Evaluating and Training Long-Context Large Language Models for Question Answering on Scientific Papers

Lukas Hilgert **Danni Liu** **Jan Niehues**
Karlsruhe Institute of Technology, Germany
{lukas.hilgert, danni.liu, jan.niehues}@kit.edu

Abstract

With the number of scientific papers published every year growing and current large language models (LLMs) showing state-of-the-art performance on natural language processing (NLP) tasks, we ask the question if LLMs could be utilized to answer questions on scientific papers. We investigate how well state-of-the-art large language models (LLMs) can answer questions on scientific paper by experimenting with long-context versions of the LLaMA 2 model and evaluating and training on the Qasper dataset. We analyze how well the LLMs handle longer papers and questions that can only be answered by accessing information from far out paragraphs. During our experiments, we see that the performance of these LLMs drops with growing length and position of relevant information. We employ different measures from simple prompts to chain-of-thought prompts and zero-shot usage to fine-tuning with QLoRA. While we still observe a performance loss with increased context length, our measures reduce the effects of this flaw, and we can achieve F_1 scores similar to bigger models like GPT-4.

1 Introduction

The number of scientific papers published every year is growing exponentially (Fire and Guestrin, 2018). This creates a problem for scientists but also the general public to keep up with the developments in science. A natural language processing (NLP) system that can reliably answer questions on scientific papers could help in this situation. Question answering (QA) systems often rely on task-specific machine learning models that can only be used for this purpose. Large Language Models (LLMs) are a newer type of deep learning model trained to be general-purpose models for NLP. Current commercial and open-source LLMs are often used in an intuitive, conversational manner as chatbots. They offer the ability to answer follow-up questions and have an intuitive interface for most users. They show state-of-the-art (SOTA) NLP performance (OpenAI, 2023; Anil et al., 2023) and even display some reasoning capabilities (Brown et al., 2020; Chowdhery et al., 2023; OpenAI, 2023; Anil et al., 2023) and would be one contender for the core of a QA system focused on scientific papers.

Scientific papers present great challenges as context for QA when using LLMs for multiple reasons: Their text part is typically about around 4,000 to 13,333 tokens long assuming that one word amounts to around $1.\overline{3}$ tokens (Björk et al., 2009; OpenAI, b). The base versions of newer commercial models like GPT-3.5 and GPT-4 have context windows of 4,096 (OpenAI, a) and 8,192 (OpenAI, a) tokens while open-source LLMs like LLaMA 2 (Touvron et al., 2023b) offer a 4,096 long context window. Also, scientific papers consist of long unstructured (except sectioning etc.) raw text making, it hard to determine which part is important to answer the question. The answer type is also not clear as the question could be about explaining some concepts presented in the paper, simple facts or even yes or no questions, or the question could be unanswerable. The unstructuredness and length of the context is especially problematic even for long-context LLMs as Liu et al. (2023) found: For multi-document QA where the LLM has to select the relevant context part from multiple options, the performance curve has a U-shape with respect to the position of the documents as the ones at the beginning and the end are better retrieved than those in the middle.

In this paper, we evaluate how well a small open-source LLM can perform as a QA system for scientific papers if used in a zero-shot manner – especially regarding long papers (>4k tokens) and those questions whose relevant paragraphs are far out token-wise. To do this, we bin the papers per length and the questions per position of the relevant paragraphs. We try to improve the performance using recent LLM adaptation techniques (prompting,

Proceedings of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U), pages 220–236
November 16, 2024 ©2024 Association for Computational Linguistics

parameter-efficient fine-tuning). We also investigate what weaknesses (e.g., instruction-following, long-context understanding) of the models specifically the fine-tuning improves. Finally, we compare our best model with bigger ones. We observe that increased context length and position of the relevant paragraphs result in worse performance even for long-context LLMs. While more sophisticated prompting does not help, fine-tuning increases overall performance significantly but mostly by improving the instruction-following of the LLMs.

2 Related Work

Large Language Models The foundation for most current Large Language Models (LLMs) like the Gemini (Anil et al., 2023), GPT (Brown et al., 2020; OpenAI, 2023), LLaMA (Touvron et al., 2023a,b), and Mistral (Jiang et al., 2023, 2024) families is pre-training Transformer decoder-only models with billions of parameters on Internet-scale data enabling them to perform tasks they were not explicitly trained on. As we want to experiment on LLMs themselves which includes fine-tuning and modifying them, we utilized available open-source models: Large Language Model Meta AI (Meta AI, 2023) (LLaMA) (Touvron et al., 2023a,b). Following the work on training-compute-optimal LLMs (Hoffmann et al., 2022), the authors of LLaMA focus on training smaller models with more data (and more compute) to achieve better inference-compute efficiency.

Vicuna is a collection of fine-tuned LLaMA models (Chiang et al.; Zheng et al., 2023). On top of models fine-tuned for better chatbot performance, there are models with longer context windows than the original LLaMA model (version 2: 4k (Touvron et al., 2023b)) with up to 32k tokens using a technique similar to Positional Interpolation developed independently (Ken; Li et al.). Chen et al. (2023) propose Positional Interpolation (PI) to easily increase the context window length: Stretching the original context window (L) to the new maximum length L' by downscaling the position indices that are the input to the positional encoding function.

Question Answering Task The type of context for Question answering (QA) can differ as it may be present as knowledge or as harder to manage raw text. Modern QA system mostly use deep learning-based models like (fine-tuned) BERT- or GPT-style models. Datasets that cover the topic of scientific papers focus on various aspects. Many focus on the review process which yields different artifacts. These enable different tasks: (Meta-) Review Generation (Wang et al., 2020; Lin et al., 2023), acceptance prediction / paper rating (Kang et al., 2018; Yang et al., 2018), Argument Pair Extraction from reviews and corresponding rebuttals (Cheng et al., 2020), and Multi-document Summarization on reviews (Li et al., 2022). But there are also datasets specifically for question answering on scientific papers (Dasigi et al., 2021).

Evaluating Long-Context Text Processing To make comparison of long-context LLMs easier, multiple benchmarks sets have been created to test their abilities across different task types. ZeroSCROLLS (Shaham et al., 2023) is a benchmark focused on long text understanding in a zero-shot setting. The included task types are summarization, question answering and aggregation. A similar benchmark called LongBench also includes Qasper (Bai et al., 2023). Opposed to ZeroSCROLLS it is bilingual and incorporates more task types. Also, the authors showed the performance of the models that they tested for context lengths of 0 – 4k, 4k – 8k and 8k+ tokens individually. They only investigated zero-shot prompting and they did not show how the position of the important information within the long context affects performance.

Han et al. (2024) presented with LM-Infinite a technique to increase the ability of LLMs to handle long-context without any parameter updates. However, their evaluation on Qasper showed only small improvements over their truncation baseline (30.1 vs. 31.3) and did not contain fine-grained analysis on Qasper.

3 Methodology

To improve the performance of the general-purpose LLMs on the task of QA on scientific papers, we apply different prompting techniques and fine-tuning. We list all our prompt templates in Appendix A.

3.1 Approaches

Simple Prompt Zero-shot prompting is a straightforward approach, where the LLM is directly used out-of-the-box at inference time. Although few-shot prompting in general improves performance (Brown et al., 2020), the long input size in our case precludes this approach. Therefore, we have to resort to zero-shot prompting which only includes the instruction for the model as a kind of learning signal. However, this generally leads to weaker instruction-following abilities. This approach with a simple prompt serves as the baseline for the other methods (using the same model).

Extract-then-Answer Prompt Chain-of-thought prompting (Wei et al., 2022) showed that splitting a task into subtasks can help LLMs to solve them. Inspired by this, we split the question answering into two tasks: First the model has to find the evidence – all relevant paragraphs to answer the question. After that we prompt it to answer the question based on the extracted paragraphs in the previous step. Extracting the relevant paragraphs is a useful task on its own: It could be useful to see the context of the answer inside the paper and improve interpretability. There are also some downsides: We have to run inference twice as this approach requires the model to generate its input for the second step. Also, as the model generates its own input (apart from the second prompt), this approach may lead to cascading errors. Similar approaches were investigated for science QA on short context (Lu et al., 2022; Wang et al., 2023; Yoran et al., 2023), for (Chinese) multi-document QA (He et al., 2023), and on smaller-scale models prior to the emergence of LLMs (Dasigi et al., 2021).

Supervised Fine-tuning We can fine-tune the LLM on supervised data with the simple prompt and the extract-then-answer prompt. For the latter, we fine-tune the model two subtasks: Evidence extraction and answer generation given evidence. By combining compute- and memory-efficient methods of implementing and training Transformer-based models, we are able to fine-tune a small LLM on long context. We replace the standard attention algorithm with FlashAttention 2 (Dao et al., 2022; Dao, 2023) and we use QLoRA (Hu et al., 2022; Dettmers et al., 2023) for fine-tuning the model.

3.2 Evaluation

Besides the standard evaluation of QA quality provided by the dataset authors, we conduct various fine-grained analyses to evaluate our approaches regarding our specific focus.

3.2.1 Analysis by Context Length / Position

In addition to evaluating QA quality, we want to evaluate per paper length and absolute evidence position. We therefore split the evaluation data into (partially) overlapping groups by the length / distance in tokens.

Paper Length We want to find out if long-context modifications enable models to process longer context as well as context within the original context window or if the performance differs per paper length. Here, we bin per paper as the length is the same for all associated questions. We count the number of tokens to get the length.

Evidence Position It is also important to find out if the position of the relevant information ("evidence") within the paper which is also provided by the dataset does affect performance. We will study the impact of the absolute token position of the evidence. For "Unanswerable" and some yes/no questions there is no evidence, we put these questions into a separate bin ("No evidence"). In contrast to the length binning, we group the evaluation data per question as the evidence positions differ in general per question and not per paper.

3.2.2 Evidence-only Prompt

We want to find out how our investigated models perform if we provide them with the evidence only – both during inference and training. This should give use an idea of the upper limits of the performance of the models as this task should be easier as the model has to process fewer tokens. Additionally, we think that a comparison between these fine-tuned models and those that received the full paper during training should indicate how much our fine-tuning improves our goal of long-context understanding and how much it just improves instruction following.

4 Experiments and Results

4.1 Experimental Setup

In the following, we will describe our experimental setup. We list utilized hard- and software and the hyperparameters we used during inference and training in Appendix C.

4.1.1 Dataset

The Qasper dataset (Dasigi et al., 2021) we used to evaluate and train the considered models consists of a total of 1,585 NLP papers with 5,049 questions on these papers. Each of these questions was formulated by an NLP practitioner. The answers were then answered by other NLP practitioners who also selected the paragraphs, figures or tables ("evidence") in the paper that are relevant to answer the question which are listed together with the

Models	dev-short		dev		test		ZC	
Questions / %	990	100	1,005	100	1,451	100	500	100
Paper length								
0k – 4k	333	34	333	33	511	35	149	30
4k – 8k	593	60	593	59	802	55	312	64
8k –	64	6	79	8	138	10	39	8
Absolute evidence position								
0k – 4k	794	80	799	80	1182	81	405	81
4k – 8k	173	17	180	18	263	18	91	18
8k –	6	1	11	1	18	1	7	1
No evidence	77	8	78	8	99	7	37	7

Table 1: Qasper dataset statistics we created for our research questions: paper length and absolute evidence position; the numbers for absolute evidence position exceed the total number of questions because the evidence for a question can be from multiple paragraphs. ZC refers to the subset of the Qasper test set used in the ZeroSCROLLS benchmark.

Q. type	Frequency								
Bin type	Full	dev-short	Length			Absolute evidence position			
Specific bin			0k – 4k	4k – 8k	8k –	0k – 4k	4k – 8k	8k –	No ev.
Extractive	51.8%	54.8%	53.3%	56.5%	47.4%	58.5%	55.1%	45.5%	0.0%
Abstractive	24.2%	24.3%	21.3%	25.5%	28.1%	25.9%	30.2%	27.3%	0.0%
Yes/No	13.9%	11.6%	13.8%	10.2%	13.2%	10.7%	11.8%	18.2%	24.6%
Unanswer.	10.2%	9.3%	11.6%	7.8%	11.4%	4.9%	3.0%	9.1%	75.4%

Table 2: Qasper dataset statistics (full dataset (full), (Dasigi et al., 2021)) and ours: question types for each dataset bin (all bins are from dev-shot)

gold answer in the dataset. There are four types of questions / answers in this dataset:

- Extractive: questions can be answered by copying chunks of the relevant paragraph
- Abstractive: free text answers that are not literally in the paper
- Yes/no or boolean questions
- Unanswerable: questions that can not be answered with the provided paper as context.

These question types appear in different frequencies (Table 2) and the authors evaluated the performance of their model for each question type individually. The dataset website[1] provides an official evaluation script. Like for the SQuAD dataset (Rajpurkar et al., 2016), the authors chose a span-level F_1 score as their metrics. If there are multiple reference answers, the maximum of the F_1 score will be used.

For the final analysis, we use a subset of the Qasper test split that is part of the ZeroSCROLLS (ZC) benchmark (Shaham et al., 2023). We saw a similar statistic for this subset as for the (custom) splits we used during development and final analysis. We therefore assume that the ZC subset of Qasper will be representative for the performance of our approaches.

4.1.2 Data Preprocessing

Five of the papers from the development / validation split of Qasper lead to out-of-memory errors during inference. We therefore exclude these five papers from our results and call the resulting split "dev-short". As these five only account for around 1.8% of the 281 papers in the dev split, we assume that this does not skew our view of the quality of the models. Also, the distribution of the length / position bins is not changed much (Table 1).

We make a similar observation for the binning itself (Table 2): The distribution of the questions types does not vary much between the length / position bins (with exception of the one for questions with no evidence). We therefore assume that our analysis of the models by binning the dataset does reflect the performance of the model for that spe-

[1] https://allenai.org/data/qasper

cific length / evidence position and is not influenced by the distribution of the questions type in that specific bin.

As training data, we use the training split of the Qasper dataset. As input, we use a prompt template (subsubsection A.1.1) from the LongBench benchmark dataset (Bai et al., 2023) where the paper text and the question are inserted the same way as for zero-shot prompting. The target is the answer from the dataset. As our tested models have text as their only modality, it cannot process the figures and tables provided with the dataset. We therefore remove all questions from the training data that mention figures or tables in their evidence field. Many questions are annotated with multiple possible answers. In some cases, they clearly heavily disagree with each other e.g., one possible answer is “Unanswerable” and the other is “Yes” or “No”. We remove these cases. We also have to limit the training data to texts with a maximum of 8k tokens as longer inputs cause out-of-memory errors even with both QLoRA and FlashAttention used.

4.1.3 Models

We use three models with different context window lengths in our experiments. The creators of FastChat (LMSYS Org) provide the Vicuna family (section 2) of LLMs. We only test the smallest available models with around 7 billion parameters for compute and memory efficient experiments and as this is the only model size that has a LongChat version. This version has a context window of 32k tokens (LC-32k). Vicuna 7B-4k (V-4k) has the same as LLaMA 2 (4k) and Vicuna 7B-16k's (V-16k) was extended to 16k. We use the models of version v1.5 which indicates that they are based on LLaMA 2 instead of LLaMA 1 like the previous versions.[2] The fine-tuning data was 370M tokens long. We omit the parameter count in the following from the models' names as they are the same of every model we tested.

4.2 Results and Discussion

We start our experiments with all available small (7B parameters) models from LMSYS Org with varying context window lengths: Vicuna-4k, Vicuna-16k, and LongChat-32k. Here, we only report the results for LongChat-32k as it showed the best long-context performance and show the others in Appendix D and Appendix E. During our experiments, we also investigated the performance by relative evidence position. However, we saw no U-shape of the performance and therefore do not include these results. This corresponds to prior work (Liu et al., 2023) which found this strong primary and recency bias only in large (>7B) models.

4.2.1 Simple Prompt

First, we run a simple zero-shot prompt and report the results in the first two columns of Table 3.

Simple zero-shot prompt struggles with unanswerable questions While LongChat is able to answer the “normal” questions, it seems to be unable to handle unanswerable questions (Table 3). These questions can not be answered with the given paper. Also, its ability to answer yes/no questions is limited. Qualitative analysis showed that LongChat almost never outputs “Unanswerable” and even if it does, the answer is a whole sentence which ignores the instruction in the prompt (examples: Appendix B).

Longer context leads to worse performance Fine-grained analysis by input length shows that after the threshold of 4k tokens, the performance begins to decrease from an F_1 score of 25.47 to 24.08 for papers with a length between 4k and 8k tokens. After 8k tokens this decrease accelerates (18.51) and is especially visible when binning the F_1 score by evidence position (F_1: 26.73 → 23.35 → 15.06). The model also especially struggles with questions that require no evidence (most of them are unanswerable). We assume that the lower F_1 score of LongChat on papers with more than 8k tokens is a result of this weakness and not a general property.

Fine-tuning: Trade-offs between generation and classification As the empirical results showed that LongChat had insufficient instruction following, we now want to see how much fine-tuning can increase the performance. Also, we want to find out how much it improves the F_1 scores for long papers and evidence at high token positions. The impact of QLoRA fine-tuning on LongChat-32k (Table 3) is that extractive, boolean and unanswerable questions substantially improve (F_1: 26.51 → 48.21, 36.79 → 76.47, 0.04 → 68.54). We assume that the F_1 scores for unanswerable questions do not improve after the first epoch because it reached the highest scores possible with this model size

[2] https://github.com/lm-sys/FastChat/blob/97065ff7caa3ae4ca28c661b7424f7ae4cca539b/docs/vicuna_weights_version.md

Training Variation	0S 1S	FT 1S	0S 2S	FT 2S	0S 2S+	FT 2S+
Answer F_1	24.19	47.02	24.94	39.08	17.85	41.18
Answer F_1 by type						
Extractive	26.51	48.21	23.19	37.50	16.37	41.82
Abstractive	20.78	20.10	17.35	14.92	16.03	19.41
Boolean	36.79	76.47	57.96	49.51	36.75	58.10
Unanswerable	0.04	68.54	11.84	89.09	5.33	69.23
Answer F_1 per paper length						
0k – 4k	25.47	52.15	25.68	41.57	19.26	44.78
4k – 8k	24.08	44.45	24.85	37.77	17.50	39.66
8k –	18.51	44.09	21.97	38.23	13.83	36.44
Answer F_1 per absolute evidence position						
0k – 4k	26.73	46.28	26.84	36.00	18.89	40.54
4k – 8k	23.35	37.74	23.40	30.31	16.64	34.97
8k –	15.06	67.94	28.96	56.19	2.75	39.78
No evidence	1.06	64.94	6.69	81.82	9.61	57.14

Table 3: LongChat, dev-short set, **simple (one-step / 1S)** and **extract-then-answer prompts (two-step, 2S)**, compare initial and advanced prompt (2-step+, 2S+), zero-shot (0S) vs. fine-tuned (FT) with QLoRA.

and pre-training and fine-tuning procedure. Here, the model has to do a trade-off between generating answers with more information (extractive, abstractive) or classify the question as unanswerable. The answers to abstractive questions see an initial quality degradation and only converge back to their initial level (F_1: 20.78 → 20.10) late in training. Our interpretation is that this is a result of the training data forcing the model to fit to the answer style for around 75% of the questions in Qasper: extracting word for word and short answers. With more epochs of fine-tuning, the model re-learns the more complex task of abstractive QA (Table 4).

Epochs	0	1	3	5
Answer F_1	24.19	41.13	44.56	47.02
Answer F_1 by type				
Extractive	26.51	41.80	45.18	48.21
Abstractive	20.78	12.59	16.59	20.10
Boolean	36.79	70.49	80.33	76.47
Unanswerable	0.04	69.57	66.67	68.54

Table 4: LongChat-32k, dev-short set, **simple prompt**, fine-tuned with QLoRA.

Fine-tuning mostly improves instruction-following While we only train with sequences of up to 8k tokens, we see an improvement across all analyzed paper lengths and evidence positions and the performance loss for papers with a length between 4k and 8k tokens and longer ones almost disappears going from 5.57 (zero-shot) to 0.36 (fine-tuned). However, we still see consistently reduced performance for papers that exceed LLaMA 2's original context window length of 4k and especially for questions where the evidence is further out than 4k.

In Appendix B, we list some qualitative example how fine-tuning did improve the model's answers.

4.2.2 Evidence-only Prompt

Our previous experiments showed that even models whose context window was extended with a technique similar to Positional Interpolation struggle with papers that exceed the original context length of LLaMA 2 of 4k tokens – especially if the evidence lies outside of that range. The question now is if these questions or at least some of them are inherently harder to answer. We evaluate if the performance varies in our analysis if the context given to the model is the evidence only instead of the full paper.

Training only on evidence performs well except for unanswerable questions When fine-tuning LongChat on the evidence only, we more quickly see better results that exceed those before (Table 5) and therefore only train for 3 epochs. After training LongChat on the evidence only, we compare its performance directly against the model that we trained on full papers: The performance of the context-length-specific model is better in general (F_1: 41.66 vs. 44.56 / 47.02) but not on all sub

Epochs	3	3	3	5
Train split	evo		fp	
Eval split	evidence	fp		
Answer F_1	57.22	41.66	44.56	47.02
Answer F_1 by type				
Extractive	62.19	47.01	45.18	48.21
Abstractive	27.01	24.15	16.59	20.10
Boolean	79.83	76.67	80.33	76.47
Unanswer.	80.56	2.70	66.67	68.54
Answer F_1 per paper length				
0k – 4k	57.93	42.68	50.22	52.15
4k – 8k	56.86	40.46	41.91	44.45
8k –	56.94	42.68	39.75	44.09
Answer F_1 per absolute evidence position				
0k – 4k	54.27	44.48	43.94	46.28
4k – 8k	50.86	37.46	34.80	37.74
8k –	63.61	43.71	64.76	67.94
No ev.	93.51	16.88	61.04	64.94

Table 5: Compare LongChat-32k, fine-tuned with QLoRA on **evidence only** (evo) or full paper (fp).

scores. When evaluating the evidence-only model on full papers we made an interesting observation: This model has equal or better F_1 scores on all question types except for unanswerable questions. The score for this type of question is probably so low as the model only learned to map the absence of evidence or the presence of a placeholder to the question being unanswerable.

Fine-tuning improves instruction-following and unanswerable question detection We assume that this result together with less than 8k tokens long training data improving performance on more than 8k tokens long evaluation data means that training the model mostly improves instruction following and does not promote better long-context understanding. But we also note that in order for the model to learn if a question is unanswerable it has to explicitly learn the mapping of no evidence in the whole paper to the question being unanswerable. During fine-grained analysis by input length, we see that the model that we trained on evidence only shows almost no performance decrease with increased paper length but also its performance for shorter papers is worse than those models that were trained on full papers. We also see that training on the full papers is useful as it dramatically improves performance for questions where no evidence is contained in the paper text.

4.2.3 Extract-then-Answer Prompt

Inspired by the results of using only the evidence as context to answer the questions, we hypothesize that a chain-of-thought prompt could increase performance: The model has to extract the relevant paragraphs first and then answer the questions based on the evidence found.

Epochs	0	1	3	5
Answer F_1	24.94	19.93	34.52	39.08
Answer F_1 by type				
Extractive	23.19	8.27	29.98	37.50
Abstractive	17.35	1.52	10.28	14.92
Boolean	57.96	23.76	52.34	49.51
Unanswerable	11.84	98.45	89.57	89.09
Evidence F_1	12.73	25.74	34.32	38.45

Table 6: LongChat-32k, dev-short set, **extract-then-answer prompt**, fine-tuned with QLoRA.

Extract-then-Answer Prompt does not improve performance During training, we saw an initial drop in performance for all question types that can be answered with the paper as context (Table 6). When looking at the evidence score and during qualitative analysis, we see that the model does not extract the correct paragraphs leading to an inability to answer most of the questions. After five epochs, for 535 out of 990 questions (∼54%) the model finds evidence. But during training, the model saw evidence for 1,607 out of 1,904 questions (∼84%). Yet after the same number of epochs as the one-step prompt model, this model still performs worse (47.02 vs. 39.08, Table 3).

Even for longer papers and evidence more difficult to reach, the extract-then-answer prompt does not improve performance as the evidence extraction also suffers on longer context and also does not help even inside the original context window. Out of 990 questions, the fine-tuned model still finds no evidence for 455 questions.

Handling Absent Evidence During training, the most common unique evidence string presented to the model is the placeholder we use for no evidence. For an improved prompt, we therefore include a prefix in the training data and as a hint in the prompt that every no empty extracted evidence starts with this prefix. We argue that this helps the model to avoid resorting to generating the "easiest" evidence which is none or the placeholder inspired by Attention Strengthening Question Answering (He et al.,

2023) which predicts the indices of the most relevant document in multi-document QA. We also adopt their approach of placing the question before and after the context.

To further reduce the number of generated empty evidence, we lower the number of examples in our training data where no evidence should be found to push the model into generating non-empty evidence more frequently. In the training data (<8k tokens), only around 16% of the questions are annotated with no evidence. However, the model that we fine-tuned on the "standard" extract-then-answer prompt generates no extracted evidence for around 40% of the questions which is 2.5 times as often. We assume a linear dependency between percentage of training answers without evidence and the percentage of generated answers without evidence. We lower the ratio of questions with no evidence in the training data to around 6% to arrive at 16% of generated empty evidence. We now employ all techniques we presented previously to improve the extract-then-answer prompt.

Adapted Prompt: Performance improves only slightly While the answer F_1 score does improve with this adapted prompt for the fine-tuned model (Table 3) when compared to the simpler extract-then-answer prompt, the evidence F_1 is lower even though the percentage of empty evidence drops from around 46% to around 22%. Also, for the zero-shot prompt all question types show worse results and the evidence score even drops to 0.0. Manual investigation shows that the model generated very long paragraphs as evidence in the zero-shot setup which led to this score. In further analysis, the advanced extract-then-answer prompt shows slightly better results for papers with under 8k tokens (F_1: 41.57 vs. 44.78, 37.77 vs. 39.66) and evidence below the same threshold (F_1: 36.00 vs. 40.54, 30.31 vs. 34.97). But the F_1 scores are still below those of the one-step prompt (47.02 vs. 41.18) as the evidence extraction also still suffers from long context.

4.3 Final Comparison against Baselines

Finally, we compare the results of our experiments against task-specific models and strong LLMs. Our comparison is on the ZeroSCROLLS subset of the Qasper test set which we believe is representative enough for the full test set (Table 1) to use it for comparison to strong LLMs. The ZeroSCROLLS subset uses a slightly different prompt for Qasper and does not include the title and abstract in the input. We compare our approaches with both prompts: ZeroSCROLLS (ZC) and LongBench (LB). With the LongBench prompt used during inference, our best approach exceeds GPT-4's F_1 score on the ZeroSCROLLS subset, comes close to the strongest model, and represents a great improvement over the zero-shot setup (Table 7). It is important to note that the ZeroSCROLLS authors mentioned that GPT-4 sometimes struggled more than other models to follow the prompt on Qasper. When we use the same prompt as the other models, both our zero-shot and the fine-tuned model lose more than 5 F_1 points showing how important prompting can be. As the performance drop is almost the same, we assume that for the fine-tuned model this is not a result of the mismatch between the training prompt and the inference prompt. The fine-tuned LongChat-32k model with the LongBench prompt is only able to almost match the task-specific model. We assume that this observation and the fact that Flan-UL2 is the best performing model are a result of these models being full transformers with an encoder and a decoder. The bidirectional encoder that processes the context together with the question and the prompt before generating the answer could help here.

Model	Prompt	Training	Answer F_1
Ours			
LongChat	ZC	0-shot	25.80
LongChat	LB	0-shot	31.07
LongChat	ZC	5 epochs	46.90
LongChat	LB	5 epochs	52.73
Existing models			
Flan-UL2	ZC	0-shot	56.90
GPT-4	ZC	0-shot	50.70
CoLT5	ZC	0-shot	53.10

Table 7: Baseline results (Flan-UL2 (Tay et al., 2023), GPT-4 (OpenAI, 2023), CoLT5 (Ainslie et al., 2023)) from ZeroSCROLLS benchmark (Shaham et al., 2023) compared to our results (LongChat-32k, 5 epochs), ZeroSCROLLS subset of Qasper test set.

5 Conclusion

We wanted to investigate how well LLMs can handle scientific papers and how we can improve their performance. We observe that the (unmodified) small open-source long-context LLMs we tested are able to process scientific papers with up to about 16k tokens from the Qasper dataset but fall short of commercial LLMs. Additionally, the performance drops after the context exceeds the original context window – especially if the relevant information to answer to question lies in that region of the paper.

When we employ the current techniques for efficient training QLoRA and FlashAttention, we can fine-tune the models on papers with a length of up to 8k tokens on a single datacenter GPU that is available to a university student for research. The performance of our fine-tuned model still increases for even longer papers without being trained on these lengths. Experiments with models that we only trained on extracted paragraphs without providing the model the full paper suggest that our training primarily improves instruction following but also improves the models' ability to determine if a question is unanswerable as it has to learn the connection between the absence of relevant information and the unanswerability of the question. When comparing our results against baselines, we saw that our best approach reaches or surpasses the result of the original GPT-4.

Limitations

This paper only investigates the Qasper dataset and the LongChat LLM. The Qasper dataset is limited to scientific papers from the NLP domain and mostly provides questions about facts and not more complex prompts like asking for new research directions based on the given paper. LongChat may have different strengths and weaknesses than other LLMs which may respond differently to the our prompts, our fine-tuning scheme, and long context in general (as seen by Liu et al. (2023)). While our resulting model is an improvement over the zero-shot LongChat, it still makes mistakes (like determining a question as unanswerable even if it is answerable).

We did not investigate all fitting configurations of our experimental setup like providing a random paragraph as evidence instead of no paragraph or how the fine-tuning for one prompt type influences the performance during inference with a different prompt type (except for the model that we fine-tuned on the evidence only).

Acknowledgment

We thank the anonymous reviewers for their valuable feedback.

References

Joshua Ainslie, Tao Lei, Michiel de Jong, Santiago Ontañón, Siddhartha Brahma, Yury Zemlyanskiy, David C. Uthus, Mandy Guo, James Lee-Thorp, Yi Tay, Yun-Hsuan Sung, and Sumit Sanghai. 2023. Colt5: Faster long-range transformers with conditional computation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 5085–5100. Association for Computational Linguistics.

Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy P. Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul Ronald Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, and et al. 2023. Gemini: A family of highly capable multimodal models. *CoRR*, abs/2312.11805.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *CoRR*, abs/2308.14508.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *CoRR*, abs/2004.05150.

Bo-Christer Björk, Annikki Roos, and Mari Lauri. 2009. Scientific journal publishing: yearly volume and open access availability. *Inf. Res.*, 14(1).

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023. Extending context window of large language models via positional interpolation. *CoRR*, abs/2306.15595.

Liying Cheng, Lidong Bing, Qian Yu, Wei Lu, and Luo Si. 2020. APE: argument pair extraction from peer review and rebuttal via multi-task learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 7000–7011. Association for Computational Linguistics.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2023. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24:240:1–240:113.

Tri Dao. 2023. Flashattention-2: Faster attention with better parallelism and work partitioning. *CoRR*, abs/2307.08691.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 4599–4610. Association for Computational Linguistics.

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

Michael Fire and Carlos Guestrin. 2018. Over-optimization of academic publishing metrics: Observing goodhart's law in action. *CoRR*, abs/1809.07841.

Chi Han, Qifan Wang, Hao Peng, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. 2024. LM-infinite: Zero-shot extreme length generalization for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3991–4008. Association for Computational Linguistics.

Junqing He, Kunhao Pan, Xiaoqun Dong, Zhuoyang Song, Yibo Liu, Yuxin Liang, Hao Wang, Qianguo Sun, Songxin Zhang, Zejian Xie, and Jiaxing Zhang. 2023. Never lost in the middle: Improving large language models via attention strengthening question answering. *CoRR*, abs/2311.09198.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training compute-optimal large language models. *CoRR*, abs/2203.15556.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *CoRR*, abs/2310.06825.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *CoRR*, abs/2401.04088.

Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard H. Hovy, and Roy Schwartz. 2018. A dataset of peer reviews (peerread): Collection, insights and NLP applications. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1647–1661. Association for Computational Linguistics.

Kaio Ken. Extending context is hard…but not impossible.

Dacheng Li, Rulin Shao, Anze Xie, Ying Shenga, Lianmin Zheng, Joseph E. Gonzalez, Ion Stoica, Xuezhe Ma, and Hao Zhang. How long can open-source llms truly promise on context length?

Miao Li, Jianzhong Qi, and Jey Han Lau. 2022. Peersum: A peer review dataset for abstractive multi-document summarization. *CoRR*, abs/2203.01769. Withdrawn.

Jialiang Lin, Jiaxin Song, Zhangping Zhou, Yidong Chen, and Xiaodong Shi. 2023. MOPRD: A multidisciplinary open peer review dataset. *Neural Comput. Appl.*, 35(34):24191–24206.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *CoRR*, abs/2307.03172.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Meta AI. 2023. Introducing llama: A foundational, 65-billion-parameter large language model. Retrieved 2023-12-11.

OpenAI. a. Gpt-3.5 turbo.

OpenAI. b. What are tokens and how to count them?

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

Uri Shaham, Maor Ivgi, Avia Efrat, Jonathan Berant, and Omer Levy. 2023. Zeroscrolls: A zero-shot benchmark for long text understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 7977–7989. Association for Computational Linguistics.

Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Denny Zhou, Neil Houlsby, and Donald Metzler. 2023. UL2: unifying language learning paradigms. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Lei Wang, Yi Hu, Jiabang He, Xing Xu, Ning Liu, Hui Liu, and Heng Tao Shen. 2023. T-sciq: Teaching multimodal chain-of-thought reasoning via large language model signals for science question answering. *CoRR*, abs/2305.03453.

Qingyun Wang, Qi Zeng, Lifu Huang, Kevin Knight, Heng Ji, and Nazneen Fatema Rajani. 2020. Reviewrobot: Explainable paper review generation based on knowledge synthesis. In *Proceedings of the 13th International Conference on Natural Language Generation, INLG 2020, Dublin, Ireland, December 15-18, 2020*, pages 384–397. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35:*

Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022.

Pengcheng Yang, Xu Sun, Wei Li, and Shuming Ma. 2018. Automatic academic paper rating based on modularized hierarchical convolutional neural network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 496–502. Association for Computational Linguistics.

Ori Yoran, Tomer Wolfson, Ben Bogin, Uri Katz, Daniel Deutch, and Jonathan Berant. 2023. Answering questions by meta-reasoning over multiple chains of thought. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 5942–5966. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023.*

A Prompts

We used the following prompts during our experiments. <CONTEXT> stands for the paper text or a shortened version of it while <QUESTION> is the placeholder for the specific question on the provided context.

A.1 ZeroSCROLLS

You are given a scientific article and a question. Answer the question as concisely as you can, using a single phrase or sentence if possible. If the question cannot be answered based on the information in the article, write "unanswerable". If the question is a yes/no question, answer "yes", "no", or "unanswerable". Do not provide any explanation.

Article: <CONTEXT>

Question: <QUESTION>

A.1.1 LongBench (our version)

You are given a scientific article and a question. Answer the question as concisely as you can, using a single phrase or sentence if possible. If the question cannot be answered based on the information in the article, write ’unanswerable’. If the question is a yes/no question, answer ’yes’, ’no’, or ’unanswerable’. Do not provide any explanation.
Article: <CONTEXT>
Answer the question based on the above article as concisely as you can, using a single phrase or sentence if possible. If the question cannot be answered based on the information in the article, write ’unanswerable’. If the question is a yes/no question, answer ’yes’, ’no’, or ’unanswerable’. Do not provide any explanation.
Question: <QUESTION>

A.2 Evidence only

You are given excerpts from a scientific article and a question. Answer the question as concisely as you can, using a single phrase or sentence if possible. If the question cannot be answered based on the information in the excerpts from an article, write ’unanswerable’. If the question is a yes/no question, answer ’yes’, ’no’, or ’unanswerable’. Do not provide any explanation.
Excerpts from Article: <CONTEXT>
Answer the question based on the above excerpts from an article as concisely as you can, using a single phrase or sentence if possible. If the question cannot be answered based on the information in the excerpts from an article, write ’unanswerable’. If the question is a yes/no question, answer ’yes’, ’no’, or ’unanswerable’. Do not provide any explanation.
Question: <QUESTION>

A.3 Two-turn

Turn 0:
You are given a scientific article and a question. Extract all paragraphs that are relevant to answer the question. Copy them word by word from the article. If there are no relevant paragraphs answer ’No relevant paragraphs found’. Do not provide any explanation.
Article: <CONTEXT>
Extract all paragraphs that are relevant to answer the question. Copy them word by word from the article. If there are no relevant paragraphs answer ’No relevant paragraphs found’.
Question: <QUESTION>

Turn 1:
Answer the question based on your extracted relevant paragraphs from the above article and answer as concisely as you can, using a single phrase or sentence if possible. If the question cannot be answered based on the information in the article (your last answer was possibly 'No relevant paragraphs found'), write 'unanswerable'. If the question is a yes/no question, answer 'yes', 'no', or 'unanswerable'. Do not provide any explanation.
Question: <QUESTION>

A.4 Two-turn, Advanced (Prefix for Evidence, Question Repeated)

Turn 0:
You are given a scientific article and a question. Extract all paragraphs that are relevant to answer the question. Copy them word by word from the article and start with 'Found these relevant paragraphs:'. If there are no relevant paragraphs answer 'No relevant paragraphs found'. Do not provide any explanation.
Question: <QUESTION>
Article: <CONTEXT>
Extract all paragraphs that are relevant to answer the question. Copy them word by word from the article. If there are no relevant paragraphs answer 'No relevant paragraphs found'.
Question: <QUESTION>

Turn 1:
Answer the question based on your extracted relevant paragraphs from the above article and answer as concisely as you can, using a single phrase or sentence if possible. If the question cannot be answered based on the information in the article (your last answer was possibly 'No relevant paragraphs found'), write 'unanswerable'. If the question is a yes/no question, answer 'yes', 'no', or 'unanswerable'. Do not provide any explanation.
Question: <QUESTION>

B Example answers

We also provide a list of qualitative examples how fine-tuning improved the performance (simple prompt, test set). We separate multiple possible answers by slashes.

- Answer more clearly to unanswerable question
    - **Question:** How does this result compare to other methods KB QA in CCKS2019?
    - **Answer (ground truth):** Unanswerable
    - **Answer (zero-shot):** The proposed method for open domain Chinese question answering in CCKS2019 achieved an F1 score of 70.45% on the test data. This result is comparable to other methods in KB QA, as it also employs a combination of information retrieval and semantic parsing. However, the specific comparison to other methods is not provided in the article, so it is unanswerable.
    - **Answer (fine-tuned):** Unanswerable

- Detect unanswerable question
    - **Question:** Which labeling scheme do they use?
    - **Answer (ground truth):** Unanswerable
    - **Answer (zero-shot):** They use a sequence labeling scheme for both constituency and dependency parsing.
    - **Answer (fine-tuned):** Unanswerable

- Answer more concisely
    - **Question:** Does regularization of the fine-tuning process hurt performance in the target domain?
    - **Answer (ground truth):** No
    - **Answer (zero-shot):** No, regularization of the fine-tuning process does not hurt performance in the target domain. In fact, it can improve performance by reducing catastrophic forgetting and preserving the source domain's performance as much as possible while keeping the target domain's performance optimal.
    - **Answer (fine-tuned):** No

- Answer more concisely and correctly
    - **Question:** what are the baselines?
    - **Answer (ground truth):** Reschke CRF, Reschke Noisy-OR, Reschke Best
    - **Answer (zero-shot):** The article does not provide specific baselines for the machine reading task. However, it mentions several related works and systems that the authors aim to improve upon. These include: 1. Reschke CRF: a conditional

random field model. 2. Reschke Noisy-OR: a sequence tagger with a N̈oisy-ORf̈orm of aggregation that discourages the model from predicting the same value for multiple slots. 3. Reschke Best: a sequence tagger using a cost-sensitive classifier, optimized with SEARN, a learning-to-search framework. These baselines represent different approaches to machine reading and event extraction from noisy sources. The authors' proposed architecture, called RAC, aims to improve upon these baselines by incorporating attention-based mention scoring, cluster-wide aggregation, and global constraints.

  - **Answer (fine-tuned):** Reschke CRF. Reschke Noisy-OR. Reschke Best.

- Answer more concisely and correctly 2
  - **Question:** How do they preprocess Tweets?
  - **Answer (ground truth):** They removed retweets, URLs, emoticons, mentions of other users, hashtags; lowercased the text and removed the punctuation. / re-tweets do not bring any additional information to our study, thus we removed them, removed any expressions considered to be semantically meaningless like URLs, emoticons, mentions of other users (denoted by the @ symbol) and hashtags, downcased and stripped the punctuation
  - **Answer (zero-shot):** The article does not provide specific details on how the tweets are preprocessed. However, it mentions that certain expressions are removed, such as URLs, emoticons, mentions of other users, and hashtags, to simplify textual pre-processing. Additionally, the tweets are downcased and stripped of punctuation.
  - **Answer (fine-tuned):** They remove retweets, expressions considered to be semantically meaningless, and downcase and strip the punctuation from the text.

- Answer correctly
  - **Question:** Do they evaluate only on English datasets?
  - **Answer (ground truth):** Yes / The twitter dataset is English-only; no information for the reddit dataset is given
  - **Answer (zero-shot):** No, they evaluate on a diverse range of social media platforms and content types, including English, Spanish, Portuguese, French, German, Italian, and Arabic.
  - **Answer (fine-tuned):** Yes

- Answer more specifically
  - **Question:** How many layers of recurrent neural networks do they use for encoding the global context?
  - **Answer (ground truth):** 2 / 8
  - **Answer (zero-shot):** The article does not provide specific information on the number of layers of recurrent neural networks used for encoding the global context. However, it mentions that they use stacked recurrent neural networks to capture the long-term dependencies and global context information from the whole sentence.
  - **Answer (fine-tuned):** Two feed-forward layers of size 400 with rectified linear units (ReLU); One bi-directional gated recurrent unit (BiGRU) layer with hidden size 200; Two feed-forward layers of hidden size 200 with rectified linear units; One BiGRU layer with hidden size 100; Two feed-forward layers of size 100 and 50 respectively with ReLU activation.

C Further Experimental Setup

Evidence type	Frequency
Text	81.6%
Table/Figure	11.6%
None	12.8%

Table 8: Qasper dataset statistics (Dasigi et al., 2021): evidence types; the percentages add to over 100% because answers can include multiple evidence types.

Minor details To bin the evaluation sets, we tokenize the whole paper texts from the JSON files that the dataset website provides. We used the tokenizer from Vicuna-16k, but the number of tokens should be the same for the other tokenizers. For the evidence positions, we determine the position of the first token of the evidence paragraphs inside the paper text. We assume this is representative as most evidence is at most a few sentences long. If there are multiple evidence paragraphs that are not continuous in the dataset, we bin potentially one paper multiple times.

A model tasked to extract the evidence should output no paragraphs if there is none for the question at hand. Instead, it should generate the string "No relevant paragraphs found" which we include in the prompts and filter out of the answers before calculating the F_1 score.

C.1 Hard- and Software

For evaluation and training of the tested models we need high-performance GPUs. Therefore, we use the bwUniCluster 2.0[3] for our experiments. Depending on availability, we use the NVIDIA A100 with 80 GB of accelerator memory or the NVIDIA H100 with 94 GB. The bwUniCluster 2.0 allows the use of NVIDIA Enroot[4] which enables running Docker[5] containers on the computing cluster. We use the PyTorch container[6] by NVIDIA to train the models in our experiments. FlashAttention is only implemented per GPU type at the moment and comes pre-installed with this container.

We run all our experiments (inference and training) with the FastChat[7] (Zheng et al., 2023) framework which is an open-source platform for "training, serving, and evaluating large language model based chatbots". It is developed by the Large Model Systems Organization (LMSYS Org).[8] The LMSYS Org also operates the LMSYS Chatbot Arena[9] (Zheng et al., 2023) which tries to compare the performance of current LLMs against each other in a chatbot setting. FastChat provides code to easily run models, feed them with input data, and store their answers. Besides regular fine-tuning it also provides a (Q)LoRA implementation that can utilize FlashAttention. This script is run with the DeepSpeed[10] library.

C.2 Hyperparameters

All following stated hyperparameters are the same on all experiments if not stated differently per experiment.

During inference, we run the models with a temperature of 0.0 which equates to greedy decoding.[11] FastChat code also uses a temperature of 0.0 for tasks like extraction and reasoning.[12] This fits our requirements as we want the most accurate and truthful answer. Also, we saw a degradation in performance when raising the temperature. We let the models generate up to 1,024 tokens.

Our training configuration is the same as the example from FastChat: We use a LoRA rank r of 8 and a LoRA Alpha of 16. Rank $r = 8$ results in 4,194,304 trainable parameters out of 6,742,609,920 for LLaMA 2 7B based models. The dropout is 0.05 and we apply no weight decay. The learning rate is initialized with 2e-5 with a warm-up ratio of 0.03 and a cosine learning rate scheduling. We do no extensive hyperparameter search because of time constraints regarding compute and because the authors of QLoRA already noted that the most important "hyperparameter" is the location of the adapted parameters inside the model. We train each model for 5 epochs on the training split after our preprocessing. We chose this duration as it could be done within a few hours on a single GPU, and we saw performance saturation within this training duration.

D Additional Evaluation Results

We provide additional evaluation results for all models – zero-shot (Table 9) and fine-tuned with QLoRA (Table 10).

We also tested if changing the temperature increases performance (Table 11): Our rationale is that the most probable evidence is none as the placeholder string for this is always the same and occurs more often during training than any other evidence string. Also, it is not that important if the found paragraphs are perfectly correct (e.g., not too long): It just has to be useful to answer the question. Yet, increasing the temperature monotonously decreases both the evidence and answer F_1 scores. On top of reduced quality, the percentage of empty evidence rises from $\sim$46% (0.0) to $\sim$66% (1.0).

[3] https://wiki.bwhpc.de/e/Main_Page

[4] https://github.com/NVIDIA/enroot

[5] https://docs.docker.com/

[6] https://catalog.ngc.nvidia.com/orgs/nvidia/containers/pytorch

[7] https://github.com/lm-sys/FastChat

[8] https://lmsys.org/

[9] https://chat.lmsys.org/

[10] https://github.com/microsoft/DeepSpeed

[11] https://huggingface.co/blog/how-to-generate

[12] https://github.com/lm-sys/FastChat/blob/085c2c37dca426059f023e2a080c45717c742fd1/fastchat/llm_judge/common.py

Models	Bin count	Vicuna-4k	Vicuna-16k	LongChat-32k
Answer F_1 per paper length				
0k – 4k	333	25.53	27.20	25.47
4k – 8k	593	0.40	24.01	24.08
8k –	64	0.00	19.55	18.51
Answer F_1 per absolute evidence position				
0k – 4k	794	9.79	25.82	26.73
4k – 8k	173	0.38	18.02	23.35
8k –	6	0.00	3.78	15.06
No evidence	77	11.80	23.38	1.06

Table 9: Analysis of the models we tested, dev-short set, **LongBench prompt** (Bai et al., 2023), zero-shot.

Models	Bin count	Vicuna-4k	Vicuna-16k	LongChat-32k
Answer F_1 per paper length				
0k – 4k	333	38.89	50.26	52.15
4k – 8k	593	18.53	43.02	44.45
8k –	64	2.48	39.55	44.09
Answer F_1 per absolute evidence position				
0k – 4k	794	23.99	43.54	46.28
4k – 8k	173	9.33	35.23	37.74
8k –	6	0.00	64.37	67.94
No evidence	77	52.81	75.32	64.94

Table 10: Models we tested, dev-short set, **LongBench prompt** (Bai et al., 2023), fine-tuned with QLoRA for 5 epochs.

We compare our best approach against the baseline model from the original publication of the Qasper dataset (Dasigi et al., 2021). Their model is the Longformer-Encoder-Decoder (LED) (Beltagy et al., 2020) in two sizes: base and large. It contains more fine-grained results than the comparison on the ZeroSCROLLS (Shaham et al., 2023) subset of Qasper. Also, they estimate a lower bound for the human performance on the test set by calculating the agreement between different annotator answers for each question. Their best model for question answering is LED-base that receives the full paper as input. One variant includes evidence extraction during training.

Our comparison (Table 12) shows that LED has a similar distribution of the F_1 scores per type. The extractive score is higher than the abstractive score and the boolean score is the highest or close to it. We can also see a similar behavior of the LED model to the extract-then-answer prompt when integrating evidence extraction into the answer generation process: The extractive and abstractive scores suffer while the model detects unanswerable questions better. Also, our best approach performs better on questions with very short answers (yes/no, unanswerable) than the lower bound for human performance. This could be an explanation of our observation that longer training does not improve these scores after they reach a certain level (trade-off: short vs. long answers). However, the quality of the abstractive answers is considerably worse (39.71 vs. 18.79).

For the evidence extraction, our best model is LongChat-32k fine-tuned with the extract-then-answer prompt. While the evidence extraction did not improve the answer quality in our case, it can be a useful addition for the user of a QA system to contextualize the answer. Here, the difference between our approach and the Qasper baseline LED-large (Table 13) is not as high as for the answer F_1 score but we still see a clear improvement over the baseline.

E Additional Training Results

Here, we list how the F_1 scores during our training runs changed compared to the zero-shot results with the same prompt. For the evidence only prompt (Table 14) and for the extract-then-answer prompt (Table 6, Table 15), we only trained LongChat-32k.

LongChat-32k	0.0	0.2	0.4	0.6	0.8	1.0
Answer F_1	39.08	37.61	35.05	33.15	30.57	29.60
Evidence F_1	38.45	37.20	35.31	33.54	31.85	29.16

Table 11: LongChat, dev-short set, **extract-then-answer prompt**, fine-tuned 5 epochs with QLoRA, varying temperatures.

Models	LongChat-32k LongBench prompt zero-shot	LongChat-32k LongBench prompt 5 epochs	LED-base without evidence extraction	LED-base with evidence extraction	Human (lower bound)
Test answer F_1	28.81	55.20	32.80	33.63	60.92
Test answer F_1 by type					
Extractive	28.39	54.89	30.96	29.97	58.92
Abstractive	20.82	18.79	15.76	15.02	39.71
Boolean	56.11	84.68	70.33	68.90	78.98
Unanswerable	2.14	86.42	26.21	44.97	69.44

Table 12: Comparison of our approaches against baselines from the Qasper paper, test set.

Models	LongChat-32k extract-then-answer prompt 5 epochs	LED-base	LED-large	Human (lower bound)
Dev evidence F_1	38.27	23.94	31.25	–
Test evidence F_1	42.57	29.85	39.37	71.62

Table 13: Comparison of our approaches against baselines from the Qasper paper, full dev and test set, evidence extraction.

LongChat-32k	Zero-shot	1 epoch	2 epochs	3 epochs
Answer F_1	36.16	55.65	56.97	57.22
Answer F_1 by type				
Extractive	37.58	61.04	61.41	62.19
Abstractive	21.80	25.20	25.60	27.01
Boolean	47.96	72.27	80.99	79.83
Unanswerable	53.48	84.00	83.33	80.56

Table 14: LongChat-32k, **evidence only** dev-short set, fine-tuned with QLoRA.

LongChat-32k	Zero-shot	1 epoch	3 epochs	5 epochs
Answer F_1	17.85	27.59	41.54	41.18
Answer F_1 by type				
Extractive	16.37	22.27	45.00	41.82
Abstractive	16.03	10.64	21.06	19.41
Boolean	36.75	59.66	68.14	58.10
Unanswerable	5.33	59.79	40.23	69.23
Evidence F_1	0.00	26.37	31.12	35.13

Table 15: LongChat, dev-short set, **extract-then-answer prompt, improved**, fine-tuned with QLoRA.