Towards Large Language Models for Everyone: Instruction Following, Knowledge Retrieval and Multilingualism

Xi Victoria Lin

A dissertation

submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

University of Washington
2024

Reading Committee:
Luke Zettlemoyer, Chair
Yejin Choi
Hannaneh Hajishirzi

Program Authorized to Offer Degree:
Computer Science and Engineering

University of Washington

Abstract

Towards Large Language Models for Everyone:
Instruction Following, Knowledge Retrieval and Multilingualism

Xi Victoria Lin

Chair of the Supervisory Committee:
Professor Luke Zettlemoyer
Computer Science and Engineering

Large language models (LLMs) have significantly advanced the field of Natural Language Processing and demonstrated the potential to fuel a variety of AI applications. Nonetheless, building them in a way that maximally benefits the very wide range of everyday use cases is challenging. Firstly, LLMs are pre-trained with the next-token prediction objective, which does not align well with specific user requests. Secondly, LLMs suffer from knowledge cut-off and tend to hallucinate about long-tail facts. Lastly, popular LLMs are trained on almost exclusively English text, making it difficult for non-English speakers to adopt them.

This thesis presents methodologies addressing all three challenges. We begin by studying the Instruction Meta-Learning (IML) approach, enabling LLMs to perform an array of tasks by fine-tuning them over pairs of natural language instructions and responses. Our study highlights the efficacy of scaling IML along three axes: fine-tuning task diversity, language diversity and model parameters. Next, we propose integrating LLMs with an external data store during IML (retrieval-augmented dual instruction tuning, RA-DIT). RA-DIT significantly improves LLM performance in scenarios that require access to large, external knowledge sources (e.g., answering information-seeking questions). Finally, we introduce a family of cross-lingual generative language models (XGLMs) pre-trained on a multilingual corpus exhibiting a heavy-tailed distribution. XGLMs demonstrate enhanced cross-lingual capabilities and few-shot generalization across medium- and low-resource languages. Together, these research strands provide core strategies for advancing

the boundaries of LLM capabilities and paving the way towards real-world deployment.

Acknowledgements

My PhD is a special chapter in my life which was filled with exhilaration, growth, personal challenges and transformation. Reflecting on this journey, I am struck by the swift passage of a decade and the evolution of my own identity from its onset to its conclusion. It is with a deep sense of gratitude that I acknowledge the abundance of blessings, both familiar and newfound, that have accompanied me through the trials and transformations.

I am deeply grateful to my advisor, Luke Zettlemoyer, for his tremendous support. His open-mindedness, approachable nature, unwavering dedication to science, and astute judgment in research have been truly inspirational. My experience at the Department of Computer Science and Engineering at the University of Washington (UW CSE) was mind-opening. Its vibrant and inclusive culture fosters innovation and collaboration, enabling me to pursue my PhD in two distinct phases: initially as a student within the department, and later as a full-time industrial researcher. I also express my heartfelt gratitude to my late former advisor, Ben Taskar, who had brought me into this dynamic community and an exceptional environment for conducting machine learning research.

I have had the privilege of being mentored by many outstanding researchers through sustained collaboration across various domains of natural language processing and artificial intelligence (AI). My gratitude extends to Ves Stoyanov, who opened my eyes to the landscape of large-scale language modeling (LLMs); to Scott Yih, for a venture into the wonder of open information retrieval and its synergistic integration with LLMs; to Richard Socher, for changing my view of neural networks as universal problem solvers; and to Caiming Xiong, for passing on acute insights into the integration of deep learning with diverse application challenges during the technology's infancy. Additionally, I am also indebted to Mike Ernst, my MSc thesis co-advisor, who encouraged my early work on the then-nascent topic of programming with natural language,

which has now evolved into a flourishing field; and to Sameer Singh, for his mentorship when I just started doing core machine learning research.

The majority of my thesis work was completed at the Fundamental AI Research team at Meta. I extend my sincere gratitude to my managers, Ves Stoyanov, Scott Yih and Daniel Li, for their firm support as I pursued my PhD concurrently with my full-time role. My gratitude also extends to Yejin Choi, Hannaneh Hajishirzi, Ves Stoyanov and Lucy Wang, for being willing to serve on my thesis committee and providing valuable feedback to this work.

Research collaboration has been an invaluable part of my PhD journey. I especially thank the following collaborators for the vibrant intellectual exchanges: Mikel Artetxe, Todor Mihaylov, Naman Goyal, Myle Ott, Stephen Roller, Srini Iyer, Ramakanth Pasunuru, Sida Wang, Tianlu Wang, Jingfei Du, Xilun Chen, Mingda Chen, Mona Diab, Omer Levy, Mike Lewis, Wang-Chiew Tan and Alon Halevy at Meta; Nazneen Rajani, Victor Zhong, Yingbo Zhou, Bryan McCann, Nitish Keskar, James Bradbury, Melvin Gruesbeck, Kazuma Hashimoto, Alex Trott, Romain Paulus, Stephen Merity, Chien-Sheng Wu, Xinyi Yang, Tian Xie, Wenpeng Yin, Tong Niu, Stephen Hoi and Mark Riedl at Salesforce; Chenglong Wang, Calvin Loncaric, Deric Pang and Kevin Vu for the partnership on Tellina; Tao Yu, Rui Zhang and Dragomir Radev for the quest on natural language interfaces to structured data and beyond. I am also grateful to my internship mentors at Microsoft Research, Kristina Toutanova, Scott Yih, Hoifung Poon, and Chris Quirk, as well as those at the Allen Institute for Artificial Intelligence (AI2), Tom Kwiatkowski, for their guidance and insights.

I was fortunate to be part of the Natural Language Processing Group at the University of Washington (UW NLP), a vibrant and diverse hub of researchers across various NLP disciplines. I thank my wonderful groupmates for their support and insightful research exchanges, especially Kenton Lee, Luheng He, Eunsol Choi, Nicholas FitzGerald, Mark Yatskar, Srini Iyer, Mandar Joshi, Omer Levy, Mike Lewis, Yannis Konstas, Chloé Kiddon, Sameer Singh, Victor Zhong, Weijia Shi, Sewon Min, Margaret Li, Ari Holtzman, Tim Dettmers, Suchin Gururangan, Huan Sun, Lingpeng Kong, Minjoon Seo, Aaron Jaech, Yi Luan, Rowan Zellers, Trang Tran, Tongshuang Wu, Akari Asai and Yizhong Wang. Additionally, a special thanks to Maarten Sap and Hannah Rashkin for preparing this widely adopted thesis template.

I also extend my gratitude to the wonderful staff at UW CSE, especially our graduate student service

(GSR) directors, Elise deGoede Dorough and Lindsay Michimoto, who have offered substantial help during each milestone of my PhD. My gradtitude also extends to my professors and mentors at the Department of Computer and Information Science at the University of Pennsylvania (UPenn CIS), who have provided invaluable guidance during the initial phase of my graduate studies.

My PhD journey was enriched by the invaluable friendship from many. I am grateful to Diyi Yang, Jessy Li, Wendy Shang, He He, Sheng Zha and Chang Liu for their enduring friendship and support through the tough times; to my officemates at UW, Ryan Mass, Dominik Moritz, Thierry Moreau, Kristi Morton, and Alex Colburn, for a dynamic working environment rich with interdisciplinary expertise; to Irene Zhang for her mentorship upon my arrival at UW, and for being an inspiring role model and advocate for women in computer science and engineering. I also thank Shu Liang, Chenglong Wang, Pavel Panchekha, Yanping Huang, Luheng He, Qiao Zhang, Xiaoyi Zhang, Shumo Chu, Tianqi Chen, Tianyi Zhou, Audrey Cook, Jennifer Gillenwater, David Weiss, Brian Dolhansky, Robert Rand, Meng Xu, Fei Miao, Congle Zhang, Yuyin Sun, Shirley Ren, Jinna Lei, Bing Xu, Mianmian Wen, Tim Shi, Navjot Matharu, Yufei Hu, Kerui Min and the graduate student communities at UW CSE and UPenn CIS for adding joy and fun to this ride.

Lastly, I am grateful to Anat Caspi and Aviv Taskar, who continue to inspire me daily. I am also thankful for Daisy, Evan, and Angela, my extended family in the Bay Area, whose companionship over the years has been invaluable in navigating the challenges of balancing a PhD and a demanding career in the industry environment of Silicon Valley. My deepest appreciation goes to my parents and grandparents, who bestowed me the freedom to pursue my dreams, all while providing unwavering love, support and inspiration.

DEDICATION

In memorial of Ben Taskar

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Large language models (LLMs) are transformer-based [Vaswani et al., 2017] architectures with billions of parameters pre-trained over tremendous amount of unlabeled data. They have shown substantial promise as general-purpose, multi-task learners [Bommasani et al., 2021], and have fueled successful products such as ChatGPT[1], Meta AI[2] and Google Bard[3]. This thesis tackles three key challenges in *enhancing the usability of pre-trained LLMs* by proposing methodologies that enable them to follow natural language instructions, access external knowledge and process input in different languages.

1.1 The Rise of Large Language Models

Contextualized language pre-training was introduced by the pioneering work of McCann et al. [2017], Peters et al. [2018] and Devlin et al. [2019]. These early work adopt an encoder-only network architecture, leaving practitioners to separately fine-tune prediction gates or a decoder for various downstream tasks. Raffel et al. [2020] and Lewis et al. [2020a] further extended this paradigm to pre-traing sequence-to-sequence architectures. Meanwhile, GPT [Radford and Narasimhan, 2018] and GPT-2 [Radford et al., 2021] demonstrated the potential for simple autoregressive language models trained with the next-token prediction objective to function as end-to-end multi-task learners. For example, given the question "*Who wrote the book the origin of species?*" as the prompt, the model generates "*Charles Darwin*" as the response. GPT-3 [Brown et al.,

[1] `https://chat.openai.com/auth/login`
[2] `https://www.facebook.com/help/messenger-app`
[3] `https://bard.google.com`

2020] further showcased the power of scaling language model pre-training along two dimensions: the model size and the amount of pre-training data. With 175 billion parameters trained over 300 billion English tokens, the model acquired many "emergent capabilities", including being able to follow few-shot examples formatted with textual templates to perform new tasks [Min et al., 2022b].

Subsequent work has further improved LLMs by scaling up both model size and pre-training data [Chowdhery et al., 2022; Touvron et al., 2023a,b; Hoffmann et al., 2022]. However, scaling LLM pre-training is extremely costly, and to date, has only been done by a small number of organizations. While some of the most powerful LLMs are proprietary, the capabilities of open-sourced LLMs have consistently risen thanks to investments from multiple institutions. The work in this thesis are based on strong open-sourced LLMs such as OPT [Zhang et al., 2022] and Llama [Touvron et al., 2023a]. Additionally, we introduce two new LLMs: the instructable LLM (OPT-IML), presented in Chapter 2, and the multilingual LLM (XGLM), introduced in Chapter 4.

1.2 The Alignment Problem

While pre-trained LLMs are capable of multi-tasking in both zero- and few-shot settings [Brown et al., 2020], their performance is extremely sensitive to the prompt formulation [Gao et al., 2021; Liu et al., 2023]. Moreover, the self-supervised learning objective, which focuses on predicting the next token, inherently results in pre-trained LLMs being particularly good at completing sentences. As a result, they are unable to directly respond to most user requests specified in natural language (Figure 1.1). To address this problem, several concurrent work [Mishra et al., 2022; Wei et al., 2022b; Sanh et al., 2022] introduced *instruction-tuning*, an approach that fine-tunes LLMs using a collection of tasks expressed via instruction and ground-truth response pairs. This improves their problem-solving abilities when provided with a task description in natural language. The resulting LLMs, commonly referred to as *instructable LLMs*, demonstrated significant performance improvement following instructions in a zero-shot manner. This enhancement in usability made these models much more practical and useful for a wide range of applications.

Initial scaling-law curves from these studies suggest the potential for improved cross-task generalization through further scaling of both *model size* and *task diversity*. We therefore pose the following research question – ***can we enable an LLM to follow instructions to solve common natural language processing***

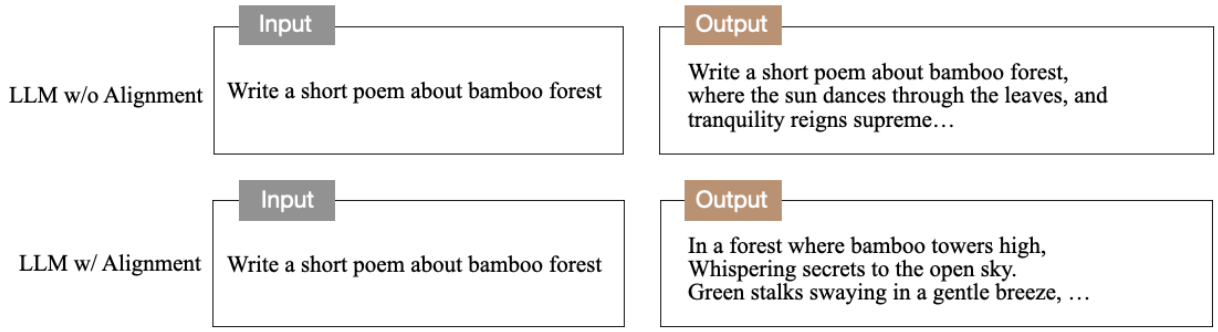

Figure 1.1: Pre-trained LLMs without instruction tuning tend to complete any input. Instruction tuning enables the model to generate content in response to the input specification (output generated using ChatGPT).

(NLP) tasks by fine-tuning it over a massive instruction dataset with high diversity? We consider NLP tasks that belong to a broad taxonomy, including *lexical tasks* (e.g. word analogy), *syntactic tasks* (e.g. part-of-speech tagging), *semantic tasks* (e.g. textual entailment) and *common NLP applications* (e.g. question answering, summarization and hate speech detection). Each level of the taxonomy includes more fine-grained task categories (with examples provided in parentheses). We use these tasks as the test bed to examine the generalization capabilities of instruction-tuned LLMs.[4]

To maximize the task and language diversity in our instruction tuning data, we compiled up to 2,000 NLP datasets from 8 instruction-tuning benchmarks previously released by the community (Table 2.1). In order to systematically evaluate the generalization capability of LLMs, we split the task collection into a training set and three evaluation sets designed to test three types of model generalization:

- to *unseen task categories (new instructions, new tasks, in- and out-of-domain examples);*
- to *unseen datasets from the same task category (new instructions, seen tasks, in- and out-of-domain examples);*
- to *unseen examples from the same dataset (seen instructions, in-domain examples).*

We fine-tuned OPT [Zhang et al., 2022] with 30B and 175B parameters on the training split of our benchmark, after carefully re-balancing the training data across multiple attributes such as task categories and annotation sources. We dubbed the resulting LLMs OPT-IML (OPT with *instruction meta learning*),

[4] Wang et al. [2022b] first introduced Super-NaturalInstructions, an instruction-tuning dataset consisting of 1,600+ NLP tasks. However, it only experimented with the encoder-decoder based T5 LMs [Raffel et al., 2020] of up to 13B parameters, and the instruction data were all taken from the same human annotation pipeline. Chung et al. [2022b] is the most closely related to our work. Our projects were initiated around the same time.

and the corresponding meta benchmark as OPT-IML Bench. ***Our experiments demonstrated the efficacy of developing instructable NLP task solvers by fine-tuning LLMs over a large and diverse instruction dataset.*** First, OPT-IML performs strongly when evaluated on unseen examples from a dataset included in the fine-tuning data. For example, OPT-IML-30B achieves > 0.85 ROUGE-L F1 metrics on SQuAD v1 [Rajpurkar et al., 2016], even though the fine-tuning data contains 1,500+ other datasets, and the model is effectively trained on only a small number of SQuAD v1 examples. This suggests that increasing the LLM size and engaging in extensive pre-training can effectively mitigate the task interference problem often observed in smaller multi-task models [McCann et al., 2018a], resulting in performance on par with state-of-the-art supervised learning models. Second, OPT-IML demonstrates strong generalization abilities over unseen instructions, both across different datasets and task categories. OPT-IML significantly improves over its base pre-trained model at both 30B and 175B scales on four different benchmarks: PromptSource [Sanh et al., 2022], FLAN [Wei et al., 2022b], Super-NaturalInstructions [Wang et al., 2022b], and UnifiedSKG [Xie et al., 2022]. Additionally, the OPT-IML models also perform competitively in comparison with each of the prior models individually tuned on these benchmarks in both zero and few-shot performance (§2.7).

We further ask the research question – ***what are the important factors that impact the effectiveness of instruction-tuning?*** Using our evaluation framework that considers multiple levels of generalization, we characterize the tradeoffs relating to different factors when scaling up instruction-tuning to the aggregate of 8 different benchmarks. We outline the tradeoffs of dataset and benchmark sampling strategies during tuning, the scaling laws with respect to tasks and categories, the effects of incorporating task demonstrations into instruction-tuning based on Min et al. [2022a], as well as instruction-tuning with specialized datasets that contain reasoning chains [Wei et al., 2022c] and dialogue. These experiments establish best practices for large-scale instruction-tuning of LLMs (§2.4).

While OPT-IML demonstrate a significant multi-tasking capability jump compared to its base model, we still observe model weaknesses similar to those observed in smaller multi-task learning models. For example, when evaluating on unseen instructions, the model performs better on instructions and tasks similar to those in the fine-tuning set. Adding more dissimilar instructions to the training set can hurt the performance. This highlights the importance of future research on multi-task learning and continuous learning using LLMs.

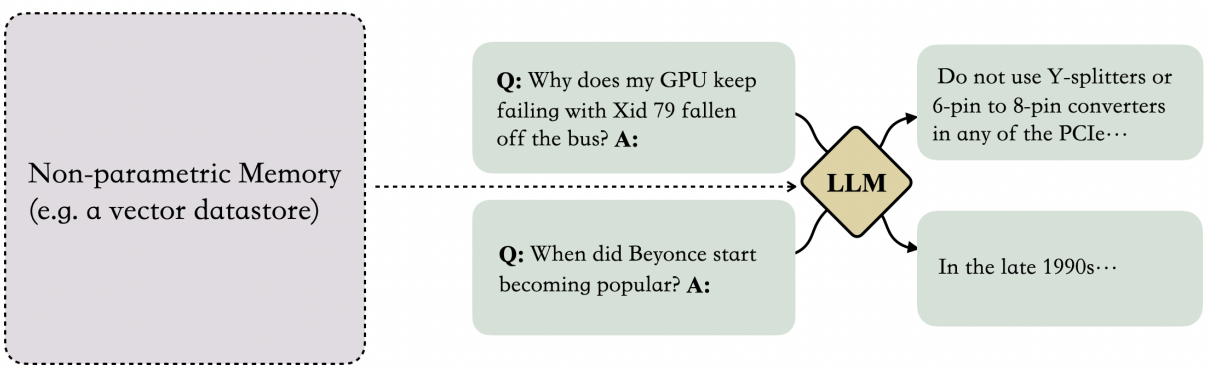

Figure 1.2: Non-parametric memory enables LLMs to solve a broader range of tasks, especially those requiring access to long-tail and private knowledge.

Two months prior to the publication of our work, Chung et al. [2022b] published FLAN-PaLM and FLAN-T5 along similar lines. FLAN-PaLM and FLAN-T5 achieve impressive gains on the challenging benchmarks of MMLU [Hendrycks et al., 2021a] and Big-Bench Hard [Suzgun et al., 2023] by instruction-tuning PaLM [Chowdhery et al., 2022] and T5 [Raffel et al., 2020] on a scaled-up collection of 1,800+ tasks. OPT-IML trained under similar settings still underperforms in comparison on these challenging benchmarks. In subsequent experiments, we determined that the primary factor contributing to the performance discrepancy between OPT-IML and FLAN-PalM is the inadequate training of the base LLM. The comparison with FLAN-T5, apart from differences in training data size, can also be attributed to architectural disparities between autoregressive LLMs and T5 [Artetxe et al., 2022b]. We believe that the insights gained from the experiments, particularly with regards to the importance of adequate pre-training, will be valuable for guiding future research in this area.

1.3 Retrieval Augmentation

Although LLMs post-alignment demonstrate strong instruction-following capabilities and are useful for various tasks, they suffer from the knowledge cutoff problem and cannot solve problems requiring knowledge beyond their pre-training cutoff date. Additionally, LLMs struggle with modeling long-tail knowledge and are inaccurate within this range. *Retrieval augmentation (RA)* enables LLMs to incorporate external knowledge for problem-solving by integrating them with non-parametric information retrieval (Figure 1.2) [Guu et al., 2020; Borgeaud et al., 2022; Shi et al., 2023b].

Previous work has applied retrieval augmentation to different language model architectures and in different model training stages. REALM [Guu et al., 2020] (encoder-only) and RETRO [Borgeaud et al., 2022] (autoregressive) opt for *end-to-end pre-training*, incorporating the retrieval component from the outset. Atlas [Izacard et al., 2022b] builds upon T5 [Raffel et al., 2020] with Fusion-in-Decoder [Izacard and Grave, 2021] modifications, and continuosly pre-trains the framework over unsupervised text. REPLUG [Shi et al., 2023b] and In-Context RALM [Ram et al., 2023] directly combine off-the-shelf LLMs and retrievers, showing that LLMs and retrievers, even when optimized independently, can be effectively fused through the emergent in-context learning capbabilities of LLMs.

However, incorporating retrieval during LLM pre-training significantly increases the system complexity and training cost, and the off-the-shelf fusion approach also has limitations, particularly as the LLMs are not inherently trained to incorporate retrieved content. Given fine-tuning is much cost-effective than pre-training, we pose the research question – ***is it possible to introduce retrieval augmentation during the LLM alignment stage while adapting the LLM to downstream tasks?*** To this end, 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 LM prediction.

The RA-DIT approach consists of two separate instruction tuning steps. For *language model fine-tuning*, we augment each fine-tuning prompt with a retrieved "background" field prepended to the instructions and fine-tune the LM using the label-loss objective [Chung et al., 2022b; Iyer et al., 2022]. 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*, we update the retriever using a generalized language-model-supervised retrieval objective [LSR, Shi et al., 2023b]. This way, we enable the retriever to yield more contextually relevant results, aligned with the preferences of the LLM. During inference, 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 demonstrate that each fine-tuning step leads to significant performance gains, and the fine-tuned LLM and retriever can be combined to achieve further improvements. We initialize the framework us-

ing pre-trained LLAMA [Touvron et al., 2023a] and a state-of-the-art dual-encoder based dense retriever, DRAGON+ [Lin et al., 2023a]. 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., 2021a] (+8.2% 0-shot; +0.7% 5-shot) and Natural Questions [Kwiatkowski et al., 2019] (+22% 0-shot; +3.8% 5-shot). RA-DIT 65B also substantially outperforms ATLAS 11B on eight 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 conducted a comprehensive model analysis, demonstrating the effectiveness of our approach across LLMs of varying sizes (7B, 13B and 65B). On average, smaller LLMs exhibited significantly larger relative performance improvements. Additionally, for simpler single-hop information retrieval tasks, both the 7B and 65B Llama models performed comparably with retrieval augmentation. However, in more complex tasks such as multi-hop question answering, retrieval augmentation failed to bridge the gap between smaller and larger LLMs, suggesting that parameter scaling confers unique capabilities to the model that cannot be replicated solely through non-parametric memory, as implemented in our approach.

We also found the retrieval component to be a fragile part of the framework, as the retriever frequently made errors and returned noisy text chunks (post fine-tuning). Through qualitative error analysis, a significant portion of our model's performance improvement can be attributed to its ability to effectively disregard irrelevant retrieval content, outperforming an LLM baseline fine-tuned with conventional instruction tuning methods. This observation highlights the importance of developing more robust and accurate retrieval mechanisms to further enhance the performance of retrieval-augmented language models. Additionally, we investigated the impact of different retrieval strategies on the overall performance, such as using different search algorithms or adjusting the number of retrieved documents. Our findings suggest that careful selection and optimization of the retrieval strategy can significantly contribute to the success of the approach.

1.4 Multilingualism

LLMs are impressive multi-task learners and can be used to solve a wide range of tasks. However, their pre-training data is predominantly English, making it difficult for non-English speakers to benefit from them. Although the training data of GPT-3 [Brown et al., 2020] contains a small percentage of non-English text (7%) allowing it to achieve some promising cross-lingual generalization, the model is almost exclusively deployed for use cases in English. Multilingual masked and sequence-to-sequence language models have been studied, including mBERT, XLM-R, mT5, and mBART Devlin et al. [2019]; Conneau et al. [2020]; Xue et al. [2021]; Liu et al. [2020]. These models are typically fine-tuned on large amount of labeled data in downstream tasks. In comparison, multilingualism for autoregressive LLMs is less well understood. This presents a significant challenge for developing LLMs that can effectively serve users across different languages and regions. Addressing this issue is crucial for ensuring that the benefits of LLM technology are accessible to a broader population and not limited to English speakers.

We pose the following research questions – ***can we train a multilingual LLM that performs competitively against state-of-the-art English LLMs while demonstrating strong multilingual and cross-lingual capabilities, particularly in medium- and low-resource languages?*** To address this, we train four multilingual generative language models (up to 7.5 billion parameters), dubbed XGLM's, and conducted a comprehensive study of multilingual zero- and in-context few-shot learning. We train the models using a large-scale corpus of 500 billion tokens comprising 30 diverse languages, upsampling the less-resourced languages to render a more balanced language representation.

We evaluate XGLMs on multiple multilingual natural language understanding (NLU) tasks, machine translation and a subset of English tasks demonstrated in [Brown et al., 2020]. The models demonstrate strong cross-lingual capabilities, with competitive zero- and few-shot learning performance when using English prompts alongside non-English examples. Our largest model (XGLM 7.5B) achieves strong zero- and few-shot learning performance on language completion and inference tasks, such as XStoryCloze (65.4% 0-shot, 66.5% 4-shot) and XNLI (46.3% 0-shot, 47.3% 4-shot). Additionally, it established a new state-of-the-art on few-shot machine translation across numerous language pairs in the FLORES-101 benchmark Goyal et al. [2022], significantly outperforming the GPT-3 model of comparable size (6.7 billion parameters).

However, we found that multilingual pre-training leads to performance drop on English. On 8 En-

glish natural language understanding tasks, XGLM 7.5B underperforms GPT-3 6.7B by an average of 10.9% in zero-shot learning. GPT-3 6.7B also outperforms XGLM 7.5B in machine translation on several high-resource language pairs, including WMT-14 English-French, WMT-16 English-German, and WMT-19 English-Chinese. There are multiple reasons why XGLM 7.5B underperforms English centric models on the English tasks. First, only 32.6% of XGLM 7.5B's 500B-token training data is English while both English-centric models are trained on close to 300B English tokens. Second, the model capacity of XGLM 7.5B is shared by 30 languages, and the "curse of multilinguality" can degrade the performance across all languages [Conneau et al., 2020]. We hypothesize that further scaling up the model capacity and training data can potentially close this gap.

We conduct an in-depth analysis of different multilingual prompting approaches and examine cross-lingual transfer through template and demonstration examples respectively. Our findings reveal that non-English templates sometimes yield unexpected low zero- and few-shot learning accuracy even when crafted by native speakers. However, using English templates or adding demonstration examples proved to be effective remedies. Interestingly, we found that using demonstration examples from another language often fails to further improve zero-shot learning performance when a strong prompting language like English is used. This suggests room for improvement in both cross-lingual pre-training and in-context transfer approaches. Our analysis highlights the importance of carefully selecting prompting languages and demonstration examples to optimize cross-lingual transfer and performance. By understanding these factors, we can develop more effective strategies for improving the multilingual capabilities of LLMs and ensuring their applicability across diverse language contexts.

1.5 Summary

In summary, this thesis focuses on addressing the challenges faced in building LLMs that can cater to a wide range of everyday use cases. The three main challenges include misalignment with user requests, external and tail-range knowledge access issues, and the predominance of English language in widely used LLMs. To tackle these challenges, we present methodologies based on instruction meta-learning (IML), retrieval-augmented dual instruction tuning (RA-DIT), and cross-lingual generative language pre-training (XGLMs). These approaches aim to enhance LLM capabilities by scaling IML along task and language

diversities, integrating external knowledge sources, and improving cross-lingual performance for medium- and low-resource languages. Together, these strategies pave the way for real-world deployment of advanced LLMs, enabling them to better serve diverse user needs.

Chapter 2

Instruction Meta-Learning

Pre-trained LLMs are not aligned to directly respond to user instructions in a manner similar to human communication. Early work adopt in-context learning and prompt engineering techniques when applying LLMs to downstream tasks [Brown et al., 2020]. As a result, the LLM discerns the task semantics during test time through demonstration examples or prompts that bear resemblance to the text that appears in the pre-training corpus. *Instruction Meta-Learning (aka. instruction-tuning)*, the approach that fine-tunes LLMs using a collection of tasks described via instructions, was introduced to improve the problem-solving abilities of LLMs when provided solely with a natural language problem description [Mishra et al., 2022; Wei et al., 2022b; Sanh et al., 2022]. Early studies in this area underscore the potential of increasing the model size and diversifying the fine-tuning tasks. Following this hypothesis, we execute an extreme scaling of the amount of fine-tuning tasks to assess the LLM's ability to concurrently learn a multitude of tasks and to generalize to new tasks via natural language [Iyer et al., 2022].

In this chapter[1], we first introduce OPT-IML Bench, a large benchmark for Instruction Meta-Learning (IML) consisting of 1,991 NLP tasks consolidated into task categories from 8 existing benchmarks. Then we present the experiments comparing different fine-tuning data sampling strategies, and also detail the scaling laws with respect to tasks and categories. Informed by the insights obtained from our ablation experiments, we applied instruction tuning to OPT models [Zhang et al., 2022] at 30B and 175B parameters, subsequently designating the resulting model as OPT-IML. On four different instruction-tuning bench-

[1]The work in this chapter was conducted at Meta AI, in collaboration with Srini Iyer*, Ram Pasunuru*, Todor Mihaylov, Tianlu Wang, Daniel Simig, Veslin Stoyanov†, Luke Zettlemoyer† and others [Iyer et al., 2022]. (*co-first authors, †research leadership)

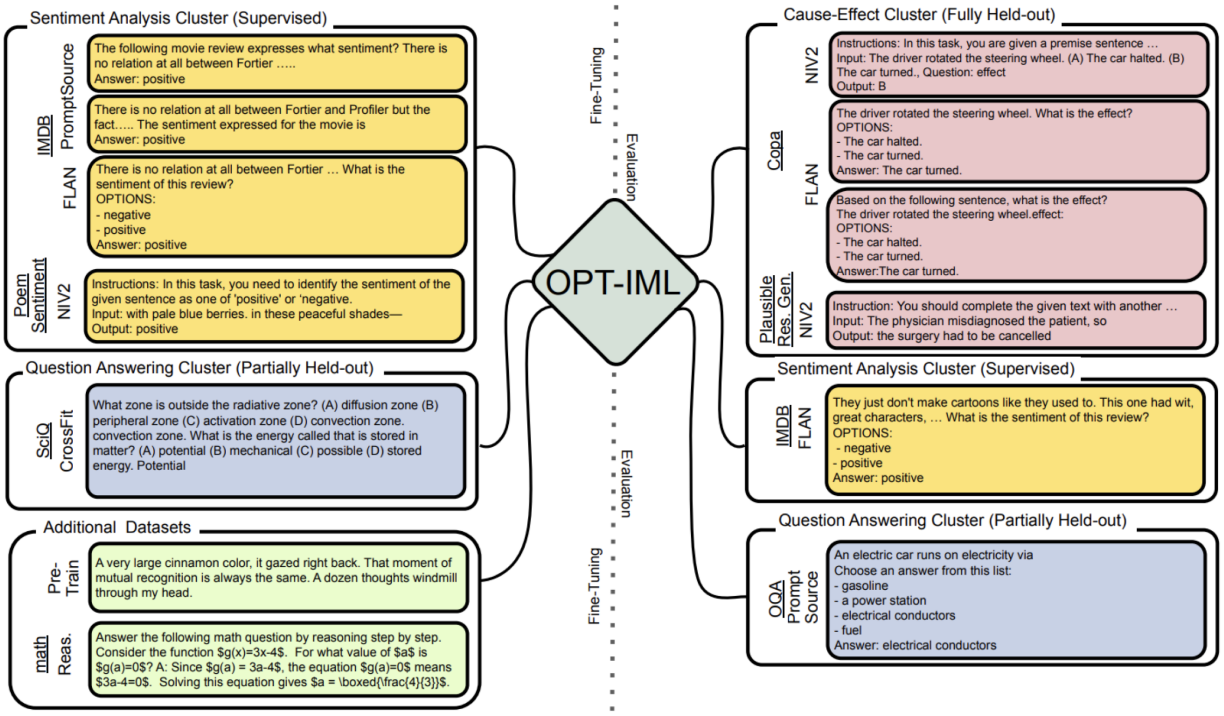

Figure 2.1: OPT-IML is fine-tuned on a large collection of 1500+ NLP tasks divided into task categories (left hand side). Each category contains multiple related tasks, as well as multiple prompts for the same task (e.g. IMDB), aggregated from multiple benchmarks. We evaluate OPT-IML on a set of evaluation categories (right hand-side) which can be disjoint, partially overlap or fully-overlap with the categories used for tuning, corresponding to evaluating model generalization to tasks from fully held-out categories, to tasks from categories seen during training, and to instances from tasks seen during training.

marks: PromptSource [Bach et al., 2022], FLAN [Wei et al., 2022b], Super-NaturalInstructions [Wang et al., 2022b], and UnifiedSKG [Xie et al., 2022], OPT-IML significantly surpassed the performance of previously instruction-tuned models that were specifically optimized for each benchmark in the majority of the settings. On complex reasoning benchmarks including MMLU [Hendrycks et al., 2021a] and Big-Bench Hard [Chung et al., 2022b], OPT-IML demonstrates competitive performance, but trails behind the state-of-the-art such as FLAN-T5 11B [Chung et al., 2022b] and OpenAI `text-davinci-003`. We discuss possible causes of the performance disparity and propose possible strategies for enhancement.

2.1 OPT-IML Bench

We expand the Super-NaturalInstructions benchmark of 1600+ tasks by Wang et al. [2022b] with the task collections from multiple existing work on *instruction-tuning*: FLAN [Chung et al., 2022b], T0 [Bach et al., 2022]; *prompt crowdsourcing*: PromptSource [Bach et al., 2022]; *cross-task transfer studies*: ExMix [Aribandi et al., 2022], T5 [Raffel et al., 2020], CrossFit [Ye et al., 2021]; and *domain-specific task consolidation*: Structured Knowledge Grounding [Xie et al., 2022], Dialogue [Shuster et al., 2022] and Chain-of-thought Reasoning [Chung et al., 2022b]. We found significant overlap between the datasets included in these benchmarks. For example, popular datasets such as SQuAD [Rajpurkar et al., 2016, 2018] appear in almost all benchmarks. In addition, while Super-NaturalInstructions, PromptSource, FLAN and Chain-of-thought Reasoning contain long-form human-authored instructions or reasoning chains, the rest of the benchmarks are designed for multi-task learning and the prompt templates often only consist of short task prefixes (e.g. "question:", "label:"). As a result, we only kept tasks from the CrossFit, ExMix and T5 collections that do not appear in any other benchmarks. We randomly take maximally 100k examples per task from all benchmarks except FLAN, where we take maximally 30k examples per task following the same practice as Wei et al. [2022b].

Benchmark	Instruct. type	# clusters	# tasks	# total examples	Avg. # prompts / task	prompt length mean	prompt length std
Super-NaturalInstructions	task inst.	76	1613	12.4M	1.0	287	882
PromptSource	instance inst.	51	280	12.8M	5.7	179	222
CrossFit	keywords	32	159	7.1M	1.0	117	258
FLAN	instance inst.	12	70	4.4M	8.5	193	375
ExMix	keywords	10	14	0.5M	1.0	132	191
T5	keywords	9	36	1.9M	1.0	111	167
UnifiedSKG	keywords	7	21	0.8M	1.0	444	297
Reasoning	task inst.	1	14	0.4M	1.0	146	122
OPT-IML Bench (train)	mixed	93†	1,545	17.9M	1.7	261	631
OPT-IML Bench (dev)	mixed	7	35	145K	2.9	–	–
OPT-IML Bench (test)	mixed	10	87	321K	4.6	–	–

Table 2.1: The statistics of each existing benchmark is calculated using the original data we downloaded. The statistics of OPT-IML Bench is calculated using the data after we performed task filtering and taking a maximum of M examples per tasks. †The estimation of the number of task clusters in our train set is based on a coarse union of the clustering tags from each original benchmark.

Train, validation and test splits. We split the collection of tasks in a way that allows us to perform massive instruction fine-tuning and evaluate the resulting model with respect to three levels of generalization. First, we hold out several task categories to evaluate model generalization to *new categories of tasks*. Second, we select a subset of the remaining categories as partially held-out categories. We divide the datasets in these categories into train and evaluation and use them to test model generalization to *new datasets from seen task categories*. [2] Finally, for a subset of the training tasks, we hold out the validation and test sets from the original data release, and use them to test model generalization in the standard multi-task learning setting, i.e. *new examples from seen tasks*. We reserve 35 evaluation tasks spanning 9 task categories from the evaluation tasks as the validation set, and use them to characterize the tradeoffs of different instruction-tuning strategies.

Task de-duplication. We make sure that the train and evaluation tasks do not overlap on the data source they were created from, to prevent leakage[3], following the practice of Wang et al. [2022b]. For each pair of train and eval tasks, we compute the fraction of examples that have any 13-gram overlap between the instantiated sequences from those examples. We manually examine every pair where more than 1% of the eval set overlaps with the training set ($\sim$14,000 pairs) to confirm whether tuning on the train task can unfairly benefit the eval task, and decide either to remove the train or the eval task in confirmed cases. The task pairs that share a broad contextual resource such as Wikipedia but otherwise contain unrelated output labels are retained. Table 2.1 shows the statistics of our task splits.

2.2 Approach

To fine-tune the OPT models, we separate the training sequence into a source (context) sequence and a target (answer) sequence and only include loss terms from the tokens in the target sequence. We refer to this method of loss calculation as "label-loss". We treat the task instructions and inputs as source tokens and the label tokens as target tokens. Formally, for a fine-tuning dataset $\mathcal{D}$ comprising source instances s_i and their corresponding target tokens $t_i = \{t_{ij}\}$, we fine-tune the pre-trained auto-regressive LM with parameters θ

[2] We select the fully and partially held-out categories by largely staying consistent with previous instruction fine-tuning work [Wang et al., 2022b; Wei et al., 2022b; Sanh et al., 2022] to allow direct comparison.

[3] This condition is maintained for our partially held-out evaluation tasks as well.

to minimize the following loss over the target tokens conditioned on the source tokens.

$$\mathcal{L}(\mathcal{D};\theta) = -\sum_{i}\sum_{j}\log p_\theta(t_{ij}|s_i, t_{i,<j}) \tag{2.1}$$

We minimize this loss across all training examples in OPT-IML Bench after mixing examples from different datasets. The mixing is done based on the sizes of the datasets and the proportions allocated to the benchmarks they originate from.

Packing and document attention. To boost computational efficiency, we pack multiple examples (source and target) together as a sequence of 2048 tokens [Raffel et al., 2020], separated by <eos> tokens. One consequence of packing is that the tokens belonging to one example can attend to tokens from previously packed examples in the same sequence. We modify the token attention mask in auto-regressive LMs to attend only to the tokens that are part of the same example, rather than all the previous tokens in the sequence. This changes the attention mask from a triangular to a block triangular mask and improves both stability and performance in our experiments.

2.3 Experimental Setup

Fine-tuning Hyperparameters. We fine-tune all 30B models on 64 40GB A100s, and 175B models on 128 40GB A100s. Following OPT, we use Fully Sharded Data Parallel [Artetxe et al., 2022a] and the Megatron-LM Tensor Parallelism [Shoeybi et al., 2019]. We use Adam [Kingma and Ba, 2015] with 32-bit state with $(\beta_1, \beta_2) = (0.9, 0.95)$, linearly warming up the learning rate for 60 steps to the maximum, followed by linearly decaying it to 0. We conduct preliminary experiments to select learning rates from $\{1e^{-5}, 3e^{-5}, 5e^{-5}, 6e^{-5}\}$ and per-GPU batch sizes from {2, 4, 8} using the validation split of OPT-IML Bench. The resulting hyperparameters are listed in Table 2.2. During fine-tuning, our models saw approximately 2 billion tokens, which is only 0.6% of the pre-training budget of OPT (Table 2.2).

Prompt construction details. To compile our train data, we merged all prompt data for a task with N examples and randomly take N prompts from the pool such that the training task distribution is kept the

Model	# Gpus	Batch Size	Learning Rate	Steps	Warm-up Steps	FT Time (h)	# Tokens
OPT-IML 30B	64	256	5e-05	4000	60	19	2B
OPT-IML 175B	128	128	5e-05	8000	60	72	2B

Table 2.2: Fine-tuning parameters for all OPT-IML models.

same regardless of how many prompts are given for the tasks. We merged the prompts for each task in a similar manner in our validation set, and randomly sample a maximum of 250 prompts per task to report the validation results. For our test tasks, we keep all prompt variations and all examples.

Generalization levels. Starting with a baseline instruction-tuned model, we independently characterize the effect of each factor, by tuning models with several variations of that factor and evaluating the models on the tasks from our validation split, separated into three generalization levels: a) tasks from clusters not included in training (Fully Held-out), b) tasks unseen during training but from seen clusters (Partially Supervised), and c) tasks seen during training (Fully Supervised). An instruction-tuning setting is desirable if it improves performance on fully held-out and partially supervised tasks without sacrificing performance on fully supervised tasks. We use average performance across all three generalization levels on both 0-shot and 5-shot settings on the validation/test sets of the tasks in the validation split to determine the best settings for each factor.

Decoding. Our evaluation data comprises tasks with answer candidates (of which one is correct), as well as tasks with multiple gold reference sequences. For the former set of tasks, we use rank classification similar to Brown et al. [2020], where we score each candidate based on their likelihood and output the highest-scoring candidate as the answer. This candidate is used to compute accuracy on the task. For tasks without candidates, we perform greedy decoding until an <eos> token is predicted or a maximum of *N=256* tokens are generated. Based on the generated sequence and the references, we then compute either Exact-match or Rouge-L F1 scores.

Model selection. For all experiments, we first aggregate results separately for 0-shot and 5-shot across task subtypes. For example, pro and anti versions of type 1 and type 2 Winobias [Zhao et al., 2018] tasks from PromptSource, and all 57 subtasks of MMLU [Hendrycks et al., 2021a], would be aggregated to get

Task	Category	Benchmark	Generalization Level	Metric
Winobias	Stereotype Detection	PromptSource	Fully Held-Out	Rouge-L F1
Winobias	Stereotype Detection	SuperNatInst	Fully Held-Out	Rouge-L F1
Bard Analogical Reasoning	Word Analogy	SuperNatInst	Fully Held-Out	Rouge-L F1
Cause-Effect	Cause Effect Classification	SuperNatInst	Fully Held-Out	Rouge-L F1
COPA	Cause Effect Classification	PromptSource	Fully Held-Out	Accuracy
COPA	Cause Effect Classification	SuperNatInst	Fully Held-Out	Rouge-L F1
COPA	Cause Effect Classification	FLAN	Fully Held-Out	Accuracy
COPA CommonSense	Cause Effect Classification	SuperNatInst	Fully Held-Out	Rouge-L F1
Glucose	Cause Effect Classification	SuperNatInst	Fully Held-Out	Rouge-L F1
Jfleg	Grammar Error Correction	SuperNatInst	Fully Held-Out	Rouge-L F1
MMLU	MMLU	MMLU	Partially Supervised	Accuracy
Civil Comments	Toxic Language Detection	SuperNatInst	Partially Supervised	Rouge-L F1
Jigsaw	Toxic Language Detection	PromptSource	Partially Supervised	Rouge-L F1
Newsroom	Summarization	FLAN	Partially Supervised	Rouge-L F1
Race	Question Answering	PromptSource	Partially Supervised	Accuracy
Race	Question Answering	SuperNatInst	Partially Supervised	Rouge-L F1
StrategyQA	Reasoning	Reasoning	Partially Supervised	Rouge-L F1
GSM8K	Reasoning	Reasoning	Partially Supervised	Rouge-L F1
SQuAD v1	Question Answering	FLAN	Fully Supervised	Rouge-L F1
SQuAD v1	Question Answering	PromptSource	Fully Supervised	Rouge-L F1
Blended Skill Talk	Dialogue Generation	PromptSource	Fully Supervised	Rouge-L F1
CNNDM	Summarization	PromptSource	Fully Supervised	Rouge-L F1

Table 2.3: Full details of validation tasks used in our experimental studies. Some of these tasks contain sub-tasks (e.g., MMLU) which we did not list in this table.

per task performance. If the same task exists across multiple benchmarks, we then average performance across benchmarks as well. We then compute 0-shot and 5-shot averages of all tasks within a category (or benchmark depending on the experiment), and finally, compute a combined average of all 0 and 5-shot scores of each category (or benchmark), which we use for model selection.

We tune each model for 4000 steps and evaluate on our validation split on both 0-shot and 5-shot settings, using 250 examples from each task for compute-efficiency. Our validation splits for each task include a mix of multiple prompts for FLAN and PromptSource. All but four validation tasks are generation-style tasks (where we report Rouge-L F1). We compute accuracy based on scoring for the remaining tasks and aggregate them together with Rouge-L for presentation purposes. Table 2.3 shows full details about the tasks in our validation split.

2.3.1 Effects of varying task mixing-rate maximum

Prior work [Raffel et al., 2020; Wei et al., 2022b] typically uses example-proportional sampling and builds batches by sampling from datasets proportional to their sizes, while enforcing a maximum size parameter (EPS) to prevent large datasets from overwhelming the batch. To understand how this maximum mixing rate (EPS) affects performance across the different generalization levels, we perform experiments with EPS $\in \{128, 256, 512, 1024, 2048, 4096, 8192, 16384, 10^6\}$ and report results in Table 2.4. An EPS of 512 causes 97% datasets to hit their maximum, while an EPS of 8192 causes 16% datasets to hit their maximum. We also experiment without using EPS i.e. EPS=100K.

	Fully Held Out				Partially Supervised					Fully Supervised		
	Cause Effect	Gram. Corr.	Stereo. Det.	Word Ana.	Reas.	MMLU	QA	Summ.	Toxic Det.	Dial ogue.	QA	Summ.
2^7	61.4/62.0	86.2/87.5	59.1/82.5	12.1/59.1	2.9/22.4	42.5/35.6	67.5/59.7	21.0	61.7/66.3	16.8/17.5	86.9/83.3	30.7
2^8	59.3/60.7	86.5/87.8	60.2/83.4	13.0/57.1	2.6/19.1	41.5/36.0	64.8/59.9	20.5	61.7/69.5	16.4/16.8	86.2/83.7	31.0
2^9	59.6/61.3	86.4/87.9	55.2/82.8	12.9/58.5	2.6/24.7	40.2/38.1	65.3/57.4	20.2	59.8/66.2	17.1/16.6	85.7/82.6	31.2
2^{10}	64.5/60.3	86.0/87.6	47.9/82.3	14.1/56.8	2.7/23.6	39.0/35.9	66.9/61.6	20.5	60.8/66.4	17.7/16.0	86.1/85.2	31.0
2^{11}	64.4/62.7	85.9/87.7	50.4/82.2	11.7/54.5	2.7/22.0	40.1/35.7	67.4/58.6	19.9	60.1/65.6	17.2/16.8	87.3/84.6	31.4
2^{12}	63.5/62.5	86.1/87.5	58.9/82.3	17.2/57.8	2.6/20.4	41.5/37.0	69.3/59.0	18.1	60.0/70.0	16.1/15.8	87.6/83.5	31.3
2^{13}	63.3/61.2	85.6/87.9	48.2/81.3	13.2/56.8	2.6/25.6	38.3/35.9	69.4/57.7	19.6	59.4/68.2	16.4/15.6	86.2/84.5	32.3
2^{14}	60.2/61.3	86.0/88.0	57.3/82.5	15.1/52.6	2.6/20.3	41.8/36.1	70.5/61.1	19.8	58.6/64.0	16.9/14.7	86.1/84.4	32.0
10^6	59.2/62.2	86.4/86.9	57.3/80.8	8.8/53.7	2.6/22.0	39.2/34.2	67.6/59.5	19.8	58.2/68.1	15.2/15.8	84.6/81.6	31.7

Table 2.4: Performance variation across different task categories with different maximum mixing rates (EPS), for each generalization level on OPT-IML 30B, after 4000 steps. Results are in the format of 0-shot/5-shot. We use only 0-shot performance for summarization tasks. Most tasks are generation tasks, for which we report Rouge-L. We report accuracy for MMLU. Some tasks in the Cause Effect Cluster also use accuracy, which is averaged with Rouge-L for presentation purposes. We select models based on their average performance aggregated per category, benchmark and shot.

Overall, we find that while EPS is important to instruction-tuning i.e. on average all models that use EPS outperform the model without it, after a certain threshold i.e. less than 4096 in our case, there is minimal variation in performance across all generalization levels. While based on the highest average performance, we choose 4096 (also corresponds to 50% of the dataset lengths being capped) for our other experiments and the final OPT-IML models, we find that all values below 4096 also perform quite well, with EPS=128 closely matching 4096. Also note that changing EPS implicitly changes the proportion of fine-tuning data from each benchmark, which we control for explicitly in the next Section.

2.3.2 Effects of varying benchmark proportions

Using multiple benchmarks for training, together with only example-proportional sampling, results in benchmarks with more tasks overwhelming the batch composition. For example, in our benchmark, 71% of training examples would come from SuperNatInst, with 18% from PromptSource, and only 5% from FLAN. Since each benchmark is associated with a specific task format, this can bias the resulting model towards certain input-output formats. We vary the proportions of different benchmarks to evaluate their effect on downstream task performance on our three generalization levels and present results in Table 2.5. For this experiment, we compare models based on their aggregate performance on each benchmark instead of task category, since we would like to choose the parameters that perform well on a maximum number of benchmarks.

	Fully Held-Out			Partially Supervised					Fully Supervised	
Benchmark Props. Crossfit/Exmix/Flan /NIV2/PS/T5/U-SKG	FLAN	NIV2	PromptS	Reas.	FLAN	MMLU	NIV2	PromptS	FLAN	PromptS
2/1/ 5/71/18/1/2	79.2/74.4	52.4/61.8	75.2/79.7	2.7/23.4	17.8	37.3/35.3	69.3/61.4	54.3/62.0	85.8/82.9	43.1/49.1
2/1/35/25/34/1/2	86.8/80.8	53.0/62.5	72.0/83.7	2.6/20.3	17.7	34.5/30.8	62.2/53.5	57.6/66.2	85.9/81.7	44.3/48.3
3/3/35/25/25/7/2	81.2/83.2	52.5/61.1	79.7/83.5	2.7/19.8	20.0	36.7/29.8	60.9/54.1	57.1/56.8	86.8/84.1	43.4/48.3
2/1/27/40/27/1/2	86.8/81.2	52.4/63.2	77.9/83.3	2.6/21.3	20.2	36.3/30.3	67.3/60.4	57.8/61.7	86.4/81.6	43.2/48.8
3/3/25/25/35/7/2	91.2/80.4	51.1/62.2	75.6/83.4	2.6/18.4	21.4	37.5/33.7	59.7/51.5	57.4/66.9	83.6/83.7	44.3/48.9
4/2/35/25/30/2/2	88.0/76.8	51.5/61.3	75.1/82.7	3.0/16.8	20.0	37.1/30.7	65.6/58.0	60.4/61.5	85.4/81.5	43.2/49.9
4/2/20/25/45/2/2	88.8/83.6	54.5/62.2	73.5/85.0	2.5/13.1	19.8	38.2/33.2	63.0/57.5	56.1/61.8	86.1/84.2	43.0/48.7
2/1/35/25/30/5/2	86.0/83.2	51.1/61.6	74.0/82.8	2.6/17.1	20.8	36.9/31.9	63.5/62.4	53.1/63.7	86.2/81.6	43.5/49.7
7/1/35/25/28/2/2	85.6/81.2	51.0/61.6	78.0/82.1	2.6/19.9	20.0	36.3/31.9	65.1/60.6	59.6/63.1	85.0/84.0	43.2/49.3
0/0/35/30/35/0/0	86.0/79.2	52.3/62.6	71.8/84.2	2.6/15.3	19.3	36.6/28.6	60.8/54.8	56.9/62.3	85.2/80.2	43.6/47.8

Table 2.5: Per-benchmark performance variation at each generalization level with varying benchmark proportions; The first row represents the original proportions in the OPT-IML benchmark. Results are in the format of 0-shot/5-shot. We use only 0-shot performance for Summarization tasks. Most tasks are generation tasks, for which we report Rouge-L. We report accuracy for MMLU. Four tasks in the Cause Effect Cluster also use accuracy, which is averaged with Rouge-L for presentation purposes. We select models based on their average performance aggregated per benchmark and shot.

First, we look at performance improvements within the same benchmark where the proportions were changed. As we increase the proportion of FLAN from 5% to 25%, its performance improves significantly on both the fully-held out and the partially held-out generalization levels, with no notable improvement on the fully-supervised tasks. SuperNatInstshows a similar trend on partially-supervised tasks, but surprisingly, not so much on fully held-out tasks. It is possible that the very specific input-output format of SuperNatInstmakes it such that changing proportions of unrelated clusters provides no benefit to its fully held-out clusters.

PromptSource is relatively unchanged on fully supervised clusters and partially supervised clusters, possibly owing to reaching performance saturation with even an 18% proportion. However, it benefits with more proportion on the fully-held out clusters.

Secondly, we also observe benchmarks complementing each other. For example, the highest accuracy on fully held-out FLAN i.e. 88.8/83.6%, is achieved, not with the highest proportion of FLAN, but with improving the proportions of PromptSource and Crossfit. Similarly, the highest generation performance on fully-held out PromptSource of 79.7/83.5% is achieved with 25% PS, and not with 45% PS proportions. We also observe certain tradeoffs, for example, the best proportions for FLAN and PromptSource result in a sharp drop in performance on reasoning datasets, and vice versa. Finally, setting Crossfit, Exmix, T5 and Unified-SKG proportions to 0 results in the worst model, demonstrating the benefits of using a diverse set of benchmarks for instruction-tuning. Based on average performance across benchmarks, "2/1/27/40/27/1/2", "7/1/35/25/28/2/2" and "4/2/20/25/45/2/2" performed the best and we choose the last one as the proportion for our final OPT-IML models. Despite our choice, instruction-tuned models with different end-goals (for example, producing reasoning chains) would benefit from choosing differently. We also explore methods to improve performance on reasoning datasets in Section 2.5.1.

2.4 Effects of Scaling up the Fine-tuning Task Set

We first conduct the experiment to verify that scaling the number of training tasks or clusters[4] improves the overall performance of the model on the fully held-out generalization setting, as implied by previous work [Wei et al., 2022b; Wang et al., 2022b]. Comparing to previous work, our study broaden the scope to more generalization settings including *fully held-out*, *partially supervised*, and *fully supervised* tasks.

Scaling number of tasks. We randomly sample 16, 64, 256, and 1024 sets of tasks such that smaller sets are subset of bigger sets, and fully supervised tasks are always selected. Figure 2.2 presents these task scaling studies on the three generalization levels, aggregated at the cluster-level for both 0 and 5-shot performance. We observe that both fully held-out and partially supervised tasks get the most improvements with the increase in the number of training tasks. In the fully held-out setting, *Cause Effect Classification* and *Word*

[4]We use task cluster/category interchangeably in this report.

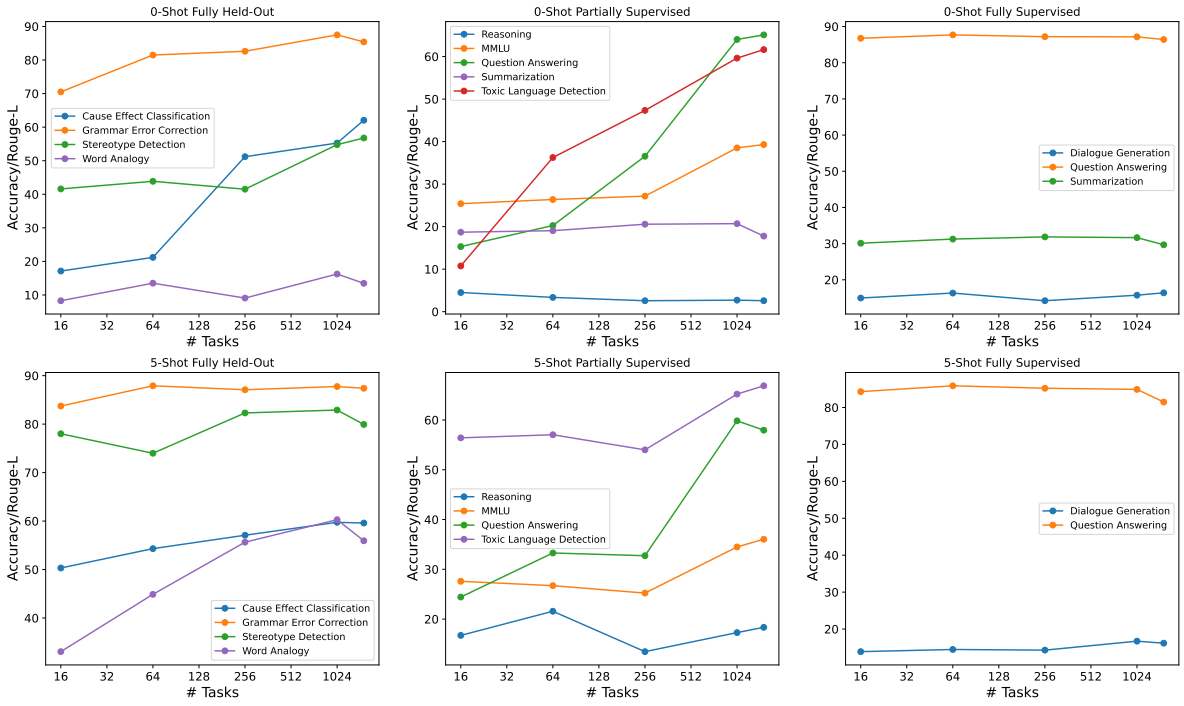

Figure 2.2: Effect of scaling the number of training tasks on each generalization level for OPT-IML 30B under both 0-shot and 5-shot settings, aggregated by task category.

Analogy clusters see the biggest improvements in zero-shot and few-shot, respectively. On the partially supervised, *Question Answering* and *Toxic Language Detection* clusters see the biggest improvements on both zero-shot and few-shot. Both *Summarization* and *Reasoning* tasks within the partially supervised tier, as well as *Word Analogy* within the fully held-out tier (0-shot setting) do not exhibit a significant performance boost as the number of tasks increases. For *Summarization* and *Word Analogy*, the stagnant performance could possibly be attributed to the nature of the tasks in our evaluation set. These tasks are constructed to assess discrete capabilities (e.g., "write a brief summary for the following article"). Consequently, fine-tuning on an increased number of tasks or tasks from the same category fails to yield additional improvement. Interestingly, the performance of fully supervised tasks remains unchanged even when more relevant tasks are seen, as we increase the amount of training tasks. Overall, we found that OPT-IML perform the best on supervisedly trained tasks, approaching the performance of previous state-of-the-art supervised models.[5] Tasks trained in a partially supervised manner comprise the second best tier. For fully held-out tasks, despite significantly outperforming the baseline models, OPT-IML's performance tends to fall considerably behind

[5]For example, it achieves an F1-score of more than 85.0 on the SQuAD v2 dataset [Rajpurkar et al., 2018].

state-of-the-art supervised models. We did not encounter a task category where the performance declined as we increase the number of fine-tuning tasks, indicating a generally positive impact of incorporating more fine-tuning tasks.

Scaling number of task categories. We also benchmarked the impact of scaling up the number tasks in a more principled way. To this end, we order the task clusters based on the decreasing order of the number of tasks present in each cluster and select the first 4, 16, 64, and 93 (all) clusters. Additionally, we make sure that *Question Answering*, *Summarization*, and *Dialogue Generation* clusters are always represented since our fully supervised validation tasks belong to these three clusters. Figure 2.3 presents the corresponding results on all three generalization levels for both zero-shot and few-shot settings. As we increase the fine-tuning clusters, the performance on fully supervised tasks either stay the same or slightly drop in the few-shot setting. On the fully held-out and partially supervised levels, the results on the zero-shot settings improve an increase in the number of clusters. However, the results for the few-shot settings are somewhat inconsistent, but they generally tend to decline with cluster scaling. Note that the first 4 clusters already cover 673 tasks (clusters belonging to the fully supervised setting have a lot of tasks). Hence, the model starts with strong performance, which might lead to the mixed results that we observe. Based on these experiments, we use all tasks and clusters to train our final OPT-IML models.

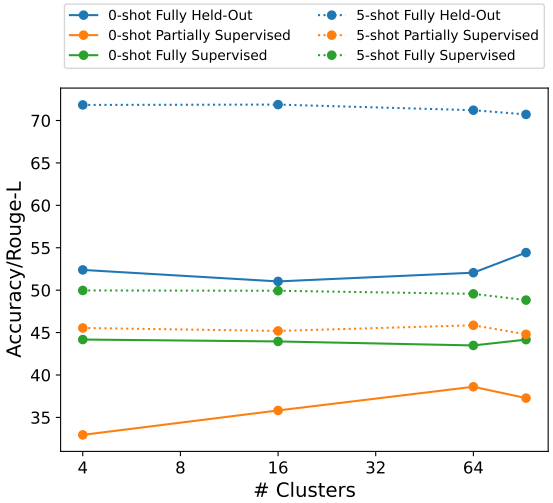

Figure 2.3: Scaling # training categories.

2.5 Impact of Special Datasets

2.5.1 Reasoning Datasets

Recent work [Wei et al., 2022c; Kojima et al., 2022] has illustrated improvements in the performance of LLMs on reasoning tasks, when prompted to generate a reasoning chain in natural language before generating the answer. Based on these findings, we attempt to explicitly fine-tune LLMs to perform reasoning

Dataset	Example (Input Prompt and *Output*)
Pre-training	*You could make it a full group party with the kids and wives. Don't make it just about books. So have A movie night My parents made a movie group they go out to dinner then see a movie then dicuss it. You could play card games. Watch some comedy. Ask the members. Do a music night when one of you has to bring a selection of their fav music.*
Reasoning	Answer the following question by reasoning step by step. How do most people feel about a person they love? popularity, know all, own house, care about, flu Output: *we care about people we love. The answer is care about*
Dialogue	*I love cats and have five of them.* *Cats are nice. How old are you?* *Old enough to work in the construction field. You?* *I am 68, been retired for a few years now.* *Great. What did you work and retire from?* *I was a tailor.*

Table 2.6: Examples from the pre-training, reasoning, and dialogue datasets. For pre-training and dialogue data, the source is empty and the entire text sequence is considered as the target.

by compiling a set of 14 reasoning datasets[6], where the output includes a rationale before the answer and by including these datasets during instruction-tuning. This set includes the 9 datasets used by [Chung et al., 2022b] in their CoT category as well as some additional datasets. Each dataset has a single prompt that uses an instruction, that explicitly asks the model to generate a reasoning chain [Kojima et al., 2022], followed by examples in the few-shot setting that illustrate how the reasoning chain should be produced before the answer. We show an example with such a prompt in Table 2.6. Using benchmark proportions of "2/1/27/40/27/1/2" as a baseline (see Section 2.3.2), we experiment with adding 1%, 2%, and 4% proportions of reasoning data (by reducing the proportion of the highest proportion benchmark i.e. SuperNatInst), and present results for the 5-shot setting in Figure 2.4 by generalization level and task category.

We see a substantial performance improvement on the 2/14 held-out validation reasoning tasks (Rouge-L from 12.2% to 31.6%) when we instruction-tune with reasoning datasets, but alongside, we also see improvements on other held-out task categories such as Cause-Effect, Stereotype Detection, Toxicity Detection, and Word Analogy. Furthermore, adding 1% reasoning data results in the largest gains overall, beyond which, the gains start to reduce on MMLU, Cause-Effect Accuracy, Toxicity, and Dialogue (averaged over

[6]GSM8K [Cobbe et al., 2021b], StrategyQA [Geva et al., 2021a], AQUA-RAT [Ling et al., 2017a], CoQA [Reddy et al., 2019], CoS-E [Rajani et al., 2019], CREAK [Onoe et al., 2021], ECQA [Aggarwal et al., 2021a], e-SNLI [Camburu et al., 2018], MATH [Hendrycks et al., 2021c], ProofWriter [Tafjord et al., 2020], QASC [Khot et al., 2020], QED [Lamm et al., 2021], SenseMaking [Wang et al., 2019], WinoWhy [Zhang et al., 2020].

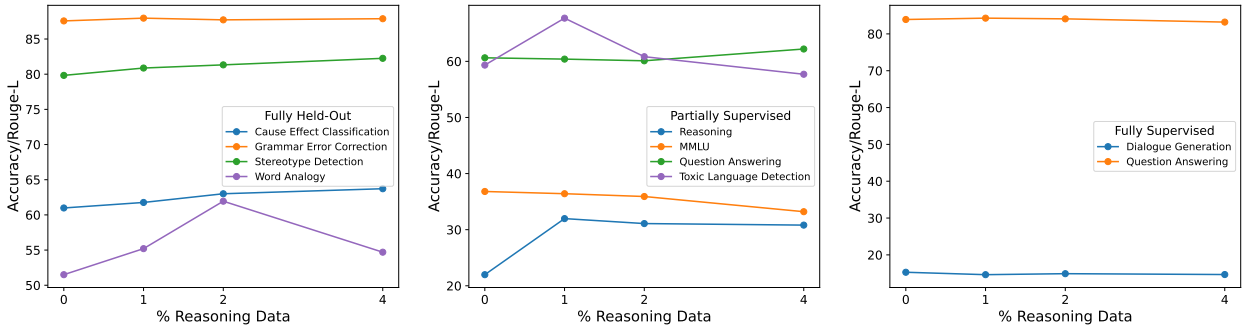

Figure 2.4: Effect of fine-tuning using reasoning datasets on each generalization level for OPT-IML 30B in a 5-shot setting, aggregated by task category. We experiment with adding 1%, 2% and 4% reasoning datasets by proportion. Note that the baseline for this experiment is based on a different proportion than other experiments.

0 and 5-shot). On the other hand, the Summarization cluster (only 0-shot) continues to benefit from higher proportions of reasoning data. Based on average performance across categories and generalization levels, we use 1% reasoning data for our final OPT-IML models.

2.5.2 Dialogue Datasets

We experiment with adding dialogues as auxiliary fine-tuning data to test if it can improve the LM's ability to respond to directional input and understand referential expressions. Another goal is to evaluate if this approach can induce chat-bot behaviors [Shuster et al., 2022] and make the resulting models more conversational. Using a subset of dialogue datasets[7] used for training BlenderBot 3 [Shuster et al., 2022], we process the dialogues into sequences of turns separated by a single newline token (see Table 2.6 for an example). The data consists of 320,543 unique dialogues and we fine-tune the model to predict the entire dialogue sequence. We set the proportion of the included dialogue data to be 0.5% and present 0 and 5-shot results by task category and generalization level on our validation split in Table 2.7.

We observe that adding even just 0.5% of the aforementioned dialogue data lowers 0-shot performance while 5-shot performance remains unchanged. Specifically, 0-shot performance suffers mainly on stereotype detection and word analogy. On examining model predictions on these categories, we found that they are primarily generation tasks whose references are either a single word or a short piece of text with a specific

[7]Wizard of Internet [Komeili et al., 2022], Wizard of Wikipedia [Dinan et al., 2019b], Blended Skill Talk [Smith et al., 2020], ConvAI2 [Dinan et al., 2019a], Multi-Session Chat [Xu et al., 2022] and Light+ Wild [Urbanek et al., 2019; Shuster et al., 2021]. These are a subset of those used by Shuster et al. [2022].

EPS	Fully Held Out				Partially Supervised				Fully Supervised				Average
	Cause Effect	Gram. Corr.	Stereo. Det.	Word Ana.	Reas.	MMLU	QA	Summ.	Toxic Det.	Dial ogue.	QA	Summ.	
Baseline	63.5/62.5	86.1/87.5	58.9/82.3	17.2/57.8	2.6/20.4	41.5/37.0	69.3/58.9	18.1	60.0/70.0	16.1/15.8	87.6/83.5	31.3	**46.0/57.6**
+ 0.5% BB3	61.7/62.2	86.1/87.4	51.9/83.4	10.4/57.5	2.6/22.2	40.2/35.4	68.9/62.5	20.6	61.9/65.4	16.1/15.2	86.4/83.7	31.1	44.8/57.5

Table 2.7: Effect of fine-tuning with 0.5% dialogue data on each generalization level for OPT-IML 30B after 4000 steps, aggregated by task category. Results are presented in the format 0-shot/5-shot. Most categories use Rouge-L F1, MMLU uses accuracy. Some Cause-Effect tasks use accuracy, which is averaged with Rouge-L F1 for presentation purposes.

format (for example, a pair of phrases from the original input that refer to each other). Training with BB3 data weakened the model's ability to conform to the required format.[8] It also significantly lowered the 5-shot performance of toxicity detection. An error analysis revealed a similar problem i.e. the model tends to perform worse on tasks that require generating a special set of decision words rather than simply generating "yes" or "no". Owing to severe model degeneration on these tasks, we do not add dialogue data while tuning OPT-IML.

2.6 Effects of Meta-Training for In-Context Learning

Recent work has shown that fine-tuning language models with demonstration examples in the instructions improves their ability to learn from the examples in context [Min et al., 2022a; Wang et al., 2022b; Chung et al., 2022b]. Both [Min et al., 2022a] and [Wang et al., 2022b] experimented with the setup where a constant number of k demonstration examples are added to each training example. The models are evaluated with the same number of k demonstration examples during inference. [Chung et al., 2022b] used a mixture of data with and without exemplars. However, the proportion of each type of data used and how many exemplars were included are not clear.

We attempt to train models that are better in-context few-shot learners, and also robust to the number of demonstration examples used during inference time.[9] We experiment with a simple way of creating training

[8] On one hand this behavior demonstrates a weakened instruction-following ability for the underlying model. On the other hand, it exposes a caveat in measuring model performance on tasks with instructions – model performance on a specific task category is often the result of multiple factors and underperforming on a particular task category may not offer a useful atomic diagnosis. As in our case, we found the model to perform worse on stereotype detection tasks because it cannot parse the required output format, not because it is a more biased model.

[9] In preliminary experiments, we found models trained with k exemplars tend to perform worse when a different number of exemplars is used during inference time.

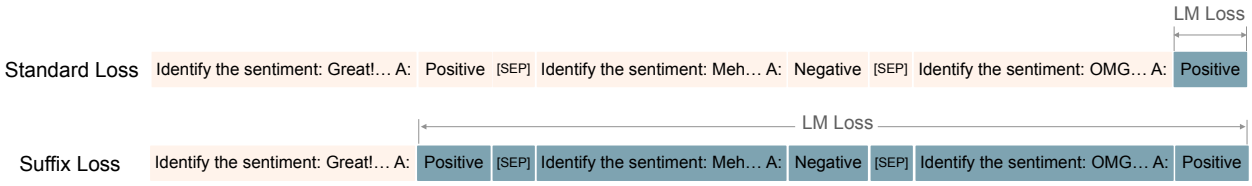

Figure 2.5: We experiment with two types of training losses for MetaICL: the generation loss over the label of the target example as proposed by [Min et al., 2022a], and the generation loss over the label of the first demonstration example and the complete sequences of the following examples.

examples that include varying numbers of demonstration examples. For each example e, we sample k from a distribution $\mathcal{D}$ with cap K, and randomly select k other examples $E_d = \{e_1, \ldots, e_k\}, e_i \neq e$, from the train set, if $k > 0$. We add E_d as the demonstration examples in e's prompt, where the examples are separated by a special token `[SEP]`. For benchmarks with task-level instructions such as Super-NaturalInstructions, we place the demonstration examples before e and after the instruction field; for benchmarks with instance-level instructions such as FLAN and PromptSource, we place the demonstration examples before e.

Because the demonstration examples significantly increase the prompt lengths, including too many few-shot training examples often leads to worse performance and reduced learning stability, owing to sparsity in the loss and lower batch diversity. As a result, we choose $\mathcal{D}$ to be the Zipf distribution[10], which can be heavily tilted towards $k = 0$. We train MetaICL models with different $\mathcal{D}$'s by adjusting the shape parameter a of the Zipf distribution. When $a = 4$, 92.5% of the examples are zero-shot examples; and when $a = 2$, 67.1% of the examples are zero-shot examples. We set $K = 5$ and use three consecutive newline tokens as `[SEP]` following [Min et al., 2022a].

MetaICL with suffix loss. To further address the loss sparsity problem, we also experiment with a variation of the original MetaICL loss, illustrated in Figure 2.5. Given an example with instructions and exemplars, rather than training the model to produce the target label, we train the model to produce the target label of the first exemplar followed by the complete sequences of the remaining exemplars. This effectively turns the demonstration examples into training examples as well, and mitigates the loss sparsity problem given it is now spread over more tokens.

[10] `https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.zipf.html`

Performance degradation on generation tasks. We present validation set results for instruction-tuning with different settings for MetaICL, aggregated by generalization level and task category under both 0 and 5-shot settings, in Table 2.8. We observe that adding MetaICL training leads to worse performance in both 0-shot and 5-shot setups in most cases, while MetaICL with the suffix loss outperforms regular MetaICL, especially in the 0-shot setup. Further examination of per-category performance reveals that while MetaICL models show reasonable improvements in multiple 5-shot evaluations, the 5-shot performances on Stereotype Detection and Word Analogy degrade significantly. An error analysis reveals a similar problem as in §2.5.2 – the MetaICL models tend to lose the ability to strictly follow the output pattern in the presence of in-context exemplars. In addition, the standard MetaICL loss significantly hurts reasoning tasks. The resulting models tend to generate short answers despite the presence of reasoning chains in the in-context learning examples. Further investigation reveals that the model could be over-fitting to the demonstration separators and modifying them at inference time can significantly mitigate these problems (Table 2.9). Fine-tuning with random demonstration separators may effectively mitigate these issues and we leave it to future work to investigate this approach. Interestingly, MetaICL degrades performance only for generation tasks, but is overall beneficial for scoring based classification tasks such as MMLU. However, owing to severe output degeneration in the regular setting, we decide to not use MetaICL to train our OPT-IML models.

EPS	Fully Held Out				Partially Supervised				Fully Supervised				Average
	Cause Effect	Gram. Corr.	Stereo. Det.	Word Ana.	Reas.	MMLU	QA	Summ.	Toxic Det.	Dial ogue.	QA	Summ.	
Baseline	62.1/59.6	85.4/87.4	56.8/79.9	13.5/55.9	2.6/18.3	39.3/36.0	65.1/58.0	17.8	61.6/66.9	16.4/16.2	86.4/81.5	29.7	44.7/**56.0**
Zipf a=4	60.5/61.4	84.7/87.5	53.0/67.6	13.8/36.5	2.9/3.3	37.9/35.9	63.6/59.7	18.8	59.5/62.2	15.5/15.3	86.1/86.3	30.2	43.9/51.6
Zipf a=4 sf.	59.8/62.0	85.1/87.2	52.9/67.6	12.2/42.9	2.7/20.7	41.0/38.7	64.3/61.6	18.4	66.3/66.2	15.9/16.2	85.9/85.2	29.5	44.5/54.8
Zipf a=2	61.6/62.0	84.2/87.0	48.0/69.1	11.0/41.2	2.6/5.2	37.9/36.4	63.7/64.9	20.2	65.1/72.8	16.1/14.5	85.6/84.8	29.8	43.8/53.8
Zipf a=2 sf.	56.1/64.3	87.6/88.1	60.8/65.9	14.5/35.9	2.6/16.9	39.7/38.0	63.4/62.1	19.1	65.2/75.3	16.2/16.9	85.4/86.2	31.5	**45.2**/55.0

Table 2.8: Effects of MetaICL fine-tuning on each generalization level for OPT-IML 30B after 2000 steps, aggregated by task category. Results are presented as 0-shot/5-shot. We underline categories where the MetaICL model outputs demonstrate severe degeneration compared to the baseline model.

2.7 OPT-IML Models

We fine-tune OPT-IML 30B and 175B models using the best settings for instruction tuning from our ablation experiments. Specifically, we set the best values for dataset size cut-off (EPS) and benchmark proportions

EPS	Fully Held Out				Partially Supervised					Fully Supervised			Avg.
	Cause Effect	Gram. Corr.	Stereo. Det.	Word Ana.	Reas.	MMLU	QA	Summ.	Toxic Det.	Dial ogue.	QA	Summ.	
Baseline	62.1/59.5	85.4/87.6	56.8/79.8	13.5/55.4	2.6/18.3	39.3/36.0	65.1/56.6	17.8/15.2	61.6/65.7	16.4/16.5	86.4/82.4	29.7/19.0	44.7/49.3†
Zipf a=4	60.5/60.6	84.7/88.1	54.1/81.2	13.8/55.8	2.9/9.7	38.4/37.3	64.4/62.8	18.8/19.5	59.5/65.8	15.5/15.3	86.1/85.4	30.2/29.5	44.1/50.9
Zipf a=4 sf.	59.8/61.5	85.1/87.4	52.9/79.4	12.2/52.8	2.7/24.6	41.0/38.7	64.3/59.6	18.4/20.3	66.3/67.3	15.9/15.9	85.9/85.0	29.5/26.9	44.5/51.6
Zipf a=2	61.6/61.9	84.2/87.7	48.0/80.1	11.0/55.2	2.6/15.1	37.9/36.4	63.7/61.9	20.2/21.5	65.1/75.3	16.1/15.3	85.6/85.0	29.8/28.1	43.8/**52.0**
Zipf a=2 sf.	56.1/63.5	87.6/88.2	60.8/75.0	14.5/44.7	2.6/20.3	39.7/38.0	63.4/60.5	19.1/20.7	65.2/76.0	16.2/16.2	85.4/86.3	31.5/28.8	**45.2**/51.5

Table 2.9: A repeat of the MetaICL experiments reported in §2.6 using "\n\n" as the example separator during inference. Under this setting, all MetaICL models outperform the baseline model. † The 5-shot baseline performance is not comparable with those in the other experiment tables since we also include the 5-shot performance on summarization tasks here.

based on the valid split performance, include all tasks in the training split, add 1% datasets with reasoning chains, and 5% data from the OPT pre-training corpus. We evaluate the final models on the OPT evaluation tasks as well as on four multi-task benchmarks from prior work [Wei et al., 2022b; Sanh et al., 2022; Wang et al., 2022b; Xie et al., 2022; Zhang et al., 2022] in both zero and 5-shot settings, directly comparing them to individual benchmark specific instruction-tuned models released by prior work.

Model	**StoryCloze**	**PIQA**	**ARC (e)**	**ARC (c)**	**OpenBookQA**	**Winograd**	**Winogrande**	
OPT 30B	80.3/84.1	77.5/78.8	63.9/72.7	43.1/45.2	57.2/60.1	83.5/83.3	69.7/71.7	
OPT-IML 30B	80.1/82.7	77.3/69.2	64.9/72.1	45.5/46.7	50.6/55.2	83.5/83.5	67.8/69.0	
OPT 175B	82.9/86.9	79.5/81.6	67.0/76.8	44.1/50.5	58.4/64.5	85.3/87.8	73.7/77.6	
OPT-IML 175B	83.3/86.4	79.8/80.5	70.8/77.2	50.9/53.2	58.2/65.0	85.7/87.5	73.0/74.4	
Model	**BoolQ**	**CB**	**COPA**	**RTE**	**WIC**	**WSC**	**MultiRC**	**Average**
OPT 30B	64.0/69.6	28.6/5.7	84.0/88.6	58.1/61.7	50.2/54.0	62.2/63.2	6.1/7.8	59.2/60.5
OPT-IML 30B	66.9/71.8	82.1/78.6	85.0/89.0	83.8/73.3	57.1/52.0	75.7/54.1	7.7/4.9	66.3/64.4
OPT 175B	60.1/76.8	46.4/70.0	87.0/91.4	60.3/71.0	56.6/54.3	51.4/75.1	7.5/14.0	61.4/69.9
OPT-IML 175B	71.4/81.7	69.6/53.6	88.0/89.0	84.8/83.8	56.1/56.1	73.0/75.7	10.3/20.4	**68.2/70.3**

Table 2.10: Accuracies of OPT-IML compared with OPT on the 14 standard NLP tasks from Zhang et al. [2022] under the fully held-out setup. The results are presented in the format of 0-shot/32-shot.

We examine these results in Table 2.10 and Figure 2.6. In particular, OPT-IML outperforms OPT on all benchmarks and is competitive with the individual benchmark specific instruction-tuned models on both zero- and few-shot performance. We conclude that supervised fine-tuning over a large number of tasks enhances the proficiency of LLMs as task solvers in various setting, especially in the zero-shot scenarios where only a natural language description of the task is provided. On the other hand, we find instruction tuning does not improve the in-context learning capabilities of LLMs, and it might even hurt this aspect. Across the

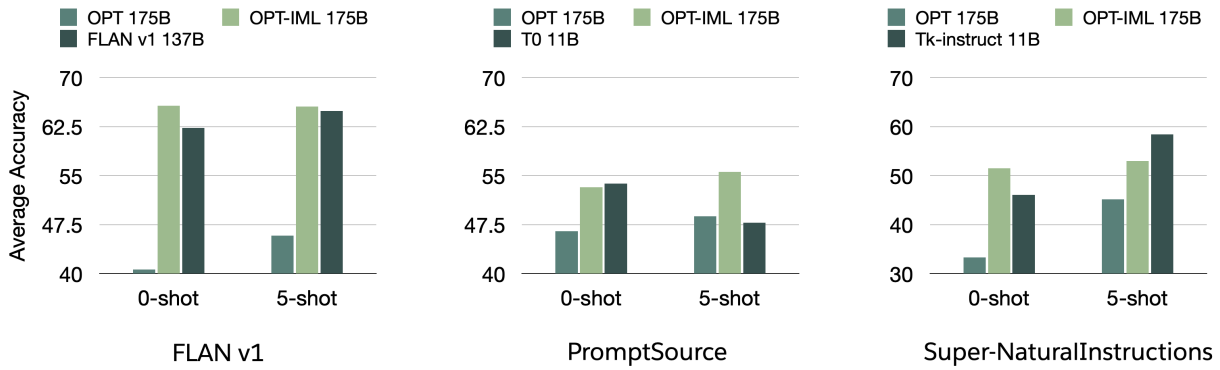

Figure 2.6: Accuracies of OPT-IML compared with OPT and other instructable LLMs fine-tuned specifically for each benchmark, under both 0-shot and 5-shot settings.

three instruction benchmarks shown in Figure 2.6, the improvement of 5-shot in-context learning over the 0-shot setting for OPT-IML is less pronounced than it is for OPT. Chung et al. [2022b] reported similar results for certain tasks. However, for the majority of tasks, their study did not offer a direct comparison between 0-shot and few-shot performances. It would be worthwhile to examine this phenomenon more carefully in future work, as it is perplexing for models designed to learn and follow natural instructions to potentially lose their in-context learning ability. We also conducted an ablation experiment, incorporating in-context learning examples in the fine-tuning prompts. However, in contrast to the findings of Chung et al. [2022b] and Min et al. [2022a], adding the in-context learning examples did not result in substantial improvement on our validation set and led to model degeneration for generative tasks. As a result, we did not use in-context learning examples to fine-tune the OPT-IML models.

2.8 Comparison with State-of-the-Art LLMs

To compare the performance of OPT-IML with other state-of-the-art instructable LLMs, we instruction-tune OPT 30B and 175B on our entire benchmark of 1,991 tasks, and compare the resulting models (dubbed OPT-IML-Max) to FLAN-T5 and FLAN-PaLM developed by Google [Chung et al., 2022b] and the Text-DaVinci APIs developed by OpenAI[11]. We evaluate the models on three demanding benchmarks that have not been utilized for fine-tuning any of the models under consideration: MMLU [Hendrycks et al., 2021a], Big Bench Hard (BBH) [Srivastava et al., 2022] and RAFT [Alex et al., 2021] using their standard test splits.

[11] `https://api.openai.com/v1/completions`

According to Table 2.11, OPT-IML significantly outperforms its base counterpart at all scales. However, while OPT-IML-Max-175B is competitive with FLAN-T5 11B on the RAFT benchmark, it trails behind both the suite of comparable FLAN models and the instruction-tuned GPT-3 models (*-davinci-*) on MMLU and BBH. We hypothesize that the most important cause of the performance difference is the quantity of the pre-training data of the respective underlying base LLMs. The T5 models were trained on 1T tokens [Raffel et al., 2020] , PaLM on 800B [Chowdhery et al., 2022], and OPT on 300B (180B if counting only unique text) [Zhang et al., 2022]. Subsequent research by Touvron et al. [2023a] demonstrated that by carefully sifting a pre-training dataset of approximately 2 trillion tokens, the resulting LLaMA-v2 65B model matched the performance of the PaLM-v1 540B model on MMLU, BBH and other representative benchmarks. Further instruction-tuning the LLaMA v2 models results in performance approaching the best instructable LLMs available.[12] There are also differences relating to the composition of the pre-training data and the respective modeling architectures. Le Scao et al. [2022] observed that encoder-decoder models can fine-tune more effectively than decoder only models at similar scales, which could partially explain why the FLAN-T5 model outperforms OPT-IML even with significantly fewer parameters.

	BBH	MMLU	RAFT
# shots	3	0/5	5
OPT 1.3B	27.9	23.5/25.9	49.1†
OPT 30B	28.4	24.2/26.1	59.1†
OPT 175B	30.2	27.3/34.2	63.2†
T5 11B	29.5	-/25.9	–
PaLM 62B	37.4	-/55.1	–
PaLM 540B	49.1	-/71.3	–
OpenAI davinci	33.6	-/32.3	64.5
OPT-IML-Max 1.3B	26.5	34.9/29.5	55.9†
OPT-IML-Max 30B	30.9	46.3/43.2	69.3†
OPT-IML-Max 175B	35.7	49.1/47.1	79.3†
T0pp 11B	13.0	46.7/33.7	56.8†
FLAN-T5 11B	45.3	53.7/54.9	79.5†
FLAN-PaLM 62B	47.5	-/59.6	–
FLAN-PaLM 540B	57.9	-/73.5	–
OpenAI text-davinci-002	48.6	-/64.5	72.1
OpenAI text-davinci-003	50.9	-/74.2	–
OpenAI code-davinci-002	52.8	-/77.4	–

Table 2.11: Test-set performance of OPT-IML-Max, trained on all tasks in our benchmark, on Big-Bench Hard, MMLU, and RAFT.

2.8.1 Discussion

The literature concerning the enhancement of instruction-following capabilities of Language Learning Models (LLMs) is expanding rapidly as the capability to follow natural instructions is fundamental to foster seam-

[12] https://tatsu-lab.github.io/alpaca_eval/

less human-machine interaction. Our study attests to the potential of supervised fine-tuning using extensive NLP benchmarks in developing robust, multi-task solvers proficient in instruction following. However, this work is not without limitations. Firstly, the OPT base models we employ are severely under-trained as per the Chinchilla Scaling Law [Hoffmann et al., 2022]. We anticipate that applying the same procedure to a well-trained base model (e.g. LLaMA v1 or v2) would yield significantly improved results. Secondly, our evaluation of the model is primarily based on accuracy-focused benchmarks, meaning the results often compound several factors. Future work could benefit from benchmarking different properties of instructable LLMs separately, such as robustness and controllability, to gain clearer insights into the progress made against language-controlled models. Lastly, we fine-tuned our model using conventional NLP datasets to minimize the cost associated with human labelling for data collection. However, training exclusively on conventional NLP datasets makes it challenging for the model to generalize to more complex practical use cases like trip planning or providing mental health consultations.[13]

Recent research indicates that high-quality, manually-authored instruction data can effectively enhance free-form generation by LLMs [Zhou et al., 2023]. Efforts like Self-Instruct [Wang et al., 2023b] and Unnatural Instructions [Honovich et al., 2023] have also shown the potential of model-generated data as a cost-effective substitute to crowdsourcing for dataset expansion and diversification. Nonetheless, community-generated alignment datasets tend to produce models that perform well in certain domains but poorly on the other (e.g. coding tasks). Therefore, determining how to collect instruction tuning data to improve model capabilities in a balanced manner remains an open question, which we leave out of the scope of this thesis.

2.9 Related Work

Instruction Tuning. Language models are trained to predict the next token in a sequence with self-supervised learning [Brown et al., 2020; Zhang et al., 2022; Chowdhery et al., 2022]. Prompt engineering and in-context learning has become a dominant approach to leverage these models to solve many NLP tasks. In order to align these models to follow natural instructions and avoid prompt engineering, recent works have proposed instruction fine-tuning [Ouyang et al., 2022b; Wei et al., 2022b; Chung et al., 2022b; Wang et al., 2022b]. Some of these works focus on fine-tuning the model on a wide range of tasks using human annotated

[13] `https://chat.openai.com/`

prompts and feedback [Ouyang et al., 2022b], whereas the others focusing on supervised fine-tuning using academic benchmarks and datasets augmented with manually or automatically generated instructions [Wang et al., 2022b; Wei et al., 2022b; Sanh et al., 2022; Zhong et al., 2021]. In our work, we focus on the second approach and consolidate a massive collection of publicly available datasets with instructions to finetune OPT. Concurrent to our work, Chung et al. [2022a] also proposes a similar instruction benchmark scaling approach to 1836 tasks from 4 benchmarks. While they focus on fine-tuning using the entire benchmark in order to push the limits of performance on several challenging held-out tasks that test the model's world knowledge and reasoning capabilities such as MMLU [Hendrycks et al., 2021a] and Big-Bench Hard (BBH) [Suzgun et al., 2023], we focus on characterizing the tradeoffs of various instruction-tuning decisions that can affect downstream performance.

Prompting and Meta-Training Zero- and few-shot learning (a.k.a. in-context learning) that leverages very few examples to solve any NLP task by effectively prompting the language models, is becoming a dominant paradigm in recent years [Brown et al., 2020]. Prompting involves modifying the input and output space of a given task that can effectively leverage the knowledge of the language model to solve it. Various approaches have proposed better prompting ways to improve generalization performance [Wei et al., 2022c; Lu et al., 2022]. Furthermore, recent developments have shown ways to improve in-context learning (ICL) by meta-tuning language models to better adapt for ICL [Min et al., 2022b,a]. In our work, we leverage both the variants of prompts available from different benchmarks, as well as meta-training with demonstrations from a large pool of tasks, to study the effective settings for instruction-based fine-tuning that induce robustness against different prompting language and setups.

Learning to Reason. Despite the progress of in-context learning, state-of-the-art LLMs still struggle with reasoning tasks such as commonsense reasoning [West et al., 2022], and math word problems [Hendrycks et al., 2021c] which require arithmetic reasoning, etc. To solve these challenging tasks, recent work used different prompting methods which include a rationale with the final answer in the form of a scratchpad for arithmetic and logical reasoning [Nye et al., 2021], provided chain-of-thought prompts in demonstrations [Wei et al., 2022c], or added trigger phrases such as *let's think step-by-step* to prompt models to generate explanations [Kojima et al., 2022]. In addition to changing prompts, Chung et al. [2022a] integrated

step-by-step explanations into the instruction tuning stage. Following [Chung et al., 2022a], we further expand the set of reasoning datasets to 14 datasets and study the effects of different proportions of reasoning data on different held-out task clusters.

Multi-task Learning. Instruction-based fine-tuning can be considered as a formulation of multi-task Learning (MTL). MTL is a popular paradigm that improves the generalization performances of a task when combined with related tasks by sharing common parameters or representations [Caruana, 1997; Kumar and Daume III, 2012]. MTL has been applied to many NLP scenarios in recent years primarily focusing on improving the performance on the training tasks or to new domains by leveraging the signal from related tasks [Collobert and Weston, 2008; McCann et al., 2018b; Raffel et al., 2020; Vu et al., 2020]. In contrast, instruction-based fine-tuning allows us to improve the generalization performance to new tasks that are never seen during training. This is achieved by unifying all the tasks into a common format [Kumar et al., 2016; Khashabi et al., 2020] via *instructions*, and training them together by sharing all the weights of the model across all tasks.

Continuous Learning. Existing work also address continuous adaptation of language models by revisiting the instructions [Yin et al., 2022] or examples [Scialom et al., 2022] of previously learned tasks when fine-tuning with a new task to prevent catastrophic forgetting. The results show that LMs can be adapted effectively to new tasks without losing sight of the previously learned tasks. Other work enable the LM to perform new tasks via arithmetic combination of learned task vectors [Ilharco et al., 2022] or soft prompts [Anonymous, 2023] patched to the base LM without changing its parameters. We focus on the (massively) multi-task adaptation setting by fine-tuning the LM with 2000 tasks at once. Continuously adapting the resulting model to new data, new tasks and new domain would be an interesting and important future direction.

2.10 Conclusions

Instruction-tuning of LLMs has emerged as an effective means to improve their zero and few-shot generalization abilities. We make three main contributions to instruction-tuning in this paper. First, we curate

a large scale benchmark for instruction-tuning comprising 2000 NLP tasks from 8 dataset collections, annotated into task categories. We strategically produce evaluation splits on this benchmark to evaluate three different types of model generalization abilities: 1) fully-supervised performance, 2) performance on unseen tasks from seen task categories, and 3) performance on tasks from completely held-out categories. Second, using our evaluation suite, we establish tradeoffs and best practices of many aspects of instruction-tuning, such as different sampling methods of fine-tuning tasks and categories, fine-tuning with task demonstrations, and fine-tuning with specialized datasets for reasoning and dialogue. Finally, using the best settings from our experiments, we train and release OPT-IML 30B and 175B instruction-tuned models based on OPT, that strongly outperform OPT on five evaluation benchmarks and are competitive with recent instruction-tuned models that are tuned on individual benchmarks.

Chapter 3

Instruction Meta-Learning with Nonparametric Memory

3.1 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 with the backbone language model, such architectures have exhibited superior performance on knowledge intensive tasks such as open-domain question answering [Lewis et al., 2020b; Izacard et al., 2022b] and live chat interactions [Liu, 2022].

Existing RALM architectures focus on two high-level challenges: (i) enhancing the LLM's capability to incorporate retrieved knowledge [Lewis et al., 2020b; Izacard et al., 2022b] and (ii) refining the retrieval component to return more relevant content [Shi et al., 2023b; Izacard et al., 2022b]. Retrieval capabilities have also been instroduced 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

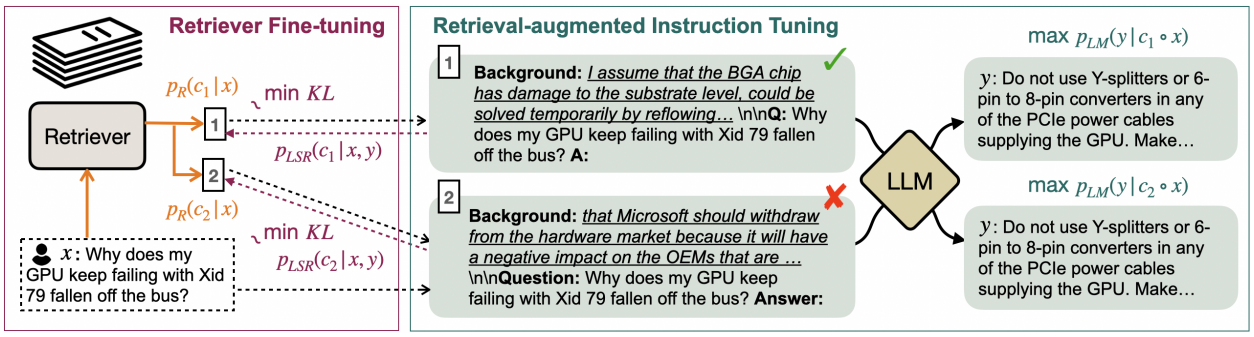

Figure 3.1: The RA-DIT approach separately fine-tunes the LLM and the retriever. For a given example, the LM-ft component updates the LLM to maximize the likelihood of the correct answer given the retrieval-augmented instructions (§3.2.3); the R-ft component updates the retriever to minimize the KL-Divergence between the retriever score distribution and the LLM preference (§3.2.4)

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.

In this chapter[1], 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 knowledge intensive scenarios. 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., 2023a]. 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 label-loss objective [Chung et al., 2022b; Iyer et al., 2022] and augment each fine-tuning prompt with

[1]The work in this chapter was conducted at Meta AI, in collaboration with Xilun Chen*, Mingda Chen*, Maria Lomeli, Rich James, Scott Wen-tau Yih† and others [Lin et al., 2023b]. (*co-first authors, †research leadership)

a retrieved "background" field prepended to the instructions (Figure 3.1). 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., 2021a] (+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.

3.2 Method

3.2.1 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.

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., 2020b; Izacard et al., 2022b; Shi et al., 2023b]. Given a corpus

Table 3.1: 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}

$\mathcal{C}$ 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:

$$s(q, c) = \mathbf{E}_q(q) \cdot \mathbf{E}_d(c). \tag{3.1}$$

We initialize the retriever using DRAGON+ [Lin et al., 2023a], 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 to the prompt[2], 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

$$p_{LM}(y|x, \mathcal{C}') = \sum_{c \in \mathcal{C}'} p_{LM}(y|c \circ x) \cdot p_R(c|x), \tag{3.2}$$

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]The complete set of our instruction-tuning templates are shown in Table 3.1.

Table 3.2: 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., 2017b]	✓		97,467
	ecqa‡	Explanations for CommonsenseQ [Aggarwal et al., 2021b]	✓		7,598
	gsm8k‡	Grade School Math 8K [Cobbe et al., 2021a]	✓		7,473
	math‡	MATH [Hendrycks et al., 2021d]	✓		7,500
	strategyqa‡	StrategyQA [Geva et al., 2021b]	✓		2,290

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

3.2.2 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 3.2, our *language model fine-tuning* datasets ($\mathcal{D}_L$) consists of 20 datasets across 5 distinct categories: dialogue, open-domain QA, reading comprehension[3], summarization and chain-of-thought 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

[3]These categories were selected due to their representativeness of practical knowledge-intensive language tasks. 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 answer only when the passage is relevant (otherwise the response is set to "I don't know"). 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 exploring this behavior to improve answer generation as a future work.

(Table 3.1). 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 3.3.4.

3.2.3 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: $\{(c_{ij} \circ x_i, y_i) | j = 1 \dots \tilde{k}\}$.[4]

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]:

$$\mathcal{L}(\mathcal{D}_L) = -\sum_i \sum_j \log p_{LM}(y_i | c_{ij} \circ x_i). \tag{3.3}$$

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 §3.5.1.

3.2.4 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-*

[4]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.

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:

$$p_{LSR}(c|x, y) = \frac{\exp\left(p_{LM}(y|c \circ x)/\tau\right)}{\sum_{c' \in \mathcal{C}} \exp\left(p_{LM}(y|c' \circ x)/\tau\right)} \approx \frac{\exp\left(p_{LM}(y|c \circ x)/\tau\right)}{\sum_{c' \in \mathcal{C}'} \exp\left(p_{LM}(y|c' \circ x)/\tau\right)}, \tag{3.4}$$

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. 3.2:

$$\mathcal{L}(\mathcal{D}_R) = \mathbb{E}_{(x,y)\in\mathcal{D}_R} KL\big(p_R(c|x) \parallel p_{LSR}(c|x, y)\big) \tag{3.5}$$

In practice, we only update the query encoder of the retriever, as fine-tuning both encoders hurts the performance (§3.5.1). While previous work [Shi et al., 2023b] relies solely on unlabeled texts (denoted as *corpus data*) for LSR training, we show that LSR can be generalized to incorporate the multi-task instruction data introduced in §3.2.2 (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 §3.5.1, combining both types of data yields the best results and outperforms using either source alone.

3.3 Experiment Setup

3.3.1 Retriever

We initialize the retriever in our framework with DRAGON+ [Lin et al., 2023a] and also use it to study various retriever configurations. 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,

Table 3.3: 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}	

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[5] released by Izacard et al. [2022b]. We obtain the retrieval queries used for our fine-tuning and evaluation tasks using manually[6] constructed templates (Table 3.1 and 3.3).

3.3.2 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

[5] https://github.com/facebookresearch/atlas

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

Task	Dataset name	Acronym	Metric	Score
Open-domain QA	MMLU [Hendrycks et al., 2021b]	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., 2019c]	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

Table 3.4: Our evaluation datasets. † indicates the development datasets we used to select fine-tuning hyperparameters.

addition, we also compare RA-DIT to ATLAS [Izacard et al., 2022b] in a 64-shot fine-tuning setting (§3.4).

3.3.3 Evaluation

We primarily conduct evaluation on knowledge-intensive tasks that are not included in our fine-tuning datasets, including MMLU [Hendrycks et al., 2021b], Natural Questions (NQ; Kwiatkowski et al., 2019), TriviaQA (TQA; Joshi et al., 2017), and a subset[7] 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. 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

[7]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., 2019c] and ELI5 [Fan et al., 2019].

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 Table 3.4.

3.3.4 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 $\eta = 7500$ based on the model performance on our development set.[8] We then randomly sample batches in accordance with this adjusted mixture probability.

64-shot Eval Task Fine-tuning Table 3.5 summarizes our hyperparameters for 64-shot fine-tuning on the 9 KILT eval tasks shown in Table 3.3 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 3.4.

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

Table 3.5: Hyperparameters for 64-shot fine-tuning on the eval tasks.

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 form a set of

[8] We did not thoroughly tune this parameter to avoid overfitting to the development sets.

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 3.2, including 10 open-domain question answering and dialog tasks, with a total of 286k training examples. As discussed in §3.5.1, 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[9]. 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 3.2.4 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. 3.4) 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., 2023a], 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. 3.2 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.[10] 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

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

[10] 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.

of the answer given "Answer:": $\frac{p(y|x)}{p(y|\text{“}Answer:\text{”})}$.

3.4 Main Results

Knowledge-Intensive Tasks We report the main results in Table 3.6. 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 and ELI5) and achieves the best performance on most datasets. This corroborates 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.[11]

We also compare with ATLAS, a state-of-the-art encoder-decoder based RALM that jointly pre-trains 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 3.6, 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 hence do not perform retrieval augmentation in this experiment. As shown in Table 3.7, 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. Maintaining the parametric knowledge in the LLM component is vital as a safety net when the retriever makes mistakes.

[11] 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.

Table 3.6: Main results: Performance on knowledge intensive tasks (test sets).

	MMLU	NQ	TQA	ELI5	HoPo	FEV	AIDA	zsRE	T-REx	WoW	Avg$^{\diamond}$	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**

$^{\diamond}$ 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.

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**

Table 3.7: Performance on commonsense reasoning tasks (dev sets) in the 0-shot setting without using retrieval augmentation.

3.5 Analysis

3.5.1 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[12] (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 3.8, 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. This observation aligns with our findings from Chapter 2. HotpotQA is an exception likely because it is from a task category

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

0 / 5-shot	HoPo	FEV	AIDA	zsRE	T-REx	WoW	Avg
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-3 chunks							
LLAMA 65B + DRAGON+	29.6 / 40.8	74.9 / 90.3	43.1 / 52.8	55.9 / 62.9	37.2 / 70.8	16.0 / 17.2	42.8 / **55.8**
IT 65B + DRAGON+	35.2 / 40.0	85.7 / 91.2	49.7 / 52.3	56.2 / 61.9	45.9 / 68.6	15.6 / 15.6	48.1 / 54.9
RA-IT 65B + DRAGON+	39.9 / 40.6	82.4 / 91.7	45.2 / 53.4	63.4 / 61.3	52.8 / 67.6	16.6 / 16.7	**50.1** / 55.2
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

Table 3.8: Ablation of language model fine-tuning strategies. All rows report dev set performance.

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.

Table 3.9: Ablation of retriever fine-tuning strategies. All rows use the LLAMA 65B model and report 5-shot performance on the dev sets.

covered in our instruction-tuning data. 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. In § 3.5.4, 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.

Retriever Fine-tuning In Table 3.9, we study different retriever fine-tuning strategies. As mentioned in §3.2.4, we explore two types of retriever fine-tuning data, the *multi-task instruction (MTI) 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 3.9 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.[13]

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 3.9 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.

3.5.2 Dual Instruction Tuning Ablation

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**

Table 3.10: The impact of LM and Retriever fine-tuning in RA-DIT. 5-shot dev set performance is reported.

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. [14] According to Table 3.10, 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

[13] 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.)

[14] 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.

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.

3.5.3 Retriever Settings

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

Table 3.11: Retriever settings: We report 5-shot dev set performance using LLAMA 65B and various retrievers in the REPLUG setting.

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 3.11 first compares DRAGON+ [Lin et al., 2023a] 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 bottom section in Table 3.11 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 (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

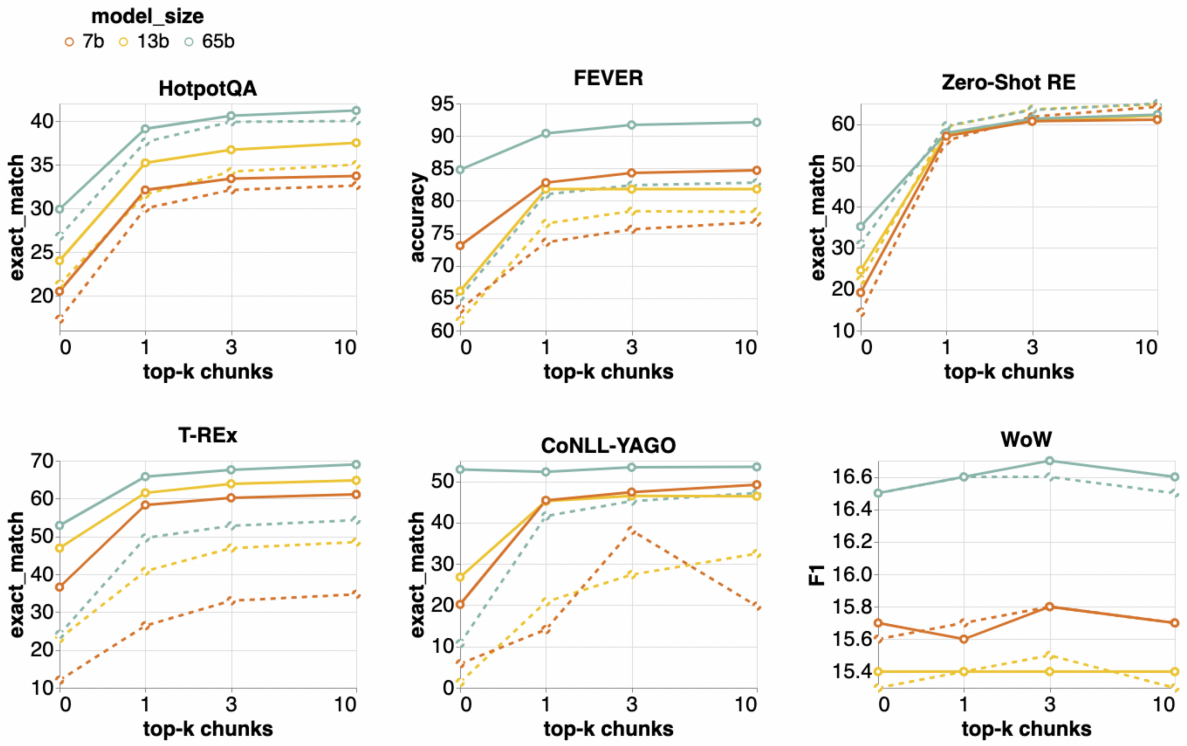

Figure 3.2: RA-IT model performance (combined with DRAGON+) across sizes 7B, 13B and 65B on our development tasks. 0-shot performance: dashed lines; 5-shot performance: solid lines.

the supplied retrieval corpus, and up-to-date information can be provided by updating the retrieval index at test time.

3.5.4 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 3.2. 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.

3.5.5 Qualitative Analysis

We analyze the performance of the two models on the development set of HotpotQA in the zero-shot setting since under this setting RA-IT 65B outperforms IT 65B by a large margin. Table 3.12 show two examples from the HotpotQA development set where RA-IT 65B makes a correct prediction while IT 65B makes a wrong prediction. First, we observed that the dense retriever struggles to return useful text chunks for the multi-hop questoins in the HotpotQA dataset and most of the returned text chunks contains no information that helps the prediction. In this case, the IT 65B model shows a stronger tendency to be misled by distractors within the retrieved text chunk, since it has not been trained with noisy passages during fine-tuning. It also tend to predict "I don't know" more frequently[15], while the RA-IT 65B can ignore the noisy passages retrieved and predict the correct answer based on its parametric knowledge [Mallen et al., 2023]. We also observe that in cases where both models generate wrong predictions because of 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 where both models ignore the noisy passages and rely on the parametric knowledge to make a prediction, RA-IT 65B outputs the correct answer with a higher probability (e.g. for the second text chunk in the first example).

3.6 Related Work

Retrieval-Augmented Language Models RALMs augment LMs with a non-parametric memory to facilitate external knowledge access and provide provenance [Guu et al., 2020; Lewis et al., 2020b; Borgeaud et al., 2022; Shi et al., 2023b]. Previous work have proposed different ways of fusing the LM and the non-parametric component. For example, RETRO [Borgeaud et al., 2022] and FiD [Izacard and Grave, 2021] leverage separate encoder modules to encode the retrieved content, which are integrated with the backbone LM via cross-attention. A more widely adopted approach directly augments the LM input with the retrieved

[15] As discussed in §3.2.2, 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.

content [Guu et al., 2020; Lewis et al., 2020b; Shi et al., 2023b]. This approach yields competitive results with a moderate inference cost increase, as the LM can effectively contextualize the retrieved content and the original prompt through multi-layer self-attention. RA-DIT is grounded in the in-context RA framework for its simplicity and practicality. Instead of performing extensive pre-training [Guu et al., 2020; Borgeaud et al., 2022; Izacard et al., 2022b], we propose a lightweight fine-tuning recipe that primarily utilizes downstream data, and demonstrate improved few-shot generalization of the fine-tuned RALM on knowledge-intensive language tasks.

Instruction Tuning Instruction tuning has been proposed to align pre-trained LLMs to follow natural language instructions and avoid extensive prompt engineering [Ouyang et al., 2022a; Wei et al., 2022a; Chung et al., 2022a; Wang et al., 2022a; 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. Concurrent work has also applied instruction tuning to other RALM architectures. Notably, Wang et al. [2023a] fine-tunes the backbone LM in the RETRO architecture while freezing the cross-attention module and the memory encoder. In comparison, RA-DIT fine-tunes both the LM and the retriever while decoupling the fine-tuning processes of the two components.[16] Asai et al. [2023] fine-tunes an LM to adaptively retrieve passages on demand and reflect on the relevancy of the retrieved passages and its generation using special-token markups. In comparison, RA-DIT is trained using a supervised fine-tuning objective and does not explicitly generate reflection tokens. The most relevant work to ours is SAIL [Luo et al., 2023], 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 for multiple retrieved passages while SAIL concatenates them in the LM context. Furthermore, RA-DIT adopts a holistic view of the RALM architecture by employing a learnable neural retriever and proposing a dual optimization framework. SAIL, in comparison, leans on non-differentiable retrievers such as BM25 and focuses on improving the LM (e.g. it proposes an in-context retrieval selection technique to guide the model focus towards informative content).

[16] Although the differences in the base LMs, fine-tuning datasets and inference settings make direct comparisons between the two models challenging, RA-DIT 65B compares favorably to InstructRetro 48B [Wang et al., 2023a] in zero-shot setting on the shared evaluation datasets.

Information Retrieval Retrieval methods include *sparse retrievers* that does matching over a sparse bag-of-words representation [Robertson and 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 and Zaharia, 2020; Li et al., 2023]. We adopt a state-of-the-art dense retriever, DRAGON [Lin et al., 2023a], 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.

3.7 Conclusion

In this chatpter, we proposed 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.

Table 3.12: 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.

Prompt	p_R	Output		nll_{LM}	
		RA-IT	IT	RA-IT	IT
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 ✗					
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 G̈rey Will Fade, on the B-side of the single T̈here'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:	0.27	1992	1997	1.16	1.01
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 g̈irl 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:	0.21	1992	1992	0.46	0.98
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:	0.13	1992	I don't know.	0.54	0.72
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. ✗					
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 Rocklea2̌014in the member for Morton's electorate2̌014which 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:	0.25	10,000	I don't know.	7.27	0.61
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 Macquarie2̌019s 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:	0.15	45,698	45,698	0.18	0.38
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 you2̌019ll be aiming for and Tamworth is the major western town of reference on the map. Once you2̌019re in the main streets of Port, you2̌019ll 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 2̌013 just perfect for all you thrill seeking middle aged folk and laid back grey nomads and let2̌019s 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:	0.12	76,000	76,000	4.85	0.93

Chapter 4

Multilingualism

In previous chapters, we demonstrated that Language Language Models (LLMs), upon appropriate adaptation, prove to be competitive in both few-shot and zero-shot learning scenarios. However, the pre-training data of these models is dominated by English [Brown et al., 2020], potentially limiting their cross-lingual generalization. Parallel to these developments, there has been a growing wave of interest within the community focused on the investigation of multilingual encoder-only and encoder-decoder language models, including mBERT, XLM-R, mT5, and mBART [Devlin et al., 2019; Conneau et al., 2020; Xue et al., 2021; Liu et al., 2020]. These models are typically fine-tuned on many labeled data when applied to downstream tasks. The multilingual few-shot learning capabilities of auto-regressive language models, especially when no parameter updates are performed during inference time, are less well understood.

In this chapter[1], we pre-train multilingual auto-regressive language models (up to 7.5B parameters) by upsampling the medium- and low-resource languages in the pre-training corpus. We evaluate the resulting models (XGLM) on multiple multilingual natural language understanding (NLU) tasks, machine translation and a subset of English tasks as featured in Brown et al. [2020]. Our findings indicate that XGLM exhibits robust few-shot learning performance across a diverse range of languages, with a small sacrifice of its English performance.

[1]The work in this chapter was conducted at Meta AI, in collaboration with Xian Li*, Todor Mihaylov, Mikel Artetxe, Shuohui Chen, Veslin Stoyanov† and others [Lin et al., 2021]. (*co-first author, †research leadership)

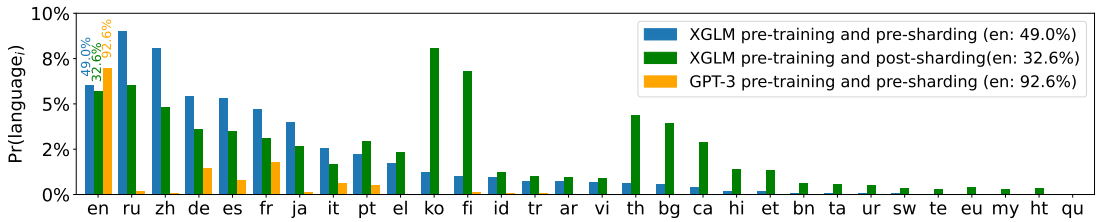

Figure 4.1: The % of each language l ($l = 1, 2, ..., 30$) in XGLM's pre-training data pre-upsampling (blue), post-upsampling (green), and its corresponding % in GPT-3's training data (orange). We truncate the y-axis at 10% to better visualize the tail distribution.

4.1 Pre-training Data

Language selection and pre-processing. We extend the pipeline used for mining the CC100 corpus [Conneau et al., 2020; Wenzek et al., 2020] to generate CC100-XL, a significantly larger multilingual dataset covering 68 Common Crawl (CC) snapshots (from Summer 2013 to March/April 2020) and 134 languages. Our pretraining data include 30 languages covering 16 language families. We group the languages into four resource tiers: *high*, *medium*, *low* and *extremely-low*, based on the amount of data they have after filtering. Let X_l be the number of tokens of a language l in the pre-training data, we define the resource tier according to the following criteria *high*: $X_l \geq 48$B, *medium*: $5\text{B} \leq X_l < 48\text{B}$, *low*: $0.1\text{B} \leq X_l < 5\text{B}$, *extemely-low*: $X_l < 0.1\text{B}$. Following previous work on multilingual pre-training [Conneau et al., 2020], we up-sampled the medium and low resource languages to create a more balanced language distribution. Figure 4.1 shows the language distribution of our pre-training data before (blue) and after (green) up-sampling.[2]

Joint sub-word vocabulary. We process all languages with a joint vocabulary of size 250k created through unigram language modeling [Kudo, 2018], using the SentencePiece library [Kudo and Richardson, 2018]. We train the unigram-LM model using 10 million sentences randomly sampled from a subset of the pre-training data, according to the multinomial distribution defined in Lample and Conneau [2019] with $\alpha = 0.3$.

[2]We inadvertently over-sampled some of the less resourced languages which is reflected in the statistics of *ko*, *fi*, *th*, *bg*, *ca*, *hi*, *et* languages, as shown in Figure 4.1. We did not ablate the effect of this mistake due to the extreme computational cost. Studying optimal language balancing is an important area for future work.

4.2 Models

We train decoder-only language models with the transformer architecture similar to GPT-3. This allows us to study the effect of scaling up model size along both width and depth dimensions. As a result, we compare four models with 564M, 1.7B, 2.9B and 7.5B parameters, respectively. The architecture details are summarized in Table 4.1. Our models match the dimension of GPT-3 models except for the additional embedding parameters from a larger vocabulary.[3] All models are trained for up to 500B tokens, with a context length of 2048 tokens.

GPT-3			XGLM		
size	l	h	*size*	l	h
125M	12	768		—	
355M	24	1024	564M	24	1024
760M	24	1536		—	
1.3B	24	2048	1.7B	24	2048
2.7B	32	2560	2.9B	48	2048
6.7B	32	4096	7.5B	32	4096

Table 4.1: Model details. *size*: number of parameters, l: layers, h: hidden dimension. Models within the same row have comparable sizes.

Pre-training details. We use the Adam optimizer [Kingma and Ba, 2015] with $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 1e-8$. We adjust the learning rate based on model size, e.g. $1.5e-3$ for the 564M and 1.7B model, $7.5e-4$ for the 2.9B model, and $1.2e-4$ for the 7.5B models. Learning rates were adjusted with a 2000 warm-up updates followed by a polynomial decay schedule. All models are trained with data parallel and an effective batch size of 4M tokens. The XGLM 7.5B model was trained on 256 A100 GPUs for about 3 weeks, at a speed of 311.6k words per second.

4.3 Multilingual and Cross-lingual Prompting

We consider three approaches for generating the prompts for evaluating the XGLM models in multilingual downstream tasks.

[3]For XGLM 2.9B we used the optimal depth-to-width parameter allocation for GPT-3 architectures based on rank bottleneck analysis [Levine et al., 2020]. This allocation is expected to have improved training efficiency. However, this setting did not converge for XGLM 7.5B in our experiments, and we fell back to the original GPT-3 setup.

Task Category	Dataset	Template	Candidate Verbalizer
Reasoning	XCOPA XStoryCloze XWinograd	*cause*: `{Sentence 1}` because `[Mask]` *effect*: `{Sentence 1}` so `[Mask]` `{Context} [Mask]` `{Context}` (*with '_' replaced by* `[Mask]`)	Identity
NLI	XNLI	`{Sentence 1}`, right? `[Mask]`, `{Sentence 2}`	*Entailment*: Yes | *Neural*: Also | *Contradiction*: No
Paraphrase	PAWS-X	`{Sentence 1}`, right? `[Mask]`, `{Sentence 2}`	*True*: Yes | *False*: No
Translation	WMT, FLORES-101	`{Source sentence}` = `[Mask]`	Identity

Table 4.2: Handcrafted (*en*) prompts for multilingual NLU and translation tasks.

- **Handcrafting prompts.** The first approach is to ask native speakers of the target language to handcraft the prompts. Prompts created this way are expected to be natural and grammatically sound. Nonetheless, language expertise is expensive and we further consider two alternatives.

- **Translating from English prompts.** We assume sourcing high-quality prompts for a given task in English is relatively straightforward [Bach et al., 2022]. As shown in Table 4.2, we characterize prompts as *verbal* and *non-verbal*. Non-verbal prompts do not contain words in any particular language (e.g. the StoryCloze and WMT prompts), while verbal prompts have different realizations in different languages (Table 4.3). Non-verbal prompts can easily generalize across languages.[4] If the prompt is verbal, we translate it into the other languages using Google translation APIs.[5]

- **Cross-lingual prompting.** The third approach which directly applies the prompts in English (or another high-resource language) to non-English examples. We expect this approach to be competitive, as a result of the cross-lingual capability of the model after being trained on a diverse set of languages.

Task	Lang	Template	Candidate Verbalizer		
			Entailment	Contradiction	Neutral
XNLI	en	`{Sentence 1}`, right? `[Mask]`, `{Sentence 2}`	Yes	No	Also
	zh	`{Sentence 1}[Mask]`, `{Sentence 2}`	由此可知，	所以，不可能	同时，
	es	`{Sentence 1}`, ¿verdad? `[Mask]`, `{Sentence 2}`	Sí	No	Además

Table 4.3: Handcrafted multilingual prompts. English (*en*), Chinese (*zh*) and Spanish (*es*) for XNLI.

[4]Different languages may have different punctuation systems. We try to ensure consistency in punctuation according to each language's standards when mapping a template across languages.

[5]`https://cloud.google.com/translate`

Temp.	en	zh	es	hi	Avg
En (HW)	**50.8/50.6**	**48.5/47.7**	37.5/44.4	**44.0/45.5**	**45.2/47.0**
Zh (HW)	33.5/35.5	33.5/36.4	34.5/34.8	36.0/34.0	34.4/35.1
Es (HW)	39.2/49.9	44.8/45.3	**46.2/48.2**	41.5/43.5	42.9/46.7
Hi (HW)	45.0/43.5	39.5/41.0	34.2/40.5	36.2/40.5	38.8/41.4
Multi. (HW)	50.8/50.6	33.5/36.4	**46.2/48.2**	36.2/40.5	41.7/43.9
Multi. (MT)	50.8/50.6	35.8/39.5	36.5/45.0	41.0/39.9	41.0/43.8
Multi. (HT)	50.8/50.6	38.5/41.2	46.0/48.1	37.5/38.9	43.1/44.7

Table 4.4: 0/4-shot performance of XGLM 7.5B, evaluated on the first 400 examples of XNLI (development set in *en*, *zh*, *es* and *hi*) using different prompting approaches. Top: all inputs are instantiated with templates in the language specified in column 1. Bottom: all inputs are instantiated with templates in the same language as themselves. HW: human-written. MT: machine-translated. HT: human-translated.

4.4 Evaluation Tasks

Multilingual tasks. We select four multilingual tasks spanning commonsense reasoning (XCOPA), anaphora resolution (XWinograd), natural language inference (XNLI) and paraphrasing (PAWS-X) for our downstream evaluation. We also created a new dataset, XStoryCloze, by professionally translating the validation split of the English StoryCloze dataset (Spring 2016 version) to 10 other typologically diverse languages (*ru*, *zh* Simplified, *es* Latin American, *ar*, *hi*, *id*, *te*, *sw*, *eu*, *my*). We further split the translated data into train and test (20% vs. 80%, respectively) for each language, keeping the parallel sentence mapping in both splits.

Machine Translation. We also report machine translation results on a subset of the FLORES-101. This dataset consists of 3001 sentences extracted from English Wikipedia which cover a variety of different topics and domains, professionally translated in 101 languages [Goyal et al., 2022].

4.5 Experiments

4.5.1 Cross-lingual Transfer through Templates

We first compare different multilingual prompting approaches proposed in §4.3 by evaluating XGLM 7.5B using different settings on XNLI. Native speakers among the authors handcrafted[6] the prompts for the fol-

[6]The native speakers were instructed to create a prompt that convert the task into a natural cloze-style question in their native language with no further restrictions.

lowing languages: *en*, *zh*, *es* and *hi*, as shown in Table 4.3. We compare the performance of these human-written prompts to English prompts, machine-translated (MT) prompts and human-translated (HT) prompts. The results are shown in Table 4.4. We observe that the English templates perform the best on average across languages. In particular, it significantly improves the performance of Chinese (*zh*) and Hindi (*hi*) over their native templates and translated templates. On the other hand, some language (*es*) still signiifcantly benefit from using the native prompt, indicating significant room for future work on language-specific prompt optimization.

We further examine if the ability of universal prompting is English specific, and in addition, what conditions can effectively induce cross-lingual transfer through templates. To this end, we apply each of the human-written non-English templates to the rest of the languages. As shown in Table 4.4, using the Spanish prompt yields competitive 0- and 4-shot performance across all languages, with the 4-shot average performance being comparable to that of the English template. The Hindi template also achieves significantly above random performance on the XNLI tasks for most languages (especially *en*). The Chinese template, however, achieves near random performance for all languages on XNLI.[7] We hypothesize that shared vocabulary and the amount of code-switching text in the pre-training data play a significant role in enabling cross-lingual transfer through template. And in general, high-resource languages with more pre-training data and higher vocabulary overlap with other languages act as better universal prompting languages. We leave the systematic verification of this hypothesis to future work.

4.5.2 Cross-lingual Transfer through Demonstration Examples

We examine XGLM 7.5B's ability of learning from cross-lingual demonstration examples on XNLI. We examine two settings for each train-eval language pair: *same-language-prompting*, where the prompt templates and the example are in the same language, and *source-languauge-prompting* where the prompt templates for both the demo and test examples are in the source language. We use the human-translated prompts for *same-language-prompting*.

Table 4.5 shows results on a subset of language pairs of XNLI, where we evaluate transfer through

[7]It is noteworthy that even when combined with *zh* examples, the *zh* template performs poorly. The same holds true for *hi*. A plausible explanation for this could be that the non-English data in XNLI, being translated from the English NLI dataset, may not adequately reflect the Chinese and Hindi culture, resulting in less authentic content in these languages.

	high									medium		low
	en								ru	tr	ar	hi
	medium					low			medium	low		
prompt type	bg	el	th	tr	vi	hi	sw	ur	bg	ur	sw	ur
Same-lang	**2.55**	**0.98**	**2.16**	**1.27**	**2.23**	**2.51**	-0.69	**1.21**	-2.49	-0.38	-1.64	**3.31**
Source-lang	-4.59	-2.44	**7.87**	-4.97	-1.08	**2.01**	-1.15	**7.42**	-1.43	**6.67**	-5.86	**2.31**

Table 4.5: Learning from cross-lingual demonstrations on XNLI, evaluated on the test set. The results are the absolute improvement over the zero-shot performance for the evaluated language using human-translated prompts. The first language group refers to the source language and the second one refers to the target language. *Same-lang* refers to a setting where the template is in the example language and *source-lang* refers to a setting where the template is only in the source language.

demonstration examples from in-context demonstration examples from high-resource languages to lower-resourced ones, and between languages that are typologically similar. We report the difference between the 32-shot learning results and the 0-shot learning results. The non-English templates in this experiment are obtained via human-translation. While they typically demonstration using demonstration examples in the same language, most cross-lingual few-shot settings significantly improve over the 0-shot setting for the target language. We found Bulgarian an interesting exception, as it does not benefit from Russian examples despite being in the same language family. Another language that does not work well in the cross-lingual settings is Swahili (*low* resource), for which we examined transfer from English (*high* resource) and Arabic (*medium* resource). In contrast, Thai (*medium*) and Urdu (*low* resource) significantly benefit from cross-lingual demonstrations[8].

4.5.3 Performance on Machine Translation

We report machine translation results on a subset of FLORES-101 [Goyal et al., 2022] in Table 4.6. We use greedy decoding for both GPT-3 and our own model, and use the same 32 examples for few-shot learning in each case. XGLM obtain solid results across the board. In addition to surpassing GPT-3 in 171 out of 182 language pairs in the FLORES-101 set, it is also competitive with the official supervised baseline for this dataset, even surpassing it in 45 language pairs. This suggests that large-scale multilingual language models

[8]Both Thai and Urdu obtained close-to-random zero-shot learning performances using the translated templates, which might make them easier to be further improved. Besides, there is inherent code switching in these languages (English presence in Thai and Urdu both lexical and morphological). Turkish and Arabic also have influence on Urdu. We hypothesize that these factors also positively impacted the cross-lingual in-context learning performance.

have a great potential for building machine translation systems for low-resource languages, even if little or no parallel data is available.

		en	de	fr	ca	fi	ru	bg	zh	ko	ar	sw	hi	my	ta	Avg.
avg out of xx	Supervised	24.0	21.0	20.4	19.1	17.5	18.6	20.2	15.5	14.9	16.1	16.6	16.2	7.2	4.8	16.6
	GPT-3 6.7B	9.9	9.1	9.4	9.3	6.4	7.0	5.5	4.9	2.4	2.9	1.7	0.5	0.2	0.3	5.0
	XGLM 7.5B	21.1	16.5	17.1	13.6	13.4	13.2	13.9	9.1	6.5	10.4	12.1	9.8	6.9	7.1	12.2
avg into xx	Supervised	26.0	20.2	26.7	20.0	16.7	18.5	24.5	14.1	13.5	11.8	16.3	19.3	2.1	2.5	16.6
	GPT-3 6.7B	18.9	9.9	14.2	9.3	4.2	4.8	2.7	4.0	0.6	0.5	0.2	0.3	0.1	0.1	5.0
	XGLM 7.5B	28.5	14.9	20.6	14.4	10.9	12.4	18.5	10.9	5.9	6.1	8.5	9.7	5.8	3.5	12.2

Table 4.6: Results on the FLORES-101 dev set. The results are measured in spBLEU computed using the implementation from Goyal et al. [2022]. GPT-3 6.7B and XGLM 7.5B use 32 examples from the dev set for few-shot learning. Supervised results correspond to the M2M-124 615M model from Goyal et al. [2022].

4.5.4 Performance on English Tasks

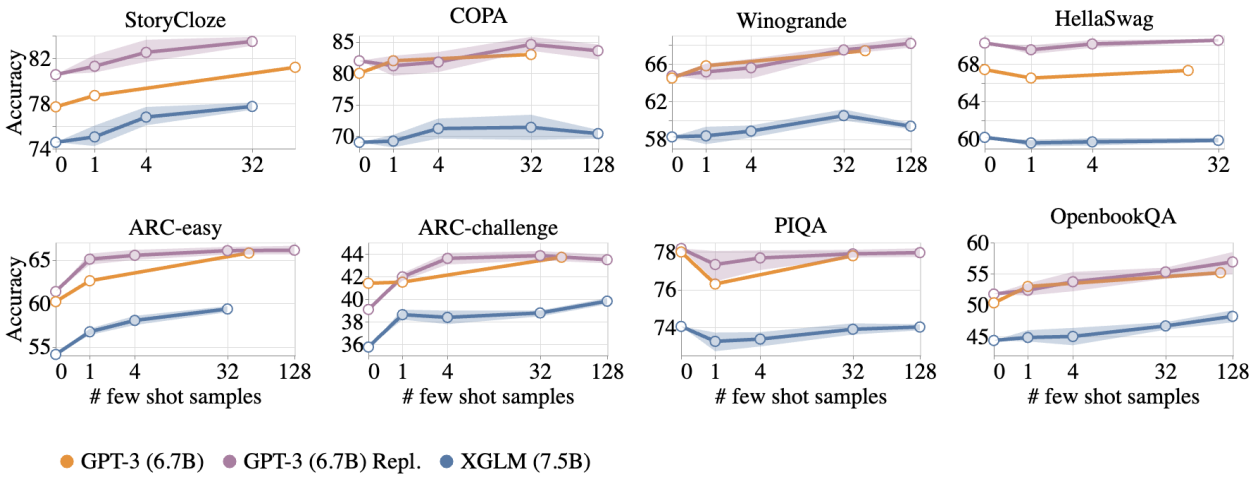

Figure 4.2: Performance on English tasks. For XGLM 7.5B and XGLM-EN, we plot the confidence interval from 5 different runs corresponding to different training sets when $k > 0$. For GPT-3 6.7B we use the performance reported by Brown et al. [2020].

We also benchmark the performance of XGLM 7.5B on English tasks. Figure 4.2 shows the comparison between XGLM 7.5B, GPT-3 6.7B and XGLM-EN on a subset of English tasks used by Brown et al. [2020]. Our replication of GPT-3 6.7B, XGLM-EN, performs better than or close to GPT-3 6.7B on all tasks. While XGLM 7.5B performs competitively on all tasks, there remains a considerable performance gap comparing to GPT-3 6.7B and XGLM-EN. On most tasks XGLM 7.5B and XGLM-EN show similar performance trend

as k changes. For example, both models show a performance dip at 1-shot on HellaSwag and PIQA, and 128-shot on COPA.

There are multiple reasons why XGLM 7.5B underperforms English centric models on the English tasks. First, only 32.6% of XGLM 7.5B's 500B-token training data is English while both English-centric models are trained on close to 300B English tokens. Second, the model capacity of XGLM 7.5B is shared by 30 languages, and the "curse of multilinguality" can degrade the performance across all languages [Conneau et al., 2020]. Further scaling up the model capacity and training data can potentially close this gap. The differences between the training corpora of the three models may have also contributed to the performance difference. While both English centric models incorporate high-quality English monolingual corpora such as BookCorpus [Zhu et al., 2015] in their training data (GPT-3 6.7B also upsamples such high-quality data), XGLM 7.5B is trained solely on data extracted from Common Crawl. However, we do not expect this to be the main impact factor. [Le Scao et al., 2022] conducted a similar experiment showing that a multilingual model (1.3B parameters) pre-trained over 13 languages also significantly underperforms an English model trained from the same data source in terms of zero-shot generalization.

4.6 Related Work

Language model prompting. Brown et al. [2020] first demonstrated in-context few-shot learning using the GPT-3 model. This method removes the need for task-specific updates to the model parameters: the few-shot examples that one would normally use for fine-tuning are provided at inference time to the same model for each task. On several high-resource Latin language pairs, GPT-3 achieves machine translation performance that is close to or better than state-of-the-art supervised models, given only a handful of demonstration examples. Such change in the learning paradigm raises new questions about multilinguality, which has not been studied as extensively. Winata et al. [2021] evaluates the in-context few-shot learning abilities of several GPT-2, GPT NEO and T5 on three additional languages (*de*, *es*, *fr*) using multiple NLU tasks, considering monolingual prompts as well as cross-lingual prompts, demonstrating the multilingual in-context learning skills of the English GPT and T5 models. Zhao and Schütze [2021] evaluated different fine-tuning and prompt-tuning [Liu et al., 2021] approaches on XLM-R and demonstrates the effectiveness of prompting in few-shot crosslingual transfer and in-language training of a multilingual masked language

model. Blevins and Zettlemoyer [2022] shows that language contamination in pre-training data can effectively boost the cross-lingual capability of English-centric language models. With a heavier tail of deliberately introduced multilingual data, PALM-540B [Chowdhery et al., 2022] later achieves even stronger few-shot machine translation performance.

Multilingual pre-training. Early multilingual pre-training work train word embeddings over multilingual corpora [Mikolov et al., 2013]. The multilingual versions of contextualized embedding models such as BERT [Devlin et al., 2019], RoBERTa [Liu et al., 2019], BART [Lewis et al., 2020a] and T5 [Raffel et al., 2020] were also developed: mBERT [Devlin et al., 2019], XLM-R [Conneau et al., 2020], mBART [Liu et al., 2020], and mT5 [Xue et al., 2021]. Such models were trained on a single, multilingual text corpus such as mC4 [Xue et al., 2021] or CC25 [Liu et al., 2020].

Several approaches have been developed to facilitate cross-lingual transfer, including sub-word tokenizers which enabled efficient, shared vocabulary learning across languages [Kudo and Richardson, 2018], joint training for efficient knowledge transfer across languages [Pires et al., 2019; Jiang et al., 2020; Kassner et al., 2021], etc. A notable concurrent work is BLOOM [9], which scales multilingual pre-training to 46 languages and 175 billion parameters.

4.7 Conclusion

We introduce four multilingual generative language models (XGLMs) at different scales, and study their in-context few- and zero-shot learning capabilities. We show that the few-shot learning capability of XGLM steadily improves as it scales. Our largest model (7.5B parameters) sets a new state of the art for few-shot learning in more than 20 languages (including mid- and low-resource languages) on commonsense reasoning, NLI and machine translation tasks. An in-depth analysis shows the models are highly cross-lingual, which leads to strong few-shot learning performance in non-English languages.

[9] https://bigscience.huggingface.co/blog/bloom

Chapter 5

Discussion and Future Work

In this chapter, we further discuss the approaches introduced in previous chapters and explore potential future directions.

Instruction Meta-Learning (a.k.a. Instruction Tuning) In Chapter 2 we conducted an ablation study on fine-tuning OPT-IML models and found that increasing the diversity of tasks during fine-tuning significantly improves cross-task generalization. In this study, we used publicly released datasets from the research community for both fine-tuning and evaluation [Iyer et al., 2022]. These datasets typically provide concise and precise answers to the given prompts. As a result, they differ in terms of task types, instruction language and input/output formats. On the other hand, commercial assistants are primarily designed to handle tasks that require long-form generation [Ouyang et al., 2022b] and chain-of-thought reasoning [Wei et al., 2022c]. Recent instruction tuning benchmarks, such as AlpacaEval [Taori et al., 2023], Vicuna [Chiang et al., 2023], LIMA [Zhou et al., 2023], and others, consist exclusively of such data. Notably, Zhou et al. [2023] demonstrated that a small set of high-quality, single-turn instruction-following questions consisting of 1,000 examples can enable an LLM to effectively respond to long-form questions from a wide range of domains, even those beyond its training set. This suggests that formulating the input and output of fine-tuning examples as fluent, long-form text can improve the efficiency of instruction-tuning and enhance the overall interaction experience with the fine-tuned LLM. One potential approach to achieve this is to convert (manually or semi-automatically) a subset of benchmark examples into the chain-of-thought format and use them for fine-tuning, which is a direction of improvement for OPT-IML.

Retrieval Augmentation In Chapter 3, we introduce the Retrieval-Augmented Dual Instruction Tuning (RA-DIT) approach, which can be used to enhance any LLM with information retrieval capabilities [Lin et al., 2023b]. We have observed that the performance of the retriever in finding useful information declines as the input task becomes more complex, such as in HotpotQA [Yang et al., 2018]. This may be due to a combination of factors, including the degradation of retriever performance for complex inputs and the need for better query synthesis methods. One possible solution is to first break down complex inputs into simpler components through a query decomposition stage, as demonstrated by Press et al. [2023] and Asai et al. [2023]. Another direction for improvement is to further enhance the capabilities of neural retrievers.

Multilinguality In Chapter 4, we present a method for improving the cross-lingual capabilities of pre-trained multilingual LLMs by up-sampling languages in the tail distribution of a multilingual pre-training corpus [Lin et al., 2021]. We evaluate the resulting XGLM in an in-context few-shot learning setup and show that they demonstrate competitive abilities for cross-lingual prompting and learning from examples in other languages. Including a heavy-tail distribution of languages has been shown to be effective in larger-scale LLM training, as demonstrated by the training data of PaLM which consists of 22% non-English languages [Chowdhery et al., 2022]. Recent work [Shaham et al., 2024] also shows that a small set of 40 multilingual examples in an English tuning set can improve the multilingual instruction following capabilities of LLMs such as PaLM-2 [Anil et al., 2023], and monolingual fine-tuning can transfer instruction-following capabilities to other languages. These findings suggest that multilingual instruction tuning is a promising next step for improving the cross-lingual capabilities of a multilingual language model.

Multimodality Finally, the approaches and models presented in this thesis are limited to the text modality only. This significantly limits the tasks that the model can solve and the interaction experience it can enable, often resulting in a less immersive experience for users. For example, in order to assist UI designers, it would be beneficial for the LLM to take entire screenshots as input and continuously monitor the user's past actions [Lee et al., 2023; Shaw et al., 2023]. Incorporating additional modalities such as images, audio, and video would also greatly enhance enhance creative generation capabilities, enabling controlled visual document generation, audio and video synthesis, and more [Yu et al., 2023; Lu et al., 2023; Team et al., 2023]. Incorporating multiple modalities into an LLM and enabling complex reasoning on top is a challenging task

and requires significant advancements in multi-modal representation learning and fusion techniques [Aghajanyan et al., 2023; Liang and Morency, 2023], which is an area that can benefit significantly from future research.

Bibliography

Shourya Aggarwal, Divyanshu Mandowara, Vishwajeet Agrawal, Dinesh Khandelwal, Parag Singla, and Dinesh Garg. 2021a. Explanations for commonsenseqa: New dataset and models. In *ACL*, pages 3050–3065.

Shourya Aggarwal, Divyanshu Mandowara, Vishwajeet Agrawal, Dinesh Khandelwal, Parag Singla, and Dinesh Garg. 2021b. Explanations for CommonsenseQA: New dataset and models. In *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*, pages 3050–3065. Association for Computational Linguistics.

Armen Aghajanyan, Lili Yu, Alexis Conneau, Wei-Ning Hsu, Karen Hambardzumyan, Susan Zhang, Stephen Roller, Naman Goyal, Omer Levy, and Luke Zettlemoyer. 2023. Scaling laws for generative mixed-modal language models. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 265–279. PMLR.

Neel Alex, Eli Lifland, Lewis Tunstall, Abhishek Thakur, Pegah Maham, C Jess Riedel, Emmie Hine, Carolyn Ashurst, Paul Sedille, Alexis Carlier, et al. 2021. Raft: A real-world few-shot text classification benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. 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), pages 2357–2367, Minneapolis, Minnesota. Association for Computational Linguistics.

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. 2023. PaLM 2 technical report.

Anonymous. 2023. Progressive prompts: Continual learning for language models without forgetting. In *Submitted to The Eleventh International Conference on Learning Representations*. Under review.

Vamsi Aribandi, Yi Tay, Tal Schuster, Jinfeng Rao, Huaixiu Steven Zheng, Sanket Vaibhav Mehta, Honglei Zhuang, Vinh Q. Tran, Dara Bahri, Jianmo Ni, Jai Prakash Gupta, Kai Hui, Sebastian Ruder, and Donald

Metzler. 2022. Ext5: Towards extreme multi-task scaling for transfer learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Mikel Artetxe, Shruti Bhosale, Naman Goyal, Todor Mihaylov, Myle Ott, Sam Shleifer, Xi Victoria Lin, Jingfei Du, Srinivasan Iyer, Ramakanth Pasunuru, Giridharan Anantharaman, Xian Li, Shuohui Chen, Halil Akin, Mandeep Baines, Louis Martin, Xing Zhou, Punit Singh Koura, Brian O'Horo, Jeffrey Wang, Luke Zettlemoyer, Mona Diab, Zornitsa Kozareva, and Veselin Stoyanov. 2022a. Efficient large scale language modeling with mixtures of experts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11699–11732, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mikel Artetxe, Jingfei Du, Naman Goyal, Luke Zettlemoyer, and Veselin Stoyanov. 2022b. On the role of bidirectionality in language model pre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3973–3985, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection.

Stephen Bach, Victor Sanh, Zheng Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, Zaid Alyafeai, Manan Dey, Andrea Santilli, Zhiqing Sun, Srulik Ben-david, Canwen Xu, Gunjan Chhablani, Han Wang, Jason Fries, Maged Alshaibani, Shanya Sharma, Urmish Thakker, Khalid Almubarak, Xiangru Tang, Dragomir Radev, Mike Tian-jian Jiang, and Alexander Rush. 2022. PromptSource: An integrated development environment and repository for natural language prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 93–104, Dublin, Ireland. Association for Computational Linguistics.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

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

Terra Blevins and Luke Zettlemoyer. 2022. Language contamination helps explains the cross-lingual capabilities of English pretrained models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3563–3574, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ B. Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen Creel, Jared Quincy Davis, Dorottya Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, and et al. 2021. On the opportunities and risks of foundation models. *CoRR*, abs/2108.07258.

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. 2022. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR.

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. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-SNLI: Natural language inference with natural language explanations. *Advances in Neural Information Processing Systems*, 31.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28:41–75.

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. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

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

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. 2022b. Scaling instruction-finetuned language models.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. 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)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

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

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. 2021a. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021b. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 160–167, New York, NY, USA. Association for Computing Machinery.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-

lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander H. Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Iulian Serban, Ryan Lowe, Shrimai Prabhumoye, Alan W. Black, Alexander I. Rudnicky, Jason D. Williams, Joelle Pineau, Mikhail S. Burtsev, and Jason Weston. 2019a. The second conversational intelligence challenge (convai2). *CoRR*, abs/1902.00098.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019b. Wizard of Wikipedia: Knowledge-powered conversational agents. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019c. Wizard of wikipedia: Knowledge-powered conversational agents. In *International Conference on Learning Representations*.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. 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)*, pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. 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. European Language Resources Association (ELRA).

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. 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.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021a. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021b. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. 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*, pages 3929–3938. PMLR.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Stein-

hardt. 2021a. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

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

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021c. Measuring mathematical problem solving with the math dataset. *NeurIPS*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021d. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1693–1701.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 782–792, Edinburgh, Scotland, UK. Association for Computational Linguistics.

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

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023. Unnatural instructions: Tuning language models with (almost) no human labor. In *Proceedings of the 61st Annual Meeting of the Association*

for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023, pages 14409–14428. Association for Computational Linguistics.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2022. Editing models with task arithmetic.

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. 2022. OPT-IML: scaling language model instruction meta learning through the lens of generalization. *CoRR*, abs/2212.12017.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022a. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

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

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

Kelvin Jiang, Dekun Wu, and Hui Jiang. 2019. 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)*, pages 318–323, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhengbao Jiang, Antonios Anastasopoulos, Jun Araki, Haibo Ding, and Graham Neubig. 2020. X-FACTR: Multilingual factual knowledge retrieval from pretrained language models. In *Proceedings of the 2020*

Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 5943–5959, Online. Association for Computational Linguistics.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. 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)*, pages 2567–2577, Hong Kong, China. Association for Computational Linguistics.

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

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

Nora Kassner, Philipp Dufter, and Hinrich Schütze. 2021. Multilingual LAMA: Investigating knowledge in multilingual pretrained language models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3250–3258, Online. Association for Computational Linguistics.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online. Association for Computational Linguistics.

Omar Khattab and Matei Zaharia. 2020. 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*, page 39–48. Association for Computing Machinery.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. QASC: A dataset for question answering via sentence composition. 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*, pages 8082–8090. AAAI Press.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*.

Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2022. Internet-augmented dialogue generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 8460–8478. Association for Computational Linguistics.

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. 2023. Openassistant conversations–democratizing large language model alignment. *arXiv preprint arXiv:2304.07327*.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical*

Methods in Natural Language Processing: System Demonstrations, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Abhishek Kumar and Hal Daume III. 2012. Learning task grouping and overlap in multi-task learning. In *ICML*.

Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1378–1387, New York, New York, USA. PMLR.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Matthew Lamm, Jennimaria Palomaki, Chris Alberti, Daniel Andor, Eunsol Choi, Livio Baldini Soares, and Michael Collins. 2021. QED: A framework and dataset for explanations in question answering. *Transactions of the Association for Computational Linguistics*, 9:790–806.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *CoRR*, abs/1901.07291.

Teven Le Scao, Thomas Wang, Daniel Hesslow, Stas Bekman, M Saiful Bari, Stella Biderman, Hady Elsahar, Niklas Muennighoff, Jason Phang, Ofir Press, Colin Raffel, Victor Sanh, Sheng Shen, Lintang Sutawika, Jaesung Tae, Zheng Xin Yong, Julien Launay, and Iz Beltagy. 2022. What language model to train if you have one million GPU hours? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 765–782, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. 2023. Pix2Struct: Screenshot parsing as pretraining for visual language understanding. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 18893–18912. PMLR.

Yoav Levine, Noam Wies, Or Sharir, Hofit Bata, and Amnon Shashua. 2020. Limits to depth efficiencies of self-attention. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

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. 2020b. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

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. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Minghan Li, Sheng-Chieh Lin, Barlas Oguz, Asish Ghoshal, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. 2023. 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)*, pages 11891–11907, Toronto, Canada. Association for Computational Linguistics.

Paul Pu Liang and Louis-Philippe Morency. 2023. Tutorial on multimodal machine learning: Principles, challenges, and open questions. In *International Conference on Multimodal Interaction, ICMI 2023, Companion Volume, Paris, France, October 9-13, 2023*, pages 101–104. ACM.

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

Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Rich James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, Luke Zettlemoyer, and Scott Yih. 2023b. Ra-dit: Retrieval-augmented dual instruction tuning.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona T. Diab, Veselin Stoyanov, and Xian Li. 2021. Few-shot learning with multilingual language models. *CoRR*, abs/2112.10668.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017a. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *ACL*.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017b. 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), pages 158–167, Vancouver, Canada. Association for Computational Linguistics.

Jerry Liu. 2022. LlamaIndex.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9):195:1–195:35.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. GPT understands, too. *CoRR*, abs/2103.10385.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Jiasen Lu, Christopher Clark, Sangho Lee, Zichen Zhang, Savya Khosla, Ryan Marten, Derek Hoiem, and Aniruddha Kembhavi. 2023. Unified-io 2: Scaling autoregressive multimodal models with vision, language, audio, and action.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

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

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memo-

ries. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6294–6305.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018a. The natural language decathlon: Multitask learning as question answering. *CoRR*, abs/1806.08730.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018b. The natural language decathlon: Multitask learning as question answering. *CoRR*, abs/1806.08730.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. 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*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Tomás Mikolov, Quoc V. Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.

Katharine Miller. 2023. How Do We Fix and Update Large Language Models?

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022a. MetaICL: Learning to learn in context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2791–2809, Seattle, United States. Association for Computational Linguistics.

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

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487, Dublin, Ireland. Association for Computational Linguistics.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. *CoRR*, abs/1611.09268.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. 2021. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*.

Yasumasa Onoe, Michael JQ Zhang, Eunsol Choi, and Greg Durrett. 2021. CREAK: A dataset for commonsense reasoning over entity knowledge. *arXiv preprint arXiv:2109.01653*.

OpenAI. 2023. Gpt-4 technical report.

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. 2022a. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. 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. 2022b. Training language models to follow instructions with human feedback. In *NeurIPS*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

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. 2021. 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*, pages 2523–2544, Online. Association for Computational Linguistics.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, Singapore. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, and Jeff Wu. 2021. Gpt-2 output dataset. `https://github.com/openai/gpt-2-output-dataset`. Last Updated: 02-17-21.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

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

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. 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)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for

machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

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

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

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

Anna Rogers, Olga Kovaleva, Matthew Downey, and Anna Rumshisky. 2020. 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*, pages 8722–8731. AAAI Press.

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

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. 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), pages 4463–4473, Hong Kong, China. Association for Computational Linguistics.

Thomas Scialom, Tuhin Chakrabarty, and Smaranda Muresan. 2022. Continual-t0: Progressively instructing 50+ tasks to language models without forgetting. *CoRR*, abs/2205.12393.

Uri Shaham, Jonathan Herzig, Roee Aharoni, Idan Szpektor, Reut Tsarfaty, and Matan Eyal. 2024. Multilingual instruction tuning with just a pinch of multilinguality.

Peter Shaw, Mandar Joshi, James Cohan, Jonathan Berant, Panupong Pasupat, Hexiang Hu, Urvashi Khandelwal, Kenton Lee, and Kristina Toutanova. 2023. From pixels to ui actions: Learning to follow instructions via graphical user interfaces.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. 2023a. Large language models can be easily distracted by irrelevant context. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 31210–31227. PMLR.

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

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*.

Kurt Shuster, Jack Urbanek, Emily Dinan, Arthur Szlam, and Jason Weston. 2021. Dialogue in the wild: Learning from a deployed role-playing game with humans and bots. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 611–624, Online. Association for Computational Linguistics.

Kurt Shuster, Jing Xu, Mojtaba Komeili, Da Ju, Eric Michael Smith, Stephen Roller, Megan Ung, Moya Chen, Kushal Arora, Joshua Lane, et al. 2022. Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage. *arXiv preprint arXiv:2208.03188*.

Eric Michael Smith, Mary Williamson, Kurt Shuster, Jason Weston, and Y-Lan Boureau. 2020. Can you put it all together: Evaluating conversational agents' ability to blend skills. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2021–2030. Association for Computational Linguistics.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

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

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, Toronto, Canada. Association for Computational Linguistics.

Oyvind Tafjord, Peter Clark, Matt Gardner, Wen-tau Yih, and Ashish Sabharwal. 2019. 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*, pages 7063–7071. AAAI Press.

Oyvind Tafjord, Bhavana Dalvi Mishra, and Peter Clark. 2020. ProofWriter: Generating implications, proofs, and abductive statements over natural language. *arXiv preprint arXiv:2012.13048*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technolo-*

gies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pages 4149–4158. Association for Computational Linguistics.

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

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul R. Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, Alexandre Frechette, Charlotte Smith, Laura Culp, Lev Proleev, Yi Luan, Xi Chen, James Lottes, Nathan Schucher, Federico Lebron, Alban Rrustemi, Natalie Clay, Phil Crone, Tomas Kocisky, Jeffrey Zhao, Bartek Perz, Dian Yu, Heidi Howard, Adam Bloniarz, Jack W. Rae, Han Lu, Laurent Sifre, Marcello Maggioni, Fred Alcober, Dan Garrette, Megan Barnes, Shantanu Thakoor, Jacob Austin, Gabriel Barth-Maron, William Wong, Rishabh Joshi, Rahma Chaabouni, Deeni Fatiha, Arun Ahuja, Ruibo Liu, Yunxuan Li, Sarah Cogan, Jeremy Chen, Chao Jia, Chenjie Gu, Qiao Zhang, Jordan Grimstad, Ale Jakse Hartman, Martin Chadwick, Gaurav Singh Tomar, Xavier Garcia, Evan Senter, Emanuel Taropa, Thanumalayan Sankaranarayana Pillai, Jacob Devlin, Michael Laskin, Diego de Las Casas, Dasha Valter, Connie Tao, Lorenzo Blanco, Adrià Puigdomènech Badia, David Reitter, Mianna Chen, Jenny Brennan, Clara Rivera, Sergey Brin, Shariq Iqbal, Gabriela Surita, Jane Labanowski, Abhi Rao, Stephanie Winkler, Emilio Parisotto, Yiming Gu, Kate Olszewska, Yujing Zhang, Ravi Addanki, Antoine Miech, Annie Louis, Laurent El Shafey, Denis Teplyashin, Geoff Brown, Elliot Catt, Nithya Attaluri, Jan Balaguer, Jackie Xiang, Pidong Wang, Zoe Ashwood, Anton Briukhov, Albert Webson, Sanjay Ganapathy, Smit Sanghavi, Ajay Kannan, Ming-Wei Chang, Axel Stjerngren, Josip Djolonga, Yuting Sun, Ankur Bapna, Matthew Aitchison, Pedram Pejman, Henryk Michalewski, Tianhe Yu, Cindy Wang, Juliette

Love, Junwhan Ahn, Dawn Bloxwich, Kehang Han, Peter Humphreys, Thibault Sellam, James Bradbury, Varun Godbole, Sina Samangooei, Bogdan Damoc, Alex Kaskasoli, Sébastien M. R. Arnold, Vijay Vasudevan, Shubham Agrawal, Jason Riesa, Dmitry Lepikhin, Richard Tanburn, Srivatsan Srinivasan, Hyeontaek Lim, Sarah Hodkinson, Pranav Shyam, Johan Ferret, Steven Hand, Ankush Garg, Tom Le Paine, Jian Li, Yujia Li, Minh Giang, Alexander Neitz, Zaheer Abbas, Sarah York, Machel Reid, Elizabeth Cole, Aakanksha Chowdhery, Dipanjan Das, Dominika Rogozińska, Vitaly Nikolaev, Pablo Sprechmann, Zachary Nado, Lukas Zilka, Flavien Prost, Luheng He, Marianne Monteiro, Gaurav Mishra, Chris Welty, Josh Newlan, Dawei Jia, Miltiadis Allamanis, Clara Huiyi Hu, Raoul de Liedekerke, Justin Gilmer, Carl Saroufim, Shruti Rijhwani, Shaobo Hou, Disha Shrivastava, Anirudh Baddepudi, Alex Goldin, Adnan Ozturel, Albin Cassirer, Yunhan Xu, Daniel Sohn, Devendra Sachan, Reinald Kim Amplayo, Craig Swanson, Dessie Petrova, Shashi Narayan, Arthur Guez, Siddhartha Brahma, Jessica Landon, Miteyan Patel, Ruizhe Zhao, Kevin Villela, Luyu Wang, Wenhao Jia, Matthew Rahtz, Mai Giménez, Legg Yeung, Hanzhao Lin, James Keeling, Petko Georgiev, Diana Mincu, Boxi Wu, Salem Haykal, Rachel Saputro, Kiran Vodrahalli, James Qin, Zeynep Cankara, Abhanshu Sharma, Nick Fernando, Will Hawkins, Behnam Neyshabur, Solomon Kim, Adrian Hutter, Priyanka Agrawal, Alex Castro-Ros, George van den Driessche, Tao Wang, Fan Yang, Shuo yiin Chang, Paul Komarek, Ross McIlroy, Mario Lučić, Guodong Zhang, Wael Farhan, Michael Sharman, Paul Natsev, Paul Michel, Yong Cheng, Yamini Bansal, Siyuan Qiao, Kris Cao, Siamak Shakeri, Christina Butterfield, Justin Chung, Paul Kishan Rubenstein, Shivani Agrawal, Arthur Mensch, Kedar Soparkar, Karel Lenc, Timothy Chung, Aedan Pope, Loren Maggiore, Jackie Kay, Priya Jhakra, Shibo Wang, Joshua Maynez, Mary Phuong, Taylor Tobin, Andrea Tacchetti, Maja Trebacz, Kevin Robinson, Yash Katariya, Sebastian Riedel, Paige Bailey, Kefan Xiao, Nimesh Ghelani, Lora Aroyo, Ambrose Slone, Neil Houlsby, Xuehan Xiong, Zhen Yang, Elena Gribovskaya, Jonas Adler, Mateo Wirth, Lisa Lee, Music Li, Thais Kagohara, Jay Pavagadhi, Sophie Bridgers, Anna Bortsova, Sanjay Ghemawat, Zafarali Ahmed, Tianqi Liu, Richard Powell, Vijay Bolina, Mariko Iinuma, Polina Zablotskaia, James Besley, Da-Woon Chung, Timothy Dozat, Ramona Comanescu, Xiance Si, Jeremy Greer, Guolong Su, Martin Polacek, Raphaël Lopez Kaufman, Simon Tokumine, Hexiang Hu, Elena Buchatskaya, Yingjie Miao, Mohamed Elhawaty, Aditya Siddhant, Nenad Tomasev, Jinwei Xing, Christina Greer, Helen Miller, Shereen Ashraf, Aurko Roy, Zizhao Zhang, Ada Ma, Angelos Filos, Mi-

los Besta, Rory Blevins, Ted Klimenko, Chih-Kuan Yeh, Soravit Changpinyo, Jiaqi Mu, Oscar Chang, Mantas Pajarskas, Carrie Muir, Vered Cohen, Charline Le Lan, Krishna Haridasan, Amit Marathe, Steven Hansen, Sholto Douglas, Rajkumar Samuel, Mingqiu Wang, Sophia Austin, Chang Lan, Jiepu Jiang, Justin Chiu, Jaime Alonso Lorenzo, Lars Lowe Sjösund, Sébastien Cevey, Zach Gleicher, Thi Avrahami, Anudhyan Boral, Hansa Srinivasan, Vittorio Selo, Rhys May, Konstantinos Aisopos, Léonard Hussenot, Livio Baldini Soares, Kate Baumli, Michael B. Chang, Adrià Recasens, Ben Caine, Alexander Pritzel, Filip Pavetic, Fabio Pardo, Anita Gergely, Justin Frye, Vinay Ramasesh, Dan Horgan, Kartikeya Badola, Nora Kassner, Subhrajit Roy, Ethan Dyer, Víctor Campos, Alex Tomala, Yunhao Tang, Dalia El Badawy, Elspeth White, Basil Mustafa, Oran Lang, Abhishek Jindal, Sharad Vikram, Zhitao Gong, Sergi Caelles, Ross Hemsley, Gregory Thornton, Fangxiaoyu Feng, Wojciech Stokowiec, Ce Zheng, Phoebe Thacker, Çağlar Ünlü, Zhishuai Zhang, Mohammad Saleh, James Svensson, Max Bileschi, Piyush Patil, Ankesh Anand, Roman Ring, Katerina Tsihlas, Arpi Vezer, Marco Selvi, Toby Shevlane, Mikel Rodriguez, Tom Kwiatkowski, Samira Daruki, Keran Rong, Allan Dafoe, Nicholas FitzGerald, Keren Gu-Lemberg, Mina Khan, Lisa Anne Hendricks, Marie Pellat, Vladimir Feinberg, James Cobon-Kerr, Tara Sainath, Maribeth Rauh, Sayed Hadi Hashemi, Richard Ives, Yana Hasson, YaGuang Li, Eric Noland, Yuan Cao, Nathan Byrd, Le Hou, Qingze Wang, Thibault Sottiaux, Michela Paganini, Jean-Baptiste Lespiau, Alexandre Moufarek, Samer Hassan, Kaushik Shivakumar, Joost van Amersfoort, Amol Mandhane, Pratik Joshi, Anirudh Goyal, Matthew Tung, Andrew Brock, Hannah Sheahan, Vedant Misra, Cheng Li, Nemanja Rakićević, Mostafa Dehghani, Fangyu Liu, Sid Mittal, Junhyuk Oh, Seb Noury, Eren Sezener, Fantine Huot, Matthew Lamm, Nicola De Cao, Charlie Chen, Gamaleldin Elsayed, Ed Chi, Mahdis Mahdieh, Ian Tenney, Nan Hua, Ivan Petrychenko, Patrick Kane, Dylan Scandinaro, Rishub Jain, Jonathan Uesato, Romina Datta, Adam Sadovsky, Oskar Bunyan, Dominik Rabiej, Shimu Wu, John Zhang, Gautam Vasudevan, Edouard Leurent, Mahmoud Alnahlawi, Ionut Georgescu, Nan Wei, Ivy Zheng, Betty Chan, Pam G Rabinovitch, Piotr Stanczyk, Ye Zhang, David Steiner, Subhajit Naskar, Michael Azzam, Matthew Johnson, Adam Paszke, Chung-Cheng Chiu, Jaume Sanchez Elias, Afroz Mohiuddin, Faizan Muhammad, Jin Miao, Andrew Lee, Nino Vieillard, Sahitya Potluri, Jane Park, Elnaz Davoodi, Jiageng Zhang, Jeff Stanway, Drew Garmon, Abhijit Karmarkar, Zhe Dong, Jong Lee, Aviral Kumar, Luowei Zhou, Jonathan Evens, William Isaac, Zhe Chen, Johnson Jia, Anselm Levskaya, Zhenkai Zhu, Chris Gorgolewski, Peter

Grabowski, Yu Mao, Alberto Magni, Kaisheng Yao, Javier Snaider, Norman Casagrande, Paul Suganthan, Evan Palmer, Geoffrey Irving, Edward Loper, Manaal Faruqui, Isha Arkatkar, Nanxin Chen, Izhak Shafran, Michael Fink, Alfonso Castaño, Irene Giannoumis, Wooyeol Kim, Mikołaj Rybiński, Ashwin Sreevatsa, Jennifer Prendki, David Soergel, Adrian Goedeckemeyer, Willi Gierke, Mohsen Jafari, Meenu Gaba, Jeremy Wiesner, Diana Gage Wright, Yawen Wei, Harsha Vashisht, Yana Kulizhskaya, Jay Hoover, Maigo Le, Lu Li, Chimezie Iwuanyanwu, Lu Liu, Kevin Ramirez, Andrey Khorlin, Albert Cui, Tian LIN, Marin Georgiev, Marcus Wu, Ricardo Aguilar, Keith Pallo, Abhishek Chakladar, Alena Repina, Xihui Wu, Tom van der Weide, Priya Ponnapalli, Caroline Kaplan, Jiri Simsa, Shuangfeng Li, Olivier Dousse, Fan Yang, Jeff Piper, Nathan Ie, Minnie Lui, Rama Pasumarthi, Nathan Lintz, Anitha Vijayakumar, Lam Nguyen Thiet, Daniel Andor, Pedro Valenzuela, Cosmin Paduraru, Daiyi Peng, Katherine Lee, Shuyuan Zhang, Somer Greene, Duc Dung Nguyen, Paula Kurylowicz, Sarmishta Velury, Sebastian Krause, Cassidy Hardin, Lucas Dixon, Lili Janzer, Kiam Choo, Ziqiang Feng, Biao Zhang, Achintya Singhal, Tejasi Latkar, Mingyang Zhang, Quoc Le, Elena Allica Abellan, Dayou Du, Dan McKinnon, Natasha Antropova, Tolga Bolukbasi, Orgad Keller, David Reid, Daniel Finchelstein, Maria Abi Raad, Remi Crocker, Peter Hawkins, Robert Dadashi, Colin Gaffney, Sid Lall, Ken Franko, Egor Filonov, Anna Bulanova, Rémi Leblond, Vikas Yadav, Shirley Chung, Harry Askham, Luis C. Cobo, Kelvin Xu, Felix Fischer, Jun Xu, Christina Sorokin, Chris Alberti, Chu-Cheng Lin, Colin Evans, Hao Zhou, Alek Dimitriev, Hannah Forbes, Dylan Banarse, Zora Tung, Jeremiah Liu, Mark Omernick, Colton Bishop, Chintu Kumar, Rachel Sterneck, Ryan Foley, Rohan Jain, Swaroop Mishra, Jiawei Xia, Taylor Bos, Geoffrey Cideron, Ehsan Amid, Francesco Piccinno, Xingyu Wang, Praseem Banzal, Petru Gurita, Hila Noga, Premal Shah, Daniel J. Mankowitz, Alex Polozov, Nate Kushman, Victoria Krakovna, Sasha Brown, MohammadHossein Bateni, Dennis Duan, Vlad Firoiu, Meghana Thotakuri, Tom Natan, Anhad Mohananey, Matthieu Geist, Sidharth Mudgal, Sertan Girgin, Hui Li, Jiayu Ye, Ofir Roval, Reiko Tojo, Michael Kwong, James Lee-Thorp, Christopher Yew, Quan Yuan, Sumit Bagri, Danila Sinopalnikov, Sabela Ramos, John Mellor, Abhishek Sharma, Aliaksei Severyn, Jonathan Lai, Kathy Wu, Heng-Tze Cheng, David Miller, Nicolas Sonnerat, Denis Vnukov, Rory Greig, Jennifer Beattie, Emily Caveness, Libin Bai, Julian Eisenschlos, Alex Korchemniy, Tomy Tsai, Mimi Jasarevic, Weize Kong, Phuong Dao, Zeyu Zheng, Frederick Liu, Fan Yang, Rui Zhu, Mark Geller, Tian Huey Teh, Jason Sanmiya, Evgeny Glad-

chenko, Nejc Trdin, Andrei Sozanschi, Daniel Toyama, Evan Rosen, Sasan Tavakkol, Linting Xue, Chen Elkind, Oliver Woodman, John Carpenter, George Papamakarios, Rupert Kemp, Sushant Kafle, Tanya Grunina, Rishika Sinha, Alice Talbert, Abhimanyu Goyal, Diane Wu, Denese Owusu-Afriyie, Cosmo Du, Chloe Thornton, Jordi Pont-Tuset, Pradyumna Narayana, Jing Li, Sabaer Fatehi, John Wieting, Omar Ajmeri, Benigno Uria, Tao Zhu, Yeongil Ko, Laura Knight, Amélie Héliou, Ning Niu, Shane Gu, Chenxi Pang, Dustin Tran, Yeqing Li, Nir Levine, Ariel Stolovich, Norbert Kalb, Rebeca Santamaria-Fernandez, Sonam Goenka, Wenny Yustalim, Robin Strudel, Ali Elqursh, Balaji Lakshminarayanan, Charlie Deck, Shyam Upadhyay, Hyo Lee, Mike Dusenberry, Zonglin Li, Xuezhi Wang, Kyle Levin, Raphael Hoffmann, Dan Holtmann-Rice, Olivier Bachem, Summer Yue, Sho Arora, Eric Malmi, Daniil Mirylenka, Qijun Tan, Christy Koh, Soheil Hassas Yeganeh, Siim Põder, Steven Zheng, Francesco Pongetti, Mukarram Tariq, Yanhua Sun, Lucian Ionita, Mojtaba Seyedhosseini, Pouya Tafti, Ragha Kotikalapudi, Zhiyu Liu, Anmol Gulati, Jasmine Liu, Xinyu Ye, Bart Chrzaszcz, Lily Wang, Nikhil Sethi, Tianrun Li, Ben Brown, Shreya Singh, Wei Fan, Aaron Parisi, Joe Stanton, Chenkai Kuang, Vinod Koverkathu, Christopher A. Choquette-Choo, Yunjie Li, TJ Lu, Abe Ittycheriah, Prakash Shroff, Pei Sun, Mani Varadarajan, Sanaz Bahargam, Rob Willoughby, David Gaddy, Ishita Dasgupta, Guillaume Desjardins, Marco Cornero, Brona Robenek, Bhavishya Mittal, Ben Albrecht, Ashish Shenoy, Fedor Moiseev, Henrik Jacobsson, Alireza Ghaffarkhah, Morgane Rivière, Alanna Walton, Clément Crepy, Alicia Parrish, Yuan Liu, Zongwei Zhou, Clement Farabet, Carey Radebaugh, Praveen Srinivasan, Claudia van der Salm, Andreas Fidjeland, Salvatore Scellato, Eri Latorre-Chimoto, Hanna Klimczak-Plucińska, David Bridson, Dario de Cesare, Tom Hudson, Piermaria Mendolicchio, Lexi Walker, Alex Morris, Ivo Penchev, Matthew Mauger, Alexey Guseynov, Alison Reid, Seth Odoom, Lucia Loher, Victor Cotruta, Madhavi Yenugula, Dominik Grewe, Anastasia Petrushkina, Tom Duerig, Antonio Sanchez, Steve Yadlowsky, Amy Shen, Amir Globerson, Adam Kurzrok, Lynette Webb, Sahil Dua, Dong Li, Preethi Lahoti, Surya Bhupatiraju, Dan Hurt, Haroon Qureshi, Ananth Agarwal, Tomer Shani, Matan Eyal, Anuj Khare, Shreyas Rammohan Belle, Lei Wang, Chetan Tekur, Mihir Sanjay Kale, Jinliang Wei, Ruoxin Sang, Brennan Saeta, Tyler Liechty, Yi Sun, Yao Zhao, Stephan Lee, Pandu Nayak, Doug Fritz, Manish Reddy Vuyyuru, John Aslanides, Nidhi Vyas, Martin Wicke, Xiao Ma, Taylan Bilal, Evgenii Eltyshev, Daniel Balle, Nina Martin, Hardie Cate, James Manyika, Keyvan Amiri, Yelin Kim, Xi Xiong, Kai Kang, Florian Luisier, Nilesh

Tripuraneni, David Madras, Mandy Guo, Austin Waters, Oliver Wang, Joshua Ainslie, Jason Baldridge, Han Zhang, Garima Pruthi, Jakob Bauer, Feng Yang, Riham Mansour, Jason Gelman, Yang Xu, George Polovets, Ji Liu, Honglong Cai, Warren Chen, XiangHai Sheng, Emily Xue, Sherjil Ozair, Adams Yu, Christof Angermueller, Xiaowei Li, Weiren Wang, Julia Wiesinger, Emmanouil Koukoumidis, Yuan Tian, Anand Iyer, Madhu Gurumurthy, Mark Goldenson, Parashar Shah, MK Blake, Hongkun Yu, Anthony Urbanowicz, Jennimaria Palomaki, Chrisantha Fernando, Kevin Brooks, Ken Durden, Harsh Mehta, Nikola Momchev, Elahe Rahimtoroghi, Maria Georgaki, Amit Raul, Sebastian Ruder, Morgan Redshaw, Jinhyuk Lee, Komal Jalan, Dinghua Li, Ginger Perng, Blake Hechtman, Parker Schuh, Milad Nasr, Mia Chen, Kieran Milan, Vladimir Mikulik, Trevor Strohman, Juliana Franco, Tim Green, Demis Hassabis, Koray Kavukcuoglu, Jeffrey Dean, and Oriol Vinyals. 2023. Gemini: A family of highly capable multimodal models.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. 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)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Kushal Tirumala, Aram H. Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. 2022. Memorization without overfitting: Analyzing the training dynamics of large language models. In *NeurIPS*.

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. 2023a. Llama: Open and efficient foundation language models.

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. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, Vancouver, Canada. Association for Computational Linguistics.

Jack Urbanek, Angela Fan, Siddharth Karamcheti, Saachi Jain, Samuel Humeau, Emily Dinan, Tim Rocktäschel, Douwe Kiela, Arthur Szlam, and Jason Weston. 2019. Learning to speak and act in a fantasy text adventure game. 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)*, pages 673–683, Hong Kong, China. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Tu Vu, Tong Wang, Tsendsuren Munkhdalai, Alessandro Sordoni, Adam Trischler, Andrew Mattarella-Micke, Subhransu Maji, and Mohit Iyyer. 2020. Exploring and predicting transferability across nlp tasks. In *EMNLP*.

Boxin Wang, Wei Ping, Lawrence McAfee, Peng Xu, Bo Li, Mohammad Shoeybi, and Bryan Catanzaro. 2023a. Instructretro: Instruction tuning post retrieval-augmented pretraining.

Cunxiang Wang, Shuailong Liang, Yue Zhang, Xiaonan Li, and Tian Gao. 2019. Does it make sense? and

why? a pilot study for sense making and explanation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4020–4026.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023b. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13484–13508. Association for Computational Linguistics.

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. 2022a. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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 Gary 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. 2022b. Super-naturalinstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 5085–5109. Association for Computational Linguistics.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M.

Dai, and Quoc V Le. 2022a. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022b. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022c. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

Peter West, Chandra Bhagavatula, Jack Hessel, Jena Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. 2022. Symbolic knowledge distillation: from general language models to commonsense models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4602–4625, Seattle, United States. Association for Computational Linguistics.

Genta Indra Winata, Andrea Madotto, Zhaojiang Lin, Rosanne Liu, Jason Yosinski, and Pascale Fung. 2021. Language models are few-shot multilingual learners. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 1–15, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I. Wang, Victor Zhong, Bailin Wang, Chengzu Li, Connor Boyle, Ansong Ni, Ziyu Yao, Dragomir Radev, Caiming Xiong, Lingpeng Kong, Rui Zhang, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2022. UnifiedSKG: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. In *Proceedings of the 2022 Conference on Empirical Meth-*

ods in Natural Language Processing, pages 602–631, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

Jing Xu, Arthur Szlam, and Jason Weston. 2022. Beyond goldfish memory: Long-term open-domain conversation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 5180–5197. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal. Association for Computational Linguistics.

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

Qinyuan Ye, Bill Yuchen Lin, and Xiang Ren. 2021. CrossFit: A few-shot learning challenge for cross-task generalization in NLP. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7163–7189, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wenpeng Yin, Jia Li, and Caiming Xiong. 2022. Contintin: Continual learning from task instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1:*

Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022, pages 3062–3072. Association for Computational Linguistics.

Lili Yu, Bowen Shi, Ramakanth Pasunuru, Benjamin Muller, Olga Golovneva, Tianlu Wang, Arun Babu, Binh Tang, Brian Karrer, Shelly Sheynin, Candace Ross, Adam Polyak, Russell Howes, Vasu Sharma, Puxin Xu, Hovhannes Tamoyan, Oron Ashual, Uriel Singer, Shang-Wen Li, Susan Zhang, Richard James, Gargi Ghosh, Yaniv Taigman, Maryam Fazel-Zarandi, Asli Celikyilmaz, Luke Zettlemoyer, and Armen Aghajanyan. 2023. Scaling autoregressive multi-modal models: Pretraining and instruction tuning.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Hongming Zhang, Xinran Zhao, and Yangqiu Song. 2020. Winowhy: A deep diagnosis of essential commonsense knowledge for answering winograd schema challenge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5736–5745.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 15–20, New Orleans, Louisiana. Association for Computational Linguistics.

Mengjie Zhao and Hinrich Schütze. 2021. Discrete and soft prompting for multilingual models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8547–8555, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ruiqi Zhong, Kristy Lee, Zheng Zhang, and Dan Klein. 2021. Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2856–2878, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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. 2023. Lima: Less is more for alignment.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*.