16.58	**29.36**
Vanilla[c]						23.63	25.94			
CCQA	†**7.15**	†**3.19**	†**15.08**	†**15.69**	†**15.85**	22.69	22.32	24.73	**16.64**	29.09

Results from [a] Lewis et al. (2021a) [b] Khashabi et al. (2021) [c] Roberts et al. (2020)

Table 3: Closed-book zero-shot and fine-tuned results. Best performance of fairly computed results per sub-table **bold**. †Zero-shot model outperforms fully fine-tuned randomly initialized transformer of same architecture. *-d* extension indicates denoising CCQA pre-training task. *AR*=Answer-level recall, *EM*=Exact Match, *RL*=Rouge-L.

4.1.5 Hyper-Parameters

We use the default parameters of the BART (Lewis et al., 2019) and T5 (Raffel et al., 2020) models for in-domain pre-training and fine-tuning whenever possible. Regarding the in-domain pre-training on our CCQA dataset, we limit training to $800k$ steps using a batch-size of $1,024$. During our fine-tuning runs, we limit the number of updates to $20k$ steps with a batch-size of 256 samples, with exception of the GooAQ dataset, which we fine-tune for $100k$ steps due to it's large size. We select the best model during our in-domain pre-training runs based on the perplexity measure, and pick the top fine-tuned model according to the final evaluation metric. We do not perform any hyper-parameter search during in-domain pre-training and fine-tuning.

For the inference step, our hyper-parameter setting is closely related to commonly used summarization parameters. We use a beam-size of 4, max-length of 140, and length-penalty of 2.0. For the fine-tuned short-form task, we choose a max-length of 30, following Xiong et al. (2020) and a length-penalty of 1.0. All model evaluations are based on Huggingface Transformers[8] (Wolf et al., 2019).

[8]Experiments are executed on Nvidia V100 32GB GPUs.

4.1.6 Results

Our main results for the closed-book question-answering task are presented in Table 3, showing the zero-shot and fine-tuned performance of the BART Large (top), T5 Small (center) and T5 Base (bottom) models for each of the 5 evaluation datasets. Even though we present a wide variety of benchmark results, from short-form factoid questions to long-form answers, the CCQA seq2seq pre-trained model consistently outperforms all other models on the zero-shot question-answering task. Even more importantly, the additional in-domain pre-training step achieves better zero-shot performance than fully fine-tuned, randomly initialized transformer models (as extensively used prior to 2018) in almost all settings. Specifically, our model outperforms the randomly initialized transformers on all benchmarks for T5 Small and T5 Base, as well as on 4 out of 5 datasets using BART Large.

Comparing the fully fine-tuned setting across models and datasets it becomes clear that, although oftentimes performing comparably, our CCQA seq2seq pre-trained model underperforms the vanilla models in most cases. Seq2seq in-domain pre-training on CCQA only reaches su-

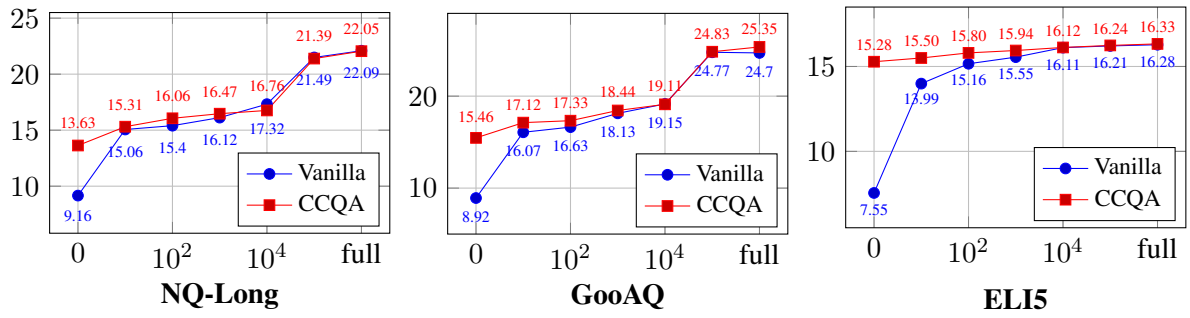

Figure 4: Low resource experiments comparing the Rouge-L score of vanilla T5 Small with our CCQA pre-trained models on NQ-long (left), GooAQ (center) and ELI5 (right).

perior performance on the ELI5 dataset for all models, as well as on the GooAQ dataset for T5 Small. Showing that seq2seq pre-training on CCQA is effective in zero-shot scenarios, however only partially improves over baselines in the fine-tuned setting, we investigate: (1) Additional experiments using the CCQA dataset for denoising-style pre-training (*-d* in Table 3) and (2) Evaluate additional low-resource scenarios, shown in Figure 4.

For our denoising-style in-domain pre-training experiment, we keep the available markup information, in line with HTLM (Aghajanyan et al., 2021b). As shown in Table 3, the in-domain CCQA denoising objective outperforms the vanilla BART Large model on 4 out of 5 benchmarks in the fine-tuned setting. We believe that this result, alongside the zero-shot performance of the seq2seq CCQA model, clearly shows the usefulness and generality of our CCQA corpus for closed-book open-domain question-answering.

Taking a closer look at low-resource scenarios, we evaluate the vanilla T5 Small model against our in-domain pre-trained approach using 5 proper subsets of the NQ-Long, GooAQ and ELI5 benchmark datasets, drawn at random. As presented in Figure 4, our CCQA model mostly outperforms the vanilla T5 Small model in low-resource scenarios with up to $10,000$ data points. While the performance of our CCQA model is consistently better on the ELI5 test-set, the vanilla baselines outperform our models fastest on the NQ-Long corpus. Additional low-resource experiments on T5 Base are shown in Table 6, in Appendix G.

4.2 Passage Retrieval

4.2.1 Task

For the passage retrieval task, an important component of most open-book QA systems (e.g., Lewis et al. (2020); Izacard and Grave (2021)), models

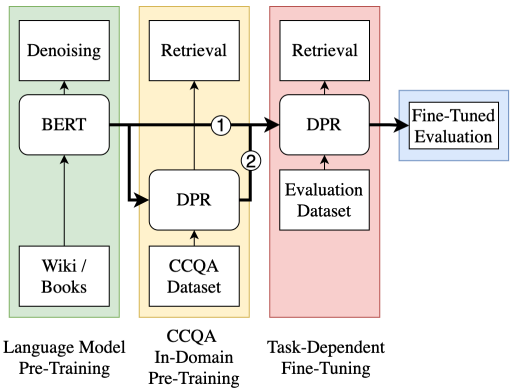

Figure 5: High-level overview of the CCQA passage retrieval in-domain pre-training step (yellow) as part of the training pipeline for DPR. Language model pre-training shown in green. Task-dependent fine-tuning presented in red. Evaluation in blue. (1) Baseline pre-training/fine-tuning pipeline, (2) In-domain pre-training/fine-tuning pipeline.

aim to extract a set of evidence passages from a large collection of documents through conditional ranking. To align our corpus with the passage retrieval task, we aggregate every question into a single data point, consisting of the question itself, alongside all available answers as either positive or negative contexts. If available, answer votes are used as a proxy to determine positive and negative (sometimes called "hard-negative") contexts. Following the practice in Fan et al. (2019), we assign every answer with at least 2 more upvotes than downvotes as a positive context and all other answer as negative. If answer votes are not available, we use the accepted/suggested label (shown in Figure 2) as an indicator for positive and negative contexts. In the absence of either criterion, we use all available answers as positive contexts.

4.2.2 Models & Training

For passage retrieval, we choose the Dense Passage Retriever (DPR) (Karpukhin et al., 2020), used in a variety of popular end-to-end open-book QA mod-

Model	TQA		NQ-Short	
	Acc@20	Acc@100	Acc@20	Acc@100
DPR	79.4	85.0	78.4	85.4
DPR v2	79.5	85.3	78.3	85.6
CCQA DPR	**80.0**	**85.6**	**79.1**	**86.3**

Table 4: Fine-tuned Dense Passage Retriever (DPR) accuracy measure on the TQA and NQ-Short datasets. DPR represents the original DPR model (Karpukhin et al., 2020), DPR v2 (Oğuz et al., 2021) indicates the updated codebase. CCQA DPR uses our CCQA pre-trained DPR model for retrieval fine-tuning.

Bench. (test)	TQA	NQ-S	NQ-L	ELI5	GooAQ
Bench. (train)	11.9	4.9	5.2	3.0	26.9
CCQA (train)	0.4	1.9	2.3	0.5	26.9

Table 5: 8-gram question overlap (in %) between training sets and benchmark test-sets (inspired by Radford et al. (2019)). *Bench (train)* refers to the overlap between the respective training- and test-portion of the benchmark datasets, *CCQA (train)* identified overlaps between our dataset and the test-splits. False positive rate upper-bound by $\frac{1}{10^8}$. All inputs are normalized and lower-cased. NQ-S=NQ-Short, NQ-L=NQ-Long.

els, such as RAG (Lewis et al., 2020) and FiD (Izacard and Grave, 2021). As shown in Figure 5, we start with the vanilla DPR model based on BERT (Devlin et al., 2018) and in-domain pre-train using questions and positive/negative passages from the CCQA dataset (yellow box in Figure 5), similar to Oğuz et al. (2021). In line with the training-flows of the closed-book models, we train DPR using either the vanilla setup (pre-training $\rightarrow$ fine-tuning) or the in-domain pre-training approach (pre-training $\rightarrow$ in-domain pre-training $\rightarrow$ fine-tuning), shown as circles (1) and (2) in Figure 5, respectively.

4.2.3 Datasets & Metrics

Following the original DPR paper (Karpukhin et al., 2020), we evaluate the passage retrieval task on the NQ-Short and TQA datasets presented in section 4.1.3, using the top-20 and top-100 retrieval accuracy (Acc@20/Acc@100) measures.

4.2.4 Hyper-Parameters

We use the default DPR hyper-parameters whenever possible (Karpukhin et al., 2020). For in-domain pre-training, we limit training to $800k$ steps using a batch-size of $1,536$ samples. During fine-tuning, we restrict the number of updates to $20k$ steps with a batch-size of 128. The best checkpoint is selected based on the Mean Reciprocal Rank (MRR) measure, following Oğuz et al. (2021). We do not perform any hyper-parameter search.

4.2.5 Results

For the passage retrieval experiments, we compare our CCQA in-domain pre-trained DPR model against the vanilla DPR model published in Karpukhin et al. (2020), as well as the recently enhanced version (Oğuz et al., 2021). Table 4 contains our empirical results, showing consistent improvements of our CCQA DPR model over the vanilla baselines. More specifically, the in-domain CCQA pre-training step increases the top-20 and top-100 accuracy score on the TQA benchmark dataset by over half a point, while the performance gap on NQ-Short shows consistent improvement of over 0.7%.

4.3 Evaluation Fairness: Dataset Overlap

With modern pre-training approaches using increasingly large datasets, accidental overlaps between pre-training corpora and benchmark datasets become more and more common (Lewis et al., 2021a). To analyze this threat to the integrity of our dataset and empirical analysis, we follow Radford et al. (2019) and evaluate the 8-gram question overlap of our CCQA training portion with the test-split of benchmark datasets using bloom filters. Table 5 shows a consistently smaller question overlap between CCQA and the benchmark test set, compared to the benchmark training split itself.

5 Conclusion and Future Work

In this work, we presented our new web-scale CCQA dataset for in-domain model pre-training. We started by showing the generation process, followed by detailed insights into key dataset dimensions of this new, large-scale, natural, and diverse question-answering corpus. In a set of empirical evaluations, we confirmed the initial intuition that the dataset presents a valuable resource for open-domain question-answering research. In our zero-shot, low-resource and fine-tuned experiments for open- and closed-book QA tasks, we show promising results across multiple model architectures. With around 130 million question-answer pairs (55M unique) as well as additional meta-data, our CCQA dataset presents a versatile source of information, which has a large variety of applications in future work (e.g., question summarization, answer rating, answer ranking and many more).

References

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. 2021a. Muppet: Massive multi-task representations with pre-finetuning. *arXiv preprint arXiv:2101.11038*.

Armen Aghajanyan, Dmytro Okhonko, Mike Lewis, Mandar Joshi, Hu Xu, Gargi Ghosh, and Luke Zettlemoyer. 2021b. Htlm: Hyper-text pre-training and prompting of language models. *arXiv preprint arXiv:2107.06955*.

Wei-Cheng Chang, Felix X Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training tasks for embedding-based large-scale retrieval. *arXiv preprint arXiv:2002.03932*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

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*.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. Eli5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*.

Gautier Izacard and Édouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880.

Robin Jia, Mike Lewis, and Luke Zettlemoyer. 2021. Question answering infused pre-training of general-purpose contextualized representations. *arXiv preprint arXiv:2106.08190*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016a. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016b. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single qa system. *arXiv preprint arXiv:2005.00700*.

Daniel Khashabi, Amos Ng, Tushar Khot, Ashish Sabharwal, Hannaneh Hajishirzi, and Chris Callison-Burch. 2021. Gooaq: Open question answering with diverse answer types. *arXiv preprint arXiv:2104.08727*.

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.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096.

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, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401*.

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. 2021a. Question and answer test-train overlap in open-domain question answering datasets. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1000–1008.

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

Barlas Oğuz, Kushal Lakhotia, Anchit Gupta, Patrick Lewis, Vladimir Karpukhin, Aleksandra Piktus, Xilun Chen, Sebastian Riedel, Wen-tau Yih, Sonal Gupta, et al. 2021. Domain-matched pre-training tasks for dense retrieval. *arXiv preprint arXiv:2107.13602*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Édouard Grave. 2020. Ccnet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4003–4012.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Wenhan Xiong, Xiang Lorraine Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Wen-tau Yih, Sebastian Riedel, Douwe Kiela, et al. 2020. Answering complex open-domain questions with multi-hop dense retrieval. *arXiv preprint arXiv:2009.12756*.

A CCQA Dataset Generation Algorithm

Algorithm 1 CCQA Dataset Generation Procedure

```
for document ∈ CommonCrawl do
    if "schema.org/Question" in document then          ▷ Webpage contains schema.org annotation
        tree ← parse_html(document)
        questions ← find_question_root(tree)
        for question_sub_tree in questions do
            question_sub_tree ← clean_question_subtree(question_sub_tree)
        end for
        questions ← convert_to_json(questions)
        save(questions)
    else
        skip document
    end if
end for

procedure FIND_QUESTION_ROOT(node)          ▷ Pre-order traversal, return when question found
    if node.itemtype == "https://schema.org/Question" then
        return node
    end if
    for child in node.children() do
        node ← find_question_root(child)
        nodes.append(node)
    end for
    return nodes
end procedure

procedure CLEAN_QUESTION_SUBTREE(node)      ▷ Post-order traversal, clean elements bottom-up
    for child in node do
        child ← clean_question_subtree(child)
    end for
    if "itemtype" | "itemprop" in node.attributes() then
        for attribute in node.attributes() do
            if not attribute.starts_with("item" | "content" | "date") then
                attribute.remove()
            end if
        end for
    else
        replace_node_with_children(node)
    end if
end procedure
```

B Dataset Format

The structured output of the dataset collection is shown in Figure 2, containing a three-level nested structure: (1) Every top-level data point represents a webpage in Common Crawl, encapsulating questions and answers found on the page, together with relevant metadata. (2) On the second level of the nested structure, every question is represented as a tuple containing the question-name (a short summary of the question) and -text (the main question). Questions also contain additional metadata as shown in Figure 2. (3) Every question can contain an arbitrary number of associated answers and answer attempts, located on the third and final level of the nested structure. An answer thereby contains a mandatory accepted/suggested label, the answer text as well as optional metadata.

With this nested structure of our CCQA dataset, we allow users to: (1) Verify question-answer pairs and their metadata on the original webpage, (2) utilize additional parts of the original webpage and (3) tackle question-answering related tasks, such as answer selection, answer rating or answer ranking.

C Detailed Topic Distribution

Topic	Top 5 Appearances in CCQA				
Top-Level	Regional (38.90%)	Society (21.14%)	Business (8.36%)	Sports (7.04%)	Rec. (6.20%)
Regional	North America (61.48%)	Europe (34.69%)	Asia (1.28%)		
Society	Issues (76.89%)	Religion (18.39%)	Philosophy (2.36%)	Law (1.41%)	
Business	Industrial Goods (13.41%)	Energy (9.75%)	Textiles (9.75%)	Construction (7.31%)	Business Services (6.09%)
Sports	Golf (81.08)	Aquatiques (10.81%)	Events (2.70%)	Water Sports (2.70%)	Lacrosse (1.35%)
Recreational	Food (56.92)	Outdoors (23.07%)	Travel (12.30%)	Motorcycles (3.07%)	Pets (1.53%)

Table 6: Fine-grained CCQA dataset topic distribution of 1000 randomly chosen domains retrieved through the DMOZ/Curlie annotation at https://curlie.org/. Only showing sub-topics with $\geq 1\%$.

D Detailed Question Word Distribution

Question-Word	What	How	When	Which	Where	Why	Who	Whose
Frequency	5.3M (36.20%)	4.3M (29.80%)	1.4M (9.68%)	1.4M (9.64%)	881k (6.04%)	717k (4.92%)	514k (3.53%)	25k (0.17%)

Table 7: Question word distribution for all 8 English question words with their number of appearance in the CCQA corpus and their relative frequency.

E HTML Markup Tag Distribution

Rank	HTML Markup Tag Distribution				
1-5	p (28.48%)	a (14.89%)	br (14.87%)	li (10.04%)	span (5.77%)
6-10	strong (4.93%)	code (4.59%)	em (2.79)	div (2.38%)	ul (2.27%)
11-15	pre (1.80%)	b (1.70%)	blockquote (1.14%)	h3 (0.89%)	td (0.88%)
16-20	h2 (0.48%)	ol (0.42%)	tr (0.42%)	h1 (0.35%)	i (0.24%)
21-25	sup (0.17%)	tbody (0.12%)	table (0.12%)	u (0.12%)	sub (0.11%)

Table 8: Distribution of the 25 most common HTML tags in CCQA.

F Sensibility and Answerability Examples

Metric	Type	Example	Explanation
Q-sensibility	Pos	What languages do you speak?	Q-Sensible, since question internally makes sense
	Neg	How blue is the number 7?	Not Q-Sensible, since question internally makes no sense
Q-answerability	Pos	How can I purchase affordable Flats in Vancouver?	Q-Answerable, since a single answer exists
	Neg	What languages do you speak?	Not Q-Answerable, since no single answer exists, but depends on the (unavailable) context
QA-sensibility	Pos	Which is the busiest month to travel from London to Delhi? $\rightarrow$ July	QA-Sensible, since question and answer make sense together
	Neg	How can I purchase affordable Flats in Vancouver? $\rightarrow$ There are many affordable Flats available.	Not QA-Sensible, since answer does not answer the question

Table 9: Examples and explanations for Question-sensibility (Q-sensibility), Question-answerability (Q-answerability) and QA-sensibility. Pos = Positive example, Neg = Negative example.

G Full Set of Low Resource Experiments

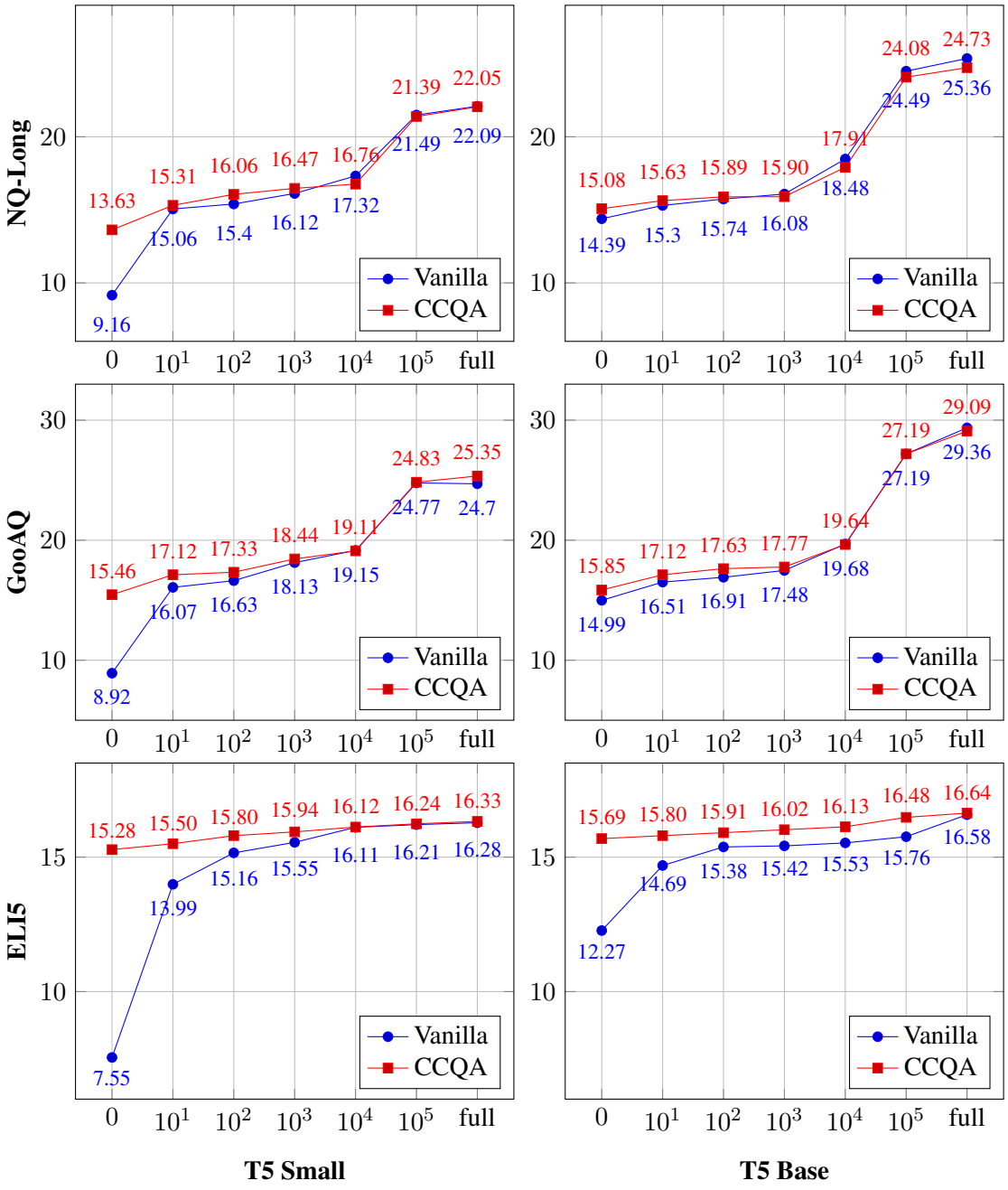

Figure 6: Low resource experiments comparing the Rouge-L score of vanilla T5 Small (left) and T5 Base (right) with our CCQA pre-trained models on NQ-long (top), GooAQ (center) and ELI5 (bottom).

H Qualitative Dataset Examples


```
{
"Language":"-",
"Fasttext_language":"en",
"URI":"https://www.geograph.ie/faq3.php?q=multiple+account",
"UUID":"a5e97da2-f688-42af-8626-73a38fa8d06f",
"WARC_ID":"CC-MAIN-20201026031408-20201026061408-00221",
"Questions":[
      {
         "name_markup":"Can I change my name to a <b>pseudonym</b> on
a submission ?",
         "Answers":[
            {
               "text_markup":"You can submit all your
photos under a pseudonym by changing the name on your
Profile<span><a>http://www.geograph.org.uk/profile.php</a></span>(link
top write on most pages). Note that by doing this, the name will be
changed on all photos you have previously submitted from the account.
These may already have been used elsewhere, crediting the name
originally shown. <br> You can change the credit on an individual
image, for instance if you asked someone else to take it for you,
but the name on your profile will still be shown on the photo page
and the photographer name will still link back to your profile. <br>
You can open another account under a pseudonym but this will need
to be done from a different email address and you will have to take
care which account you are signed in with before submitting, making
changes or posting in the forums.",
               "status":"acceptedAnswer"
            }
         ]
      }
   ]
}
```

```
{
"Language":"en-US",
"Fasttext_language":"en",
"URI":"https://www.catholicfaithstore.com/Store/Products/SKU/b0d/
       St-Olgas-Cross-Medal.html",
"UUID":"94def557-e521-493a-babd-b63c5e030e62",
"WARC_ID":"CC-MAIN-20210308174330-20210308204330-00337",
"Questions":[
      {
         "name_markup":"How do I care for my sterling silver?",
         "Answers":[
            {
               "text_markup":"<p>Sterling Silver Cleaning
Instructions</p><ul><li>NEVER use a sterling silver cleaning
solution on your jewelry. It will take off the protective
coating.</li><li>Take a half cup of warm water and a few drops of
mild dishwashing liquid soap and mix together.</li><li>With a soft
clean cotton cloth dip the cloth into the soapy water getting
it moist.</li><li>Use the moist cloth to wipe the surface of your
sterling silver jewelry.</li><li>Take the just cleaned jewelry
and run under clear water for a few seconds to wash away any
soap.</li><li>Allow jewelry to dry before storing</li></ul><p>Other
things to remember: When not wearing your sterling silver jewelry,
keep it in an air-tight container or zip lock bag. Avoid household
clean products getting in contact with the jewelry. And take off your
jewelry when you swim, shower or are washing dishes.</p><p>For a more
detailed explanation see<a>5 Easy-To-Follow Steps for Cleaning Your
Sterling Silver Jewelry</a></p>",
               "status":"acceptedAnswer"
            }
         ]
      }
   ]
}
```

```
{
"Language":"-",
"Fasttext_language":"en",
"URI":"https://quant.stackexchange.com/questions/39510/
       software-for-american-basket-option-pricing-using-longstaff
       -schwartz-least-squar",
"UUID":"e059deaf-3d73-4517-88a0-8abb8ad74972",
"WARC_ID":"CC-MAIN-20210305183324-20210305213324-00585",
"Questions":[
      {
         "author":"Bananach",
         "name_markup":"<a>Software for American basket option
pricing using Longstaff-Schwartz/Least Squares Monte Carlo
method</a>",
         "text_markup":"<p>Is there free software (preferably
in Python) that computes American basket (high-dimensional!)
option prices in the Black Scholes model using the
Longstaff-Schwartz algorithm (also known as Least Squares Monte
Carlo)?</p>~<p>Optimally, I want to be able to control the number
of basis functions, the number of Monte Carlo samples and the number
of time steps used.</p>",
         "date_created":"2018-04-30T09:16:33",
         "upvote_count":"1",
         "answer_count":"1",
         "Answers":[
            {
               "author":"byouness",
               "text_markup":"<p>QuantLib is what
you are looking for. It is free/open source library
written in C++, it is available in Python as well (via
SWIG):<a>https://www.quantlib.org/install/windows-python.shtml
</a></p>~<p>Examples are shipped with QuantLib and among
them some show how to price options.</p><p>To get a feel
for what it's like, you can check this blog post, explaining
how to price an American option on a single asset using a
binomial tree in Python:~<a>http://gouthamanbalaraman.com/blog/
american-option-pricing-quantlib-python.html</a></p>",
               "status":"acceptedAnswer",
               "upvote_count":"1",
               "comment_count":"1"
            }
         ]
      }
   ]
}
```

```
{
"Language":"en",
"Fasttext_language":"en",
"URI":"https://wwwmybizpro.invoicera.com/expense-management.html",
"UUID":"8cfe986c-4f33-4a2a-98f1-32aab3811533",
"WARC_ID":"CC-MAIN-20210512100748-20210512130748-00544",
"Questions":[
      {
         "name_markup":"Do I need any new IT infrastructure to get
the best use out of this software?",
         "Answers":[
            {
               "text_markup":"NO! Invoicera simply integrates with
your current ERP and CRM. It comes with the simplest self-explanatory
user-interface for you to use. You do not need any extra guidance
with your Invoicera.",
               "status":"acceptedAnswer"
            }
         ]
      }
   ]
}
```