RA-DIT: Retrieval-Augmented Dual Instruction Tuning

Xi Victoria Lin* **Xilun Chen*** **Mingda Chen***
Weijia Shi **Maria Lomeli** **Rich James** **Pedro Rodriguez** **Jacob Kahn**
Gergely Szilvasy **Mile Lewis** **Luke Zettlemoyer** **Scott Yih**
{victorialin,xilun,mingdachen,scottyih}@meta.com

Abstract

Retrieval-augmented language models (RALMs) improve performance by accessing long-tail and up-to-date knowledge from external data stores, but are challenging to build. Existing approaches require either expensive retrieval-specific modifications to LM pre-training or use post-hoc integration of the data store that leads to suboptimal performance. We introduce **R**etrieval-**A**ugmented **D**ual **I**nstruction **T**uning (RA-DIT), a lightweight fine-tuning methodology that provides a third option by retrofitting any large language model (LLM) with retrieval capabilities. Our approach operates in two distinct fine-tuning steps: (1) one updates a pre-trained LM to better use retrieved information, while (2) the other updates the retriever to return more relevant results, as preferred by the LM. By fine-tuning over tasks that require both knowledge utilization and contextual awareness, we demonstrate that each stage yields significant performance improvements, and using both leads to additional gains. Our best model, RA-DIT 65B, achieves state-of-the-art performance across a range of knowledge-intensive zero- and few-shot learning benchmarks, significantly outperforming existing in-context RALM approaches by up to +8.9% in 0-shot setting and +1.4% in 5-shot setting on average.

1 Introduction

Large language models (LLMs) excel as zero- and few-shot learners across various tasks (Brown et al., 2020; Chowdhery et al., 2022; Touvron et al., 2023a;b; Anil et al., 2023; OpenAI, 2023). However, because knowledge is represented only in the model parameters, they struggle to capture long-tail knowledge (Tirumala et al., 2022; Sun et al., 2023) and require substantial resources to be kept up-to-date (Miller, 2023). Retrieval-Augmented Language Modeling (RALM) integrates LLMs with non-parametric information retrieval to overcome these limitations (Guu et al., 2020; Borgeaud et al., 2022; Izacard et al., 2022b; Shi et al., 2023b; Ram et al., 2023). By explicitly decoupling knowledge retrieval from the backbone language model, such architectures have exhibited superior performance on knowledge intensive tasks such as open-domain question answering (Lewis et al., 2020; Izacard et al., 2022b) and live chat interactions (Liu, 2022).

Existing efforts in RALM development primarily focus on two high-level challenges: (i) enhancing the LLM's capability to incorporate retrieved knowledge (Lewis et al., 2020; Izacard et al., 2022b; Luo et al., 2023) and (ii) refining the retrieval component to return more relevant content (Shi et al., 2023b; Izacard et al., 2022b). Retrieval capabilities have also been introduced at different stages of the model training process. REALM (Guu et al., 2020) and RETRO (Borgeaud et al., 2022) opt for *end-to-end pre-training*, incorporating the retrieval component from the outset. ATLAS (Izacard et al., 2022b) builds upon the T5 language model (Raffel et al., 2020), and *continuosly pre-trains* the framework over unsupervised text. REPLUG (Shi et al., 2023b) and In-Context RALM (Ram et al., 2023) combine *off-the-shelf* LLMs with general-purpose retrievers, showing that LLMs and retrievers, even when optimized independently, can be effectively fused through the emergent in-context learning capbabilities of LLMs. However, extensive pre-training of such architectures incurs high computational costs, and the off-the-shelf fusion approach also has limitations, particularly as the LLMs are not inherently trained to incorporate retrieved content.

*Equal contribution

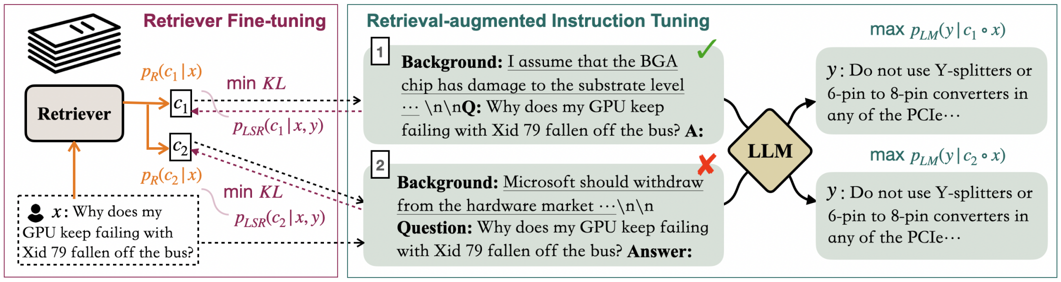

Figure 1: The RA-DIT approach separately fine-tunes the LLM and the retriever. For a given example, the LM-ft component updates the LLM to maximize the likelihood of the correct answer given the retrieval-augmented instructions (§2.3); the R-ft component updates the retriever to minimize the KL-Divergence between the retriever score distribution and the LLM preference (§2.4)

In this work, we show lightweight instruction tuning (Chung et al., 2022b; Iyer et al., 2022; Zhou et al., 2023) alone can significantly boost the performance of RALMs, especially in scenarios that require access to large, external knowledge sources. We propose **R**etrieval-**A**ugmented **D**ual **I**nstruction **T**uning (RA-DIT), an approach that retrofits any LLM with retrieval capabilities via fine-tuning over a set of tasks selected to cultivate knowledge utilization and contextual awareness in the language model predictions. We initialize the framework using pre-trained LLAMA (Touvron et al., 2023a) and a state-of-the-art dual-encoder based dense retriever, DRAGON+ (Lin et al., 2023). Following Shi et al. (2023b), we retrieve relevant text chunks based on the language model prompt. Each retrieved chunk is prepended to the prompt, and the predictions from multiple chunks are computed in parallel and ensembled to produce the final output.

We perform instruction-tuning in two separate steps. For *language model fine-tuning* (LM-ft), we adopt the supervised fine-tuning objective (Chung et al., 2022b; Iyer et al., 2022) while augmenting each fine-tuning prompt with a retrieved "background" field prepended to the instructions (Figure 1). We also leverage the design of existing NLP tasks and populate this field with the ground truth context for tasks such as reading comprehension and summarization. By incorporating the background text during fine-tuning, we guide the LLM to optimally utilize the retrieved information and ignore distracting content (Shi et al., 2023a). For *retriever fine-tuning* (R-ft), we update the query encoder using a generalized *LM-Supervised Retrieval* (LSR, Shi et al., 2023b) training objective computed over a combination of supervised tasks and unsupervised text completion. This way we enable the retriever to yield more contextually relevant results, aligned with the preferences of the LLM.

We demonstrate that each fine-tuning step offers significant performance gains, and that the fine-tuned LLM and retriever can be combined to achieve further improvements. Our largest model, RA-DIT 65B, attains state-of-the-art performance in zero- and few-shot settings on knowledge intensive benchmarks, notably surpassing the un-tuned in-context RALM approach on datasets including MMLU (Hendrycks et al., 2021b) (+8.2% 0-shot; +0.7% 5-shot) and Natural Questions (Kwiatkowski et al., 2019) (+22% 0-shot; +3.8% 5-shot). In addition, RA-DIT 65B also substantially outperforms ATLAS 11B on 8 knowledge-intensive tasks (+7.2% on average in the 64-shot fine-tuning setting). This suggests that language models and retrievers, when optimized independently and then fused through instruction-tuning, can compete effectively with RALMs that have undergone extensive continuous pre-training. We further conduct a comprehensive model analysis, showing the effectiveness of our approach across LLMs of varying sizes, as well as evaluating the influence of different fine-tuning strategies and retriever configurations.

2 METHOD

2.1 ARCHITECTURE

Language Model We focus on retrieval-augmenting pre-trained auto-regressive language models (Brown et al., 2020). In particular, we use LLAMA (Touvron et al., 2023a), a family of open-sourced language models pre-trained on trillions of tokens.

Table 1: Our intruction tuning datasets. All datasets are downloaded from Hugging Face (Lhoest et al., 2021), with the exception of those marked with ‡, which are taken from Iyer et al. (2022).

Task	HF identifier	Dataset name	$\mathcal{D}_L$	$\mathcal{D}_R$	#Train
Dialogue	oasst1	OpenAssistant Conversations Dataset (Köpf et al., 2023)	✓	✓	31,598
Open-Domain QA	commonsense_qa	CommonsenseQA (Talmor et al., 2019)	✓	✓	9,741
	math_qa	MathQA (Amini et al., 2019)	✓	✓	29,837
	web_questions	Web Questions (Berant et al., 2013)	✓	✓	3,778
	wiki_qa	Wiki Question Answering (Yang et al., 2015)	✓	✓	20,360
	yahoo_answers_qa	Yahoo! Answers QA	✓	✓	87,362
	freebase_qa	FreebaseQA (Jiang et al., 2019)		✓	20,358
	ms_marco*	MS MARCO (Nguyen et al., 2016)		✓	80,143
Reading Comprehension	coqa	Conversational Question Answering (Reddy et al., 2019)	✓		108,647
	drop	Discrete Reasoning Over Paragraphs (Dua et al., 2019)	✓		77,400
	narrativeqa	NarrativeQA (Kočiský et al., 2018)	✓		32,747
	newsqa	NewsQA (Trischler et al., 2017)	✓		74,160
	pubmed_qa	PubMedQA (Jin et al., 2019)	✓	✓	1,000
	quail	QA for Artificial Intelligence (Rogers et al., 2020)	✓		10,246
	quarel	QuaRel (Tafjord et al., 2019)	✓	✓	1,941
	squad_v2	SQuAD v2 (Rajpurkar et al., 2018)	✓		130,319
Summarization	cnn_dailymail	CNN / DailyMail (Hermann et al., 2015)	✓		287,113
Chain-of-thought Reasoning	aqua_rat‡	Algebra QA with Rationales (Ling et al., 2017)	✓		97,467
	ecqa‡	Explanations for CommonsenseQ (Aggarwal et al., 2021)	✓		7,598
	gsm8k‡	Grade School Math 8K (Cobbe et al., 2021)	✓		7,473
	math‡	MATH (Hendrycks et al., 2021c)	✓		7,500
	strategyqa‡	StrategyQA (Geva et al., 2021)	✓		2,290

* We only used the question-and-answer pairs in the MS MARCO dataset.

Retriever We adopt a dual-encoder based retriever architecture, since it can be easily fine-tuned and is efficient at the inference stage (Lewis et al., 2020; Izacard et al., 2022b; Shi et al., 2023b). Given a corpus $\mathcal{C}$ and a query q, the document encoder maps each *text chunk* $c \in \mathcal{C}$ to an embedding $\mathbf{E}_d(c)$ and the query encoder maps q to an embedding $\mathbf{E}_q(q)$. The top-k relevant text chunks for q are retrieved based on the query-document embedding similarity, which is often computed via dot product:

$$s(q, c) = \mathbf{E}_q(q) \cdot \mathbf{E}_d(c). \tag{1}$$

We initialize the retriever using DRAGON+ (Lin et al., 2023), a state-of-the-art dual-encoder model trained with a contrastive learning objective and large-scale data augmentation.

Parallel In-Context Retrieval-Augmentation Following Shi et al. (2023b), for a given language model prompt x, we retrieve the top-k relevant text chunks $\mathcal{C}' \subset \mathcal{C}, |\mathcal{C}'| = k$. To stay within the context window size limit, each retrieved chunk is prepended individually to the prompt[1], and the language model predictions from multiple augmented prompts are computed in parallel. The final output probability is a mixture of the probability from each augmented prompt weighted by the chunk relevance score

$$p_{LM}(y|x, \mathcal{C}') = \sum_{c \in \mathcal{C}'} p_{LM}(y|c \circ x) \cdot p_R(c|x), \tag{2}$$

where $\circ$ denotes sequence concatenation, and $p_R(c|x) = \frac{\exp s(x,c)}{\sum_{c' \in \mathcal{C}'} \exp s(x,c')}$ are the retriever scores re-normalized among top-k relevant chunks.

2.2 Fine-tuning Datasets

We choose a set of fine-tuning tasks aimed at boosting the language model's ability to utilize knowledge effectively and improving its contextual awareness in generating predictions. As shown in Table 1, our *language model fine-tuning* datasets ($\mathcal{D}_L$) consists of 20 datasets across 5 distinct categories: dialogue, open-domain QA, reading comprehension[2], summarization and chain-of-thought

[1]We use a pair of start ("Background:") and end ("\n\n") tokens to demarcate the retrieved segment in the augmented prompt. The complete set of our instruction-tuning templates are shown in Appendix C.

[2]Our reading comprehension (RC) fine-tuning datasets include SQuAD 2.0 (Rajpurkar et al., 2018), which trains the model to determine whether a question can be answered using a given passage, and to provide an

reasoning. For *retriever fine-tuning* datasets $\mathcal{D}_R$, we opt for the QA datasets in our collection featuring standalone questions, and we additionally include two QA datasets, FreebaseQA (Jiang et al., 2019) and MS-MARCO (Nguyen et al., 2016). The examples of each dataset are serialized for instruction tuning using manually compiled templates (Table 10). For tasks in $\mathcal{D}_L \cap \mathcal{D}_R$, we use the same template for both fine-tuning steps. In addition, we observe that supplementing the instruction-tuning data with unsupervised text leads to additional performance gains for both language model and retriever fine-tuning, and we detail data mixture used in Appendix B.

2.3 Retrieval Augmented Language Model Fine-tuning

To improve the language model's ability to utilize retrieved information, we fine-tune it on the selected datasets $\mathcal{D}_L$ with in-context retrieval augmentation. Formally, we separate each fine-tuning sequence into an instruction segment (x) and an output segment (y). For each example $(x_i, y_i) \in \mathcal{D}_L$, we retrieve the top-$\tilde{k}$ relevant text chunks $\mathcal{C}_i \subset \mathcal{C}$ based on x_i. Mirroring the inference-time handling, for each retrieved chunk $c_{ij} \in \mathcal{C}_i$, we create a separate fine-tuning example by prepending it to the instructions as a background field, resulting in $\tilde{k}$ independent fine-tuning instances per original example: $\{(c_{ij} \circ x_i, y_i) | j = 1 \ldots \tilde{k}\}$.[3]

We fine-tune the language model using the next-token prediction objective and minimize the loss from tokens in the output segment of each instance (Iyer et al., 2022):

$$\mathcal{L}(\mathcal{D}_L) = -\sum_i \sum_j \log p_{LM}(y_i | c_{ij} \circ x_i). \tag{3}$$

Integrating in-context retrieval augmentation during fine-tuning gives a twofold benefit. First, it adapts the LLM to better utilize relevant background knowledge to make a prediction. Secondly, even state-of-the-art retrievers can falter and return inaccurate results. By training the LLM to make correct predictions when a wrong retrieved chunk is given, we enable the LLM to ignore misleading retrieval content and lean into its parametric knowledge in such cases. The efficacy of this fine-tuning strategy is empirically demonstrated in §5.1.

2.4 Retriever Fine-tuning

In addition to fine-tuning the language model with retrieval augmentation, we also fine-tune the retriever to better align its output with the language model. In particular, we adopt a generalized version of LSR (*LM-Supervised Retrieval*, Shi et al., 2023b) training that leverages the language model itself to provide supervision for retriever fine-tuning.

For a training sample (x, y) in the retriever fine-tuning dataset $\mathcal{D}_R$, we define the LSR score for a retrieved chunk c as follows:

$$p_{LSR}(c|x, y) = \frac{\exp\left(p_{LM}(y|c \circ x)/\tau\right)}{\sum_{c' \in \mathcal{C}} \exp\left(p_{LM}(y|c' \circ x)/\tau\right)} \approx \frac{\exp\left(p_{LM}(y|c \circ x)/\tau\right)}{\sum_{c' \in \mathcal{C}'} \exp\left(p_{LM}(y|c' \circ x)/\tau\right)}, \tag{4}$$

where τ is a temperature hyperparameter, and $\mathcal{C}' \subset \mathcal{C}$ denotes the top-k retrieved chunks for x. A higher LSR score indicates that c is more effective at improving the language model's chance of predicting the correct answer. The goal of LSR training is for the retriever to assign higher scores to chunks that can improve the LLM's likelihood of generating the correct answer. To achieve this, we minimize the KL-divergence between p_{LSR} and the retriever scores p_R defined in Eq. 2:

$$\mathcal{L}(\mathcal{D}_R) = \mathbb{E}_{(x,y) \in \mathcal{D}_R} KL\big(p_R(c|x) \parallel p_{LSR}(c|x, y)\big) \tag{5}$$

In practice, we only update the query encoder of the retriever, as fine-tuning both encoders hurts the performance (§5.1). While previous work (Shi et al., 2023b) relies solely on unlabeled texts

answer only when the passage is relevant (otherwise the response is set to "I don't know"). As shown in Appendix F, fine-tuning on this dataset promotes a desirable behavior: the instruction-tuned model tends to respond with "I don't know" when the retriever presents an incorrect passage. We leave further exploration of this behavior to improve answer generation as a future work.

[3] The exceptions are summarization tasks and RC tasks with context dependent questions (e.g. "when was the writer born?"), where we do not perform retrieval and create the fine-tuning instances using the given background text instead. For RC tasks with self-contained questions, we use the retrieved chunks in addition to the given background text to create fine-tuning instances, resulting in $\tilde{k} + 1$ of them per original example.

(denoted as *corpus data*) for LSR training, we show that LSR can be generalized to incorporate the multi-task instruction data introduced in §2.2 (denoted as *MTI data*). The MTI data provide direct supervision to the retriever to return relevant information that enhances the language model in various downstream tasks. As shown in §5.1, combining both types of data yields the best results and outperforms using either source alone.

3 Experiment Setup

3.1 Retriever

We initialize the retriever in our framework with Dragon+ (Lin et al., 2023) and also use it to study various retriever configurations. To build the retrieval corpus, we combine the text chunks (37M) from the Dec. 20, 2021 Wikipedia dump released by Izacard et al. (2022b) with additional ones (362M) from the 2017-2020 CommonCrawl dumps. We detail the corpus pre-processing and indexing in Appendix A. Our final retrieval data store, with the two data sources combined, contain 399M text chunks with a maximum length of 200 words. In §5.3, we conduct an analysis on the impact of retrieval corpus using various subsets of our retrieval index, as well as different Wikipedia snapshots. We obtain the retrieval queries used for our fine-tuning and evaluation tasks using manually[4] constructed templates (Table 10 and 12).

3.2 Baselines

We focus on comparing our approach to the base Llama models (Touvron et al., 2023a) and RePlug (Shi et al., 2023b), a state-of-the-art approach that integrates off-the-shelf LLMs and retrievers, in the zero-shot and in-context few-shot learning settings. We instantiate RePlug using Llama and Dragon+. In addition, we also compare RA-DIT to Atlas (Izacard et al., 2022b) in a 64-shot fine-tuning setting (§4).

3.3 Evaluation

We primarily conduct evaluation on knowledge-intensive tasks that are not included in our fine-tuning datasets, including MMLU (Hendrycks et al., 2021a), Natural Questions (NQ; Kwiatkowski et al., 2019), TriviaQA (TQA; Joshi et al., 2017), and a subset[5] of the tasks in the KILT benchmark (Petroni et al., 2021). We use the development split of six of the KILT tasks (excluding ELI5) to determine fine-tuning hyperparameters (Appendix B). This enables us to report genuine few-shot evaluation results for four of the ten evaluation tasks. For the remaining tasks, we report few-shot results assuming access to in-domain development data. We randomly select few-shot examples from the official training splits of the KILT tasks, except for FEV, NQ and TQA, where we use the 64-shot examples released by Izacard et al. (2022b). For these three datasets, we also ensure that the 5-shot examples are subsets of the 64 examples. In our retrieval augmented models, we use the top-1 most relevant chunk for the in-context few-shot examples. In addition, we also evaluate models on commonsense reasoning tasks to evaluate the impact of retrieval-augmented instruction tuning on the LLM's parametric knowledge and reasoning capabilities. Here we report results on the entire development sets. Details of our evaluation datasets, including the evaluation metrics, template and the scoring functions used, can be found in in Appendix D.

4 Main Results

Knowledge-Intensive Tasks We report the main results in Table 2. In particular, RA-DIT is compared to Llama (Touvron et al., 2023a) as well as RePlug (Shi et al., 2023b), in both 0-shot and 5-shot settings. We first observe that RePlug works much better than the base Llama 65B, confirming the benefits of RALMs on knowledge-intensive tasks. Furthermore, RA-DIT significantly outperforms RePlug (+8.9% in 0-shot and +1.4% in 5-shot on average over MMLU, NQ, TQA

[4]We leave automatically generating task-specific retrieval queries to future work.

[5]The subset consists of seven tasks: HotpotQA (Yang et al., 2018), FEVER (Thorne et al., 2018), AIDA CoNLL-YAGO (Hoffart et al., 2011), Zero-Shot RE (Levy et al., 2017), T-REx (Elsahar et al., 2018), Wizard of Wikipedia (Dinan et al., 2019) and ELI5 (Fan et al., 2019).

Table 2: Main results: Performance on knowledge intensive tasks (test sets).

	MMLU	NQ	TQA	ELI5	HoPo	FEV	AIDA	zsRE	T-REx	WoW	Avg◇	Avg
					0-shot							
LLAMA 65B	51.2	5.2	55.8	19.5	12.5	59.3	0.6	6.7	1.3	15.6	32.9	22.8
LLAMA 65B REPLUG	59.7	28.8	72.6	19.1	32.0	73.3	41.8	50.8	36.3	16.1	45.1	43.1
RA-DIT 65B	**64.6**	**35.2**	**75.4**	**21.2**	**39.7**	**80.7**	**45.1**	**73.7**	**53.1**	**16.4**	**49.1**	**50.5**
					5-shot in-context							
LLAMA 65B	63.4	31.6	71.8	22.1	22.6	81.5	48.2	39.4	52.1	**17.4**	47.2	45.0
LLAMA 65B REPLUG	64.4	42.3	74.9	22.8	**41.1**	89.4	46.4	60.4	**68.9**	16.8	51.1	52.7
RA-DIT 65B	**64.9**	**43.9**	**75.1**	**23.2**	40.7	**90.7**	**55.8**	**72.4**	68.4	17.3	**51.8**	**55.2**

64-shot fine-tuned	NQ	TQA	HoPo	FEV	AIDA	zsRE	T-REx	WoW	Avg
ATLAS†	42.4	**74.5**	34.7	**87.1**	66.5	74.9	58.9	15.5	56.8
RA-DIT 65B	**43.5**	72.8	**36.6**	86.9	**80.5**	**78.1**	**72.8**	**15.7**	**60.9**

◇ Average of MMLU, NQ, TQA, and ELI5.

† ATLAS conducts 64-shot fine-tuning for each individual task and reports the performance of task-specific models. For RA-DIT, we perform multi-task fine-tuning using a compilation of 64-shot examples from each task, and report the performance of a unified model across tasks.

Table 3: Performance on commonsense reasoning tasks (dev sets) without retrieval augmentation.

0-shot	BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-E	ARC-C	OBQA	Avg
LLAMA 65B	85.3	82.8	52.3	84.2	77.0	78.9	56.0	**60.2**	72.1
RA-DIT 65B	**86.7**	**83.7**	**57.9**	**85.1**	**79.8**	**83.7**	**60.5**	58.8	**74.5**

and ELI5) and achieves the best performance on most datasets. This supports our claim that combining off-the-shelf LLMs and retrievers is sub-optimal, and our dual instruction tuning approach is an effective way of retrofitting LLMs with retrieval capabilities.[6]

We also compare with ATLAS, a state-of-the-art encoder-decoder based RALM that jointly pretrains the language model and the retriever. Here we adopt a 64-shot setting similar to Izacard et al. (2022b) with the following differences. While ATLAS conducts 64-shot fine-tuning for each individual task and reports the performance of task-specific models, we continuously fine-tune the RA-DIT checkpoint using the 64-shot examples from all tasks combined, and report the performance of a single model across tasks. As shown in Table 2, despite using a single model, RA-DIT outperforms ATLAS by an average of 4.1 points, achieving higher performance on 6 out of the 8 datasets.

Commonsense Reasoning We benchmark RA-DIT 65B on a set of commonsense reasoning tasks to evaluate the impact of retrieval-augmented instruction tuning on the LLM's parametric knowledge and reasoning capabilities. We do not perform retrieval augmentation in this experiment. As shown in Table 3, RA-DIT demonstrates improvements over the base LLAMA models on 7 out of 8 evaluation datasets, indicating that the parametric knowledge and reasoning capabilities of the LLM component are in general preserved. As discussed in Appendix F, maintaining the parametric knowledge in the LLM component is vital as a fallback when the retriever makes mistakes.

5 ANALYSIS

In this section, we present a set of analyses of various modeling decisions.

Table 4: Ablation of language model fine-tuning strategies. All rows report dev set performance.

0 / 5-shot	HoPo	FEV	AIDA	zsRE	T-REx	WoW	Avg
No retrieval							
LLAMA 65B	12.5 / 23.8	59.6 / 83.7	0.9 / 64.1	9.7 / 36.0	1.2 / 52.3	15.7 / 17.4	16.6 / **46.2**
IT 65B	20.0 / 30.0	67.8 / 83.2	8.9 / 58.5	19.0 / 35.4	17.3 / 53.5	16.4 / 16.5	24.9 / **46.2**
RA-IT 65B	26.8 / 29.9	65.2 / 84.8	10.7 / 52.9	30.9 / 35.2	24.1 / 52.9	16.5 / 16.5	**29.0** / 45.4
top-1 chunk							
LLAMA 65B + DRAGON+	25.8 / 39.4	72.8 / 89.8	39.1 / 50.7	48.8 / 59.6	31.4 / 69.1	15.8 / 17.1	39.0 / **54.3**
IT 65B + DRAGON+	33.3 / 38.8	84.0 / 90.1	43.9 / 50.3	56.8 / 58.2	44.7 / 66.4	15.7 / 15.6	46.4 / 53.2
RA-IT 65B + DRAGON+	37.6 / 39.1	81.0 / 90.4	41.6 / 52.3	59.6 / 57.9	49.6 / 65.8	16.6 / 16.6	**47.7** / 53.7
top-10 chunks							
LLAMA 65B + DRAGON+	31.0 / 41.6	75.4 / 90.8	44.8 / 54.0	58.6 / 63.7	40.2 / 71.9	16.0 / 17.8	44.3 / **56.6**
IT 65B + DRAGON+	33.9 / 40.6	87.0 / 91.8	50.5 / 53.8	53.9 / 62.5	45.7 / 69.4	15.6 / 15.7	47.8 / 55.6
RA-IT 65B + DRAGON+	40.0 / 41.2	82.8 / 92.1	47.2 / 53.5	65.0 / 62.3	54.3 / 69.0	16.5 / 16.6	**51.0** / 55.8

5.1 FINE-TUNING STRATEGIES

Language Model Fine-tuning We compare LLAMA instruction-tuned with retrieval-augmentation (RA-IT 65B) to the base language model, as well as LLAMA that is instruction-tuned conventionally[7] (IT 65B) on the same set of tasks. We evaluate all models with in-context retrieval augmentation using the DRAGON+ retriever, adjusting the number of retrieved chunks to 0, 1 or 10. As shown in Table 4, while both instruction tuning methods substantially enhance the 0-shot performance, they offers marginal improvements or even hurt the model performance in the 5-shot setting for most tasks except for HotpotQA[8]. When in-context retrieval-augmentation is applied, all models show substantial gains in both settings, even when limited to the top-1 chunk. The model performance consistently improves as we include more retrieved chunks. In the 0-shot setting with top-10 retrieved chunks, the RA-IT 65B model outperforms the IT 65B model by a large margin (51.0% vs. 47.7%). Under this setting, we observe that retrieval-augmented instruction tuning significantly enhances the LLM's ability to integrate information from the retrieved text chunks. The model is able to extract the correct answers from relevant chunks with greater confidence, while effectively leaning on its parametric knowledge for prediction when an irrelevant text chunk is present (Appendix F). In Appendix E.1, we also discuss the performance of RA-IT models when applied to smaller LLAMA models (7B and 13B), showing that it offers even larger performance boost in those cases.

Table 5: Ablation of retriever fine-tuning strategies. All rows use the LLAMA 65B model and report 5-shot performance on the dev sets.

5-shot	MMLU	NQ	TQA	HoPo	FEV	AIDA	zsRE	T-REx	WoW	Avg$^\diamond$	Avg
DRAGON+	62.6	41.8	72.9	41.5	90.6	54.1	63.7	72.1	17.5	56.6	57.4
MTL instruction tuning data	61.1	43.6	74.0	36.5	91.4	64.6	56.7	72.1	17.1	56.4	57.5
corpus data (FT both encoders)	61.7	43.2	73.8	37.5	88.2	69.8	53.5	57.2	17.5	54.0	55.8
corpus data	62.9	43.0	74.3	41.1	91.6	54.4	63.4	71.8	17.4	56.6	57.8
95% corpus + 5% MTL data	63.0	42.1	74.9	41.2	91.6	54.9	65.2	71.6	17.5	**57.0**	**58.0**

$^\diamond$ Average over the 6 KILT development tasks.

Retriever Fine-tuning In Table 5, we study different retriever fine-tuning strategies. As mentioned in §2.4, we explore two types of retriever fine-tuning data, the *multi-task instruction (MTI)*

[6]In comparison to Touvron et al. (2023a), we report lower 0-shot performance for LLAMA 65B on NQ and TQA. By examining the model generation, we think Touvron et al. (2023a) reported the ratio of responses that contain the ground truth answer string in the 0-shot setting. We do not post-process the model predictions and report exact match instead.

[7]Since our instruction tuning datasets include reading comprehension and summarization, the IT models are also exposed to problem types that depend on background knowledge.

[8]This observation aligns with the findings from previous instruction-tuning literature (Iyer et al., 2022). HotpotQA is an exception likely because it is from a task category covered in our instruction-tuning data.

data and the *corpus data*. We observe that fine-tuning the retriever with the corpus data alone improves over the base DRAGON+ model by an average of 0.4 points, whereas fine-tuning using only the MTI data improves by a smaller margin of 0.1 points. While fine-tuning with the MTI data yields good performance on certain datasets such as NQ (possibly due to its similarity to the MTI data), fine-tuning with the corpus data appears to generalize better and leads to stronger overall performance. Furthermore, we experiment with fine-tuning using both the MTI and corpus data. Table 5 shows that fine-tuning with "95% corpus data + 5% MTI data" achieves the best accuracy across all models, outperforming the non-finetuned baseline by 0.6 points on average.[9]

Finally, we also compare jointly fine-tuning both the query and document encoders with only fine-tuning the query encoder while freezing the document encoder. Table 5 shows this experiment conducted using the corpus data, where freezing the document encoder produces significantly better performance. As a result, we only fine-tune the query encoder in this work.

5.2 DUAL INSTRUCTION TUNING ABLATION

Table 6: The impact of LM and Retriever fine-tuning in our RA-DIT method, comparing the REPLUG baseline, LM-ft only, R-ft only, and RA-DIT. 5-shot dev set performance is reported.

5-shot	MMLU	NQ	TQA	ELI5	HoPo	FEV	AIDA	zsRE	T-REx	WoW	Avg
LLAMA 65B + DRAGON+	61.7	41.7	73.0	22.1	41.6	90.8	54.0	63.7	71.9	17.2	53.8
LLAMA 65B + FTed DRAGON+	63.0	42.2	74.9	22.2	41.4	91.6	54.9	65.2	71.4	17.4	54.4
RIT 65B + DRAGON+	64.8	42.8	73.1	23.6	41.2	92.1	53.5	62.3	69.0	16.6	53.9
RIT 65B + FTed DRAGON+	64.3	43.8	75.0	23.3	42.0	92.3	52.8	65.2	70.1	17.3	**54.6**

We isolate the impact of the language model fine-tuning from retriever fine-tuning in our RA-DIT method, and illustrate the benefit of each. [10] According to Table 6, both LM-ft and R-ft are beneficial when used alone, and outperform the REPLUG using LLAMA 65B and the DRAGON+ retriever. On the other hand, the most gain can be achieved when combining LM-ft and R-ft in our RA-DIT method, which outperforms the REPLUG baseline by 0.8 points on average. In our prelimary experiments, we also attempted iterative dual instruction tuning by fine-tuning the retriever using LSR scores from the RA-IT LM or conduct the RA-IT step using passages returned by the fine-tuned retriever, for one or two such iterations, but did not observe further gains. We leave the exploration of multi-step RA-DIT to future work.

5.3 RETRIEVER SETTINGS

In this section, we study the impact of various retriever choices in our framework. We use LLAMA 65B as the language model and combine it with different retrievers. Table 7 first compares DRAGON+ (Lin et al., 2023) with other state-of-the-art retrievers such as Contriever (Izacard et al., 2022a). All retrieval-augmented models substantially improve over the LLAMA baseline, and DRAGON+ significantly outperforms both Contriever and Contriever-MSMARCO. We hence adopt DRAGON+ as our base retriever in all experiments.

The middle section in Table 7 shows the impact of varying the retrieval corpora. In particular, we consider several subsets of our 399M retrieval corpus, namely CommonCrawl only (362M) and Wikipedia only (with and without infoboxes). We further compare with another Wikipedia snapshot (Wiki 2018) commonly used in the literature (Karpukhin et al., 2020). We observe that retrieving from Wikipedia only is beneficial for a number of KILT tasks such as AIDA and zsRE, as Wikipedia was the intended corpus for KILT tasks. We find that Wiki 2018 works better for NQ since the corpus is closer to the date of its data collection, similar to the observations by Izacard et al. (2022b). This indicates that our retrieval-augmented LM is faithful to the supplied retrieval corpus, and up-to-date information can be provided by updating the retrieval index at test time.

[9] In early experiments, we also tested other mixtures and found that using 5% or 10% MTI data worked the best. (They perform similarly to each other.)

[10] Minor performance differences may be observed for the LLAMA 65B + DRAGON+ model in different ablations due to the differences in few-shot example truncation in long prompts. We ensure all rows within each table are comparable.

Table 7: Retriever settings: We report 5-shot dev set performance using LLAMA 65B and various retrievers in the REPLUG setting.

5-shot	MMLU	NQ	TQA	HoPo	FEV	AIDA	zsRE	T-REx	WoW	ELI5	Avg
LLAMA 65B	61.3	30.9	70.6	23.8	83.7	50.2	36.0	52.3	17.4	23.4	45.0
Retriever ablation using LLAMA *65B and the 399M CC + Wiki corpus*											
Contriever	59.3	41.2	73.0	32.4	88.1	45.0	40.8	56.1	17.2	21.6	47.5
Contriever-msmarco	62.0	42.1	74.1	38.7	89.3	49.3	60.2	62.9	17.4	21.8	51.8
DRAGON+	61.7	41.7	73.0	40.8	90.8	48.8	63.7	71.9	17.8	23.8	53.4
Retriever corpus ablation using LLAMA *65B and the* DRAGON+ *retriever*											
CC only	62.8	39.6	72.6	34.4	89.5	54.8	30.3	46.2	17.1	22.9	47.0
Wiki 2021 + infobox	62.2	42.0	71.2	41.8	89.8	62.2	65.3	73.1	17.7	22.2	54.8
Wiki 2021	62.2	41.8	71.0	41.7	89.7	62.1	65.2	73.3	17.6	22.2	54.7
Wiki 2018	61.5	42.6	70.7	40.4	90.8	62.1	51.3	59.8	17.6	22.5	51.9
Number of retrieved chunks ablation using LLAMA *65B and the* DRAGON+ *retriever*											
top-1 chunks	60.5	36.6	69.2	39.4	89.8	48.6	59.6	69.1	17.1	22.2	51.2
top-3 chunks	62.1	39.6	71.3	40.8	90.3	49.8	62.9	70.8	17.2	22.7	52.8
top-10 chunks	61.7	41.7	73.0	40.8	90.8	48.8	63.7	71.9	17.8	23.8	53.4

Finally, we experiment with the number of retrieved passages supplied to LLAMA during generation. Table 7 shows that even retrieving the top-1 passage significantly improves LLAMA's average performance from 45.0 to 51.2, and it continues to increase as more retrieved passages are used. Due to diminishing return and inference cost, we adopt 10 retrieved passages by default in our experiments.

6 RELATED WORK

Retrieval-Augmented Language Models RALMs fuse language models (LMs) with a retrieval module that explicitly augments the LM with information retrieved from external knowledge stores (Guu et al., 2020; Lewis et al., 2020). One mainstream type of RALM follows the "retrieve-and-read" paradigm, where the retrieval module supplies external knowledge as additional context which the LM (reader) leverages to produce the final output (Izacard et al., 2022b; Borgeaud et al., 2022; Shi et al., 2023b; Ram et al., 2023). Some existing work focuses on pre-training the LM to better utilize retrieved knowledge. For example, REALM (Guu et al., 2020) and RETRO (Borgeaud et al., 2022) incorporate retrieval from the beginning and conduct end-to-end retrieval-augmented pre-training, whereas ATLAS (Izacard et al., 2022b) continuously pre-trains a T5 LM (Raffel et al., 2020) jointly with a retriever. Others assume black-box access to an LM and combine it with either off-the-shelf or fine-tuned retrievers (Shi et al., 2023b; Ram et al., 2023). Our approach adopts lightweight fine-tuning to effectively retrofit any pre-trained LLM with retrieval capacity. This approach offers efficiency compared to methods involving extensive pre-training and demonstrates superior effectiveness compared to the off-the-shelf fusion approach.

Independent to our work, Luo et al. (2023) proposes SAIL, an approach that fine-tunes the LM with instructions augmented with retrieved content, and examines it on public instruction following datasets (Taori et al., 2023; Chiang et al., 2023) using a moderately sized model (7B parameters). In comparison, RA-DIT conducts parallel retrieval-augmentation by generating distinct prompts for each retrieved passage and subsequently aggregating the outcomes; SAIL, on the other hand, concatenates the top retrieved passages in the augmentation. Furthermore, RA-DIT adopts a holistic view of the RALM architecture, employing a learnable neural retriever and proposing a dual optimization framework. SAIL, in comparison, leans on commercial search engines and BM25 and focuses on the LM-side enhancement (e.g. it proposes an in-context retrieval selection technique to guide the model focus towards informative content).

Another family of RALMs incorporate retrieval in the output distribution of the LM (Khandelwal et al., 2020; Zhong et al., 2022). Such models retrieve a set of k nearest-neighbor tokens using the LM context representation, and interpolate this distribution of retrieved tokens with the LM output distribution to generate the next token at inference time. Alternatively, the retrieved token distribution can be used alone to make a non-parametric LM (Min et al., 2023).

Instruction Tuning Instruction fine-tuning has been proposed to align pre-trained LLMs to follow natural language instructions and avoid extensive prompt engineering (Ouyang et al., 2022; Wei et al., 2022; Chung et al., 2022a; Wang et al., 2022; Iyer et al., 2022). We propose retrieval-augmented instruction tuning (RA-IT) as part of our *dual instruction tuning* framework to improve the LM's ability to leverage retrieved information.

Information Retrieval Retrieval methods include *sparse retrievers* that does matching over a sparse bag-of-words representation (Robertson & Zaragoza, 2009; Formal et al., 2021), *dense retrievers* that embed queries and documents into a fixed-size dense vector for nearest-neighbor search (Karpukhin et al., 2020; Xiong et al., 2021), and *multi-vector retrievers* which uses multiple vectors as the representation and more complex search algorithms for increased accuracy (Khattab & Zaharia, 2020; Li et al., 2023). We adopt a state-of-the-art dense retriever, DRAGON (Lin et al., 2023), as our base retriever, because of its simplicity, state-of-the-art accuracy, high retrieval efficiency on GPUs, and the ease of further fine-tuning.

7 CONCLUSION

In this paper, we propose RA-DIT, a lightweight Retrieval-Augmented Dual Instruction Tuning framework that can effectively retrofit any pre-trained LLM with retrieval capabilities. RA-DIT updates the LLM with *retrieval-augmented instruction tuning* to make better use of retrieved knowledge and ignore irrelevant or distracting information. It also fine-tunes the retriever with supervision from the LLM to retrieve texts that can better help the LLM generate correct outputs. RA-DIT achieves state-of-the-art performance in zero- and few-shot evaluations on knowledge intensive benchmarks, surpassing un-tuned in-context RALM approaches such as REPLUG and compete effectively against methods that require extensive pre-training such as ATLAS.

REFERENCES

Shourya Aggarwal, Divyanshu Mandowara, Vishwajeet Agrawal, Dinesh Khandelwal, Parag Singla, and Dinesh Garg. Explanations for CommonsenseQA: New dataset and models. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pp. 3050–3065. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.238. URL `https://doi.org/10.18653/v1/2021.acl-long.238`.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2357–2367, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1245. URL `https://aclanthology.org/N19-1245`.

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex

Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. PaLM 2 technical report, 2023.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1533–1544, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1160.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. Improving language models by retrieving from trillions of tokens. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 2206–2240. PMLR, 2022. URL https://proceedings.mlr.press/v162/borgeaud22a.html.

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. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

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, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

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. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022a.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022b. URL `https://arxiv.org/abs/2210.11416`.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2924–2936, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL `https://aclanthology.org/N19-1300`.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? Try arc, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021. URL `https://arxiv.org/abs/2110.14168`.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. Wizard of wikipedia: Knowledge-powered conversational agents. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=r1l73iRqKm`.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2368–2378, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1246. URL `https://aclanthology.org/N19-1246`.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. T-REx: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). URL `https://aclanthology.org/L18-1544`.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3558–3567, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1346. URL `https://aclanthology.org/P19-1346`.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. Splade: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, 2021. ISBN 9781450380379. doi: 10.1145/3404835.3463098. URL `https://doi.org/10.1145/3404835.3463098`.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 2021. doi: 10.1162/tacl_a_00370. URL `https://aclanthology.org/2021.tacl-1.21`.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3929–3938. PMLR, 2020. URL `http://proceedings.mlr.press/v119/guu20a.html`.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021a. URL `https://openreview.net/forum?id=d7KBjmI3GmQ`.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021b.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In Joaquin Vanschoren and Sai-Kit Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021c. URL `https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html`.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pp. 1693–1701, 2015. URL `https://proceedings.neurips.cc/paper/2015/hash/afdec7005cc9f14302cd0474fd0f3c96-Abstract.html`.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 782–792, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. URL `https://aclanthology.org/D11-1072`.

Srinivasan Iyer, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Daniel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, Xian Li, Brian O'Horo, Gabriel Pereyra, Jeff Wang, Christopher Dewan, Asli Celikyilmaz, Luke Zettlemoyer, and Ves Stoyanov. OPT-IML: scaling language model instruction meta learning through the lens of generalization. *CoRR*, abs/2212.12017, 2022. doi: 10.48550/arXiv.2212.12017. URL `https://doi.org/10.48550/arXiv.2212.12017`.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*, 2022a. ISSN 2835-8856. URL `https://openreview.net/forum?id=jKN1pXi7b0`.

Gautier Izacard, Patrick S. H. Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Few-shot learning with retrieval augmented language models. *CoRR*, abs/2208.03299, 2022b. doi: 10.48550/arXiv.2208.03299. URL `https://doi.org/10.48550/arXiv.2208.03299`.

Kelvin Jiang, Dekun Wu, and Hui Jiang. FreebaseQA: A new factoid QA data set matching trivia-style question-answer pairs with Freebase. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 318–323, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1028. URL `https://aclanthology.org/N19-1028`.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. PubMedQA: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2567–2577, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1259. URL `https://aclanthology.org/D19-1259`.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HklBjCEKvH.

Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 39–48. Association for Computing Machinery, 2020. ISBN 9781450380164. doi: 10.1145/3397271.3401075. URL https://doi.org/10.1145/3397271.3401075.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, 2018. doi: 10.1162/tacl_a_00023. URL https://aclanthology.org/Q18-1023.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, et al. Openassistant conversations–democratizing large language model alignment. *arXiv preprint arXiv:2304.07327*, 2023.

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. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pp. 333–342, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/K17-1034. URL https://aclanthology.org/K17-1034.

Patrick S. H. 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. Retrieval-augmented generation for knowledge-intensive NLP tasks. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 175–184, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-demo.21. URL https://aclanthology.org/2021.emnlp-demo.21.

Minghan Li, Sheng-Chieh Lin, Barlas Oguz, Asish Ghoshal, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. CITADEL: Conditional token interaction via dynamic lexical routing for efficient and effective multi-vector retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11891–11907, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.663. URL https://aclanthology.org/2023.acl-long.663.

Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oguz, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. How to train your DRAGON: Diverse augmentation towards generalizable dense retrieval. *arXiv preprint arXiv:2302.07452*, 2023.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 158–167, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1015. URL https://aclanthology.org/P17-1015.

Jerry Liu. LlamaIndex, 11 2022. URL https://github.com/jerryjliu/llama_index.

Hongyin Luo, Yung-Sung Chuang, Yuan Gong, Tianhua Zhang, Yoon Kim, Xixin Wu, Danny Fox, Helen Meng, and James R. Glass. SAIL: search-augmented instruction learning. *CoRR*, abs/2305.15225, 2023. doi: 10.48550/arXiv.2305.15225. URL https://doi.org/10.48550/arXiv.2305.15225.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9802–9822, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.546. URL https://aclanthology.org/2023.acl-long.546.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1260. URL https://aclanthology.org/D18-1260.

Katharine Miller. How Do We Fix and Update Large Language Models?, 2023. URL https://hai.stanford.edu/news/how-do-we-fix-and-update-large-language-models.

Sewon Min, Weijia Shi, Mike Lewis, Xilun Chen, Wen-tau Yih, Hannaneh Hajishirzi, and Luke Zettlemoyer. Nonparametric masked language modeling. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 2097–2118, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.132. URL https://aclanthology.org/2023.findings-acl.132.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. Ms marco: A human generated machine reading comprehension dataset. *CoRR*, abs/1611.09268, 2016. URL http://dblp.uni-trier.de/db/journals/corr/corr1611.html#NguyenRSGTMD16.

OpenAI. Gpt-4 technical report, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2523–2544, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.200. URL https://aclanthology.org/2021.naacl-main.200.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 784–789, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2124. URL https://aclanthology.org/P18-2124.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. In-context retrieval-augmented language models, 2023.

Siva Reddy, Danqi Chen, and Christopher D Manning. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, 2019.

Stephen E. Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009. doi: 10.1561/1500000019. URL https://doi.org/10.1561/1500000019.

Anna Rogers, Olga Kovaleva, Matthew Downey, and Anna Rumshisky. Getting closer to AI complete question answering: A set of prerequisite real tasks. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 8722–8731. AAAI Press, 2020. URL https://aaai.org/ojs/index.php/AAAI/article/view/6398.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*, 2019.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4463–4473, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1454. URL https://aclanthology.org/D19-1454.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 31210–31227. PMLR, 23–29 Jul 2023a. URL https://proceedings.mlr.press/v202/shi23a.html.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. REPLUG: retrieval-augmented black-box language models. *CoRR*, abs/2301.12652, 2023b. doi: 10.48550/arXiv.2301.12652. URL https://doi.org/10.48550/arXiv.2301.12652.

Kai Sun, Yifan Ethan Xu, Hanwen Zha, Yue Liu, and Xin Luna Dong. Head-to-tail: How knowledgeable are large language models (llm)? a.k.a. will llms replace knowledge graphs?, 2023.

Oyvind Tafjord, Peter Clark, Matt Gardner, Wen-tau Yih, and Ashish Sabharwal. QUAREL: A dataset and models for answering questions about qualitative relationships. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pp. 7063–7071. AAAI Press, 2019. doi: 10.1609/aaai.v33i01.33017063. URL https://doi.org/10.1609/aaai.v33i01.33017063.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4149–4158. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1421. URL https://doi.org/10.18653/v1/n19-1421.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 809–819, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1074. URL https://aclanthology.org/N18-1074.

Kushal Tirumala, Aram H. Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. Memorization without overfitting: Analyzing the training dynamics of large language models. In *NeurIPS*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/fa0509f4dab6807e2cb465715bf2d249-Abstract-Conference.html.

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

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, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023b.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. NewsQA: A machine comprehension dataset. In *Proceedings of the*

2nd Workshop on Representation Learning for NLP, pp. 191–200, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-2623. URL https://aclanthology.org/W17-2623.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 5085–5109, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.340. URL https://aclanthology.org/2022.emnlp-main.340.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=gEZrGCozdqR.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=zeFrfgyZln.

Yi Yang, Wen-tau Yih, and Christopher Meek. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2013–2018, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1237. URL https://aclanthology.org/D15-1237.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472.

Zexuan Zhong, Tao Lei, and Danqi Chen. Training language models with memory augmentation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 5657–5673, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.382. URL https://aclanthology.org/2022.emnlp-main.382.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. Lima: Less is more for alignment, 2023.

A Retrieval Corpus

We combine the text chunks from the Dec. 20, 2021 Wikipedia dump released by Izacard et al. (2022b) with additional ones from the 2017-2020 CommonCrawl dumps. The Wikipedia dump includes lists and infoboxes in addition to regular articles. The articles are split by section, where long sections are further split into text chunks of equal sizes and contain less than 200 words, leading to a total of 37M text chunks. We randomly sample a subset of articles from the CommonCrawl dumps, and split them into equal-sized text chunks that contain less than 100 white-space-separated words, leading to a total of 362M text chunks.

We use a GPU-based exact k-nearest-neighbor search index implementation[11] released by Izacard et al. (2022b).

B Implementation Details

Retrieval-augmented LM Fine-tuning We use the top-3 retrieved text chunks for a given example (i.e. $\tilde{k} = 3$) to generate the fine-tuning instances. To improve fine-tuning efficiency, we pack multiple examples up to the language model context window limit (2048 tokens). Each example is demacrate by a pair of <bos> and <eos> tokens, and we adopt the document attention masking (Iyer et al., 2022) such that a token only attends to the previous tokens in the same example. We use a dataset mixture that contains 10% unsupervised text and 5% OASST-1 data. For the remaining datasets, we establish a cap on the number of examples per dataset at $\eta = 7500$. We then randomly sample batches in accordance with this adjusted mixture probability.

We fine-tune the 7B, 13B and 65B Llama models using 8, 16 and 64 A100 GPUs, respectively. The fine-tuning hyperparameters are detailed in Table 8. Similar to Zhou et al. (2023), we found that the best generalization performance on the dev set can be achieved using a small number of fine-tuning steps. We evaluate the models every 100 steps, and select the best checkpoint based on the average dev set performance over the 6 development KILT tasks shown in Table 11 (early stopping).

Table 8: Hyperparameters for retrieval-augmented LM fine-tuning.

Model	peak lr	end lr	lr scheduler	warm-up	# steps	early stopping	batch size	model parallel	seq len
RA-DIT 7B	1e-5	1e-7	cosine	200	500	500	64	1	2048
RA-DIT 13B	1e-5	1e-7	cosine	200	500	400	128	2	2048
RA-DIT 65B	1e-5	1e-7	cosine	200	500	300	128	8	2048

64-shot Eval Task Fine-tuning Table 9 summarizes our hyperparameters for 64-shot fine-tuning on the 9 KILT eval tasks shown in Table 12 except for MMLU. Given the small amount of examples used ($64 \times 9 = 576$), we fine-tune for a significantly less number of steps at this stage without using warm-up. We evaluate the model every 50 steps, and select the best checkpoint based on the average dev set performance over the 6 development KILT tasks shown in Table 11.

Table 9: Hyperparameters for 64-shot fine-tuning on the eval tasks.

Model	peak lr	end lr	lr scheduler	warm-up	# steps	early stopping	batch size	model parallel	seq len
Llama 65B	1e-5	1e-6	linear	0	100	100	8	8	2048
RA-DIT 13B	1e-5	1e-6	linear	0	100	50	32	2	2048
RA-DIT 65B	1e-5	1e-6	linear	0	100	50	32	8	2048

Retriever Fine-tuning We employ both unsupervised text and downstream tasks for retriever fine-tuning. For the *corpus data*, we randomly sample 900k text chunks from our retrieval corpus to

[11] `https://github.com/facebookresearch/atlas`

form a set of self-supervised data, using the first 50 tokens of each chunk as the input x and the last 50 tokens as the ground-truth output y. In addition, we leverage the multi-task instruction tuning datasets (MTI data) as shown in Table 1, including 10 open-domain question answering and dialog tasks, with a total of 286k training examples. As discussed in §5.1, we observe that, when used alone, the corpus data works slightly better than the downstream tasks. However, combining both types of fine-tuning data yields the best results and outperforms using either source alone. Therefore, we adopt a mixture of 95% corpus data and 5% downstream tasks for retriever fine-tuning in our final model.

We fine-tune the DRAGON+ retriever on 16 A100 GPUs using the dpr-scale codebase[12]. The retriever is fine-tuned using a learning rate of 1e-5 with 1237 warmup steps (DRAGON default), a per-GPU batch size of 32, and a temperature $\tau = 0.01$, for a single epoch over a combination of *5% MTI data* and 95% *corpus data*. We adopt the KL-divergence loss as discussed in Section 2.4 using the top-10 retrieved chunks for each example. For simplicity and efficiency, we produce the top-10 retrieved chunks and their LSR scores (Eqn. 4) using LLAMA 65B and DRAGON+, and do not update them during R-ft. Furthermore, as only the query encoder is fine-tuned, there is no need to update the chunk embeddings in the retriever index. Model validation is performed once every 500 steps using the same mean reciprocal rank (MRR) metric as in the original DRAGON paper (Lin et al., 2023), on a combined validation set from the 10-task MTI data.

Inference Without further specification, we use the top-10 retrieved text chunks for a given example (i.e. k = 10) and ensemble their predictions during inference. For multi-choice tasks, we compute the weighted average probability of each choice items according to Eq. 2 and select the choice with the highest probability. For generation tasks, we perform decoding using each augmented prompt independently, compute the weighted average probability of each unique generated answer, and output the answer with the highest probability.[13] When computing probabilities of output answers, we use several scoring functions: "nll", "nll_char", "nll_token", and "nll_compl". "nll" is the sum of negative log likelihood across all tokens in the sequence. "nll_char" and "nll_token" are "nll" divided by the numbers of characters and subword units in output answers respectively. "nll_compl" selects answers based on the probability divided by the probability of the answer given "Answer:": $\frac{p(y|x)}{p(y|\text{"}Answer\text{:"})}$.

C FINE-TUNING DATASET TEMPALTES

Table 10: Instruction template used for our fine-tuning datasets. <inst_s>, <inst_e> and <answer_s> are special markers denoting the start and the end of a field.

Category	Instruction Tuning Template	Query Template
Dialogue	Background: {retrieved passage}\n\nQ: {turn$_1$} A: {turn$_2$} Q: {turn$_3$} A: ...	{turn$_1$} {turn$_2$} {turn$_3$} ...
Open-domain QA	Background: {retrieved passage}\n\n<inst_s> {question} <inst_e> <answer_s> {answer}	{question}
Reading Comprehension	Background: {context}\n\n<inst_s> {question} <inst_e> <answer_s> {answer}	{question}
Summarization	Background: {context}\n\nSummarize this article: <inst_e> <answer_s> {summary}	
Chain-of-thought Reasoning	Background: {retrieved passage}\n\n<inst_s> {instructions} {reasoning chain} <answer_s> {answer}	{question}

Table 10 shows the templates we used to serialize our instruction tuning datasets. Following Chung et al. (2022b) and Iyer et al. (2022), we randomize the field markers used during training to avoid overfitting. In pariticular, when serializing a task example, we randomly sample from {"Q:", "Ques-

[12] https://github.com/facebookresearch/dpr-scale

[13] A more sophisticated implementation of ensembling for generation tasks involves computing a weighted ensemble of the output distribution at every step and then sampling from this distribution. However, we opt for the simpler implementation as it performs reasonably well and allows us to execute inference with fewer GPUs.

Table 11: Our evaluation datasets. † indicates the development datasets we used to select fine-tuning hyperparameters.

Task	Dataset name	Acronym	Metric	Score
Open-domain QA	MMLU (Hendrycks et al., 2021a)	MMLU	Acc.	nll
	Natural Questions (Kwiatkowski et al., 2019)	NQ	EM	nll
	TriviaQA (Joshi et al., 2017)	TQA	EM	nll
	†HotpotQA (Yang et al., 2018)	HoPo	EM	nll
	ELI5 (Fan et al., 2019)	ELI5	Rouge-L	nll_token
Fact Checking	†FEVER (Thorne et al., 2018)	FEV	Acc.	nll
Entity Linking	†AIDA CoNLL-YAGO (Hoffart et al., 2011)	AIDA	Acc.	nll
Slot Filling	†Zero-Shot RE (Levy et al., 2017)	zsRE	Acc.	nll
	†T-REx (Elsahar et al., 2018)	T-REx	Acc.	nll
Dialogue	†Wizard of Wikipedia (Dinan et al., 2019)	WoW	F1	nll_token
Commonsense Reasoning	BoolQ (Clark et al., 2019)	BoolQ	Acc.	nll_compl
	PIQA (Bisk et al., 2020)	PIQA	Acc.	nll_char
	SIQA (Sap et al., 2019)	SIQA	Acc.	nll_char
	HellaSwag (Zellers et al., 2019)	HellaSwag	Acc.	nll_char
	WinoGrande (Sakaguchi et al., 2019)	WinoGrande	Acc.	nll_char
	ARC-Easy (Clark et al., 2018)	ARC-E	Acc.	nll_char
	ARC-Challenge (Clark et al., 2018)	ARC-C	Acc.	nll_char
	OpenBookQA (Mihaylov et al., 2018)	OBQA	Acc.	nll_compl

tion: ", and ""} for <inst_s>, set <inst_e> to "\n" and randomly sample from {"A:", "Answer:"} for <answer_s>.

D Evaluation Datasets and Templates

Table 12: Language model prompts and retriever query templates used for our evaluation datasets. We did not perform retrieval for commonsense reasoning tasks evaluation.

Task	LLM Prompt Template	Query Template
Knowledge-Intensive Tasks		
MMLU	Background: {retrieved passage}\n\nQuestion: {question}\nA. {choice}\nB. {choice}\nC. {choice}\nD. {choice}\nA: {answer}	{question}\nA. {choice}\nB. {choice}\nC. {choice}\nD. {choice}
NQ, TQA, ELI5, HoPo, zsRE	Background: {retrieved passage}\n\nQ: {question}\nA: {answer}	{question}
AIDA	Background: {retrieved passage}\n\n{context}\nOutput the Wikipedia page title of the entity mentioned between [START_ENT] and [END_ENT] in the given text\nA: {answer}	{context} tokens between [START_ENT] and [END_ENT]
FEV	Background: {retrieved passage}\n\nIs this statement true? {statement} {answer}	{statement}
T-REx	Background: {retrieved passage}\n\n{entity_1} [SEP] {relation} \nA: {answer}	{entity_1} [SEP] {relation}
WoW	Background: {retrieved passage}\n\nQ: {turn$_1$}\nA: {turn$_2$}\nQ: {turn$_3$} ...\nA: {answer}	{turn$_1$} {turn$_2$} {turn$_3$} ...
Commonsense Reasoning Tasks		
ARC-E, ARC-C	Question: {question}\nAnswer: {answer}	
BoolQ	{context}\nQuestion: {question}\nAnswer: {answer}	
HellaSwag	{context} {ending}	
OpenbookQA	{question} {answer}	
PIQA	Question: {question}\nAnswer: {answer}	
SIQA	{context} Q: {question} A: {answer}	
WinoGrande	{prefix} {answer} {suffix}	

Table 11 shows the evaluation datasets used in our experiments. For dev set evaluation, we use a maximum of 2500 randomly sampled examples from the respective official dev sets to reduce the computational cost. For test set evaluation, we use the full set to ensure fair comparison with

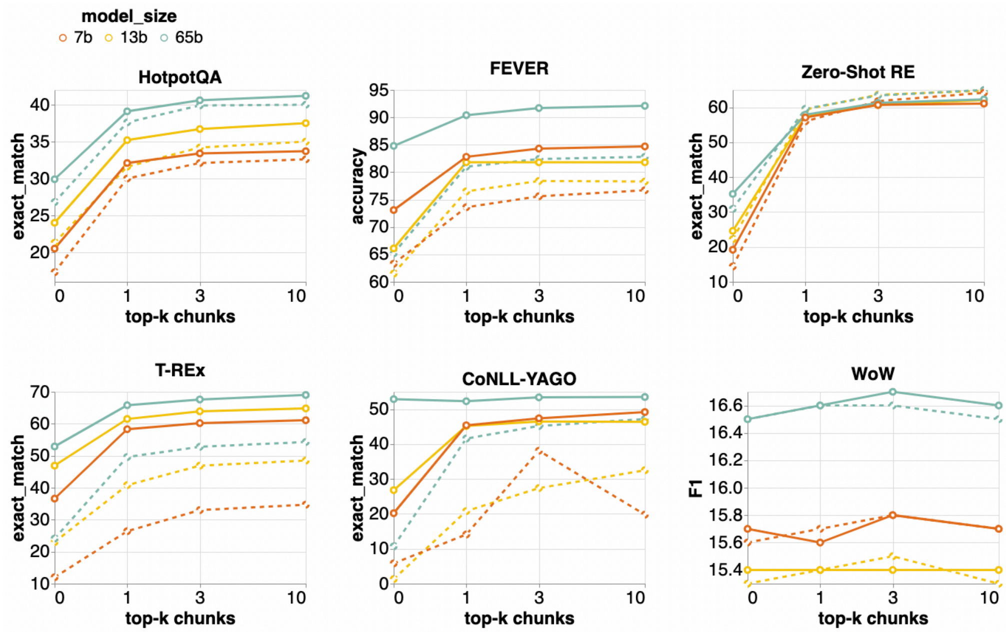

Figure 2: RA-IT model performance (combined with DRAGON+) across sizes 7B, 13B and 65B on our development tasks. 0-shot performance: dashed lines; 5-shot performance: solid lines.

previous work. We also describe the language model instruction templates and retriever queries used in our evaluation in Table 12.

E Additional Experiments

E.1 Scaling Laws of Retrieval Augmented Language Model Fine-tuning

We investigate the impact of the base language model size when retrieval-augmented instruction tuning is applied, and summarize the results in Figure 2. We combine the fine-tuned models with the base DRAGON+ retriever in this set of experiments.

Overall, all models substantially benefit from retrieval augmentation, with smaller models witnessing even bigger improvements. We further note that retrieval augmentation can be an effective strategy for enhancing the performance of smaller models (hence reducing pre-training and inference costs), given the 7B model leveraging > 1 retrieved chunks surpassed the performance of the vanilla 65B model on several tasks. This trend also differs across tasks. For tasks that primarily measure one-hop fact look-up abilities (such as Zero-Shot RE and T-REx), retrieval augmentation provides significant improvements across all model sizes and can bring the performance of smaller models closer to that of their larger counterparts. For more complex tasks (such as HotpotQA and WoW), the advantage of using a larger LLM remains prominent.

F Examples

In this section, we show the task prompts, the corresponding retrieved passages and model predictions generated by LLAMA 65B instruction-tuned with retrieval augmentation (RA-IT 65B) and LLAMA 65B instruction-tuned conventionally (IT 65B) on selected task examples.

F.1 HotpotQA

We analyze the performance of the two models on the development set of HotpotQA in the zero-shot setting, under which RA-IT 65B outperforms IT 65B by a large margin (Table 4). Table 13 show two examples from the HotpotQA development set where RA-IT 65B makes a correct prediction while IT 65B makes a wrong one. First, we observed that the dense retriever struggles to return helpful text chunks for the multi-hop questoins in the HotpotQA dataset. Most of the returned text chunks do not contain information that helps the prediction. In this case, the IT 65B model shows a stronger tendency to be misled by distractors in the retrieved text chunk, since it has not been exposed to noisy passages during fine-tuning. It also tends to predict "I don't know" more frequently[14], while the RA-IT 65B can ignore the noisy text chunks retrieved and leans on its parametric knowledge to predict the correct answer (Mallen et al., 2023). We also observe that when both models generate wrong predictions due to the distractors (e.g. for the third text chunk in the second example), the generation probability of the wrong answer from RA-IT 65B is much lower; and in cases when both models ignore the noisy text chunks and utilize the parametric knowledge, RA-IT 65B generates the correct answer with a higher probability (e.g. for the second text chunk in the first example).

[14]As discussed in §2.2, this behavior is induced by fine-tuning on SQuAD v2.0 (Rajpurkar et al., 2018), which trains the model to predict "I don't know" for passages that does not match with the given question.

Table 13: Example predictions in HotpotQA (dev set) in the 0-shot setting ensembling 10 retrieved text chunks. The top-3 retrieved chunks and the corresponding model predictions are shown. RA-IT 65B and IT 65B are used to generate these outputs.

<table>
<tr><th rowspan="2">Prompt</th><th rowspan="2">p_R</th><th colspan="2">Output</th><th colspan="2">nll$_{LM}$</th></tr>
<tr><th>RA-IT</th><th>IT</th><th>RA-IT</th><th>IT</th></tr>
<tr><td colspan="6">Input: Charlotte Hatherley initially came to prominence in a band formed in what year? Label: 1992.
RA-IT 65B final prediction: 1992 ✓
IT 65B final prediction: 1997 ✗</td></tr>
<tr><td>Background: Charlotte Hatherley Born in London, Hatherley was brought up in West London and attended Chiswick Community School. Her music career began at the age of 15, when she joined British punk band Nightnurse. Two years later, with Ash looking for a guitarist to add to their live sound, Hatherley was hired after frontman Tim Wheeler saw her play at a Nightnurse gig. Hatherley's Ash debut was at Belfast's Limelight on 10 August 1997, and the following week the new lineup played the 1997 V Festival in front of 50,000 people. Her recording career with the band began later that year on the single Ä Life Less Ordinaryänd continued on the album Nu-Clear Sounds in 1998. Hatherley was a full-time member of Ash for eight years, playing on three studio albums, and wrote a handful of the band's songs, most notably Grey Will Fade, on the B-side of the single There's a Star. The song was a cult favourite among fans, and eventually became the title track of Hatherley's debut solo album. On 20 January 2006 it was announced that Hatherley would be leaving Ash in an amicable breakup.\n\nQ: Charlotte Hatherley initially came to prominence in a band formed in what year?\nA:</td><td>0.27</td><td>1992</td><td>1997</td><td>1.16</td><td>1.01</td></tr>
<tr><td>Background: WM: Charlotte Hatherley only... so CD fans might still have to shell out big bucks for an import. Oh, in case you were wondering who Hatherley is, I first heard of her as the ëgirl guitaristïn the band Ash - a band that I have been a fan of since the early 90s when I was getting into all these Britpop-type bands. She naturally started doing her own solo material and left the band a few years ago. The last I heard of her was she was in the band new waver Client with Kate Holmes (not to be confused with the\n\nQ: Charlotte Hatherley initially came to prominence in a band formed in what year?\nA:</td><td>0.21</td><td>1992</td><td>1992</td><td>0.46</td><td>0.98</td></tr>
<tr><td>Background: Charlotte Hatherley Charlotte Franklin Hatherley (born 20 June 1979) is an English singer, songwriter, guitarist and soundtrack composer. She initially came to prominence as guitarist and backing vocalist for alternative rock band Ash. Since leaving Ash in 2006, she has pursued a solo career and acted as a touring instrumentalist for Bryan Ferry, KT Tunstall, Bat for Lashes, Cold Specks, Rosie Lowe and Birdy. Hatherley has also been a touring member of NZCA Lines and is currently musical director for South African artist Nakhane.\n\nQ: Charlotte Hatherley initially came to prominence in a band formed in what year?\nA:</td><td>0.13</td><td>1992</td><td>I don't know.</td><td>0.54</td><td>0.72</td></tr>
<tr><td colspan="6">Input: Oxley Highway ends at a coastal town that had how many inhabitants in June 2016 ? Label: 45,698.
RA-IT 65B final prediction: 45,698 ✓
IT 65B final prediction: I don't know. ✗</td></tr>
<tr><td>Background: Oxley Electorate: Ipswich Motorway: 1 Dec 2016: House debates (OpenAustralia.org) Oxley Electorate: Ipswich Motorway The Ipswich Motorway is a vital link supporting the Queensland economy. It forms part of the national land freight network providing connectivity for industry to the Acacia Ridge intermodal facility, the major industrial area of Wakool and the Brisbane markets at Rocklea2014in the member for Morton's electorate2014which are the state's largest fruit and vegetable markets and a major centre for produce on the east coast. The section of the motorway is over capacity with 93,000 vehicles on average each day, including up to 12,000 freight vehicles. Numbers are increasing each year at an average of four\n\nQ: Oxley Highway ends at a coastal town that had how many inhabitants in June 2016 ?\nA:</td><td>0.25</td><td>10,000</td><td>I don't know.</td><td>7.27</td><td>0.61</td></tr>
<tr><td>Background: Post Offices For Sale NSW — Lotto — Newsagencies — Marlow & Co South Wales about 390 km north of Sydney, and 570 km south of Brisbane. The town is located on the Tasman Sea coast, at the mouth of the Hastings River, and at the eastern end of the Oxley Highway. The town with its suburbs had a population of 45,698 in June 2016. Port Macquarie is a retirement destination, known for its extensive beaches and waterways. Port Macquarie has a humid sub-tropical climate with warm, humid summers and mild winters, with frequent rainfall spread throughout the year. Port Macquarie2019s central business district contains two shopping centres, a marina, the beginnings of\n\nQ: Oxley Highway ends at a coastal town that had how many inhabitants in June 2016 ?\nA:</td><td>0.15</td><td>45,698</td><td>45,698</td><td>0.18</td><td>0.38</td></tr>
<tr><td>Background: The Long Paddock - THE LONG PADDOCK The Long Paddock 4x4, 4WD, caravan, camper trailer, camping products reviews, tests, comparisons by Mark Allen The Long Paddock west, the Oxley Highway is the track you2019ll be aiming for and Tamworth is the major western town of reference on the map. Once you2019re in the main streets of Port, you2019ll wonder no more why in excess of 76,000 people now call the area home. As a rough breakdown, the majority of locals are 25 to 44, followed closely by the 45 to 64 year old bracket 2013 just perfect for all you thrill seeking middle aged folk and laid back grey nomads and let2019s not forget about the younger set that now have oodles of schooling and after-schooling\n\nQ: Oxley Highway ends at a coastal town that had how many inhabitants in June 2016 ?\nA:</td><td>0.12</td><td>76,000</td><td>76,000</td><td>4.85</td><td>0.93</td></tr>
</table>