

---

Informatics Project Proposal 2025

---

# Multi-LLM tool use: Analysing task divisions and PEFT strategies

---

Patryk Kuchta

---

Tool use by large language models (LLMs) is an expanding direction for utilizing the abilities of those models to answer natural language queries. The ability of LLMs to reason is highly dependent on the number of parameters they possess. Recent improvements in the field have shown that it's possible to improve the performance of small models by splitting tasks among separately fine-tuned agents. This study aims to extend the previously mentioned concept by proposing new task splits and other parameter-efficient fine-tuning (PEFT) techniques for multi-LLM tool use. Through this analysis, it is anticipated that the capacity of small models to effectively employ tools will be improved.

Supervisor  
**Dr Pasquale Minervini**

IPP Tutor  
**Xingran Ruan**



THE UNIVERSITY of EDINBURGH  
**informatics**

## Table of contents

List of figures	i
List of tables	i
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Literature review . . . . .	3
<b>2 Method or approach</b>	<b>4</b>
<b>3 Work plan</b>	<b>6</b>
3.1 Plan, milestones, risks, and deliverables . . . . .	6
<b>4 Responsible Research</b>	<b>7</b>
<b>5 References</b>	<b>8</b>

## List of figures

1 Project Gantt chart . . . . .	9
---------------------------------	---

## List of tables

1 Project milestones, deliverables and risks . . . . .	8
--	---

# 1 Introduction

## 1.1 Motivation

The integration of Large Language Models (LLMs) within question-answering (QA) systems, further enhanced by tool augmentation, presents a significant opportunity for the development of a ubiquitous technology with wide-ranging applications. This has been shown by the popularity of smart speakers such as Google Assistant and Amazon Alexa (Canalys, 2021), which are natural language assistants, that can help users with tasks such as checking the weather, by summarising the output from a small predefined list API, that the assistant selects from. Unfortunately, their inability to reason and limited natural language understanding capability limit their usefulness. These issues can be addressed to a large extent by introducing LLMs into their pipelines, as they can perform complex reasoning (Wei et al., 2023) and are excellent at understanding natural language queries.

Furthermore, the growth in popularity of instruction fine-tuned Large Language Models, which in their pure form can answer many natural language questions (Brown et al., 2020; Gao et al., 2024) encourages the need for further developments in this area. Furthermore, people with disabilities can particularly benefit from such tools as their responses do not rely heavily on visual clues to convey information, unlike the case of other well-adopted tools such as web browsers.

The popularity of LLM-based systems persists, despite the fact that most freely available models rely on parametric knowledge, which can become outdated over time (Izacard et al., 2022), whilst costing exorbitant amounts of energy to re-train (Patterson et al., 2021; Thoppilan et al., 2022). Furthermore, those models have a large tendency to hallucinate (Dahl et al., 2024; Huang et al., 2023; Ji et al., 2023; Mallen et al., 2023), which is the phenomenon of generation of facts and statements that are not based on the input data or the model's training. Hallucinations are especially noteworthy, the prevalence of those in tools used by people today raises major concerns even in professionals outside of the computer science field (Curran, Lansley and Bethell, 2023).

LLMs can be trained to retrieve documents (Asai et al., 2024; Guu et al., 2020; Lewis et al., 2021), perform complex reasoning within their output (Wei et al., 2023), write code that solves the question (Li et al., 2023) or use external tools in the form of a large set of APIs to retrieve a variety of information

based on the task (Qin et al., 2023; Tang et al., 2023). Each of these approaches broadens the ability of the LLM to produce accurate information and can prevent hallucinations because the model can reference an external source of information, which discourages hallucinating information (Béchar and Ayala, 2024; Ding et al., 2024). Tool use with very large language models, such as GPT-3, works very well even with very few examples (Brown et al., 2020), but as they are not an open source and the associated expense of computing response with them, it tends to be very computationally to deploy in practical applications, resulting in high financial and environmental costs (Patterson et al., 2021). The emergence of compact open-source language model alternatives, such as Llama 2 (Touvron et al., 2023), mitigates prevailing challenges, albeit accompanied by substantial performance degradation when utilized in their raw form (Kaplan et al., 2020; Wei et al., 2022).

There have been several approaches to improve their performance. In their work Shen et al. (2024) they propose a modification to the regular tool use pipeline. In line with the heuristics of *ReACT* (Yao et al., 2023), the authors have distinguished 3 tasks that the model has to undertake when using tools: Planning (also called *Reasoning*), Calling (also called *ACTing*) and Summarizing. In a traditional tool-use pipeline, each of these phases is performed by the same model that has been trained to perform all three tasks. (Shen et al., 2024)  $\alpha$ -UMI the authors propose to enhance the performance of the model by fine-tuning three separate models, where the split was designed based on the aforementioned heuristics of *ReACT* (Yao et al., 2023). This when combined with inference time parameter patching, using the Low-Rank Adaptation (LoRA) (Hu et al., 2021), means computationally the resulting pipeline requires only negligibly more parameters to be stored in memory, whilst resulting in a large performance improvement at certain stages of the pipeline (for instance an 8.4% improvement over the Single-LLM approach in terms of Plan Accuracy on ToolBench (Qin et al., 2023)). These specialised networks will be referred to as agents in this work. It has to be noted that the training time of three specialised agents is longer than the training time in the case of training a single network to perform all three kinds of operations. In their study, the  $\alpha$ -UMI pipeline required roughly 50% more training time than the Single-LLM network.

It is postulated that the heuristic division of tasks among agents may not represent the optimal efficiency in relation to training duration and performance. There may exist alternative, coarser divisions that yield comparable performance levels. On the other hand, the contrary might be true, where a

finer split of tasks might have a large positive impact on the performance of the pipeline, whilst not prolonging the training time by a significant margin. Finally, there are potential advantages to another parameter efficient fine turning (PEFT) strategy in certain settings (Han et al., 2024), hence I hypothesise that the final performance could be improved or at least validated to be LoRA. Amongst the reviewed PEFT strategies ReLoRA (Lialin et al., 2023) is the most promising as it is an expansion of the aforementioned LoRA, promising performance improvements. On the other hand, QLoRA (Dettmers et al., 2023), is a promising approach for cases with limited training resources, especially when the number of agents grows rapidly.

The study aims to find the best task splits for the tool-use pipeline or validate the existing proposed split as the best option (as seen in Shen et al. (2024)). Further, the study also aims to find the best PEFT technique for this task. The last aim is to validate the repeatability of the results presented in work presented by Shen et al. (2024).

## 1.2 Literature review

LLM tool use is a very dynamic field of study with most of the crucial developments in the area happening in the last few years. Several factors contributed to this expansion of research in this field, string with the recent large growth of the capability of LLMs, especially when it comes to their ability to learn from even a small number of samples (Brown et al., 2020). These benefits are the most characteristic of the very large models, whilst the smaller modern models such as Llama 2 (Touvron et al., 2023), rely on fine-tuning on the samples to achieve good performance. The next advancement that has moved the tool use field forward was the idea of generating training data using instruction fine-tuned LLMs (Wang et al., 2021), by utilizing extremely capable models such as Chat-GPT 4 (OpenAI, 2024). This approach was used by the authors of Qin et al. (2023) to create example usages of available APIs simply based on their documentation with no need for manual human annotation. This allowed for creating a larger dataset and also expanding the number of tools the model can learn to use.

A strong foundation for the field in the shape of a large dataset, that covers a wide array of tools and the availability of highly capable foundation models is crucial to the field (Zhou et al., 2023), but there are still several open questions and dilemmas that require the attention of researchers to create practically viable tools. The dilemma that is especially important in

the context of the climate crisis, is the trade-off between capability and power efficiency, as the larger models possess a higher degree of competence (Kaplan et al., 2020; Wei et al., 2022). Their monetary and climate cost cannot be underestimated as the power consumption of the LLM and the implications of that are large (Patterson et al., 2021).

Furthermore, handling the user's request in the tool use context often necessitates multi-step reasoning such as the approach shown by Yao et al. (2023), which separates the step of reasoning and acting, hence the name ReACT. This idea works much better than simply performing the answering of the question in one response. This approach can be used for tool use and is often used as a multi-step ReACT, where the agent will plan for the initial API request, execute it, consider the implications of the response, and whether to continue with the next planning step or end the execution by providing the user and answer to their original question.

This pipeline has a major issue, which is the fact that a single error or hallucination at one step is likely to lead to an incorrect response or an error. Qin et al. (2023) in addition to procuring a dataset for the task they have also tasked themselves with addressing this issue. In their alternative to the standard application of ReACT, called DFSDT, they propose allowing the model to effectively roll back some of the steps and try again. Another interesting approach is representing the APIs and their outcomes as a graph, and allowing the model to perform its reasoning over the graph (Liu et al., 2023). This is a much more complicated but also flexible structure, that enables the LLM to have a better understanding of the interaction and potential use cases of each tool, rather than considering it as a simple linear structure (ReACT) or tree structure (DFSDT).

## 2 Method or approach

Achieving the aim set out in this research necessitates reproducing the results of work undertaken by Shen et al. (2024) as this research merits attention under the assumption that the basic heuristic split can yield performance benefits under the right training regime. Secondly, as the ethos of the work is improving the quality of the response when using smaller models, the computational cost of training and inference has to be measured, therefore a metric for evaluating that has to be devised and used for each of the proposed modification of the original pipeline to provide context to the evaluation in terms cost.

In order to reproduce the work from (Shen et al., 2024) and expand it further the pipeline shown in their paper has to be implemented and further expanded to fit the needs of changing the splits in the tasks and inserting differing PEFT techniques. The choice of foundation model used in this study will remain consistent with the model used in the original study which was the Llama 2 model (Touvron et al., 2023), which is not the newest model from the Llama family since the recent release of the Llama 3 model. This choice was made to consistency and ability to compare with the existing research which is mainly based on the older version of the model.

The proposed splits have to be considered from the theoretical point of view and the model ought to be assessed based on which part of the model is the most responsible for failures of the entire pipeline. Such an evaluation regime can then be applied to the original  $\alpha$ -UMI pipeline, which can be evaluated in those terms to produce potential avenues for improvement, by splitting struggling agents and merging agents that are performing well.

This evaluation can be achieved by using a subset of training data, and then identifying where the models' behaviour deviated from the correct sample, and then recording the model responsible for the mishap. Crediting the error can be achieved using simple heuristics, such as the Planner is responsible for choosing the right API, the Caller is responsible for getting an error-free response from the API, and the Summarizer produces the correct answer to the user. The cases where the previous steps did not have issues, whilst the current step did not match the expectation can be counted and stored as a rate for each of the agents in the system. Ruan et al. (2023) present a technique for evaluating this aspect of LLMs using tools, and other works focus more of their attention on evaluating the tool choice in particular (Huang et al., 2024).

Afterwards, the new splits have to be evaluated and the respective fault rates can be measured to suggest new agent splits to test. Consideration of the performance and pattern of shifting performance based on the characteristics of each split ought to be considered as well. A similar approach ought to be undertaken for the various PEFT methods and their implications in terms of performance. The final resulting contribution from both of these analyses will be an improvement in the performance of the final model, which will aid small models to be more competitive in terms of their ability to use tools when compared to models that require more computational resources to run. Further, enrichment in terms of the rationale behind selecting such

non-heuristic splits for models of this kind and the PEFT techniques in the context of small models is another desired outcome of this study.

### 3 Work plan

Implementing the methods requires further research of evaluation methods and benchmarks that assess tool use in addition to the more thorough review of metrics of accessing the point of failure in the pipeline. This along with the research conducted for this Research Proposal can be classed as ‘Research’ in terms of work grouping.

Further, the initial benchmarks and evaluation metrics outlined in Section 2 will need to be implemented before the testing of the baseline methodology. With those pipelines in place, the baseline methodology presented by Shen et al. (2024), both in the Single-LLM and  $\alpha$ -UMI variants, will be evaluated. Once the performance of those is validated, the selection of new experimental to be proposed and then evaluated using the metrics as mentioned earlier. Developing the required code for these tasks can be generally referred to as ‘Implementation’. In this phase, the tangible deliverable will be the software and pipelines procured for this task.

A substantial portion of the time has to be dedicated to training the model and evaluating the aforementioned models. This can be done largely without human supervision therefore in terms of the Work assignment, the time assigned to this aspect of the project will be shared with other work. This grouping will be collectively known as ‘Training’.

The results of this evaluation can be used to further suggest other splits in the data can be trailed to find the most viable and efficient solution to splitting the task. Furthermore, those results ought to be considered and discussed in full in the report in the search for patterns are new split heuristics based on the experimentation conducted. Within a similar vein of work, work on writing the report will be also merged into this grouping. This grouping will be referred to as ‘Results & Report’.

#### 3.1 Plan, milestones, risks, and deliverables

The plan for the project has been visually represented as a Gantt Chart in Figure 1. This representation of the plan more clearly represents the dependencies in the project and tasks which are characterised by a degree of concurrency. Milestones are highlighted in the plan by rhombus shape in



the chart and the corresponding  $M_x$  label. Each Milestone's details and the associated Deliverable are shown in Table 1. This table also includes details of the Risks associated with the project.

The plan largely follows the steps outlined in section 3. The milestones that were highlighted for this research were the finalization of the initial research, and the completion of the code implementation, which will result in the final software repository. The next major milestone is the completion of the training/fine-tuning of all of the models required for the analysis. The final milestone is the production of the final report, which comes with comes with a distinct deliverable of the final report.

In terms of risks, the major risk is the inability to effectively assess the practical usability of the final model. Human annotated evaluation is the most indicative of actual improvements in the quality of the responses, but it is outside of the financial scope of this project. Some of the other evaluation schemes rely on access to a large number of Chat-GPT 4 API calls to access the model's ability to recover from errors (Wang et al., 2024), but the cost of using such API can be extortionate. Addressing this risk requires thorough research into multiple evaluation strategies that are computationally and financially viable. The second risk comes if the performance shown in the work of Shen et al. (2024) is unreproducible, which would put the premise of the entire research into uncertainty as if the proposed split in the paper did not improve the performance, it is unlikely that other splits will improve the performance beyond the single-LLM baseline. The next risk is the fact that the evaluation metrics proposed to advise the further splits, do not yield useful insights. This risk is quite minor and might be a partial indication that there is a limited gain to be had from changing the splits of the data. In the case of this risk becoming a reality, the research can focus on new task splits that heuristically make the most sense. Another factor to consider is that the end of the dissertation period may be a particularly busy time for the supervisor, due in part to their responsibilities to other students. As a result, it may be more challenging to receive timely feedback on the final report. To mitigate this, it might be helpful to plan ahead and allow for extra time for feedback and revisions.

## 4 Responsible Research

The research undertaken in this study is based on my best understanding is free of major ethical risks associated. The datasets used in this study (Qin

et al., 2023; Tang et al., 2023) have been acquired through ethical means, such as annotation by Chat-GPT 4, based on publicly available API documentation. The list of APIs that the model is trained on is provided by RapidAPI. Service providers on the RapidAPI platform are liable for ensuring the APIs do not infringe on any IP (intellectual property) (RapidAPI, 2023). The dataset does not include any sensitive information, as it is simply based on publicly available API documentation and the resulting dataset is itself public as well.

	Event	Period	Description	Deliverable
Milestone	M <sub>1</sub>	26.05	End of Initial Research	
	M <sub>2</sub>	29.06	Implementation completed	Software repository
	M <sub>3</sub>	<b>20.07</b>	<b>All models trained</b>	
	M <sub>4</sub>	19.08	Report finalised	<b>Final Report</b>
Risk	R <sub>1</sub>	14.05-21.05	Limited benchmarks within computational budget	
	R <sub>2</sub>	25.05-14.06	Performance in Shen et al. (2024) work proves unreproducible	
	R <sub>3</sub>	<b>03.06-20.06</b>	<b>Baseline results prove not useful for split selection</b>	
	R <sub>4</sub>	10.08-20.08	Busy period for supervisor	

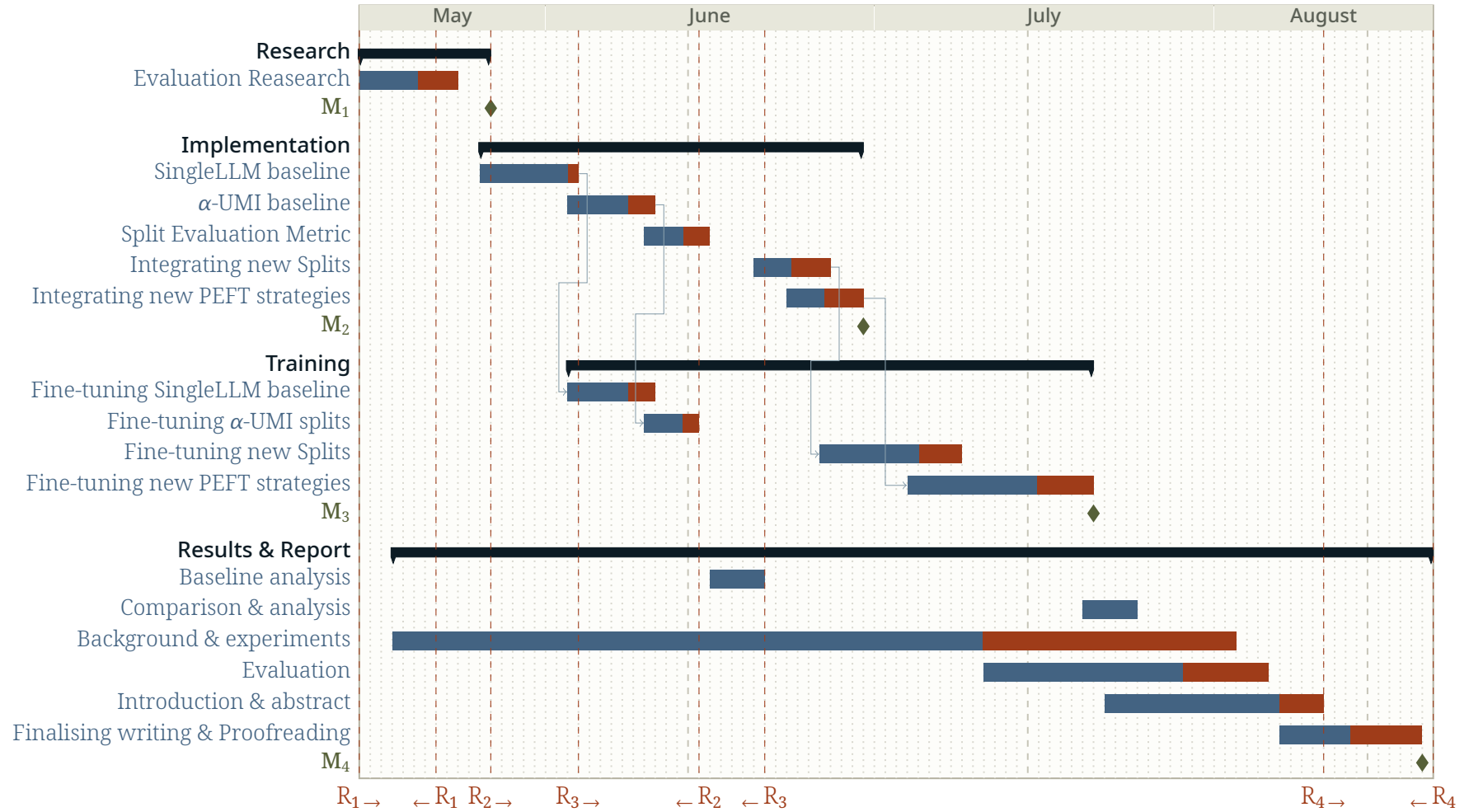
**Table 1** Project milestones, deliverables, and risks (**bold** = main); see Figure 1.

## 5 References

Asai, Akari, Zhong, Zexuan, Chen, Danqi, Koh, Pang Wei, Zettlemoyer, Luke, Hajishirzi, Hannaneh and Yih, Wen-tau, 2024. **Reliable, adaptable, and attributable language models with retrieval**. arXiv: 2403.03187 [cs.CL] (cited on page 1).

Béchar, Patrice and Ayala, Orlando Marquez, 2024. **Reducing hallucination in structured outputs via retrieval-augmented generation**. arXiv: 2404.08189 [cs.LG] (cited on page 2).

Brown, Tom B., Mann, Benjamin, Ryder, Nick, Subbiah, Melanie, Kaplan, Jared, Dhariwal, Prafulla, Neelakantan, Arvind, Shyam, Pranav, Sastry, Girish, Askell, Amanda, Agarwal, Sandhini, Herbert-Voss, Ariel, Krueger, Gretchen, Henighan, Tom, Child, Rewon, Ramesh, Aditya, Ziegler, Daniel M., Wu, Jeffrey, Winter, Clemens, Hesse, Christopher, Chen, Mark, Sigler, Eric, Litwin, Mateusz, Gray, Scott, Chess, Benjamin, Clark, Jack, Berner, Christopher, McCandlish, Sam, Radford, Alec, Sutskever,



**Figure 1** Gantt Chart of project activities, see Table 1 for details. Overarching bars coloured ■ represent the maximum time (including contingency) allocated per phase, ■ represents the scheduled time for a task and ■ its contingency time. Arrowed lines indicate task dependencies.

- Ilya and Amodei, Dario, 2020. **Language models are few-shot learners**. arXiv: 2005.14165 [cs.CL] (cited on pages 1, 2, 3).
- Canalys, 2021. **Global smart speaker market set to reach 163 million units in 2021** [Online]. canalys.com. Available from: [https://canalys-prod-public.s3.eu-west-1.amazonaws.com/static/press\\_release/2020/SSPR2020Q2.pdf](https://canalys-prod-public.s3.eu-west-1.amazonaws.com/static/press_release/2020/SSPR2020Q2.pdf) [Accessed 5 April 2024] (cited on page 1).
- Curran, Shawn, Lansley, Sam and Bethell, Oliver, 2023. **Hallucination is the last thing you need**. arXiv: 2306.11520 [cs.CL] (cited on page 1).
- Dahl, Matthew, Magesh, Varun, Suzgun, Mirac and Ho, Daniel E., 2024. **Large legal fictions: profiling legal hallucinations in large language models**. arXiv: 2401.01301 [cs.CL] (cited on page 1).
- Dettmers, Tim, Pagnoni, Artidoro, Holtzman, Ari and Zettlemoyer, Luke, 2023. **Qlora: efficient finetuning of quantized llms**. arXiv: 2305.14314 [cs.LG] (cited on page 3).
- Ding, Hanxing, Pang, Liang, Wei, Zihao, Shen, Huawei and Cheng, Xueqi, 2024. **Retrieve only when it needs: adaptive retrieval augmentation for hallucination mitigation in large language models**. arXiv: 2402.10612 [cs.CL] (cited on page 2).
- Gao, Jie, Gebreegziabher, Simret Araya, Choo, Kenny Tsu Wei, Li, Toby Jia-Jun, Perrault, Simon Tangi and Malone, Thomas W, 2024. A taxonomy for human-llm interaction modes: an initial exploration. **Arxiv preprint arxiv:2404.00405** (cited on page 1).
- Guu, Kelvin, Lee, Kenton, Tung, Zora, Pasupat, Panupong and Chang, Ming-Wei, 2020. **Realm: retrieval-augmented language model pre-training**. arXiv: 2002.08909 [cs.CL] (cited on page 1).
- Han, Zeyu, Gao, Chao, Liu, Jinyang, Zhang, Jeff and Zhang, Sai Qian, 2024. **Parameter-efficient fine-tuning for large models: a comprehensive survey**. arXiv: 2403.14608 [cs.LG] (cited on page 3).
- Hu, Edward J., Shen, Yelong, Wallis, Phillip, Allen-Zhu, Zeyuan, Li, Yuezhi, Wang, Shean, Wang, Lu and Chen, Weizhu, 2021. **Lora: low-rank adaptation of large language models**. arXiv: 2106.09685 [cs.CL] (cited on page 2).
- Huang, Lei, Yu, Weijiang, Ma, Weitao, Zhong, Weihong, Feng, Zhangyin, Wang, Haotian, Chen, Qianglong, Peng, Weihua, Feng, Xiaocheng, Qin, Bing and Liu, Ting, 2023. **A survey on hallucination in large language models: principles, taxonomy, challenges, and open questions**. arXiv: 2311.05232 [cs.CL] (cited on page 1).
- Huang, Yue, Shi, Jiawen, Li, Yuan, Fan, Chenrui, Wu, Siyuan, Zhang, Qihui, Liu, Yixin, Zhou, Pan, Wan, Yao, Gong, Neil Zhenqiang and Sun, Lichao, 2024. **Metatool benchmark for large language models: deciding whether to use tools and which to use**. arXiv: 2310.03128 [cs.SE] (cited on page 5).

- Izacard, Gautier, Lewis, Patrick, Lomeli, Maria, Hosseini, Lucas, Petroni, Fabio, Schick, Timo, Dwivedi-Yu, Jane, Joulin, Armand, Riedel, Sebastian and Grave, Edouard, 2022. **Atlas: few-shot learning with retrieval augmented language models**. arXiv: 2208.03299 [cs.CL] (cited on page 1).
- Ji, Ziwei, Lee, Nayeon, Frieske, Rita, Yu, Tiezheng, Su, Dan, Xu, Yan, Ishii, Etsuko, Bang, Ye Jin, Madotto, Andrea and Fung, Pascale, 2023. Survey of hallucination in natural language generation. **Acm computing surveys** [Online], 55(12), March, pp.1–38. Available from: <https://doi.org/10.1145/3571730> (cited on page 1).
- Kaplan, Jared, McCandlish, Sam, Henighan, Tom, Brown, Tom B., Chess, Benjamin, Child, Rewon, Gray, Scott, Radford, Alec, Wu, Jeffrey and Amodei, Dario, 2020. **Scaling laws for neural language models**. arXiv: 2001.08361 [cs.LG] (cited on pages 2, 4).
- Lewis, Patrick, Perez, Ethan, Piktus, Aleksandra, Petroni, Fabio, Karpukhin, Vladimir, Goyal, Naman, Küttler, Heinrich, Lewis, Mike, Yih, Wen-tau, Rocktäschel, Tim, Riedel, Sebastian and Kiela, Douwe, 2021. **Retrieval-augmented generation for knowledge-intensive nlp tasks**. arXiv: 2005.11401 [cs.CL] (cited on page 1).
- Li, Chengshu, Liang, Jacky, Zeng, Andy, Chen, Xinyun, Hausman, Karol, Sadigh, Dorsa, Levine, Sergey, Fei-Fei, Li, Xia, Fei and Ichter, Brian, 2023. **Chain of code: reasoning with a language model-augmented code emulator**. arXiv: 2312.04474 [cs.CL] (cited on page 1).
- Lialin, Vladislav, Shivagunde, Namrata, Muckatira, Sherin and Rumshisky, Anna, 2023. **Relora: high-rank training through low-rank updates**. arXiv: 2307.05695 [cs.CL] (cited on page 3).
- Liu, Zhaoyang, Lai, Zeqiang, Gao, Zhangwei, Cui, Erfei, Li, Ziheng, Zhu, Xizhou, Lu, Lewei, Chen, Qifeng, Qiao, Yu, Dai, Jifeng and Wang, Wenhai, 2023. **Controllm: augment language models with tools by searching on graphs**. arXiv: 2310.17796 [cs.CV] (cited on page 4).
- Mallen, Alex, Asai, Akari, Zhong, Victor, Das, Rajarshi, Khashabi, Daniel and Hajishirzi, Hannaneh, 2023. When not to trust language models: investigating effectiveness of parametric and non-parametric memories. In: Anna Rogers, Jordan Boyd-Graber and Naoaki Okazaki, eds. **Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)** [Online]. Toronto, Canada: Association for Computational Linguistics, July, pp.9802–9822. Available from: <https://doi.org/10.18653/v1/2023.acl-long.546> (cited on page 1).
- OpenAI, 2024. **Gpt-4 technical report**. arXiv: 2303.08774 [cs.CL] (cited on page 3).
- Patterson, David, Gonzalez, Joseph, Le, Quoc, Liang, Chen, Munguia, Lluís-Miquel, Rothchild, Daniel, So, David, Texier, Maud and Dean, Jeff, 2021. **Carbon emissions and large neural network training**. arXiv: 2104.10350 [cs.LG] (cited on pages 1, 2, 4).

Qin, Yujia, Liang, Shihao, Ye, Yining, Zhu, Kunlun, Yan, Lan, Lu, Yaxi, Lin, Yankai, Cong, Xin, Tang, Xiangru, Qian, Bill, Zhao, Sihan, Hong, Lauren, Tian, Runchu, Xie, Ruobing, Zhou, Jie, Gerstein, Mark, Li, Dahai, Liu, Zhiyuan and Sun, Maosong, 2023. **Toollm: facilitating large language models to master 16000+ real-world apis**. arXiv: 2307.16789 [cs.AI] (cited on pages 2, 3, 4, 7).

RapidAPI, 2023. **RapidAPI Terms of Service** [Online]. Accessed: 2024-04-04. Available from: <https://rapidapi.com/terms/> (cited on page 8).

Ruan, Jingqing, Chen, Yihong, Zhang, Bin, Xu, Zhiwei, Bao, Tianpeng, Du, Guoqing, Shi, Shiwei, Mao, Hangyu, Li, Ziyue, Zeng, Xingyu and Zhao, Rui, 2023. **Tptu: large language model-based ai agents for task planning and tool usage**. arXiv: 2308.03427 [cs.AI] (cited on page 5).

Shen, Weizhou, Li, Chenliang, Chen, Hongzhan, Yan, Ming, Quan, Xiaojun, Chen, Hehong, Zhang, Ji and Huang, Fei, 2024. **Small llms are weak tool learners: a multi-llm agent**. arXiv: 2401.07324 [cs.AI] (cited on pages 2, 3, 4, 5, 6, 7, 8).

Tang, Qiaoyu, Deng, Ziliang, Lin, Hongyu, Han, Xianpei, Liang, Qiao, Cao, Boxi and Sun, Le, 2023. **Toolalpaca: generalized tool learning for language models with 3000 simulated cases**. arXiv: 2306.05301 [cs.CL] (cited on pages 2, 8).

Thoppilan, Romal, Freitas, Daniel De, Hall, Jamie, Shazeer, Noam, Kulshreshtha, Apoorv, Cheng, Heng-Tze, Jin, Alicia, Bos, Taylor, Baker, Leslie, Du, Yu, Li, YaGuang, Lee, Hongrae, Zheng, Huaixiu Steven, Ghafouri, Amin, Menegali, Marcelo, Huang, Yanping, Krikun, Maxim, Lepikhin, Dmitry, Qin, James, Chen, Dehao, Xu, Yuanzhong, Chen, Zhifeng, Roberts, Adam, Bosma, Maarten, Zhao, Vincent, Zhou, Yanqi, Chang, Chung-Ching, Krivokon, Igor, Rusch, Will, Pickett, Marc, Srinivasan, Pranesht, Man, Laichee, Meier-Hellstern, Kathleen, Morris, Meredith Ringel, Doshi, Tulsee, Santos, Renelito Delos, Duke, Toju, Soraker, Johnny, Zevenbergen, Ben, Prabhakaran, Vinodkumar, Diaz, Mark, Hutchinson, Ben, Olson, Kristen, Molina, Alejandra, Hoffman-John, Erin, Lee, Josh, Aroyo, Lora, Rajakumar, Ravi, Butryna, Alena, Lamm, Matthew, Kuzmina, Viktoriya, Fenton, Joe, Cohen, Aaron, Bernstein, Rachel, Kurzweil, Ray, Aguera-Arcas, Blaise, Cui, Claire, Croak, Marian, Chi, Ed and Le, Quoc, 2022. **Lamda: language models for dialog applications**. arXiv: 2201.08239 [cs.CL] (cited on page 1).

Touvron, Hugo, Martin, Louis, Stone, Kevin, Albert, Peter, Almahairi, Amjad, Babaei, Yasmine, Bashlykov, Nikolay, Batra, Soumya, Bhargava, Prajjwal, Bhosale, Shruti, Bikel, Dan, Blecher, Lukas, Ferrer, Cristian Canton, Chen, Moya, Cucurull, Guillem, Esiobu, David, Fernandes, Jude, Fu, Jeremy, Fu, Wenying, Fuller, Brian, Gao, Cynthia, Goswami, Vedanuj, Goyal, Naman, Hartshorn, Anthony, Hosseini, Saghar, Hou, Rui, Inan, Hakan, Kardas, Marcin, Kerkez, Viktor, Khabsa, Madian, Kloumann, Isabel, Korenev, Artem, Koura, Punit Singh, Lachaux, Marie-Anne, Lavril, Thibaut, Lee, Jenya, Liskovich, Diana, Lu, Yinghai, Mao, Yuning, Martinet, Xavier, Mihaylov, Todor, Mishra, Pushkar, Molybog, Igor, Nie, Yixin, Poulton, Andrew, Reizenstein, Jeremy, Rungta, Rashi, Saladi, Kalyan, Schelten, Alan, Silva, Ruan, Smith, Eric Michael, Subramanian, Ranjan, Tan, Xiaoqing Ellen, Tang, Binh, Taylor, Ross, Williams,

Adina, Kuan, Jian Xiang, Xu, Puxin, Yan, Zheng, Zarov, Iliyan, Zhang, Yuchen, Fan, Angela, Kambadur, Melanie, Narang, Sharan, Rodriguez, Aurelien, Stojnic, Robert, Edunov, Sergey and Scialom, Thomas, 2023. **Llama 2: open foundation and fine-tuned chat models**. arXiv: 2307.09288 [cs.CL] (cited on pages 2, 3, 5).

Wang, Xingyao, Wang, Zihan, Liu, Jiateng, Chen, Yangyi, Yuan, Lifan, Peng, Hao and Ji, Heng, 2024. **Mint: evaluating llms in multi-turn interaction with tools and language feedback**. arXiv: 2309.10691 [cs.CL] (cited on page 7).

Wang, Zirui, Yu, Adams Wei, Firat, Orhan and Cao, Yuan, 2021. **Towards zero-label language learning**. arXiv: 2109.09193 [cs.CL] (cited on page 3).

Wei, Jason, Tay, Yi, Bommasani, Rishi, Raffel, Colin, Zoph, Barret, Borgeaud, Sebastian, Yogatama, Dani, Bosma, Maarten, Zhou, Denny, Metzler, Donald, Chi, Ed H., Hashimoto, Tatsunori, Vinyals, Oriol, Liang, Percy, Dean, Jeff and Fedus, William, 2022. **Emergent abilities of large language models**. arXiv: 2206.07682 [cs.CL] (cited on pages 2, 4).

Wei, Jason, Wang, Xuezhi, Schuurmans, Dale, Bosma, Maarten, Ichter, Brian, Xia, Fei, Chi, Ed, Le, Quoc and Zhou, Denny, 2023. **Chain-of-thought prompting elicits reasoning in large language models**. arXiv: 2201.11903 [cs.CL] (cited on page 1).

Yao, Shunyu, Zhao, Jeffrey, Yu, Dian, Du, Nan, Shafran, Izhak, Narasimhan, Karthik and Cao, Yuan, 2023. **React: synergizing reasoning and acting in language models**. arXiv: 2210.03629 [cs.CL] (cited on pages 2, 4).

Zhou, Ce, Li, Qian, Li, Chen, Yu, Jun, Liu, Yixin, Wang, Guangjing, Zhang, Kai, Ji, Cheng, Yan, Qiben, He, Lifang, Peng, Hao, Li, Jianxin, Wu, Jia, Liu, Ziwei, Xie, Pengtao, Xiong, Caiming, Pei, Jian, Yu, Philip S. and Sun, Lichao, 2023. **A comprehensive survey on pretrained foundation models: a history from bert to chatgpt**. arXiv: 2302.09419 [cs.AI] (cited on page 3).