

Self-Training Large Language Models for Tool-Use Without Demonstrations

Ne Luo¹ Aryo Pradipta Gema¹ Xuanli He² Emile van Krieken¹

Pietro Lesci³ Pasquale Minervini^{1,4}

¹University of Edinburgh ²University College London ³University of Cambridge ⁴Miniml.AI
n.luo-5@sms.ed.ac.uk p.minervini@ed.ac.uk

Abstract

Large language models (LLMs) remain prone to factual inaccuracies and computational errors, including hallucinations and mistakes in mathematical reasoning. Recent work augmented LLMs with tools to mitigate these shortcomings, but often requires curated gold tool-use demonstrations. In this paper, we investigate whether LLMs can learn to use tools *without* demonstrations. First, we analyse zero-shot prompting strategies to guide LLMs in tool utilisation. Second, we propose a self-training method to synthesise tool-use traces using the LLM itself. We compare supervised fine-tuning and preference fine-tuning techniques for fine-tuning the model on datasets constructed using existing Question Answering (QA) datasets, *i.e.*, TriviaQA and GSM8K. Experiments show that tool-use enhances performance on a long-tail knowledge task: 3.7% on PopQA, which is used solely for evaluation, but leads to mixed results on other datasets, *i.e.*, TriviaQA, GSM8K, and NQ-Open. Our findings highlight the potential and challenges of integrating external tools into LLMs without demonstrations.¹

1 Introduction

Large language models (LLMs) have shown state-of-the-art performance in many natural language processing tasks and demonstrated “emergent abilities”: the capability to perform new tasks without updating their parameters via gradient descent (Brown et al., 2020; Scao et al., 2023; Chowdhery et al., 2023; Touvron et al., 2023). Specifically, by simply being provided with task instructions, sometimes supplemented with a few examples, LLMs can achieve comparable performance to fine-tuning-based methods (Brown et al., 2020; Wei et al., 2022; Akyürek et al., 2023; Von Oswald et al., 2023). Despite the remarkable performance, LLMs may still generate inaccurate or unfactual

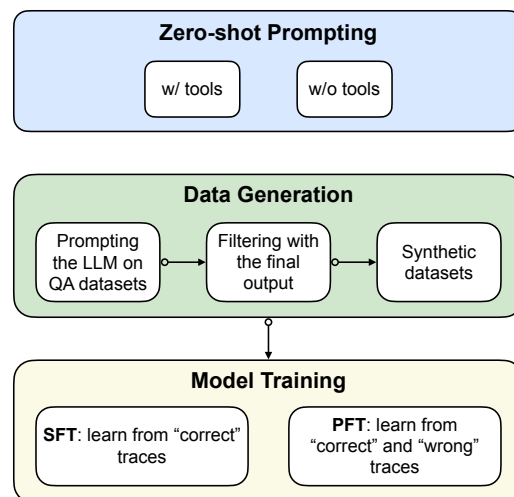


Figure 1: The overall workflow of our work. SFT: supervised fine-tuning; PFT: preference fine-tuning.

texts, *i.e.*, hallucinations (Lewis et al., 2020; Hong et al., 2024) or perform incorrect computations (Gao et al., 2023) if they solely rely on their internal parametric knowledge. Motivated by these shortcomings, we explore approaches that augment LLMs with external tools (Mialon et al., 2023)—such as a calculator or a search engine—to enhance their reasoning and problem-solving abilities.

Augmenting LLMs with tools has become an active research area in recent years (Mialon et al., 2023; Schick et al., 2023; Qin et al., 2024; Li et al., 2024). Current methods primarily follow two approaches (Wang et al., 2024b): (i) **prompting**, which leverages the in-context learning ability of large-scale models (Yao et al., 2023; Lu et al., 2023); and (ii) **fine-tuning**, with a primary focus on Supervised Fine-Tuning (SFT), which trains LLMs on datasets of tool-use examples. These datasets are typically sampled from large-scale data (Schick et al., 2023) or generated from more advanced LLMs, such as ChatGPT (Yang et al., 2023;

¹Code available at github.com/neneluo/llm-tool-use.

Qin et al., 2024; Li et al., 2024). While these approaches are highly effective and demonstrate impressive performance, they are resource-intensive and challenging to generalise.

Additionally, SFT with tool-use datasets alone can be suboptimal, as tool-use for LLMs is an inherently open-ended task: there is no oracle tool-use trace that specifies a single, unique solution to a given problem with specific tools. Many tools share overlapping functionality, and each tool can be used in multiple ways. Also, the helpfulness of tools depends on the LLM itself (Schick et al., 2023). This led us to consider optimising the model in a way that better captures the open-ended feature of tool-use, where Preference Fine-Tuning (PFT) generally helps. While concurrent work applied PFT to augment the LLMs’ ability in specific domains, *e.g.*, as math agents (Xiong et al., 2025; Wang et al., 2024a), its potential in general tool-use scenario remains under-explored.

Inspired by these challenges, we explore whether LLMs can learn tool-use without demonstrations, leveraging different optimisation objectives; that is, we answer the question: *can we teach LLMs to use tools without oracle tool-use traces?*

First, we study zero-shot prompting approaches that utilise LLMs’ instruction-following ability. This serves as a baseline for the tool-use performance of LLMs without performing gradient updates. Second, we propose a self-training approach to synthesise datasets containing tool-use traces via the LLM itself, which could be used for model fine-tuning. To improve data quality, we further explore employing different filtering strategies based on the final output and additional criteria. Then, we explore different fine-tuning objectives: SFT and PFT. Figure 1 shows an overview of our work.

Our experimental results across multiple Question Answering (QA) datasets, including TriviaQA, GSM8K, NQ-Open and PopQA, show that zero-shot prompting alone enables LLMs to use tools to some extent but may lead to degraded performance, depending on the model scale and tasks. The proposed self-training method for tool-use, which trains the model on the synthetic datasets generated from TriviaQA and GSM8K, improves model performance on a long-tail knowledge task PopQA but yields mixed performance on other datasets. These findings suggest that the LLMs learn tool-use even without explicit demonstrations, but the performance gain is mainly shown when the model’s knowledge may be insufficient to solve a task,

while inappropriate tool-use can introduce additional challenges.

2 Background

In this section, we review two common methods used to fine-tune LLMs.

Supervised Fine-Tuning. SFT is performed after the pre-training of LLMs and generally enhances the model performance in specific downstream tasks, such as summarization, QA, etc (Brown et al., 2020). Instruction fine-tuning is a special form of SFT that aims to optimise the LLMs to follow human instructions, treating instruction following as a type of downstream task (Wei et al., 2022). Given an instruction fine-tuning dataset, such as SUP-NATINST (Wang et al., 2022), the training of the LLM under SFT optimises model parameters θ by minimising the negative log-likelihood loss \mathcal{L}_{SFT} , defined as follows:

$$\mathcal{L}_{\text{SFT}} = - \sum_{i=1}^n \log P(w_i | s, w_0, \dots, w_{i-1}; \theta), \quad (1)$$

where s is an instruction, w_i is the i -th token in the response, and n is the response length.

Preference Fine-Tuning. PFT is a post-training technique used in addition to SFT, aligning model responses with human preferences by fine-tuning LLMs on pairwise preference data. The method is initially introduced as Reinforcement Learning from Human Feedback (RLHF; Ouyang et al., 2022). Given an instruction, preference data are collected by annotating the human preference between two responses. Instead of strictly optimising LLMs to follow specific responses, LLMs are optimised to adhere to a policy of favouring human-preferred responses. In this way, LLMs learn to better capture the open-ended characteristics of conversations and better align with human values.

Rafailov et al. (2023) introduce Direct Preference Optimization (DPO), an RL-free algorithm that directly optimises the policy π_θ of LLMs by implicitly adjusting the likelihood of preferred and dispreferred responses. The DPO loss is derived by directly approximating the optimal policy according to the preference data, significantly simplifying the RLHF pipeline. Given the preference dataset \mathcal{D} , instruction x , preferred response y_p and dispre-

ferred response y_d , the DPO loss is defined as:

$$\mathcal{L}_{\text{DPO}} = - \mathbb{E}_{(x, y_p, y_d) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_{\theta}(y_p | x)}{\pi_{\text{REF}}(y_p | x)} - \beta \log \frac{\pi_{\theta}(y_d | x)}{\pi_{\text{REF}}(y_d | x)} \right) \right] \quad (2)$$

where σ is the sigmoid function, β controls the divergence between the policy π_{θ} to be optimised and the reference policy π_{REF} , a lower value indicates higher divergence.

3 Approach

In this section, we introduce the tools used to augment LLMs, and describe the procedure for generating tool-use datasets for model fine-tuning.

3.1 Tools

Inspired by the tool-use setting in (Schick et al., 2023; Li et al., 2024), we developed a set of tools that can be used by LLMs, including a calculator, a Wikipedia search engine, and a machine translator. The tools serve the purpose of aiding LLMs in various areas, such as mathematical calculation, real-world information retrieval, and low-resource language understanding.

Calculator. The calculator assists LLMs by producing accurate mathematical calculation results. It supports basic operations between numbers like addition, subtraction, etc. Given a mathematical formula, the output is the computation result.

Wikipedia Search Engine. The Wikipedia search engine assists LLMs in searching for relevant information from Wikipedia documents. The Wikipedia search engine was implemented with a BM25-based information retrieval model. Given a query, the information retrieval model retrieves the most relevant Wikipedia documents.²

Machine Translator. The machine translator is defined as the setting of translating low-resource languages into English, which can potentially aid LLMs in understanding low-resource languages. The machine translator was implemented with the open-source multilingual machine translation model No Language Left Behind (NLLB; Costa-Jussà et al., 2022), which supports 200 languages.

²The documents are from a pre-built Wikipedia dump wikipedia-kilt-doc with index version lucene-index.wikipedia-kilt-doc.20210421.f29307.tar.gz.

We used a distilled version³ for computational efficiency. Given a query, the machine translator outputs its corresponding English translation.

3.2 Synthesising Tool-use Dataset

Given an instruction fine-tuned LLM⁴, we generated synthetic tool-use datasets via the model itself through the following steps:

1. **Tools collection:** First, we created a tool pool by defining a set of functions that the model could utilise as tools. For each tool, we provided a concise usage description, showing the type of problem the tool can help with, as well as its input and output format.
2. **QA datasets collection:** Second, we collected some QA datasets from existing NLP datasets, *e.g.*, GSM8K, that were likely to benefit from tool integration. These datasets include questions that external tools, *e.g.*, calculator, could potentially improve the model’s ability to produce accurate answers.
3. **Data generation:** Third, we generated the synthetic dataset with the instruction fine-tuned LLM. Specifically, for each question from the QA dataset, we prompted the model with instructions that describe the available tools, encouraging the model to provide an answer that potentially utilises tools.
4. **Data filtering:** Then, we designed a data filtering process to ensure data quality. In this step, we used the correctness of the answer given a question to serve as a proxy for identifying “correct” tool-use traces, similar to Zelikman et al. (2022). If the model provided a correct final answer, we inferred the solution path is valid. The specific data filtering strategies for SFT and PFT are described in §3.3.

3.3 Learning to Use Tools

We describe the prompting approach we use to enable zero-shot tool-use and outline how we create synthetic datasets for fine-tuning LLMs through SFT and PFT.

³huggingface.co/facebook/nllb-200-distilled-600M.

⁴We used the Llama-3-8B-Instruct model as the LLM to synthesise data in the following experiments. The model is available at: huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct.

You are an advanced AI agent designed to answer questions. You can use your own knowledge to answer the questions, or use external tools to gather information before answering. However, you can only request the use of tools once. Answer in a few words. Let’s think step by step.

Respond in the following format:

Thought: decide whether to answer using your own knowledge or utilise external tools.

Action: specify the tool here using the format ‘Tool-Name[query]’ if you decide to use tools.

Rationale: justify your answer by providing intermediate reasoning steps for your answer, based either on your own knowledge or the received tool responses.

Answer: (1) if using your own knowledge, provide your answer here; (2) if using tools, leave this part empty until the tool’s response is received.

Below are the external tools you can use:

1. Calculator[query]: this tool helps you perform simple mathematical computations with real numbers. Use the formula as the input query, the tool response will be the result.
 2. WikipediaSearch[query]: this tool helps you search for information from Wikipedia. Use a short keyword as the input query, the tool response will be the corresponding information.
 3. MachineTranslator[query]: this tool helps you understand low-resource languages by translating them to English. Use the sentence you want to translate as the input query, the tool response will be the translation.
-

Table 1: System prompt for single-step tool-use.

Prompting. As the instruction fine-tuned LLMs were already optimised to follow human instructions, we hypothesised that the LLM could obtain the tool-use ability to some extent by solely learning from instructions, *i.e.*, prompts. We used the following prompt format: a short QA task description, an expected response format, and a brief description of the tools’ applicable domain. The task description suggests the goal of the task is to answer questions briefly. We employed zero-shot prompting without providing any tool-use examples and adopted the zero-shot Chain-of-Thought (CoT) method (Kojima et al., 2022). The LLMs’ response format design was inspired and adapted from (Yao et al., 2023; Li et al., 2024). Table 1 provides an example of the tool-use prompt that enables the LLM to perform single-step tool-use, either to request a single tool or multiple tools simultaneously. Variants of the prompt used for ablation studies are shown in Appendix F.

Supervised Fine-Tuning. For SFT, we experimented with two training data filtering strategies under the same data generation procedure (prompting with Table 1): (i) SFT (tools data): the training

data contains the instances from the training set of QA datasets where the model uses tools and provides a correct answer; and (ii) SFT (mixture data): the training data contains the instances where the model answers correctly regardless of whether or not the tool is used during the process. After data filtering, we re-constructed the full conversation history of the filtered cases to be the training data.

Preference Fine-Tuning. The triplet format data are required to form the PFT dataset: a prompt (the question), a preferred response and a dispreferred response. To generate this dataset, we conducted system inference on the training sets of QA datasets under two conditions: with and without the use of tools. We then filtered the generated data based on the following criteria: (i) the model provided the correct answer with access to tools (using the prompt in Table 1); (ii) the model provided the wrong answer for the same question without access to tools (using the prompt in Table 13). Then, we experimented with fine-tuning the model on tool-use data with DPO under two data format settings: (i) DPO (conversation): the preferred and dispreferred data encompass the entire conversation following the question; (ii) DPO (response): the preferred and dispreferred data are the model’s single response following the question.

4 Experimental Setup

In this section, we describe the datasets we use for experiments and detail our research questions.

4.1 Datasets

To effectively benchmark the tool-use LLMs, we employed datasets from two categories of tasks that could potentially benefit from tool integration. The detailed dataset statistics are shown in Appendix A.

Open-domain QA. We experimented with three open-domain QA datasets: TriviaQA (Joshi et al., 2017), Natural Question-Open (NQ-Open; Lee et al., 2019) and PopQA (Mallen et al., 2023). Questions in TriviaQA require trivia knowledge. NQ-Open contains real-world questions asked by actual Google Search users. PopQA consists of questions requiring knowledge of long-tail Wikipedia entities. Thus, these datasets can potentially benefit from the use of knowledge from external tools, *e.g.*, Wikipedia search engine and machine translator. We used NQ-Open and PopQA solely for model evaluation. For model training, we randomly sam-

pled a subset from the original training set for TriviaQA to match the scale of another dataset category. Similarly, we randomly selected portions from the original validation sets or test sets to be validation sets and test sets.

Mathematical Reasoning. We experimented with the Grade School Math dataset (GSM8K; Cobbe et al., 2021), a QA dataset composed of grade school math word problems, which could potentially benefit from the usage of a calculator. For our experiments, we used the original training set for model training and randomly split the original test set into the validation and test set in approximately equal proportions.

4.2 Experiments

In the following section, we break the main research question into three sub-research questions and conduct corresponding experiments:

- **RQ1 (§5.1):** Can we instruct LLMs to use tools without training them?
- **RQ2 (§5.2):** Can we further improve LLMs’ tool-use ability by conducting SFT in a self-training manner?
- **RQ3 (§5.3):** Can we use PFT to teach LLMs to use tools?

All experiments were conducted on the Llama 3 instruction fine-tuned models (Grattafiori et al., 2024). The model fine-tuning details on the synthetic datasets are shown in Appendix B. Inference details are shown in Appendix C. For evaluation, we employed two sets of automatic evaluation metrics to evaluate both the tool-use ability and generation quality of LLMs, which are defined in Appendix D.

5 Results and Analysis

In this section, we present the experimental results and analysis of approaches for teaching LLMs to use tools without demonstrations.

5.1 Prompting

The experimental results in Table 2 and Table 3 showed an initial effort to instruct LLMs to use tools by prompting the instruction fine-tuned Llama 3 models with different instructions. The detailed prompts are shown in Appendix F.

Prompt	Size	TriviaQA		GSM8K	
		EM	Acc	EM	Acc
No tool	8B	62.6	73.6	1.4	36.5
No tool + CoT	8B	52.8	77.9	8.6	66.9
Tools + Single-step	8B	56.9	75.8	22.0	64.2
- w/o Rationale	8B	35.7	72.1	2.9	56.6
Tools + Multi-step	8B	27.7	54.8	19.1	53.1
No tool	70B	70.4	87.8	3.1	66.0
No tool + CoT	70B	79.6	88.8	21.2	45.8
Tools + Single-step	70B	57.6	77.7	51.8	75.1

Table 2: Experimental results of Llama 3 instruction fine-tuned models on the validation sets in zero-shot setting given different prompts. EM: exact match; Acc: accuracy. The detailed prompts are shown in Appendix F. All numbers shown in the table are in percentages.

Prompt	Size	TriviaQA			GSM8K		
		IR	PR	AR	IR	PR	AR
Tools + Single-step	8B	14.5	99.3	55.2	7.2	72.3	12.8
- w/o Rationale	8B	29.2	99.7	55.5	13.5	75.0	13.6
Tools + Multi-step	8B	78.3	99.4	63.9	55.7	72.9	16.6
Tools + Single-step	70B	48.5	99.0	63.1	53.8	70.6	36.9

Table 3: Experimental results of Llama 3 instruction fine-tuned models on the validation sets in zero-shot setting given different prompts. IR: invoke rate; PR: pass rate; AR: answerable rate. The detailed prompts are shown in Appendix F. All numbers shown in the table are in percentages.

Results on Llama-3-8B-Instruct. When we prompted the LLM to answer questions with its own knowledge, zero-shot CoT prompting showed an improvement compared to the prompt of not using CoT, especially in the math dataset, *i.e.*, GSM8K, where the model benefited greatly from using explicit intermediate reasoning steps. Under the tool-use setting, results in Table 3 show that the LLM had a positive invoke rate and pass rate for tools, indicating that the LLM was aware of the existence of tools and knew how to use them to some extent. Table 4 shows an example that the model called the calculator correctly and yielded the right answer after getting responses from the tool, although we had not fine-tuned the LLM on tool-use datasets yet. However, we observed a performance drop in Table 2 when we allowed the model to use tools in both single-step and multi-step tool-use scenarios. This phenomenon indicates that the model can be harmed if it uses tools inappropriately. We provide extended analysis on the results of single-step tool-use in Appendix E.

NON-TOOL-USE EXAMPLE**User:**

Hannah slips on a banana peel and breaks her arm. The doctor charges her \$200 for the cast, \$300/hour for a 30-minute visit, \$4/pill for 30 painkillers, and \$6/hour for a mandatory minimum of 2 hours of parking. How much does the doctor's visit cost total?

Assistant:

Thought: Let's break down the costs into separate components and calculate the total.

Answer: \$200 (cast) + \$45 (30-minute visit) + \$120 (30 painkillers) + \$12 (2 hours of parking) = **\$477. (wrong answer)**

TOOL-USE EXAMPLE**User:**

Hannah slips on a banana peel and breaks her arm. The doctor charges her \$200 for the cast, \$300/hour for a 30-minute visit, \$4/pill for 30 painkillers, and \$6/hour for a mandatory minimum of 2 hours of parking. How much does the doctor's visit cost total?

Assistant:

Thought: I will use external tools to help me with the calculation.

Action: Calculator[(30*300)/60 + 200 + (30*4) + (6*2)]

Rationale: To calculate the total cost, I will first calculate the doctor's visit fee, which is 30 minutes * \$300/hour = \$150. Then, I will add the cost of the cast (\$200), the painkillers (\$4/pill * 30 = \$120), and the parking fee (\$6/hour * 2 hours = \$12).

Answer:

User:

Response from tool Calculator[(30*300)/60 + 200 + (30*4) + (6*2)] are: 482.0

Assistant:

Answer: **482.0 (correct answer)**

Table 4: An example of the model getting the right answer when using tools from the validation set of GSM8K.

Results on Llama-3-70B-Instruct. Results in Table 2 show the accuracy of the 70B model with tool-use dropped on TriviaQA and increased on GSM8K compared to not using tools. The invoke rates showed in Table 3 were much higher for the 70B model than the 8B model, indicating increased tool integration in larger models. In TriviaQA, the model shows a similar pattern as in the 8B model. However, in GSM8K, we notice that the 70B model showed abnormally lower accuracy compared to the 8B model in no tool CoT prompting, where the model was also asked to follow a specific answer format. This could be because the provided human-written prompts were suboptimal and had hurt the model's performance by constraining it from following the designed answer format, which aligns with prior research that constrained answer formats can negatively influence model performance (Tam et al., 2024). Therefore, the 70B model performed worse because its better instruction-following ability makes it more sensitive to prompts. The tools

appeared helpful for the 70B model in GSM8K, suggesting that even with prompting-based methods alone, tools can be beneficial for large models in complex reasoning tasks.

5.2 Supervised Fine-Tuning

SFT on Tools Data. For the model trained on tool-use data, the results in Table 5 show that the model performance on TriviaQA, GSM8K and NQ-Open has degraded. The results in Table 6 show that the tool invoke rates increased significantly after SFT, indicating that training on tool-use data made the model learn to use tools intensively. While SFT teaches the LLM the behaviour of using tools, simply using these tools does not directly correlate with improved answers. This could be because (i) misusing tools may hurt model performance; (ii) the LLM might already have the essential knowledge to answer most of the questions from these two datasets, as these are typically composed of popular knowledge that could have been learned from the pre-training process. This assumption can be supported by the strong performance of the model given the prompt that no tool is allowed from Table 5. Also, based on the results from GSM8K, where no tool-use was involved, we observe the LLM can perform simple mathematical calculations, though not entirely error-free. As a result, the helpfulness of the provided tools can be limited in TriviaQA, GSM8K and NQ-Open. However, the results on PopQA, which contains questions involving long-tail knowledge, showed a consistent increase after using tools and fine-tuning the model on tool-use data, suggesting that the model is likely to benefit from tools when questions are less likely to be answered by its internal knowledge alone.

SFT on Mixture Data. For the model trained on the mixture of tool-use and non-tool-use data, the results in Table 5 show the model outperformed the one fine-tuned on tool-use data in all datasets in terms of accuracy. The model also showed a minor accuracy improvement over the not fine-tuned system. However, the tool usage has significantly declined in TriviaQA, GSM8K and NQ-Open, and slightly decreased in the PopQA dataset. To determine whether the performance improvement was due to enhanced tool-use ability or simply training on a larger dataset, we conducted an additional experiment using only no tool data for training. We can observe that the accuracy of models trained with tool data on PopQA is higher than the model of

System	TriviaQA		GSM8K		NQ-Open		PopQA	
	EM	Acc	EM	Acc	EM	Acc	EM	Acc
Prompting (no tool)	52.8	77.9	8.6	66.9	11.6	40.3	18.0	31.6
Prompting (tools)	56.9	75.8	22.0	64.2	15.6	38.9	22.3	34.2
SFT (tools data)	35.0	65.9	16.6	48.6	10.1	37.7	18.1	35.0
SFT (mixture data)	56.1	75.9	17.5	64.6	14.0	40.3	22.4	35.7
SFT (no tool data)	56.8	77.6	17.8	64.6	15.0	39.5	22.3	33.6
DPO (conversation)	56.1	74.5	21.1	62.2	14.5	38.4	22.3	34.6
SFT + DPO	56.8	73.6	17.4	64.2	15.5	39.1	22.3	33.0
DPO (response)	56.4	75.4	18.2	62.9	14.4	37.8	22.6	35.1
DPO $\beta = 0.01$	54.8	73.3	19.1	64.0	14.7	38.0	23.6	35.6
DPO $\beta = 0.5$	53.9	73.0	17.7	64.2	14.9	38.7	23.0	35.6

Table 5: Experimental results of Llama-3-8B-Instruct models on the validation sets. EM: exact match; Acc: accuracy. Prompting (no tool): the non-tool-use CoT prompt (Table 14), other prompts: the single-step tool-use prompt (Table 1). The β (defined in eq. (2)) for DPO was set to 0.1 unless specified. All numbers shown in the table are in percentages.

System	TriviaQA			GSM8K			NQ-Open			PopQA		
	IR	PR	AR	IR	PR	AR	IR	PR	AR	IR	PR	AR
Prompting (tools)	14.5	99.3	55.2	7.2	72.3	12.8	16.5	98.8	41.2	36.2	99.4	37.8
SFT (tools data)	67.0	99.3	64.6	75.2	68.1	20.4	62.1	98.9	44.9	84.8	99.8	37.9
SFT (mixture data)	6.6	98.5	54.5	5.1	72.7	21.2	6.4	98.4	45.3	34.7	99.4	37.5
SFT (no tool data)	1.5	100.0	53.3	0.5	100.0	0.0	1.8	100.0	44.4	9.6	100.0	36.5
DPO (conversation)	15.7	99.4	55.4	7.8	62.7	15.7	19.2	100.0	50.5	33.8	99.7	37.6
SFT + DPO	15.1	99.3	57.6	9.5	74.2	19.4	18.5	100.0	46.5	33.5	99.7	37.0
DPO (response)	16.3	98.8	61.3	8.9	79.3	17.2	17.3	99.4	45.1	38.9	99.7	38.6
DPO $\beta = 0.01$	18.1	97.8	59.7	10.6	72.5	17.4	20.4	100.0	46.6	36.9	99.7	41.7
DPO $\beta = 0.5$	16.9	100.0	58.0	9.7	63.5	11.1	16.9	100.0	41.4	37.0	99.5	40.8

Table 6: Experimental results of Llama-3-8B-Instruct models on the validation sets. IR: invoke rate; PR: pass rate; AR: answerable rate. All prompts are the single-step tool-use prompt (Table 1). The β (defined in eq. (2)) for DPO was set to 0.1 unless specified. All numbers shown in the table are in percentages.

SFT on no tool data, suggesting the improvement stems from better tool-use ability, likely because the model has learned when to use tools.

5.3 Preference Fine-Tuning

PFT on tool-use Data with Conversation Format. From the experimental results of the model optimised with DPO on conversation format tool-use data, we can observe a performance decline in the TriviaQA, GSM8K and NQ-Open, and a marginal accuracy improvement in PopQA, which shows a pattern similar to the model trained with SFT on tool-use data: When we used DPO with the SFT model trained on the mixture of tool-use and non-tool-use data as the base model, the results did not improve in all datasets when we compare to basing DPO on the instruction fine-tuned model.

PFT on tool-use Data with Single Response Format. We can observe that PFT on a single response yields better results in terms of accuracy compared to the model fine-tuned on the full conversation in TriviaQA, GSM8K and PopQA. This

could be attributed to the fact that the DPO loss calculated on a whole conversation includes responses from tools that are redundant in the loss calculation, as we want to optimise the LLM’s behaviour. Setting the preference to a single response is also suboptimal, as this did not allow the model to learn how to answer a question based on the tool responses. Therefore, how to optimise the model in the multi-turn dialogues with PFT remains an open research question. We also conducted an ablation study on the hyperparameter β by exploring different values, showing that the choice of β slightly impacts model accuracy.

5.4 Results on the Test Sets

We evaluated the proposed approaches of teaching LLMs to use tools on the test sets of TriviaQA, GSM8K, NQ-Open and PopQA, and the experimental results are shown in Table 7 and Table 8. The SFT system was fine-tuned on the mixture of tool-use and non-tool-use data, and the PFT system was fine-tuned with tool-use data. For the PFT system,

System	TriviaQA		GSM8K		NQ-Open		PopQA	
	EM	Acc	EM	Acc	EM	Acc	EM	Acc
Prompting (no tool)	52.2	77.5	6.9	63.2	10.5	41.1	17.1	31.6
Prompting (tools)	56.0	75.0	17.8	58.4	13.8	38.3	21.2	33.1
SFT (mixture data)	56.0	78.9	14.9	61.3	12.2	40.6	20.3	35.3
PFT (tools data)	54.8	73.9	17.3	57.1	12.3	37.8	20.2	33.5

Table 7: Experimental results of Llama-3-8B-Instruct models on the test sets. EM: exact match; Acc: accuracy. Prompting (no tool): the non-tool-use CoT prompt (Table 14), other prompts: the single-step tool-use prompt (Table 1). All numbers shown in the table are in percentages.

System	TriviaQA			GSM8K			NQ-Open			PopQA		
	IR	PR	AR	IR	PR	AR	IR	PR	AR	IR	PR	AR
Prompting (tools)	14.5	98.6	53.1	10.3	69.6	13.0	16.9	100.0	38.5	37.0	98.1	35.4
SFT (mixture data)	7.3	97.3	50.7	4.6	64.5	25.8	7.9	98.7	39.2	35.5	98.9	30.1
PFT (tools data)	17.3	98.3	57.8	11.2	65.3	17.3	18.5	100.0	41.1	38.0	98.9	32.4

Table 8: Experimental results of Llama-3-8B-Instruct models on the test sets. IR: invoke rate; PR: pass rate; AR: answerable rate. All prompts are the single-step tool-use prompt (Table 1). All numbers shown in the table are in percentages.

the data was constructed with the single response setting and trained with $\beta = 0.5$ for DPO. The experimental results mostly showed a similar pattern to those on the validation sets, where the accuracy of tool-use systems on the TriviaQA, GSM8K, and NQ-Open was lower than the non-tool-use system, and the accuracy on the PopQA was better, except for the SFT system. Among the methods of teaching LLMs to use tools, the model trained with SFT showed the best accuracy, suggesting that SFT is a reasonable method to approach the task when using a limited number and variety of tools. The model trained with PFT showed better accuracy in PopQA than the prompting-based system under our experimental setting, showing the potential of using PFT to teach LLMs to use tools.

6 Related work

Prompting LLMs to Use Tools. One line of research focused on investigating prompting-based methods to teach LLMs to use tools by providing tool documentation (Hsieh et al., 2023) or tool descriptions and few-shot examples, *e.g.*, ReAct (Yao et al., 2023), Chameleon (Lu et al., 2023), Hugging-GPT (Shen et al., 2023), etc. In these work, large-scale models, such as PaLM-540B (Chowdhery et al., 2023) and ChatGPT (OpenAI, 2023), were prompted to use tools. These studies suggested the feasibility and benefits of integrating LLMs with external tools. However, a gap remains in exploring whether a smaller model can effectively learn to

use tools from prompting. Compared to prior work, our work evaluated the effectiveness of prompting LLMs across different scales to use tools.

SFT for Tool Learning. Another line of research applied fine-tuning-based methods to teach smaller models to use tools with curated tool-use datasets. Toolformer (Schick et al., 2023) utilised the few-shot in-context learning ability of LLMs to generate tool-use datasets by sampling on the pre-training data and then applied data filtering. In other work where pre-training data of LLMs were inaccessible, they mainly employed more advanced LLMs, such as ChatGPT, as a teacher model to synthesise tool-use datasets and conducted supervised fine-tuning on the collected datasets (*e.g.*, ToolLLaMA (Qin et al., 2024), Gorilla (Patil et al., 2023), GPT4Tools (Yang et al., 2023), *inter alia*). In contrast, our work began with zero-shot prompting and then leveraged tool-use datasets generated by the model itself, thereby alleviating the need for accessing tool-use examples.

RLHF and Tool Learning. The intersection between RLHF and tool learning is a promising yet under-explored area. TARM (Li et al., 2024) showed augmenting the Reward Model (RM) in RLHF with tools enhances the agreement of RM and human judgement. TRICE (Qiao et al., 2024) leveraged tool execution feedback with reinforcement learning for tool learning to mitigate the problem of tool misuse adversely influencing model

performance. However, an advanced LLM was still employed to synthesise tool-use datasets. Some concurrent work explored applying preference fine-tuning methods, *e.g.*, DPO and its variant, on learning to use tools to improve mathematical reasoning ability of LLMs (Xiong et al., 2025; Wang et al., 2024a), showcasing the benefit of utilising preference to guide model behaviour. Our work differs from these works in two aspects: (i) our work alleviates the reliance on tool-use datasets synthesised from advanced LLMs; (ii) our work explores a more comprehensive fine-tuning framework for tool learning across a broader range of tasks.

7 Conclusion

In this work, we studied methods for teaching LLMs to use tools without demonstrations. First, we explored teaching LLMs to use tools solely from instruction. Then, we proposed a self-training approach to synthesise datasets containing tool-use traces by instructing the LLM to use tools on two existing QA datasets, *i.e.*, TriviaQA and GSM8K, and applying filtering strategies to improve data quality based on the final output and additional criteria. We then investigated methods to improve LLM’s tool-use ability by fine-tuning the model with the synthetic datasets. Starting from the standard SFT objective, we then studied an under-explored approach for teaching LLMs to use tools: PFT. Experimental results suggest that proposed approaches are feasible for teaching LLMs to use tools. However, while tool-use enhances the performance of LLMs on a long-tail QA dataset, *i.e.*, PopQA, it leads to mixed results on other datasets, *i.e.*, TriviaQA, GSM8K and NQ-Open.

Limitations

This work has several limitations. First, we employed a limited number of tools. Although we believe the selected tools are representative of real-world applications, as they have a relatively large action space, the generalisation of tools across various domains remains a significant research topic, which could be investigated in future work by including a broader range of tools. Second, the current tool-use dataset size in our experiments is relatively small. Although the results show the potential of using PFT to teach LLMs to use tools, future work could benefit from exploring a larger and more diverse training set, better ways of constructing training data and better loss estimation

methods to fully release the power of DPO and further verify its effectiveness. Third, the self-training method, *i.e.*, using the data generated by the models themselves to improve them, typically contains multiple iterations. While we only experimented with the first iteration, future work could potentially benefit from the multiple-iteration setting.

Ethics Statement

This work generally does not raise ethical concerns. The proposed approaches for augmenting LLMs with external tools could potentially reduce the risk of LLMs generating inaccurate information. However, there remains a possibility of potential misuse by malicious individuals using this method to enable LLMs to interact with tools for harmful purposes.

Acknowledgement

Aryo Pradipta Gema was supported by the United Kingdom Research and Innovation (grant EP/S02431X/1), UKRI Centre for Doctoral Training in Biomedical AI at the University of Edinburgh, School of Informatics. Xuanli He was funded by an industry grant from Cisco. Emile van Krieken was funded by ELIAI (The Edinburgh Laboratory for Integrated Artificial Intelligence), EPSRC (grant no. EP/W002876/1). Pietro Lesci was funded by the European Research Council (ERC) under the European Union’s Horizon 2020 Research and Innovation programme grant AVeriTeC (Grant agreement No. 865958). Pasquale Minervini was partially funded by ELIAI, an industry grant from Cisco, and a donation from Accenture LLP. This work was supported by the Edinburgh Compute and Data Facility (ECDF), the Edinburgh International Data Facility (EIDF) and the Data-Driven Innovation Programme at the University of Edinburgh.

References

- Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2023. [What learning algorithm is in-context learning? Investigations with linear models](#). In *The Eleventh International Conference on Learning Representations*.
- 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.
- 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 Sankaranarayanan Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2023. [PaLM: Scaling language modeling with pathways](#). *Journal of Machine Learning Research*, 24(240):1–113.
- 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. 2021. [Training verifiers to solve math word problems](#). *arXiv preprint 2110.14168*.
- Marta R. Costa-Jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. [No language left behind: Scaling human-centered machine translation](#). *arXiv preprint 2207.04672*.
- Tri Dao. 2024. [FlashAttention-2: Faster attention with better parallelism and work partitioning](#). In *The Twelfth International Conference on Learning Representations*.
- Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. [PAL: Program-aided language models](#). In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 10764–10799. PMLR.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Alonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kam-badur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seo-

- hyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Rappathy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gouget, Virginie Do, Vish Vogeti, Vitor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkan Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippas Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabza, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. [The Llama 3 herd of models](#). *arXiv preprint 2407.21783*.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. [The curious case of neural text de-generation](#). In *International Conference on Learning Representations*.
- Giwon Hong, Aryo Pradipta Gema, Rohit Saxena, Xiaotang Du, Ping Nie, Yu Zhao, Laura Perez-Beltrachini, Max Ryabinin, Xuanli He, Clémentine Fourier, and Pasquale Minervini. 2024. [The halluci-](#)

- nations leaderboard: An open effort to measure hallucinations in large language models. *arXiv preprint 2404.05904*.
- Cheng-Yu Hsieh, Si-An Chen, Chun-Liang Li, Yasuhisa Fujii, Alexander Ratner, Chen-Yu Lee, Ranjay Krishna, and Tomas Pfister. 2023. [Tool documentation enables zero-shot tool-usage with large language models](#). *arXiv preprint 2308.00675*.
- Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *International Conference on Learning Representations*.
- 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.
- Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. [Large language models are zero-shot reasoners](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.
- Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. [Latent retrieval for weakly supervised open domain question answering](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. [Retrieval-augmented generation for knowledge-intensive NLP tasks](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.
- Lei Li, Yekun Chai, Shuohuan Wang, Yu Sun, Hao Tian, Ningyu Zhang, and Hua Wu. 2024. [Tool-augmented reward modeling](#). In *The Twelfth International Conference on Learning Representations*.
- Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. [Lost in the middle: How language models use long contexts](#). *Transactions of the Association for Computational Linguistics*, 12:157–173.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *International Conference on Learning Representations*.
- Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. 2023. [Chameleon: Plug-and-play compositional reasoning with large language models](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 43447–43478. Curran Associates, Inc.
- 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 memories](#). 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.
- Grégoire Mialon, Roberto Dessi, Maria Lomeli, Christoforos Nalmpantis, Ramakanth Pasunuru, Roberta Raileanu, Baptiste Roziere, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, Edouard Grave, Yann LeCun, and Thomas Scialom. 2023. [Augmented language models: A survey](#). *Transactions on Machine Learning Research*.
- OpenAI. 2023. [GPT-4 technical report](#).
- OpenAI. 2024. [Chatgpt \(gpt-4o\)](#).
- 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. 2022. [Training language models to follow instructions with human feedback](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.
- Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. 2023. [Gorilla: Large language model connected with massive apis](#). *arXiv preprint 2305.15334*.
- Shuofei Qiao, Honghao Gui, Chengfei Lv, Qianghuai Jia, Huajun Chen, and Ningyu Zhang. 2024. [Making language models better tool learners with execution feedback](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3550–3568, Mexico City, Mexico. Association for Computational Linguistics.
- Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, dahai li, Zhiyuan Liu, and Maosong Sun. 2024. [ToolLLM: Facilitating large language models to master 16000+ real-world APIs](#). In *The Twelfth International Conference on Learning Representations*.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. [Direct preference optimization: Your language model is secretly a reward model](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc.

Teven Le Scao, Angela Fan, Christopher Akiki, Elie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klammer, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, Dragomir Radev, Eduardo González Ponferrada, Efrat Levkovich, Ethan Kim, Eyal Bar Natan, Francesco De Toni, Gérard Dupont, Germán Kruszewski, Giada Pistilli, Hady Elsahar, Hamza Benyamina, Hieu Tran, Ian Yu, Idris Abdulmumin, Isaac Johnson, Itziar Gonzalez-Dios, Javier de la Rosa, Jenny Chim, Jesse Dodge, Jian Zhu, Jonathan Chang, Jörg Froberg, Joseph Tobing, Joydeep Bhattacharjee, Khalid Almubarak, Kimbo Chen, Kyle Lo, Leandro Von Werra, Leon Weber, Long Phan, Loubna Ben allal, Ludovic Tanguy, Manan Dey, Manuel Romero Mu noz, Maraim Masoud, María Grandury, Mario Šaško, Max Huang, Maximin Coavoux, Mayank Singh, Mike Tian-Jian Jiang, Minh Chien Vu, Mohammad A. Jauhar, Mustafa Ghaleb, Nishant Subramani, Nora Kassner, Nurulqilla Khamis, Olivier Nguyen, Omar Espejel, Ona de Gibert, Paulo Villegas, Peter Henderson, Pierre Colombo, Priscilla Amuok, Quentin Lhoest, Rhea Harliman, Rishi Bommasani, Roberto Luis López, Rui Ribeiro, Salomey Osei, Sampo Pyysalo, Sebastian Nagel, Shamik Bose, Shamsuddeen Hassan Muhammad, Shanya Sharma, Shayne Longpre, So-maieh Nikpoor, Stanislav Silberberg, Suhas Pai, Sydney Zink, Tiago Timponi Torrent, Timo Schick, Tristan Thrush, Valentin Danchev, Vassilina Nikoulina, Veronika Laippala, Violette Lepercq, Vrinda Prabhu, Zaid Alyafeai, Zeerak Talat, Arun Raja, Benjamin Heinzerling, Chenglei Si, Davut Emre Taşar, Elizabeth Salesky, Sabrina J. Mielke, Wilson Y. Lee, Abheesht Sharma, Andrea Santilli, Antoine Chaffin, Arnaud Stiegler, Debajyoti Datta, Eliza Szczechla, Gunjan Chhablani, Han Wang, Harshit Pandey, Hendrik Strobelt, Jason Alan Fries, Jos Rozen, Leo Gao, Lintang Sutawika, M Saiful Bari, Maged S. Al-shaibani, Matteo Manica, Nihal Nayak, Ryan Teehan, Samuel Albanie, Sheng Shen, Srulik Ben-David, Stephen H. Bach, Taewoon Kim, Tali Bers, Thibault Fevry, Trishala Neeraj, Urmish Thakker, Vikas Raunak, Xiangru Tang, Zheng-Xin Yong, Zhiqing Sun, Shaked Brody, Yallow Uri, Hadar Tojarieh, Adam Roberts, Hyung Won Chung, Jaesung Tae, Jason Phang, Ofir Press, Conglong Li, Deepak Narayanan, Hatim Bourfoune, Jared Casper, Jeff Rasley, Max Ryabinin, Mayank Mishra, Minjia Zhang, Mohammad Shoeybi, Myriam Peyrounette, Nicolas Patry, Nouamane Tazi, Omar Sanseviero,

Patrick von Platen, Pierre Cornette, Pierre François Lavallée, Rémi Lacroix, Samyam Rajbhandari, San-chit Gandhi, Shaden Smith, Stéphane Requena, Suraj Patil, Tim Dettmers, Ahmed Baruwa, Amanpreet Singh, Anastasia Cheveleva, Anne-Laure Ligozat, Arjun Subramonian, Aurélie Névél, Charles Lovering, Dan Garrette, Deepak Tunuguntla, Ehud Reiter, Ekaterina Taktasheva, Ekaterina Voloshina, Eli Bogdanov, Genta Indra Winata, Hailey Schoelkopf, Jan-Christoph Kalo, Jekaterina Novikova, Jessica Zosa Forde, Jordan Clive, Jungo Kasai, Ken Kawamura, Liam Hazan, Marine Carpuat, Miruna Clinciu, Na-joung Kim, Newton Cheng, Oleg Serikov, Omer Antverg, Oskar van der Wal, Rui Zhang, Ruochen Zhang, Sebastian Gehrmann, Shachar Mirkin, Shani Pais, Tatiana Shavrina, Thomas Scialom, Tian Yun, Tomasz Limisiewicz, Verena Rieser, Vitaly Protasov, Vladislav Mikhailov, Yada Pruksachatkun, Yonatan Belinkov, Zachary Bamberger, Zdeněk Kasner, Alice Rueda, Amanda Pestana, Amir Feizpour, Ammar Khan, Amy Faranak, Ana Santos, Anthony Hevia, Antigona Uldreaj, Arash Aghagol, Arezoo Abdollahi, Aycha Tammour, Azadeh HajiHosseini, Bahareh Behroozi, Benjamin Ajibade, Bharat Saxena, Carlos Mu noz Ferrandis, Daniel McDuff, Danish Contractor, David Lansky, Davis David, Douwe Kiela, Duong A. Nguyen, Edward Tan, Emi Baylor, Ez-inwanne Ozoani, Fatima Mirza, Frankline Onon-iwu, Habib Rezanejad, Hessie Jones, Indrani Bhat-tacharya, Irene Solaiman, Irina Sedenko, Isar Ne-jadgholi, Jesse Passmore, Josh Seltzer, Julio Bonis Sanz, Livia Dutra, Mairon Samagaio, Maraim El-badri, Margot Mieskes, Marissa Gerchick, Martha Akinlolu, Michael McKenna, Mike Qiu, Muhammed Ghauri, Mykola Burynok, Nafis Abrar, Nazneen Rajani, Nour Elkott, Nour Fahmy, Olanrewaju Samuel, Ran An, Rasmus Kromann, Ryan Hao, Samira Alizadeh, Sarmad Shubber, Silas Wang, Sourav Roy, Sylvain Viguier, Thanh Le, Tobi Oye-bade, Trieu Le, Yoyo Yang, Zach Nguyen, Abhinav Ramesh Kashyap, Alfredo Palasciano, Al-ison Callahan, Anima Shukla, Antonio Miranda-Escalada, Ayush Singh, Benjamin Beilharz, Bo Wang, Caio Brito, Chenxi Zhou, Chirag Jain, Chuxin Xu, Clémentine Fourrier, Daniel León Peri nán, Daniel Molano, Dian Yu, Enrique Manjavacas, Fabio Barth, Florian Fuhrmann, Gabriel Altay, Giyased-din Bayrak, Gully Burns, Helena U. Vrabec, Imane Bello, Ishani Dash, Ji Hyun Kang, John Giorgi, Jonas Golde, Jose David Posada, Karthik Rangasai Sivara-man, Lokesh Bulchandani, Lu Liu, Luisa Shinzato, Madeleine Hahn de Bykhovetz, Maiko Takeuchi, Marc Pàmies, Maria A Castillo, Marianna Nezhurina, Mario Sängler, Matthias Samwald, Michael Cullan, Michael Weinberg, Michiel De Wolf, Mina Mihalj-cic, Minna Liu, Moritz Freidank, Myungsun Kang, Natasha Seelam, Nathan Dahlberg, Nicholas Michio Broad, Nikolaus Mueller, Pascale Fung, Patrick Haller, Ramya Chandrasekhar, Renata Eisenberg, Robert Martin, Rodrigo Canalli, Rosaline Su, Ruisi Su, Samuel Cahyawijaya, Samuele Garda, Shlok S Deshmukh, Shubhanshu Mishra, Sid Kiblawi, Si-mon Ott, Sinee Sang-aaroonsiri, Srishti Kumar, Ste-fan Schweter, Sushil Bharati, Tanmay Laud, Théo

- Gigant, Tomoya Kainuma, Wojciech Kusa, Yanis Labrak, Yash Shailesh Bajaj, Yash Venkatraman, Yifan Xu, Yingxin Xu, Yu Xu, Zhe Tan, Zhongli Xie, Zifan Ye, Mathilde Bras, Younes Belkada, and Thomas Wolf. 2023. [BLOOM: A 176B-parameter open-access multilingual language model](#). *arXiv preprint 2211.05100*.
- Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. [Toolformer: Language models can teach themselves to use tools](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 68539–68551. Curran Associates, Inc.
- Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. [Hugging-GPT: Solving AI tasks with ChatGPT and its friends in Hugging Face](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 38154–38180. Curran Associates, Inc.
- Zhi Rui Tam, Cheng-Kuang Wu, Yi-Lin Tsai, Chieh-Yen Lin, Hung-yi Lee, and Yun-Nung Chen. 2024. [Let me speak freely? A study on the impact of format restrictions on large language model performance](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1218–1236, Miami, Florida, US. Association for Computational Linguistics.
- 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. 2023. [LLaMA: Open and efficient foundation language models](#). *arXiv preprint 2302.13971*.
- Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, Joao Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2023. [Transformers learn in-context by gradient descent](#). In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 35151–35174. PMLR.
- Tianduo Wang, Shichen Li, and Wei Lu. 2024a. [Self-training with direct preference optimization improves chain-of-thought reasoning](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11917–11928, Bangkok, Thailand. 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, Krma 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, Sujaan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022. [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.
- Zhiruo Wang, Zhoujun Cheng, Hao Zhu, Daniel Fried, and Graham Neubig. 2024b. [What are tools anyway? A survey from the language model perspective](#). In *First Conference on Language Modeling*.
- Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. [Finetuned language models are zero-shot learners](#). In *International Conference on Learning Representations*.
- Wei Xiong, Chengshuai Shi, Jiaming Shen, Aviv Rosenberg, Zhen Qin, Daniele Calandriello, Misha Khalman, Rishabh Joshi, Bilal Piot, Mohammad Saleh, Chi Jin, Tong Zhang, and Tianqi Liu. 2025. [Building math agents with multi-turn iterative preference learning](#). In *The Thirteenth International Conference on Learning Representations*.
- Rui Yang, Lin Song, Yanwei Li, Sijie Zhao, Yixiao Ge, Xiu Li, and Ying Shan. 2023. [GPT4Tools: Teaching large language model to use tools via self-instruction](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 71995–72007. Curran Associates, Inc.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023. [ReAct: Synergizing reasoning and acting in language models](#). In *The Eleventh International Conference on Learning Representations*.
- Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. [STaR: Bootstrapping reasoning with reasoning](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 15476–15488. Curran Associates, Inc.
- Yuchen Zhuang, Yue Yu, Kuan Wang, Haotian Sun, and Chao Zhang. 2023. [ToolQA: A dataset for llm question answering with external tools](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 50117–50143. Curran Associates, Inc.

A Dataset Statistics

The dataset statistics are shown in Table 9.

B Training Details

Supervised Fine-Tuning. During experiments, the LLM was trained for 3 epochs over the curated dataset with the auto-regressive language

Dataset	Type	Training Set	Validation Set	Test Set
<i>Open-domain QA</i>				
TriviaQA	trivia knowledge questions	10,000	1,000	1,000
NQ-Open	real users questions	-	1,000	1,000
PopQA	long-tail Wikipedia knowledge questions	-	1,000	1,000
<i>Mathematical reasoning</i>				
GSM8K	grade school math questions	7,473	650	669

Table 9: The statistics of the datasets used in experiments.

modelling objective. We utilised Low-rank Adaptation (LoRA; [Hu et al., 2022](#)), which is one type of the Parameter-Efficient Fine-Tuning (PEFT) method, as fine-tuning the full parameters of LLMs would be expensive and time-consuming. LoRA accelerates model training by adapting the low-rank decomposition to leverage the burden of updating full parameters to update two trainable low-rank matrices instead. For LoRA training, the hyperparameter r was set to 16, and the target fine-tuning modules were set to `q_proj` and `v_proj`, which are the default settings. As a result, around 0.08% parameters out of the total parameters were trained. The optimiser was AdamW ([Loshchilov and Hutter, 2019](#)). The training loss was computed on the completion only, *i.e.*, the non-LLM response messages, such as tool responses, were disregarded during loss calculation. During model training, the maximum sequence length was set to 8192 tokens. The batch size for the model was set to 4, with gradient accumulation steps of 4, so the effective batch size was 16. We also used gradient checkpointing and FlashAttention-2 ([Dao, 2024](#)) for more memory-efficient model training. All experiments were conducted on 2 A100 GPUs.

Preference Fine-Tuning. The model was trained for 3 epochs over the curated dataset with the DPO objective. We employed DPO instead of RLHF, as the DPO pipeline is more straightforward and requires fewer computation resources. The same optimiser, LoRA setting, maximum sequence length restriction, and memory-efficient tricks as in the SFT experiments were used for model training. The batch size was set to 1 with gradient accumulation steps of 16, leading to an effective batch size of 16, which is also the same as in the SFT experiments. The maximum prompt length was set to 128, which was the default setting. All experiments were conducted on 2 A100 GPUs.

C Inference Details

We experimented with the instruction fine-tuned Llama 3 models with 8B⁵ and 70B⁶ parameters. For LLM inference, we used a default decoding setting: batch size 1, Nucleus Sampling method ([Holtzman et al., 2020](#)) with a temperature of 0.6 and `top_p` of 0.9. The maximum generated token length was set to 512. For the 70B model, we truncated the conversation length to a maximum of 8192 tokens for computational efficiency and kept other inference settings the same as the 8B model.

The conversation list fed to LLM was in a standard chat format, composing three role components: system, user, and assistant. If tool-use is allowed, the system message was the prompt suggesting the task description, the assistant answer format, and tool lists with short descriptions if tools are allowed. The LLM then decided whether to use tools, which tools to use, and what arguments to pass. The LLM’s response can contain the calling of one tool or a sequence of multiple tools. We applied a post-processing function on the LLM’s response to parse it and extract tool usages with a regular expression. If tool-use information is detected, the corresponding tools will be executed with the arguments written by the LLM. Next, the tool responses, along with the past conversation history, were fed back into the LLM to generate the subsequent response. We experimented with two types of tool-use scenarios: single-step tool-use and multi-step tool-use. In the single-step tool-use scenario, the LLM can only get tool responses once but can ask for multiple tool calls simultaneously. In the multi-step tool-use scenario, the LLM can receive tool responses as many times as it wants.

⁵huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct.

⁶huggingface.co/meta-llama/Meta-Llama-3-70B-Instruct.

D Evaluation Metrics

D.1 Measuring tool-use Ability

Invoke Rate: This is a rate to measure the frequency of LLMs calling external tools when responding. We calculated this metric based on the tool usage amount in the first response from LLMs. The metric is defined as:

$$\text{invoke_rate} = \frac{\text{\#tool_usage}}{\text{\#response}} \times 100. \quad (3)$$

Pass Rate: This is a rate to measure the percentage of LLMs successfully executing the tools, regardless of whether the content of the tool responses was related to the current conversation. We considered all tool-use instances that returned non-error responses as successful tool usage. The metric is defined as:

$$\text{pass_rate} = \frac{\text{\#pass}}{\text{\#tool_usage}} \times 100. \quad (4)$$

Answerable Rate: This is a rate to measure the percentage of tool responses containing the ground truth answers, *i.e.*, determine whether the question is answerable based on the tool responses. The metric can be partial, as the LLM may employ tools to conduct intermediate steps during the question-solving process. The metric is defined as:

$$\text{answerable_rate} = \frac{\text{\#answerable}}{\text{\#tool_usage}} \times 100. \quad (5)$$

D.2 Measuring Generation Quality

Exact Match: This is an accuracy commonly adopted in the realm of QA, evaluating whether the model answer is exactly the same as one of the ground truth answers. Following the normalisation process provided in the TriviaQA codebase (Joshi et al., 2017), the answers were normalised by removing underscores, converting into lower-case characters, removing punctuations, removing articles and then removing extra whitespaces. We applied the same normalisation process to answers of all datasets. When the normalised model answer matches one of the normalised answers from the ground truth answer list, the Exact Match score (Score_{EM}) of the sample equals 1; otherwise, 0. Then, we computed the Exact Match (EM) based on the following equation:

$$\text{EM} = \left(\frac{\sum_{i=1}^N \text{Score}_{\text{EM},i}}{N} \right) \times 100, \quad (6)$$

where N is the sample amount, and $\text{Score}_{\text{EM},i}$ is the EM score for the i -th sample in the dataset.

Accuracy: This is an accuracy to comparing the model answer and ground truth answers. We computed the Accuracy in slightly different manners given different task types to match task scenarios better. The details are as follows:

- **Open-domain QA:** For open-domain QA datasets, *i.e.*, TriviaQA, NQ-Open and PopQA, the accuracy is calculated based on the subspan score ($\text{Score}_{\text{subspan}}$), *i.e.*, check whether the model answer contains the ground truth answers, similar to Liu et al. (2024). The intuition is that the model answer may contain some descriptive sentences, *e.g.*, the background of the question. If the normalised model answer contains one of the normalised answers from the ground truth answer list, $\text{Score}_{\text{subspan}}$ equals 1, otherwise 0. Then, the Accuracy (Acc_{QA}) is computed over the whole dataset, defined as follows:

$$\text{Acc}_{\text{QA}} = \left(\frac{\sum_{i=1}^N \text{Score}_{\text{subspan},i}}{N} \right) \times 100, \quad (7)$$

where N is the sample amount, and $\text{Score}_{\text{subspan},i}$ is the subspan score for the i -th sample in the dataset.

- **Mathematical Reasoning:** For the dataset that involves mathematical computation, *i.e.*, GSM8K, the accuracy is computed based on the EM score of normalised extracted answer ($\text{Score}_{\text{EM}_{\text{extract}}}$) compared to the normalised ground truth answer. The intuition is that the model answer might contain some intermediate reasoning steps that divide the questions into several sub-questions to compute the mathematical computation step-by-step in order to generate the final answer. We extracted the last digit number as the model answer, as this is the case in most cases where the model puts the answer. The Accuracy (Acc_{Math}) is computed on the whole dataset, defined as follows:

$$\text{Acc}_{\text{Math}} = \left(\frac{\sum_{i=1}^N \text{Score}_{\text{EM}_{\text{extract}},i}}{N} \right) \times 100, \quad (8)$$

where N is the sample amount, and $\text{Score}_{\text{EM}_{\text{extract}},i}$ is the EM score of the extract answer for the i -th sample in the dataset.

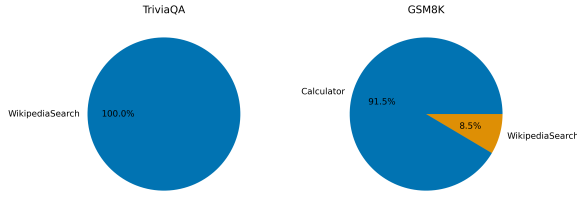


Figure 2: The distribution of tool usage by the Llama-3-8B-Instruct model on the validation sets of TriviaQA and GSM8K.

Error type	TriviaQA	GSM8K
No error	0	5
Hallucination	23	0
Reasoning error	0	23
Argument error	0	1
Low-quality retrieval	7	0
Too-long context	0	0
Infeasible actions	0	1
Misunderstanding the tool’s response	3	0

Table 10: The statistics of the erroneous instances randomly selected from the Llama-3-8B-Instruct model’s output on the validation sets of TriviaQA and GSM8K. The instance is considered erroneous when Accuracy (defined in Appendix D.2) equals 0. “No error” indicates that the model answer is correct, but marked incorrect due to parsing error during the answer extraction process or the ground truth answer is erroneous.

E Extended Results Analysis

Does the model understand the functionality of the provided tools? We analysed the tool usage distribution on the results, shown in Figure 2. The distribution suggests that the LLM understands the functionality of given tools under our setting in most cases. Given the TriviaQA dataset, the Wikipedia search engine would be a natural choice for tools to search for trivia knowledge, and the model chose to use it at all times. In the GSM8K dataset, the calculator would be the most beneficial tool. Therefore, it is also good to see that the tool usage on the dataset was dominated by it. In addition, we can observe that the model did not use the machine translator frequently, though it was provided in the tool list. This phenomenon also supports that the LLM did not randomly select tools from the provided tool list, but more likely used them based on their functionality.

Error type analysis. We examined the error type of the model under both tool-use and non-tool-use settings from randomly selected erroneous instances from the validation sets of TriviaQA (30

instances) and GSM8K (30 instances) on the Llama-3-8B-Instruct model, and summarised the number of common error types into Table 10. The categories of error types were adapted and modified from Zhuang et al. (2023). We used ChatGPT (OpenAI, 2024) to assist with the annotation process. For TriviaQA, we can observe that the error is dominated by “Hallucination”, which is when the model answers with made-up facts. Also, there were several instances in which both the tools responded with low-quality information, and the model misunderstood the low-quality tool responses led to incorrect answers. The results suggest that the model would benefit from improving the tool-use ability, *i.e.*, deciding when to invoke external tools and understanding tool responses better. For GSM8K, the main error type is “Reasoning error”, in which the model performed inaccurate intermediate steps during the problem-solving process or performed incorrect calculations. Also, there were instances in which the model performed infeasible actions or passed wrong arguments to tools. The results suggest that the model could benefit from improving reasoning ability and tool-use ability.

Case study. We examined the instances when tool-use led to inaccurate answers on the Llama-3-8B-Instruct model, and showed how low-quality retrieval and inappropriate tool-use affect the answer quality. An example from Table 11 shows the LLM chose not to provide an answer when the tool response was irrelevant information, while the model provided the correct answer when not using tools. The example suggests that low-quality tool responses might adversely affect the performance of the LLM. An example in Table 12 shows that the model provided a correct answer when it did not have access to tools. However, the tool response misled the reasoning process when the model accessed tools and sent an inappropriate query to the tool, although the reasoning steps were correct. It suggests that using tools might mislead the reasoning process of the model if it misuses tools.

F Prompts

In addition to the single-step tool-use prompts, shown in Table 1, we experimented with some variations based on it for ablation. Table 13 shows a prompt instructing the model to answer without tool-use and answer format. Table 14 shows a prompt with zero-shot CoT instructing the model to answer without tool-use, but with answer format.

NON-TOOL-USE EXAMPLE**User:**

Although both her parents are Australian, in which US state was Nicole Kidman born?

Assistant:

Thought: I will think about the birthplace of Nicole Kidman, considering she is an Australian-born actress.

Answer: **Hawaii (correct answer)**

TOOL-USE EXAMPLE**User:**

Although both her parents are Australian, in which US state was Nicole Kidman born?

Assistant:

Thought: I don't have the information about Nicole Kidman's birthplace, so I'll use external tools to gather the information.

Action: WikipediaSearch[Nicole Kidman]

Rationale: I'll use the WikipediaSearch tool to find information about Nicole Kidman's birthplace.

Answer:

User:

Response from tool WikipediaSearch[Nicole Kidman] are: Rabbit Hole (film) Rabbit Hole is a 2010 American tragedy film starring Nicole Kidman, Aaron Eckhart, and Dianne Wiest, and directed by John Cameron Mitchell; ...

Assistant:

Thought: The text provided by the WikipediaSearch tool is about the movie "Rabbit Hole" (2010) starring Nicole Kidman, but it does not directly answer the question about Nicole Kidman's birthplace.

Action: None needed, as the information provided is not relevant to the question.

Rationale: The text provides information about the movie "Rabbit Hole", including its plot, cast, production, and reception. However, it does not mention Nicole Kidman's birthplace.

Answer: **None (wrong answer)**

Table 11: An example from the validation set of TriviaQA where a tool response misleads the Llama-3-8B-Instruct model. The latter part of the tool response is abbreviated with "..." for space efficiency.

NON-TOOL-USE EXAMPLE**User:**

Zack's locker is half as big as Timothy's locker. Peter's locker is 1/4 as big as Zack's locker. If Peter's locker is 5 cubic inches, how big is Timothy's locker in cubic inches?

Assistant:

Thought: To solve this problem, I'll start by identifying the given information. Peter's locker is 5 cubic inches, and it's 1/4 as big as Zack's locker. This means Zack's locker is $5 \times 4 = 20$ cubic inches. Since Zack's locker is half as big as Timothy's locker, Timothy's locker is $2 \times 20 = 40$ cubic inches.

Answer: **40 cubic inches. (correct answer)**

TOOL-USE EXAMPLE**User:**

Zack's locker is half as big as Timothy's locker. Peter's locker is 1/4 as big as Zack's locker. If Peter's locker is 5 cubic inches, how big is Timothy's locker in cubic inches?

Assistant:

Thought: I will use external tools to solve this problem.

Action: Calculator[1/4 * 5] to find the size of Zack's locker.

Rationale: Since Peter's locker is 1/4 as big as Zack's locker, and we know Peter's locker is 5 cubic inches, we can multiply 5 by 4 to find the size of Zack's locker.

Answer: (I will wait for the tool's response before providing the answer)

User:

Response from tool Calculator[1/4 * 5] are: 1.25

Assistant:

Thought: Now that I have the size of Zack's locker, I will use external tools to find the size of Timothy's locker.

Action: Calculator[2 * 1.25] to find the size of Timothy's locker.

Rationale: Since Zack's locker is half as big as Timothy's locker, and we know Zack's locker is 1.25 cubic inches, we can multiply 1.25 by 2 to find the size of Timothy's locker.

Answer: **2.5 (wrong answer)**

Table 12: An example of the Llama-3-8B-Instruct model getting the wrong answer when using tools from the validation set of GSM8K.

Table 15 shows a prompt instructing the model to answer questions with single-step tool-use but without showing "Rationale". Table 16 is a multi-step variant of the single-step tool-use prompt.

You are an advanced AI agent designed to answer questions. Please use your own knowledge to answer the question. Answer in a few words.

Table 13: System prompt for no tool-use without answer format.

You are an advanced AI agent designed to answer questions. Please use your own knowledge to answer the question. Answer in a few words. Let's think step by step.

Respond in the following format:

Thought: describe your thoughts on how to solve the question.

Answer: provide your answer here.

Table 14: System prompt for no tool-use with zero-shot CoT.

You are an advanced AI agent designed to answer questions. You can use your own knowledge to answer the questions, or use external tools to gather information before answering. However, you can only request the use of tools once. Answer in a few words. Let's think step by step.

Respond in the following format:

Thought: describe your thoughts on how to solve the question, and decide whether to answer using your own knowledge or utilise external tools.

Action: specify the tool here using the format 'ToolName[query]' if you decide to use tools.

Answer: (1) if using your own knowledge, provide your answer here; (2) if using tools, leave this part empty until the tool's response is received.

Below are the external tools you can use:

1. Calculator[query]: this tool helps you perform simple mathematical computations with real numbers. Use the formula as the input query, the tool response will be the result.
 2. WikipediaSearch[query]: this tool helps you search for information from Wikipedia. Use a short keyword as the input query, the tool response will be the corresponding information.
 3. MachineTranslator[query]: this tool helps you understand low-resource languages by translating them to English. Use the sentence you want to translate as the input query, the tool response will be the translation.
-

Table 15: System prompt for single-step tool-use without using "Rationale" in the response format.

You are an advanced AI agent designed to answer questions. You can use your own knowledge to answer the questions, or use external tools to gather information before answering. You can request the use of tools as many times as you want. Answer in a few words. Let's think step by step.

Respond in the following format:

Thought: decide whether to answer using your own knowledge or utilise external tools.

Action: specify the tool here using the format 'ToolName[query]' if you decide to use tools.

Rationale: justify your answer by providing intermediate reasoning steps for your answer, based either on your own knowledge or the received tool responses.

Answer: (1) if using your own knowledge, provide your answer here; (2) if using tools, leave this part empty until the tool's response is received.

Below are the external tools you can use:

1. Calculator[query]: this tool helps you perform simple mathematical computations with real numbers. Use the formula as the input query, the tool response will be the result.
 2. WikipediaSearch[query]: this tool helps you search for information from Wikipedia. Use a short keyword as the input query, the tool response will be the corresponding information.
 3. MachineTranslator[query]: this tool helps you understand low-resource languages by translating them to English. Use the sentence you want to translate as the input query, the tool response will be the translation.
-

Table 16: System prompt for multi-step tool-use.