Can Few-shot Work in Long-Context? Recycling the Context to Generate Demonstrations

Anonymous ACL submission

Abstract

Despite recent advancements in Large Language Models (LLMs), their performance on tasks involving long contexts remains sub-optimal. In-Context Learning (ICL) with few-shot examples may be an appealing solution to enhance LLM performance in this scenario; However, naïvely adding ICL examples with long context introduces challenges, including substantial token overhead added for each few-shot example and context mismatch between the demonstrations and the target query. In this work, we propose to automatically generate few-shot examples for long context QA tasks by *recycling* contexts. Specifically, given a long input context (1-3k tokens) and a query, we generate additional query-output pairs from the given context as few-shot examples, while introducing the context only once. This ensures that the demonstrations are leveraging the same context as the target query while only adding a small number of tokens to the prompt. We further enhance each demonstration by instructing the model to *explicitly* identify the relevant paragraphs before the answer, which improves performance while providing fine-grained attribution to the answer source. We apply our method on multiple LLMs and obtain substantial improvements (+16 absolute points on average across models) on various QA datasets with long context, especially when the answer lies within the middle of the context. Surprisingly, despite introducing only single-hop ICL examples, LLMs also successfully generalize to multi-hop long-context QA using our approach.

1 Introduction

Long contexts are prevalent in various domains, ranging from legal documents and scientific articles to lengthy reports and novels. These may consist of a single extensive document or multiple passages, typically retrieved through specific retrieval mechanisms (e.g., RAG; Lewis et al., 2020).

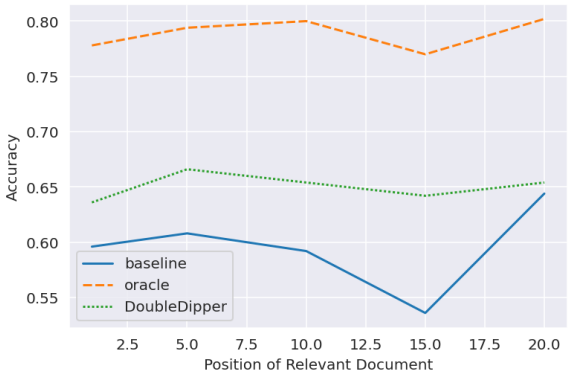

Figure 1: Performance of Gemini Flash (v1.5) on a sample of the Lost-in-the-middle dataset (Liu et al., 2023). The X-axis is the position of the relevant passage in the context. The baseline (blue line) displays a U-shaped curve, performing well only when the relevant passage is at the beginning or end of the input. The oracle (green line) shows significant performance gain when the relevant passage ID is provided in the prompt, showing that the identification of supporting evidence(s) is a major challenge. DOUBLEDIPPER (our method, orange line) flattens this U-shaped trend.

Yet, while Large Language Models (LLMs) have demonstrated impressive capabilities in a variety of tasks including answering questions requiring one or multiple reasoning steps, they often struggle to answer simple questions when faced with long contexts. Despite substantial engineering efforts (Chen et al., 2023) to extend the context window of LLMs to extremely long inputs (32k and even 1M tokens), these models continue to struggle with much shorter inputs, comprising only a few thousand tokens.

In order to answer questions from long inputs, models should *implicitly* identify relevant information segments and then reason over these segments to formulate an answer. It has been shown that LLMs struggle when the relevant information is buried in the middle of the context (Liu et al., 2023) or obscured by numerous irrelevant details (Levy

et al., 2024). Our analysis (illustrated in Figure 1) identifies the identification of relevant information as a major performance bottleneck of current models in long contexts.

In this work, we introduce a novel method to enhance the QA performance of LLMs in long input setups (to allow direct comparisons across a wide swath of models, we limit "long context" here to 1-3k tokens). Our approach, termed DOUBLEDIPPER, leverages LLMs' In-Context Learning (ICL) capability and is based on two principles. First, instead of typical ICL, where each few-shot example is standalone with a separate length context and a question-answer (QA) pair, we propose to *recycle* the given input context and automatically generate few-shot examples from this context. Specifically, we randomly select a few paragraphs from the given input context and generate QA pairs for each passage. These generated QAs serve as demonstration examples and are placed between the input context and the target input question. Figure 2 illustrates the differences between the traditional ICL with few-shot examples and DOUBLEDIPPER. Second, we further enrich each ICL example to instruct the model to *explicitly* identify the paragraph(s) containing the relevant information before generating the answer. This can be regarded as a structured Chain of Thought that incentivizes the model to pinpoint relevant information before reasoning, an essential capability for long-context processing.

By generating few-shot demonstrations from various sections of the input context while instructing the model to identify relevant passages, DOUBLEDIPPER encourages the model to develop deeper reading comprehension skills specific to the given input evidence. This, in turn, allows the model to answer subsequent queries with higher accuracy. DOUBLEDIPPER presents several advantages. In terms of efficiency, since each example does not include its own input context, our method adds to the original prompt a minimal number of tokens, resulting in a substantially cheaper inference than traditional ICL. Additionally, recycling the same context for ICL demonstrations ensures that the few-shot examples refer to the same domain as the input question, thus obviating the need for external retrieval processes. Finally, DOUBLEDIPPER generates answers with attribution to relevant paragraphs, improving the model's lookup ability and offering transparency, which substantially simplifies human evaluation (Slobodkin et al., 2024).

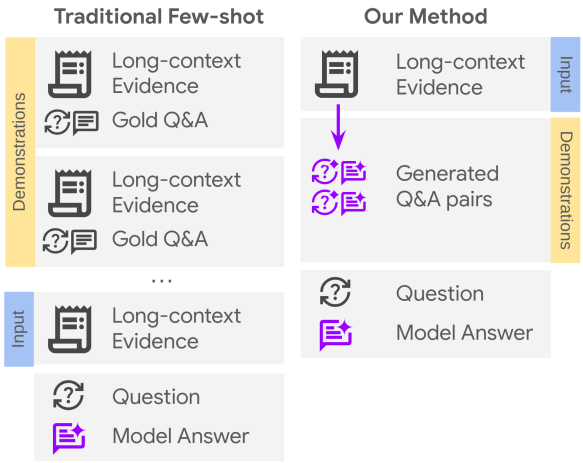

Figure 2: Comparison of traditional In-Context-Learning (ICL) and our new method. In traditional ICL (left), each example comprises a possibly lengthy context, accompanied by a query and an answer, typically derived from the training dataset. Conversely, our approach (right) simplifies each example to just a question and an answer, both of which are generated directly from the provided input context.

We applied DOUBLEDIPPER to a variety of LLMs, both commercial (Gemini Pro, Nano and Flash; Reid et al., 2024) and open-source (Llama 3.1 (Dubey et al., 2024), Mistral (Jiang et al., 2023), Mixtral (Jiang et al., 2024) and Gemma (Riviere et al., 2024)), and evaluate it on various QA datasets with long inputs, including common multi-hop QA datasets. Our experiments demonstrate that with only 3 self-generated few-shot examples, DOUBLEDIPPER consistently outperforms the baseline on our evaluation set by 16 absolute points on average across models. In addition, for some models, DOUBLEDIPPER enhances the robustness to the position of the relevant information within the text. Interestingly, while our few-shot examples focus on single-paragraph answers, DOUBLEDIPPER generalizes well to multi-hop QAs and where the answer requires information from multiple passages.

2 Background

Challenges in Long Context for Language Modeling. LLMs have been well-documented to struggle when input length grows (An et al., 2023), and especially so when it exceeds input lengths seen during training (Anil et al., 2022). Various methods have been proposed to advance long-context capabilities: Architectural, e.g., to augment the embedding layer to cleverly extrapolate to un-

seen lengths (Vaswani et al., 2017; Press et al., 2021; Caciularu et al., 2022; Hsieh et al., 2024); and via data, e.g., to incorporate longer inputs and more challenging long-context scenarios into training (He et al., 2023; Chen et al., 2023). However, this challenging problem stubbornly remains in competitive models today (Liu et al., 2023; Bishop et al., 2023; Levy et al., 2024). In contrast to the above methods, DOUBLEDIPPER does not involve training or architectural changes.

In documenting and exploring LLM performance in long-context settings, many different benchmarks targeting it have been proposed, such as Scrolls and Zero-Scrolls (Shaham et al., 2022, 2023), Loogle (Li et al., 2023), LongBench (Bai et al., 2023), inter alia. The problem of designing informative and reliable benchmarks in long-context is an an active, ever-changing area of research (Goldman et al., 2024; Yen et al., 2024). We describe the most relevant evaluation benchmarks used in this work in Section 4.

In-Context Learning The area of in-context learning (ICL) is a class of prompting techniques where demonstrations are added to the prompt in order to steer or improve model behavior (Min et al., 2022a). Typically, in-context learning involves hand-crafted demonstrations (Song et al., 2022), automatic retrieval of demonstrations from a larger set (Paranjape et al., 2023), or instructing the model to perform various tasks one after another in a pipeline (Gao et al., 2022). Recent improvements in long-context capabilities of LLMs have also had effect on improving the yield from in-context learning by simply using more short-length demonstrations (Agarwal et al., 2024).

While such methods are widely used for their effectiveness (Brown et al., 2020b; an Luo et al., 2024), they remain under-explored in settings of long-context. The reason is simple: If the context is already extremely long, adding additional demonstrations comparable in length to the input context will likely amplify the existing limitations of long context handling (Li et al., 2024c).

More related to our work, a few recent studies propose to prompt LLMs for automatically generating in-context demonstrations for various short context tasks (Kim et al., 2022; Yasunaga et al., 2024; Li et al., 2024a,b). However, applying those methods to long contexts is not straightforward because LLMs would need to generate long demonstrations.

In this work, we tackle these challenges and present a novel ICL method that recycles the given long context and further instructs models to identify the relevant information before generating the answer.

3 DOUBLEDIPPER

Recall that our work focuses on the task of question answering (QA) with long input context comprising multiple paragraphs. In addition to the answer, we aim to identify the supporting paragraphs in order to provide attribution. Formally, given a long input text D composed of n paragraphs $D = \{p_1, p_2, ..., p_n\}$ and a question q, the goal is to generate the answer a and identify the set(s) of paragraphs that support the answer $S = \{s_1, ..., s_k\}$. The number of the supporting paragraphs is not known in advance and can be one or more.

We describe DOUBLEDIPPER, an efficient method for improving the performance of large language models (LLMs) when dealing with long contexts. The core principles of DOUBLEDIPPER involve: (1) recycling the input context to automatically generate few-shot examples, and (2) "teaching" the model via in-context learning (ICL) to *explicitly* pinpoint the supporting paragraphs before generating the answer.

Figure 3 illustrates DOUBLEDIPPER. Starting with the input paragraphs D, we initially select k paragraphs at random (e.g., paragraphs 15, 5, and 17, for $k := 3$). For each chosen paragraph, we prompt the model to formulate a question that pertains to the specific paragraph, accompanied by an appropriate answer (for further details on prompt specifications, refer to Appendix A). Each generated QA pair is directly associated with its origin paragraph, enabling us to assemble the following structured in-context demonstration, shown as the DOUBLEDIPPER block in Figure 3:

$$\text{Question} : q_i$$
$$\text{Evidence} : p_i$$
$$\text{Answer} : a_i$$

Here, p_i indicates the index of the paragraph associated with the QA pair (q_i, a_i). Given a test question q, we then form a QA prompt by concatenating the original input context D, the compiled demonstrations and q. The model first generates the one or more indices of the supporting paragraph(s), followed by the answer.

Unlike traditional few shot examples that instruct the model about a specific *task*, DOUBLEDIPPER

Input

Instructions: [...]

[0]: The Parc botanique de Neuvic (6 hectares) is a botanical garden located in Neuvic-Sur-L'Isle [...]
⋮
[5] Santa Cruz de las Flores is the name of a town located south of Tlajomulco de Zuñiga, in the state of Jalisco, Mexico. It has been called Xochitlan, meaning "Place of Flowers"
[6]: Graft-De Rijp is a former municipality in the Netherlands, in the province of North Holland.
⋮
[15]: The Jardin Botanique de l'Université de Strasbourg (3.5 hectares) is a botanical garden at 28 rue Goethe, Strasbourg, Bas-Rhin, Alsace, France. It is open daily without charge.
⋮
[17]: Marquette is an unincorporated community in [...], located on Illinois Route 29, east of De Pue.
[18]: The capital and seat of the provincial government is Haarlem, and the province's largest city is the Netherlands' capital Amsterdam. The King's Commissioner of North Holland is Johan Remkes, serving since 2010.

DoubleDipper

See below a few examples:
Question: Is there an admission fee for the Jardin botanique de l'Université de Strasbourg?
Evidence: [15]
Answer: No, it is open daily without charge.
Question: What is the name of the town located south of Tlajomulco de Zuñiga?
Evidence: [5]
Answer: Santa Cruz de las Flores
Question: What is the name of the community that is west of Marquette?
Evidence: [17]
Answer: De Pue

Task

Who was in charge of the state where Graft-De Rijp is located?
Evidence: [6, 18]
Answer: Johan Remkes

Figure 3: Example of DOUBLEDIPPER applied to the MuSique dataset. Given 20 passages as input, DOUBLEDIPPER randomly selects 3 passages (specifically passages 15, 5, 17) and automatically generates a question-pair for each one. As each QA is associated with its respective paragraph, we form the demonstrations to instruct the model to identify the relevant passage(s) and the correct answer.

aims to coach the model on how to "handle" the specific input context. This is achieved by guiding the model to explicitly localize relevant information before generating the answer. Also, by randomly sampling multiple paragraphs from the input, DOUBLEDIPPER guarantees that the ICL demonstrations involve reading different parts of the context, allowing the model to better comprehend the input text. Beyond improving the performance of the QA task, instructing the model to provide the supporting paragraphs is valuable on its own as it offers transparency and substantially eases human evaluation (Slobodkin et al., 2024).

DOUBLEDIPPER offers several advantages. First, as each example in the demonstration consists only of a question, an answer and the ID of relevant passage, the number of added tokens due to the extra demonstrations is minimal, leading to a low additional cost and computation compared to the traditional In-Context-Learning. Furthermore, by reusing the same context to generate demonstrations, our approach guarantees that all few shot examples are derived from the exact same domain as the input query (Rubin et al., 2022).

Finally, we observe that, although the QA pairs in the demonstration are confined to individual paragraphs, the actual query q may require reasoning over multiple paragraphs (i.e., multi-hop QA). Surprisingly, LLMs can generalize from DOUBLEDIPPER *local* examples to these complex, *global* questions and successfully generate indices to multiple paragraphs (see Section 5).

	Size	Context Window
Gemini v1.5	Nano	32k
	Flash	32k
	Pro	32k
Llama 3.1	7B	128k
Mistral v0.3	7B	32k
Mixtral v0.1	7x8B	32k
Gemma 2	2B	8k
	9B	8k

Table 1: Models used with DOUBLEDIPPER.

4 Experiments

Datasets We apply our method to various datasets, each presenting its own domain-specific challenges. We selected these datasets because the supporting paragraphs are also annotated. Overall

Dataset	# Instances	Avg. # tokens
Lost-in-the-middle	2,500	2,815
FLenQA	1,500	3,225
HotpotQA	500	1,646
2Wiki	500	1,222
MuSiQue	500	2,549

Table 2: Evaluation datasets in our experiments. The average number of tokens is computed according to Gemma's tokenization of the simple prompt.

our evaluation set includes 5.5K instances, with statistics of each dataset given in Table 2.

The Lost-in-the-middle dataset (Liu et al., 2023) includes examples from NaturalQuestions-Open (Kwiatkowski et al., 2019; Lee et al., 2019). Each instance consists of twenty Wikipedia passages, with only one passage containing the answer to the query. The remaining passages are distractors that are lexically similar but do not contain the answer. To assess the robustness of large language models (LLMs) to the position of relevant information, Liu et al. (2023) evaluated cases where the relevant passage appeared in positions 1, 5, 10, 15, and 20. Following their methodology, we sampled 500 instances for each position, resulting in a total of 2,500 instances.

FLenQA (Levy et al., 2024) is a benchmark that includes simple questions with answers of either "True" or "False" based on two key sentences. FLenQA includes three subtasks. The first subtask is Monotone Relations (MonoRel), where each instance asks whether a transitive relation between two entities holds based on the context (e.g., "Is X younger than Y?" based on the sentences "X is younger than Z" and "Z is younger than Y"). The second subtask, People In Rooms (PIR), involves one key sentence indicating that a person is in a specific room and another key sentence describing a property of this room. The question asks whether the person is in a room with the described property. The final subtask is Simplified Rule Taker (SRT), based on RuleTaker (Clark et al., 2020). Each instance consists of a logical rule, two sentences each introducing a fact, and a question over the rule and facts. For each subtask, FLenQA includes contexts with varying lengths, from 50 to 3,000 tokens, by simply adding irrelevant text, demonstrating consistent performance degradation with increased input length. In our experiments, we sampled 250 instances for each subtask with input lengths of 2,000 and 3,000 tokens, leading to a total of 1,500 instances for FLenQA.

In addition, we evaluate our method on common multi-hop QA benchmarks. We sampled 500 instances from each of the following datasets: HotPotQA (Yang et al., 2018), 2Wiki (Ho et al., 2020), and MuSiQue (Trivedi et al., 2021). In all these datasets, the input text includes multiple passages, and models need to perform at least two steps of reasoning over different passages in order to answer the question.

Models We apply DOUBLEDIPPER to a variety of models, both commercial and open-source. The commercial models include Gemini 1.5 Pro, Gemini 1.5 Flash and Gemini 1.0 Nano (Reid et al., 2024). The open-source models we tested are Llama 3.1 8B (Dubey et al., 2024), Gemma 2B (v2) and Gemma 9B (v2) (Riviere et al., 2024), Mistral-7B-Instruct (v0.2) (Jiang et al., 2023) and Mixtral-8x7B-Instruct (v0.1) (Jiang et al., 2024). Details about models' size and context window are shown in Table 1.

Few-shot generation in DOUBLEDIPPER is an auxiliary task and should ideally run in an efficient time without requiring heavy resources. Therefore, in our main experiments, we employ Gemma 2B to generate the demonstrations at it is the smallest and most efficient model used in our experiments. See Section 6 for an ablation analysis of the effect of the chosen model for generating the demonstrations.

Baselines We compare DOUBLEDIPPER to the vanilla baseline, which takes as input the entire context D and the query q and generates only the answer a. We also compare to *Zero-shot + Evidence Retrieval*, which prompts the model in a zero-shot setting to identify the relevant passage(s) before generating the answer.

Evaluation We evaluate each dataset with the original evaluation metrics. Namely, we report Accuracy for Lost-in-the-middle (Liu et al., 2023) and FLenQA (Levy et al., 2024), and Token F1 for HotPotQA (Yang et al., 2018), 2Wiki (Ho et al., 2020) and MuSique (Trivedi et al., 2021).

In addition to the task's accuracy, we also evaluate the performance of the identification of the supporting paragraph(s), by computing the F1 score on the predicted set of supporting passages compared to the ground truth (Yang et al., 2018; Ho et al., 2020; Trivedi et al., 2021).

	Avg.	2Wiki	MonoRel	PIR	SRT	HotPotQA	Lost	MuSique
Gemini Pro (vanilla)	60.5	24.9	95.0	97.6	64.4	46.7	71.6	23.1
Zero-shot + Evidence Retrieval	62.3	32.5	94.6	95.8	62.4	49.6	**74.8**	26.5
DOUBLEDIPPER	**70.4**	**46.8**	**97.4**	**99.0**	**79.6**	**60.9**	72.4	**36.4**
Gemini Flash (vanilla)	42.9	10.2	70.0	86.0	57.6	10.0	59.5	7.3
Zero-shot + Evidence Retrieval	58.2	30.2	78.8	90.6	65.0	44.9	**67.4**	30.2
DOUBLEDIPPER	**66.1**	**48.0**	**85.8**	**95.0**	**68.6**	**60.6**	65.0	**39.7**
Gemini Nano (vanilla)	41.6	10.8	72.2	66.8	55.4	21.3	59.6	5.2
Zero-shot + Evidence Retrieval	56.5	32.0	82.4	82.4	**56.4**	56.7	**60.5**	25.2
DOUBLEDIPPER	**62.1**	**40.6**	**86.6**	**95.4**	56.2	**65.1**	**60.5**	**30.4**
Gemma 2 2B (v2) (vanilla)	38.6	8.9	71.8	68.6	**51.2**	13.3	**49.6**	6.5
Zero-shot + Evidence Retrieval	42.0	22.3	66.8	70.6	40.2	30.6	47.6	16.2
DOUBLEDIPPER	**49.5**	**23.7**	**85.8**	**81.6**	50.0	**39.9**	46.7	**18.8**
Gemma 2 9B (v2) (vanilla)	44.0	11.4	74.8	81.8	55.6	13.8	61.0	9.3
Zero-shot + Evidence Retrieval	58.7	38.6	82.0	83.4	**59.4**	56.1	**64.4**	26.8
DOUBLEDIPPER	**61.2**	**41.7**	**84.0**	**95.0**	51.4	**61.2**	61.8	**33.3**
Llama 3.1 8B (vanilla)	37.2	11.8	56.2	52.2	48.6	20.7	**63.5**	7.4
Zero-shot + Evidence Retrieval	53.1	**42.2**	71.8	65.4	50.8	55.3	62.0	24.5
DOUBLEDIPPER	**59.9**	38.7	**91.2**	**90.6**	**51.0**	**59.3**	58.1	**30.1**
Mistral 7B (v0.3) (vanilla)	37.4	14.1	59.2	57.4	50.2	15.8	**60.8**	4.6
Zero-shot + Evidence Retrieval	44.0	23.8	66.2	62.4	49.6	34.1	58.6	13.6
DOUBLEDIPPER	**51.0**	**28.6**	**68.4**	**88.8**	**50.6**	**43.4**	60.7	**16.7**
Mixtral 7x8B (v0.1) (vanilla)	42.6	13.7	73.0	66.2	**51.0**	18.2	67.7	8.4
Zero-shot + Evidence Retrieval	47.4	18.8	81.8	73.6	50.6	26.3	**67.9**	13.1
DOUBLEDIPPER	**52.2**	**22.3**	**91.8**	**86.0**	47.8	**35.1**	66.6	**16.0**

Table 3: Accuracy of the QA task for the vanilla baseline (prompting the model to only answer the question), *Zero-shot + Evidence Retrieval* (prompting the model to explicitly identify the relevant passage(s) before generating the answer) and DOUBLEDIPPER with 3 demonstrations generated by Gemma 2 2B.

Implementation Details We randomly select three passages from the input (see Section 6 for an analysis of the number of self-generated demonstrations on the performance), each containing at least two sentences, and ask the model to generate a single QA pair for each passage (see Appendix A for the exact prompt). For all experiments, including few-shot generation and question-answering, we use a temperature setting of 0.

5 Results

Result 1: DOUBLEDIPPER offers a substantial performance boost. Table 3 presents the QA performance of the baseline, *Zero-shot + Evidence Retrieval* and DOUBLEDIPPER on our evaluation set. The results first show that prompting models to explicitly identify the relevant paragraphs before generating the answer (*Zero-shot + Evidence Retrieval*) leads to a performance improvement of 9.7 points on average across models over the vanilla baseline. DOUBLEDIPPER offers an additional substantial boost of 6.3 points for all models on average, culminating in a overall improvement of 16 absolute points over the vanilla baseline. Notably, while DOUBLEDIPPER produces simple QAs answerable from a single paragraph, *it always surpasses the baseline in multi-hop QA datasets* (HotPotQA, 2Wiki and MuSique). Likewise, DOUBLEDIPPER outperforms the baseline also on the FLenQA datasets (PIR, MonoRel and SRT), which involve synthetic True/False questions although the demonstrations in DOUBLEDIPPER are typically simple factoid questions.

Result 2: Learning to retrieve the evidence(s) with DOUBLEDIPPER is more effective in commercial models than in open source. Table 4 presents the performance of the supporting paragraphs prediction for the *Zero-shot + Evidence Retrieval* and DOUBLEDIPPER on our evaluation set. For all commercial models and Gemma 2B, DOUBLEDIPPER predicts better the supporting paragraphs than in the zero-shot setting (+2.6 F1 for Gemini Pro, +3.4 F1 for Gemini Flash, +1.8 F1 for Gemini Nano and +6.5 F1 for Gemma 2B). On the other hand, DOUBLEDIPPER slightly hurts the performance of common open source models (e.g., -2.8 F1 for Mistral and -0.3 F1 for Llama 3.1). This difference is because DOUBLEDIPPER's demonstrations are based on a *single* paragraph, where commercial models can generalize better and predict multiple evidences. Indeed, Gemini Pro predicts on average 2 evidences for the datasets 2Wiki, MuSique, PIR, MonoRel, SRT and HotPotQA whereas Gemma 9B predicts 1.2 evidences for each instance. In fact, the only dataset with a single evidence is "Lost" and almost all models

		Avg.	2Wiki	MonoRel	PIR	SRT	HotPotQA	Lost	MuSique
Gemini Pro	*Zero-shot + Evidence Retrieval*	83.7	**96.7**	97.7	**97.5**	62.8	**92.1**	63.6	75.3
	DOUBLEDIPPER	**86.3**	94.4	**99.8**	97.1	**80.9**	90.0	**66.4**	**75.4**
Gemini Flash	*Zero-shot + Evidence Retrieval*	75.7	82.3	90.5	72.6	70.4	80.9	**67.3**	65.9
	DOUBLEDIPPER	**79.1**	**83.7**	**98.3**	**80.2**	**71.2**	**84.6**	66.1	**69.5**
Gemini Nano	*Zero-shot + Evidence Retrieval*	68.1	76.3	77.6	71.6	**66.0**	74.1	55.8	55.2
	DOUBLEDIPPER	**69.9**	76.6	**86.6**	**74.5**	59.8	**76.0**	**59.2**	**56.7**
Gemma 2B	*Zero-shot + Evidence Retrieval*	39.1	**57.6**	49.4	44.3	**18.9**	53.1	14.0	36.4
	DOUBLEDIPPER	**45.6**	56.5	**60.7**	**57.5**	13.1	**58.1**	**33.1**	**39.9**
Gemma 9B	*Zero-shot + Evidence Retrieval*	**61.9**	**76.7**	**69.3**	**60.3**	**43.2**	76.5	51.5	55.9
	DOUBLEDIPPER	57.0	74.2	52.5	59.0	23.1	**78.4**	**55.3**	**56.8**
Llama 3.1 8B	*Zero-shot + Evidence Retrieval*	**61.7**	53.2	**86.5**	**66.9**	**67.8**	63.2	41.1	53.5
	DOUBLEDIPPER	61.4	**68.9**	71.4	54.9	52.8	**73.7**	**53.0**	**54.8**
Mistral 7B (v0.3)	*Zero-shot + Evidence Retrieval*	**46.6**	62.4	**49.8**	**46.2**	**17.5**	64.0	43.4	**43.0**
	DOUBLEDIPPER	43.8	**63.4**	33.8	42.5	4.2	**66.7**	**55.6**	40.2
Mixtral 7x8B v(0.1)	*Zero-shot + Evidence Retrieval*	**60.0**	**72.4**	69.4	**64.6**	**43.4**	**76.9**	40.6	52.4
	DOUBLEDIPPER	58.9	70.4	**81.6**	63.6	18.3	75.0	**50.4**	**53.2**

Table 4: Performance (F1) of supporting paragraph(s) prediction.

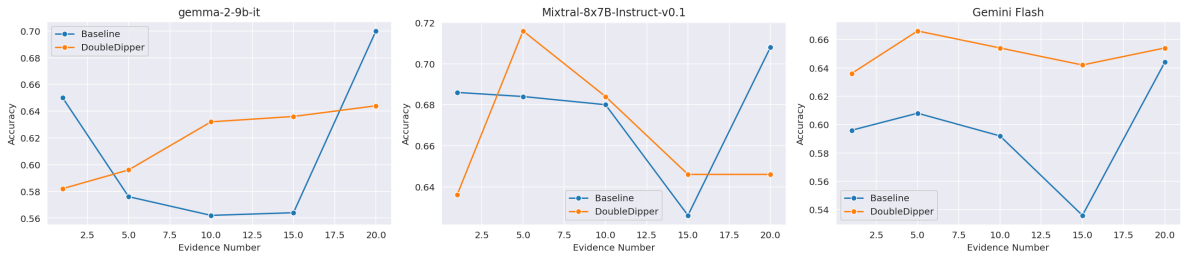

Figure 4: Performance (accuracy) of Gemma 2 9B, Mixtral 8x7B and Gemini Flash with and without DOUBLEDIPPER on our sample of the Lost-in-the-middle dataset (Liu et al., 2023) according to the position of the document that contains the answer.

highly benefit from DOUBLEDIPPER (e.g, +11.9 F1 for Llama 3.1, +12.2 F1 for Mistral 7B and +9.8 for Mixtral 7x8B).

Result 3: DOUBLEDIPPER makes models more robust to the position of relevant information. Following Liu et al. (2023), Figure 4 shows the performance of Gemma 2 9B, Mixtral 8x7B and Gemini Flash for both the baseline and DOUBLEDIPPER on our sample of the Lost-of-the-middle dataset, according to the position of the document that contains the answer. See Appendix B for the performance curve of the other tested models, which show similar trends.

Overall, the performance curve for DOUBLEDIPPER consistently surpasses the baseline when the relevant information appears "in the middle" and sometimes also at the beginning and/or the end (e.g., Gemini Flash). This variation can likely be attributed to the inherent biases of LLMs towards the beginning and end of inputs, while adding in context demonstrations mitigates this bias. This reveals that beyond improving performance, DOUBLEDIPPER can make the model more robust to the position of the relevant document.

	Gemma 2B	Self	Gemini Pro
Gemini Pro	70.4	**71.6**	**71.6**
Gemini Nano	62.1	61.7	**62.8**
Gemini Flash	66.1	67.5	**68.1**
Gemma 2B	49.5	49.5	**51.0**
Gemma 9B	61.2	62.0	**63.5**
Llama 3.1	59.9	60.1	**61.5**
Mistral v0.3	51.0	49.8	**52.6**
Mixtral	52.2	49.8	**54.4**

Table 5: Average performance of DOUBLEDIPPER with different models for generating the demonstrations. See Appendix C.2 for the results on each evaluation dataset.

6 Ablation Studies

How many examples are needed? In Table 6, we explore the impact of varying k, the number of self-generated few-shot examples in DOUBLEDIPPER to 1, 3, 5, and 10. On average, a single demonstration already provides an improvement over the baseline. Adding 3 demonstrations adds another

	$k=1$	$k=3$	$k=5$	$k=10$
Gemini Nano	60.0	62.1	62.2	62.3
Gemini Flash	65.3	66.1	65.9	66.1
Gemma 2B (v2)	47.0	49.5	49.6	49.9
Gemma 9B (v2)	58.7	61.2	61.4	61.3
Llama 3.1	57.7	59.9	60.6	61.4
Mistral 7B (v0.3)	48.9	51.0	51.1	51.4
Mixtral 7x8B (v0.1)	49.3	52.2	51.7	52.2

Table 6: Average performance on our evaluation set with various numbers of self-generated few shot demonstrations (k) in DOUBLEDIPPER. See Appendix C.1 for the results on each evaluation dataset.

boost of 2 points, while increasing the number of demonstrations to 5 and 10 leads to a marginal improvement. This finding is in line with previous work (Brown et al., 2020a; Min et al., 2022b). We conclude that a small number of examples carries most of the benefit with our method, but given additional computation budget, adding more examples does carry additional minor benefit.

Investigating the effect of the few-shot generator To understand the impact of the default chosen model (Gemma 2 2B) for generating the demonstrations, we conducted two additional experiments. The first experiment is SELF in which we use the same model for generating the demonstrations and for answering the original question. In the second experiment, we generate the demonstrations with the best LLM used in our experiments, namely Gemini Pro. The average results are reported in Table 5 and the performance for each evaluation dataset is presented in Appendix C.2. The results show that generating the demonstrations with Gemma 2 or SELF achieves similar performance, while Gemini Pro leads to a consistent increase in performance across models, indicating that future better models can improve further the performance.

As mentioned in Section 3, DOUBLEDIPPER promotes also efficiency by adding to the original prompt only a few extra tokens, leading to a significantly cheaper inference than the traditional ICL.

DOUBLEDIPPER without identification of supporting paragraphs To ablate the second principle in DOUBLEDIPPER, which is the explicit identification of the supporting paragraphs before generating the answer, we prompt the open source models with self-generated few shot examples that comprise only question-answer pairs (without instructing the model to retrieve the relevant passage(s)). For all models, the average QA performance of DOUBLEDIPPER without the evidence is lower than DOUBLEDIPPER with evidence identification. When averaging results across models and datasets, we report a substantial F1 drop from 54.8 to 46.6. Full results are available in Appendix C.3 in Table 9.

Qualitative analysis: Correctness of the generated QA pairs We manually analyze 150 QAs generated by Gemma 2B as demonstrations. Our review confirms that 93.5% of these self-generated QAs are correct, meaning that the question is meaningful and the answer could be found in the corresponding paragraph.

7 Conclusion

We develop DOUBLEDIPPER, a straightforward method for enhancing the performance of Question Answering with long context and providing attribution to the relevant paragraph(s) in the input. By recycling the input context to generate the few shot examples, each demonstration includes solely a question, an answer and a pointer to the relevant paragraph, without a separate context, thus effectively addressing the challenging of In-Context-Learning with long context. Experimental results show that our approach ourperforms both the vanilla LLM and the traditional ICL in various QA settings, including distractor passages in the input, True/False questions and multi-hop QA.

8 Limitations

One notable limitation of our approach is the extended inference time required for generating question-answer pairs. Future research could mitigate this issue by developing smaller, specialized models specifically tailored for QA generation.

Additionally, our evaluation set is constrained to instances that are solely in English and range between 1,000 to 4,000 tokens. Expanding the diversity of languages and token ranges could enhance the robustness and applicability of our findings.

Lastly, although we employ a strategy of randomly sampling k paragraphs from the input to ensure the model engages with varied segments of the text, we did not optimize the selection of these paragraphs. Future work could explore more strategic methods for paragraph selection to potentially enhance the efficacy and relevance of the generated examples.

References

Rishabh Agarwal, Avi Singh, Lei M. Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, John D. Co-Reyes, Eric Chu, Feryal Behbahani, Aleksandra Faust, and Hugo Larochelle. 2024. Many-shot in-context learning. *Preprint*, arXiv:2404.11018.

Chenxin An, Shansan Gong, Ming Zhong, Xingjian Zhao, Mukai Li, Jun Zhang, Lingpeng Kong, and Xipeng Qiu. 2023. L-eval: Instituting standardized evaluation for long context language models. *Preprint*, arXiv:2307.11088.

an Luo, Xin Xu, Yue Liu, Panupong Pasupat, and Mehran Kazemi. 2024. In-context learning with retrieved demonstrations for language models: A survey. *ArXiv*, abs/2401.11624.

Cem Anil, Yuhuai Wu, Anders Andreassen, Aitor Lewkowycz, Vedant Misra, Vinay Venkatesh Ramasesh, Ambrose Slone, Guy Gur-Ari, Ethan Dyer, and Behnam Neyshabur. 2022. Exploring length generalization in large language models. *ArXiv*, abs/2207.04901.

Yushi Bai, Xin Lv, Jiajie Zhang, Hong 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. *ArXiv*, abs/2308.14508.

Jennifer A Bishop, Qianqian Xie, and Sophia Ananiadou. 2023. Longdocfactscore: Evaluating the factuality of long document abstractive summarisation. *ArXiv*, abs/2309.12455.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020a. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

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, Jeff 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. 2020b. Language models are few-shot learners. *ArXiv*, abs/2005.14165.

Avi Caciularu, Ido Dagan, Jacob Goldberger, and Arman Cohan. 2022. Long context question answering via supervised contrastive learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2872–2879, Seattle, United States. Association for Computational Linguistics.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2023. Longlora: Efficient fine-tuning of long-context large language models. *ArXiv*, abs/2309.12307.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2020. Transformers as soft reasoners over language. In *International Joint Conference on Artificial Intelligence*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Cantón Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab A. AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriele Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guanglong Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Laurens Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Ju-Qing Jia, Kalyan Vasuden Alwala, K. Upasani, Kate Plawiak, Keqian Li, Ken-591 neth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline C. Muzzi, Mahesh Babu Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melissa Hall Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri S. Chatterji, Olivier Duchenne, Onur

cCelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasić, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Chandra Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yiqian Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zhengxu Yan, Zhengxing Chen, Zoe Papakipos, Aaditya K. Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adi Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Ben Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Shang-Wen Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzm'an, Frank J. Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory G. Sizov, Guangyi Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Han Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kaixing(Kai) Wu, U KamHou, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, A Lavender, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollár, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sung-Bae Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Andrei Poenaru, Vlad T. Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xia Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The llama 3 herd of models. *ArXiv*, abs/2407.21783.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent

Zhao, N. Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. 2022. Rarr: Researching and revising what language models say, using language models. In *Annual Meeting of the Association for Computational Linguistics*.

Omer Goldman, Alon Jacovi, Aviv Slobodkin, Aviya Maimon, Ido Dagan, and Reut Tsarfaty. 2024. Is it really long context if all you need is retrieval? towards genuinely difficult long context nlp. *ArXiv*, abs/2407.00402.

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. *Preprint*, arXiv:2311.09198.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Cheng-Yu Hsieh, Yung-Sung Chuang, Chun-Liang Li, Zifeng Wang, Long Le, Abhishek Kumar, James Glass, Alexander Ratner, Chen-Yu Lee, Ranjay Krishna, and Tomas Pfister. 2024. Found in the middle: Calibrating positional attention bias improves long context utilization. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 14982–14995, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

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, L'elio 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. *ArXiv*, abs/2401.04088.

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

Hyuhng Joon Kim, Hyunsoo Cho, Junyeob Kim, Taeuk Kim, Kang Min Yoo, and Sang goo Lee. 2022. Self-generated in-context learning: Leveraging auto-regressive language models as a demonstration generator. *ArXiv*, abs/2206.08082.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 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, Florence, Italy. Association for Computational Linguistics.

Mosh Levy, Alon Jacoby, and Yoav Goldberg. 2024. Same task, more tokens: the impact of input length on the reasoning performance of large language models. *ArXiv*, abs/2402.14848.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Jiaqi Li, Mengmeng Wang, Zilong Zheng, and Muhan Zhang. 2023. Loogle: Can long-context language models understand long contexts? *ArXiv*, abs/2311.04939.

Junlong Li, Jinyuan Wang, Zhuosheng Zhang, and Hai Zhao. 2024a. Self-prompting large language models for zero-shot open-domain QA. 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 296–310, Mexico City, Mexico. Association for Computational Linguistics.

Rui Li, Guoyin Wang, and Jiwei Li. 2024b. Are human-generated demonstrations necessary for in-context learning? In *The Twelfth International Conference on Learning Representations*.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. 2024c. Long-context llms struggle with long in-context learning. *ArXiv*, abs/2404.02060.

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. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022a. Rethinking the role of demonstrations: What makes in-context learning work? *ArXiv*, abs/2202.12837.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022b. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Bhargavi Paranjape, Scott M. Lundberg, Sameer Singh, Hannaneh Hajishirzi, Luke Zettlemoyer, and Marco Tulio Ribeiro. 2023. Art: Automatic multi-step reasoning and tool-use for large language models. *ArXiv*, abs/2303.09014.

Ofir Press, Noah A. Smith, and Mike Lewis. 2021. Train short, test long: Attention with linear biases enables input length extrapolation. *ArXiv*, abs/2108.12409.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy P. Lillicrap, Jean-Baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, Ioannis Antonoglou, Rohan Anil, Sebastian Borgeaud, Andrew M. Dai, Katie Millican, Ethan Dyer, Mia Glaese, Thibault Sottiaux, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, James Molloy, Jilin Chen, Michael Isard, Paul Barham, Tom Hennigan, Ross McIlroy, Melvin Johnson, Johan Schalkwyk, Eli Collins, Eliza Rutherford, Erica Moreira, Kareem W. Ayoub, Megha Goel, Clemens Meyer, Gregory Thornton, Zhen Yang, Henryk Michalewski, Zaheer Abbas, Nathan Schucher, Ankesh Anand, Richard Ives, James Keeling, Karel Lenc, Salem Haykal, Siamak Shakeri, Pranav Shyam, Aakanksha Chowdhery, Roman Ring, Stephen Spencer, Eren Sezener, Luke Vilnis, Oscar Chang, Nobuyuki Morioka, George Tucker, Ce Zheng, Oliver Woodman, Nithya Attaluri, Tomás Kociský, Evgenii Eltyshev, Xi Chen, Timothy Chung, Vittorio Selo, Siddhartha Brahma, Petko Georgiev, Ambrose Slone, Zhenkai Zhu, James Lottes, Siyuan Qiao, Ben Caine, Sebastian Riedel, Alex Tomala, Martin Chadwick, J Christopher Love, Peter Choy, Sid Mittal, Neil Houlsby, Yunhao Tang, Matthew Lamm, Libin Bai, Qiao Zhang, Luheng He, Yong Cheng, Peter Humphreys, Yujia Li, Sergey Brin, Albin Cassirer, Ying-Qi Miao, Lukás Zilka, Taylor Tobin, Kelvin Xu, Lev Proleev, Daniel Sohn, Alberto Magni, Lisa Anne Hendricks, Isabel Gao, Santiago Ontan'on, Oskar Bunyan, Nathan Byrd, Abhanshu Sharma, Biao Zhang, Mario Pinto, Rishika Sinha, Harsh Mehta, Dawei Jia, Sergi Caelles, Albert Webson, Alex Morris, Becca Roelofs, Yifan Ding, Robin Strudel, Xuehan Xiong, Marvin Ritter, Mostafa Dehghani, Rahma Chaabouni, Abhijit Karmarkar, Guangda Lai, Fabian Mentzer, Bibo Xu, YaGuang Li, Yujing Zhang, Tom Le Paine, Alex Goldin, Behnam Neyshabur, Kate Baumli, Anselm Levskaya, Michael Laskin, Wenhao Jia, Jack W. Rae, Kefan Xiao, Antoine He, Skye Giordano, Lakshman Yagati, Jean-Baptiste Lespiau, Paul Natsev, Sanjay Ganapathy, Fangyu Liu, Danilo Martins, Nanxin Chen, Yunhan Xu, Megan Barnes, Rhys May, Arpi Vezer, Junhyuk Oh, Ken Franko, Sophie Bridgers, Ruizhe Zhao, Boxi Wu, Basil Mustafa, Sean Sechrist, Emilio Parisotto, Thanumalayan Sankaranarayana Pillai, Chris Larkin, Chenjie Gu, Christina Sorokin, Maxim Krikun, Alexey Guseynov, Jessica Landon, Romina Datta, Alexander Pritzel, Phoebe Thacker, Fan Yang, Kevin Hui, A.E. Hauth, Chih-Kuan Yeh, David Barker, Justin Mao-Jones, Sophia Austin, Hannah Sheahan, Parker Schuh, James Svensson, Rohan Jain, Vinay Venkatesh Ramasesh, Anton Briukhov, Da-Woon Chung, Tamara von Glehn, Christina Butterfield, Priya Jhakra, Matt Wiethoff, Justin Frye, Jordan Grimstad, Beer Changpinyo, Charline Le Lan, Anna Bortsova, Yonghui Wu, Paul Voigtlaender, Tara N. Sainath, Charlotte Smith, Will Hawkins, Kris Cao, James Besley, Srivatsan Srinivasan, Mark Omernick, Colin Gaffney, Gabriela de Castro Surita, Ryan Burnell, Bogdan Damoc, Junwhan Ahn, Andrew Brock, Mantas Pajarskas, Anastasia Petrushkina, Seb Noury, Lorenzo Blanco, Kevin Swersky, Arun Ahuja, Thi Avrahami, Vedant Misra, Raoul de Liedekerke, Mariko Iinuma, Alex Polozov, Sarah York, George van den Driessche, Paul Michel, Justin Chiu, Rory Blevins, Zach Gleicher, Adrià Recasens, Alban Rrustemi, Elena Gribovskaya, Aurko Roy, Wiktor Gworek, S'ebastien M. R. Arnold, Lisa Lee, James Lee-Thorp, Marcello Maggioni, Enrique Piqueras, Kartikeya Badola, Sharad Vikram, Lucas Gonzalez, Anirudh Baddepudi, Evan Senter, Jacob Devlin, James Qin, Michael Azzam, Maja Trebacz, Martin Polacek, Kashyap Krishnakumar, Shuo yiin Chang, Matthew Tung, Ivo Penchev, Rishabh Joshi, Kate Olszewska, Carrie Muir, Mateo Wirth, Ale Jakse Hartman, Joshua Newlan, Sheleem Kashem, Vijay Bolina, Elahe Dabir, Joost R. van Amersfoort, Zafarali Ahmed, James Cobon-Kerr, Aishwarya B Kamath, Arnar Mar Hrafnkelsson, Le Hou, Ian Mackinnon, Alexandre Frechette, Eric Noland, Xiance Si, Emanuel Taropa, Dong Li, Phil Crone, Anmol Gulati, S'ebastien Cevey, Jonas Adler, Ada Ma, David Silver, Simon Tokumine, Richard Powell, Stephan Lee, Michael B. Chang, Samer Hassan, Diana Mincu, Antoine Yang, Nir Levine, Jenny Brennan, Mingqiu Wang, Sarah Hodkinson, Jeffrey Zhao, Josh Lipschultz, Aedan Pope, Michael B. Chang, Cheng Li, Laurent El Shafey, Michela Paganini, Sholto Douglas, Bernd Bohnet, Fabio Pardo, Seth Odoom, Mihaela Rosca, Cicero Nogueira dos Santos, Kedar Soparkar, Arthur Guez, Tom Hudson, Steven Hansen, Chulayuth Asawaroengchai, Ravichandra Addanki, Tianhe Yu, Wojciech Stokowiec, Mina Khan, Justin Gilmer, Jaehoon Lee, Carrie Grimes Bostock, Keran Rong, Jonathan Caton, Pedram Pejman, Filip Pavetic, Geoff Brown, Vivek Sharma, Mario Luvci'c, Rajkumar Samuel, Josip Djolonga, Amol Mandhane, Lars Lowe Sjosund, Elena Buchatskaya, Elspeth White, Natalie Clay, Jiepu Jiang, Hyeontaek Lim, Ross Hemsley, Jane Labanowski, Nicola De Cao, David Steiner, Sayed Hadi Hashemi, Jacob Austin, Anita Gergely, Tim Blyth, Joe Stanton, Kaushik

Shivakumar, Aditya Siddhant, Anders Andreassen, Carlos L. Araya, Nikhil Sethi, Rakesh Shivanna, Steven Hand, Ankur Bapna, Ali Khodaei, Antoine Miech, Garrett Tanzer, Andy Swing, Shantanu Thakoor, Zhufeng Pan, Zachary Nado, Stephanie Winkler, Dian Yu, Mohammad Saleh, Lorenzo Maggiore, Iain Barr, Minh Giang, Thais Kagohara, Ivo Danihelka, Amit Marathe, Vladimir Feinberg, Mohamed Elhawaty, Nimesh Ghelani, Dan Horgan, Helen Miller, Lexi Walker, Richard Tanburn, Mukarram Tariq, Disha Shrivastava, Fei Xia, Chung-Cheng Chiu, Zoe C. Ashwood, Khuslen Baatarsukh, Sina Samangooei, Fred Alcober, Axel Stjerngren, Paul Komarek, Katerina Tsihlas, Anudhyan Boral, Ramona Comanescu, Jeremy Chen, Ruibo Liu, Dawn Bloxwich, Charlie Chen, Yanhua Sun, Fangxiaoyu Feng, Matthew Mauger, Xerxes Dotiwalla, Vincent Hellendoorn, Michael Sharman, Ivy Zheng, Krishna Haridasan, Gabriel Barth-Maron, Craig Swanson, Dominika Rogozi'nska, Alek Andreev, Paul Kishan Rubenstein, Ruoxin Sang, Dan Hurt, Gamaleldin Elsayed, Ren shen Wang, Dave Lacey, Anastasija Ili'c, Yao Zhao, Woohyun Han, Lora Aroyo, Chimezie Iwuanyanwu, Vitaly Nikolaev, Balaji Lakshminarayanan, Sadegh Jazayeri, Raphael Lopez Kaufman, Mani Varadarajan, Chetan Tekur, Doug Fritz, Misha Khalman, David Reitter, Kingshuk Dasgupta, Shourya Sarcar, T. Ornduff, Javier Snaider, Fantine Huot, Johnson Jia, Rupert Kemp, Nejc Trdin, Anitha Vijayakumar, Lucy Kim, Christof Angermueller, Li Lao, Tianqi Liu, Haibin Zhang, David Engel, Somer Greene, Anais White, Jessica Austin, Lilly Taylor, Shereen Ashraf, Dangyi Liu, Maria Georgaki, Irene Cai, Yana Kulizhskaya, Sonam Goenka, Brennan Saeta, Kiran Vodrahalli, Christian Frank, Dario de Cesare, Brona Robenek, Harry Richardson, Mahmoud Alnahlawi, Christopher Yew, Priya Ponnapalli, Marco Tagliasacchi, Alex Korchemniy, Yelin Kim, Dinghua Li, Bill Rosgen, Kyle Levin, Jeremy Wiesner, Praseem Banzal, Praveen Srinivasan, Hongkun Yu, cCauglar Unlu, David Reid, Zora Tung, Daniel F. Finchelstein, Ravin Kumar, Andre Elisseeff, Jin Huang, Ming Zhang, Rui Zhu, Ricardo Aguilar, Mai Gim'enez, Jiawei Xia, Olivier Dousse, Willi Gierke, Soheil Hassas Yeganeh, Damion Yates, Komal Jalan, Lu Li, Eri Latorre-Chimoto, Duc Dung Nguyen, Ken Durden, Praveen Kallakuri, Yaxin Liu, Matthew Johnson, Tomy Tsai, Alice Talbert, Jasmine Liu, Alexander Neitz, Chen Elkind, Marco Selvi, Mimi Jasarevic, Livio Baldini Soares, Albert Cui, Pidong Wang, Alek Wenjiao Wang, Xinyu Ye, Krystal Kallarackal, Lucia Loher, Hoi Lam, Josef Broder, Daniel Niels Holtmann-Rice, Nina Martin, Bramandia Ramadhana, Daniel Toyama, Mrinal Shukla, Sujoy Basu, Abhi Mohan, Nicholas Fernando, Noah Fiedel, Kim Paterson, Hui Li, Ankush Garg, Jane Park, Donghyun Choi, Diane Wu, Sankalp Singh, Zhishuai Zhang, Amir Globerson, Lily Yu, John Carpenter, Félix de Chaumont Quitry, Carey Radebaugh, Chu-Cheng Lin, Alex Tudor, Prakash Shroff, Drew Garmon, Dayou Du, Neera Vats, Han Lu, Shariq Iqbal, Alexey Yakubovich, Nilesh Tripuraneni, James Manyika, Haroon Qureshi, Nan Hua, Christel Ngani, Maria Abi Raad, Hannah Forbes, Anna Bulanova, Jeff Stanway, Mukund Sundararajan, Victor Ungureanu, Colton Bishop, Yunjie Li, Balaji Venkatraman, Bo Li, Chloe Thornton, Salvatore Scellato, Nishesh Gupta, Yicheng Wang, Ian Tenney, Xihui Wu, Ashish Shenoy, Gabriel Carvajal, Diana Gage Wright, Ben Bariach, Zhuyun Xiao, Peter Hawkins, Sid Dalmia, Cl'ement Farabet, Pedro Valenzuela, Quan Yuan, Christoper A. Welty, Ananth Agarwal, Mianna Chen, Wooyeol Kim, Brice Hulse, Nandita Dukkipati, Adam Paszke, Andrew Bolt, Elnaz Davoodi, Kiam Choo, Jennifer Beattie, Jennifer Prendki, Harsha Vashisht, Rebeca Santamaria-Fernandez, Luis C. Cobo, Jarek Wilkiewicz, David Madras, Ali Elqursh, Grant Uy, Kevin Ramirez, Matt Harvey, Tyler Liechty, Heiga Zen, Jeff Seibert, Clara Huiyi Hu, A. Ya. Khorlin, Maigo Le, Asaf Aharoni, Megan Li, Lily Wang, Sandeep Kumar, Alejandro Lince, Norman Casagrande, Jay Hoover, Dalia El Badawy, David Soergel, Denis Vnukov, Matt Miecnikowski, Jiří ima, Anna Koop, Praveen Kumar, Thibault Sellam, Daniel Vlasic, Samira Daruki, Nir Shabat, John Zhang, Guolong Su, Kalpesh Krishna, Jiageng Zhang, Jeremiah Liu, Yi Sun, Evan Palmer, Alireza Ghaffarkhah, Xi Xiong, Victor Cotruta, Michael Fink, Lucas Dixon, Ashwin Sreevatsa, Adrian Goedeckemeyer, Alek Dimitriev, Mohsen Jafari, Remi Crocker, Nicholas Fitzgerald, Aviral Kumar, Sanjay Ghemawat, Ivan Philips, Frederick Liu, Yannie Liang, Rachel Sterneck, Alena Repina, Marcus Wu, Laura Knight, Marin Georgiev, Hyo Lee, Harry Askham, Abhishek Chakladar, Annie Louis, Carl Crous, Hardie Cate, Dessie Petrova, Michael Quinn, Denese Owusu-Afriyie, Achintya Singhal, Nan Wei, Solomon Kim, Damien Vincent, Milad Nasr, Christopher A. Choquette-Choo, Reiko Tojo, Shawn Lu, Diego de Las Casas, Yuchung Cheng, Tolga Bolukbasi, Katherine Lee, Saaber Fatehi, Rajagopal Ananthanarayanan, Miteyan Patel, Charbel El Kaed, Jing Li, Jakub Sygnowski, Shreyas Rammohan Belle, Zhe Chen, Jaclyn Konzelmann, Siim Poder, Roopal Garg, Vinod Koverkathu, Adam Brown, Chris Dyer, Rosanne Liu, Azade Nova, Jun Xu, Junwen Bai, Slav Petrov, Demis Hassabis, Koray Kavukcuoglu, Jeffrey Dean, Oriol Vinyals, and Alexandra Chronopoulou. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *ArXiv*, abs/2403.05530.

Gemma Team Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, L'eonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ram'e, Johan Ferret, Peter Liu, Pouya Dehghani Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stańczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot,

Boxi Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Christoper A. Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozi'nska, D. Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Pluci'nska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost R. van Amersfoort, Josh Gordon, Josh Lipschultz, Joshua Newlan, Junsong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, L. Sifre, L. Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Gorner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, S. Mc Carthy, Sarah Perrin, S'ebastien Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomás Kociský, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Brian Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeffrey Dean, Demis Hassabis, Koray Kavukcuoglu, Cl'ement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. 2024. Gemma 2: Improving open language models at a practical size. *ArXiv*, abs/2408.00118.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671, Seattle, United States. 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*, pages 7977–7989, Singapore. Association for Computational Linguistics.

Uri Shaham, Elad Segal, Maor Ivgi, Avia Efrat, Ori Yoran, Adi Haviv, Ankit Gupta, Wenhan Xiong, Mor Geva, Jonathan Berant, and Omer Levy. 2022. SCROLLS: Standardized CompaRison over long language sequences. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 12007–12021, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Aviv Slobodkin, Eran Hirsch, Arie Cattan, Tal Schuster, and Ido Dagan. 2024. Attribute first, then generate: Locally-attributable grounded text generation. *ArXiv*, abs/2403.17104.

Yisheng Song, Ting-Yuan Wang, Puyu Cai, Subrota Kumar Mondal, and Jyoti Prakash Sahoo. 2022. A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities. *ACM Computing Surveys*, 55:1 – 40.

H. Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2021. Musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Neural Information Processing Systems*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Michihiro Yasunaga, Xinyun Chen, Yujia Li, Panupong Pasupat, Jure Leskovec, Percy Liang, Ed H. Chi, and Denny Zhou. 2024. Large language models as analogical reasoners. In *The Twelfth International Conference on Learning Representations*.

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. 2024. Helmet: How to evaluate long-context language models effectively and thoroughly.

A Prompts

Figure 5 shows the zero-shot prompt we use for generating the question-answer pairs in DOUBLEDIPPER. For the QA prompts, we use the same instructions and prompt template as the original papers (Lost-in-the-middle and FLenQA) and add a simple line for the instructions in other multi-hop QA datasets: *"Please answer the question based on the given passages below."*. For MuSique, since the dataset includes questions that are not answerable, we add the following sentence to the prompt: *"If the question can't be answered given the given passages, please write "unanswerable""*.

B Lost-in-the-middle

Figure 6 shows the QA accuracy of the models Gemma 2 2B, Mistral 7B, Gemini Nano and Gemini Pro on our subset of the "Lost-in-the-middle" dataset.

C Analysis

C.1 Impact of the Number of Demonstrations in DOUBLEDIPPER

Table 7 presents the results of DOUBLEDIPPER with 1, 3 (main experiment in the paper), 5 and 10 generated demonstrations. For all these experiments, the demonstrations were generated by Gemma 2 2B.

Figure 7 shows the QA accuracy of DOUBLEDIPPER on "Lost" according to the position of the relevant passage for each $k \in \{1, 3, 5, 10\}$.

C.2 Impact of the few-shot generator

Table 8 presents the detailed QA performance of all models with different models for generating DOUBLEDIPPER's demonstrations. As mentioned in the paper (Section 6), generating the demonstrations with the best model (ie. Gemini Pro) achieves the best performance overall.

C.3 Impact of the identification of supporting paragraphs in the QA generation

Table 9 compares the performance of DOUBLEDIPPER to DOUBLEDIPPER without evidence identification.

		Avg.	2Wiki	MonoRel	PIR	SRT	HotPotQA	Lost	MuSique
Gemini Nano	$k = 1$	60.03	37.68	85.20	88.20	56.40	62.50	60.68	29.56
	$k = 3$	62.12	40.55	86.60	95.40	56.20	65.12	60.52	30.44
	$k = 5$	62.25	41.79	87.00	95.60	55.20	65.05	60.56	30.52
	$k = 10$	62.33	43.16	86.00	96.20	55.20	65.37	60.44	29.96
Gemini Flash	$k = 1$	65.34	48.83	84.80	90.80	66.40	60.17	65.16	41.19
	$k = 3$	66.11	48.03	85.80	95.00	68.60	60.58	65.04	39.71
	$k = 5$	65.87	47.49	87.20	95.00	66.60	61.13	64.20	39.48
	$k = 10$	66.08	46.13	86.20	95.20	69.00	62.03	63.80	40.18
Gemma 2B (v2)	$k = 1$	47.05	24.05	73.60	77.00	50.20	41.32	47.04	16.13
	$k = 3$	49.48	23.66	85.80	81.60	50.00	39.85	46.68	18.77
	$k = 5$	49.59	26.70	85.40	80.40	48.40	40.34	46.56	19.30
	$k = 10$	49.92	26.43	85.80	81.40	49.00	40.91	47.68	18.26
Gemma 9B (v2)	$k = 1$	58.77	37.58	81.00	87.60	51.60	58.46	63.44	31.68
	$k = 3$	61.20	41.70	84.00	95.00	51.40	61.21	61.80	33.31
	$k = 5$	61.40	44.42	84.20	94.80	50.00	62.99	60.76	32.60
	$k = 10$	61.34	43.78	84.40	95.80	50.40	63.01	59.96	32.05
Llama 3.1 8B	$k = 1$	57.74	38.20	85.40	84.80	51.80	57.50	60.96	25.49
	$k = 3$	59.86	38.75	91.20	90.60	51.00	59.29	58.12	30.09
	$k = 5$	60.58	41.56	89.80	91.40	51.00	61.31	57.68	31.28
	$k = 10$	61.41	44.68	89.00	91.00	51.00	63.88	56.36	33.98
Mistral 7B	$k = 1$	48.91	28.86	67.60	77.40	50.00	40.62	62.00	15.91
	$k = 3$	51.03	28.65	68.40	88.80	50.60	43.44	60.68	16.66
	$k = 5$	51.12	29.86	69.20	89.00	49.40	43.83	60.36	16.23
	$k = 10$	51.44	30.14	68.20	89.40	49.00	46.49	59.96	16.91
Mixtral 7x8B	$k = 1$	49.26	19.83	86.00	74.80	49.40	31.73	69.00	14.07
	$k = 3$	52.22	22.31	91.80	86.00	47.80	35.10	66.56	15.97
	$k = 5$	51.65	23.55	91.40	82.40	47.20	34.64	64.92	17.46
	$k = 10$	52.18	23.49	91.80	84.40	47.00	36.23	64.00	18.32

Table 7: Performance of DOUBLEDIPPER on our evaluation set with various numbers of self-generated few shot demonstrations (k).

		Avg.	2Wiki	MonoRel	PIR	SRT	HotPotQA	Lost	MuSique
Gemini Pro	Gemma 2B	70.4	46.8	97.4	99.0	**79.6**	60.9	72.4	36.4
	Self	**71.6**	**49.6**	**98.2**	**99.6**	78.8	**62.5**	**72.5**	**40.1**
Gemini Flash	Gemma 2B	66.1	48.0	85.8	95.0	68.6	60.6	65.0	39.7
	Self	67.5	49.5	88.6	**96.8**	65.6	64.3	**65.2**	**42.5**
	Gemini Pro	**68.1**	**50.3**	**90.4**	96.0	**69.0**	**64.8**	64.8	41.6
Gemini Nano	Gemma 2B	62.1	**40.6**	86.6	95.4	**56.2**	65.1	60.5	30.4
	Self	61.7	39.8	85.8	95.2	**56.2**	64.3	**60.7**	30.0
	Gemini Pro	**62.8**	40.4	**87.2**	**99.0**	54.4	**66.8**	60.4	**31.7**
Gemma 2 2B	Self	49.5	23.7	**85.8**	81.6	**50.0**	39.9	**46.7**	**18.8**
	Gemini Pro	**51.0**	**25.2**	83.8	**93.4**	48.0	**41.7**	46.2	18.4
Gemma 2 9B	Gemma 2B	61.2	41.7	84.0	95.0	51.4	61.2	61.8	33.3
	Self	62.0	44.8	82.4	97.0	50.6	62.9	61.5	**34.9**
	Gemini Pro	**63.5**	**46.2**	**86.4**	**99.2**	**52.2**	**64.4**	**61.4**	34.8
Llama 3.1 8B	Gemma 2B	59.9	38.7	**91.2**	90.6	51.0	59.3	**58.1**	30.1
	Self	60.1	**41.3**	88.6	89.8	50.8	61.5	57.8	31.0
	Gemini Pro	**61.5**	41.0	89.8	**95.6**	**51.6**	**63.5**	57.5	**31.7**
Mistral 7B (v0.3)	Gemma 2B	51.0	28.6	68.4	88.8	**50.6**	43.4	60.7	16.7
	Self	49.8	25.2	**68.8**	93.8	49.6	36.5	60.6	14.3
	Gemini Pro	**52.6**	**31.4**	67.2	**96.4**	48.2	**45.7**	**62.1**	**17.0**
Mixtral 7x8B (v0.1)	Gemma 2B	52.2	**22.3**	91.8	86.0	47.8	35.1	66.6	16.0
	Self	49.8	16.0	91.8	86.0	48.8	25.5	**67.8**	12.8
	Gemini Pro	**54.4**	20.8	**94.8**	**96.2**	**49.0**	**36.3**	66.6	**16.9**

Table 8: Performance of DOUBLEDIPPER according to the model for generating the demonstrations (Gemma 2B, Self or Gemini Pro).

```
Given the following TEXT, please write a simple question whose answer appears verbatim in
    the text.The question should include enough information so that it can be understood
    without the text.The answer should be concise.Please write both the question ans answer
    in the following format:
Q:
A:
TEXT: [PARAGRAPH]
```

Figure 5: Template prompt for generating the QA pairs in DOUBLEDIPPER.

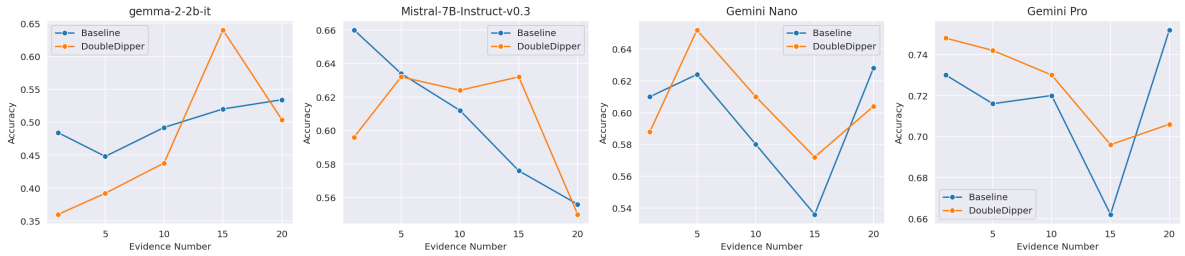

Figure 6: Performance (accuracy) of Gemma 2 2B, Mistral 7B, Gemini Nano and Gemini Pro with and without DOUBLEDIPPER on our sample of the Lost-in-the-middle dataset (Liu et al., 2023) according to the position of the document that contains the answer.

		Avg.	2Wiki	MonoRel	PIR	SRT	HotPotQA	Lost	MuSique
Gemma 2B	DOUBLEDIPPER (QA only)	46.4	16.0	84.6	79.2	49.6	31.1	**49.5**	14.7
	DOUBLEDIPPER	**49.5**	**23.7**	**85.8**	**81.6**	**50.0**	**39.9**	46.7	**18.8**
Gemma 9B	DOUBLEDIPPER (QA only)	55.1	22.5	**91.0**	**96.6**	**52.0**	42.9	61.0	20.0
	DOUBLEDIPPER	**61.2**	**41.7**	84.0	95.0	51.4	**61.2**	**61.8**	**33.3**
Mistral 7B (v0.3)	DOUBLEDIPPER (QA only)	49.1	21.8	**73.4**	**89.8**	48.6	38.6	60.3	11.5
	DOUBLEDIPPER	**51.0**	**28.6**	68.4	88.8	**50.6**	**43.4**	**60.7**	**16.7**
Mixtral 8x7B (v0.1)	DOUBLEDIPPER (QA only)	49.5	17.0	**91.4**	**86.8**	50.8	22.9	65.8	11.9
	DOUBLEDIPPER	**52.2**	**22.3**	91.8	86.0	**47.8**	**35.1**	**66.6**	**16.0**
Llama 3.1 8B	DOUBLEDIPPER (QA only)	32.7	25.0	9.2	41.4	46.6	33.5	56.7	16.6
	DOUBLEDIPPER	**59.9**	**38.7**	**91.2**	**90.6**	**51.0**	**59.3**	**58.1**	**30.1**

Table 9: Performance of DOUBLEDIPPER and DOUBLEDIPPER without instructing the models to retrieve the evidence (QA only) on the QA datasets.

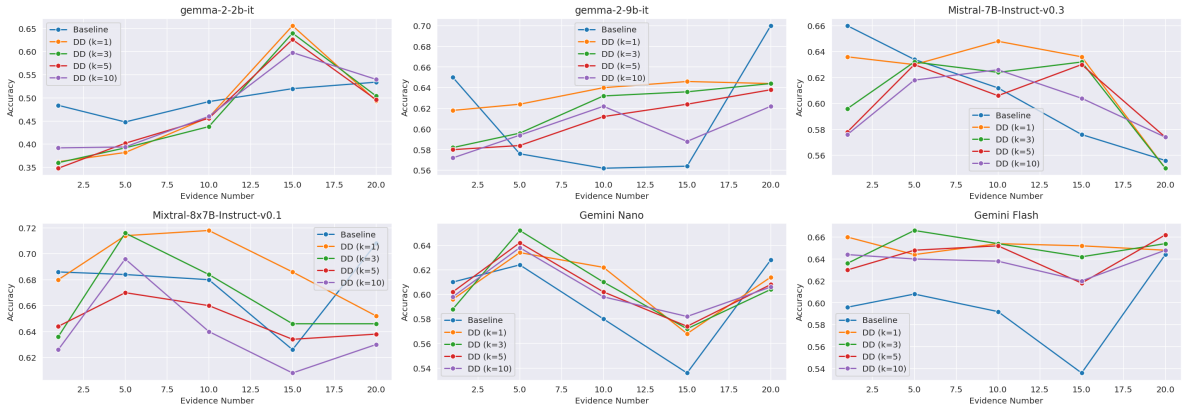

Figure 7: Performance (accuracy) of several models with DOUBLEDIPPER according to the number of self-generated demonstrations in the prompt.