LLaSA: Large Language and E-Commerce Shopping Assistant

Shuo Zhang
Zhejiang University
Hangzhou, China
shuo.zhang@zju.edu.cn

Boci Peng
Peking University
Beijing, China
bcpeng@stu.pku.edu.cn

Xinping Zhao
Harbin Institute of Technology (Shenzhen)
Shenzhen, China
zhaoxinping@stu.hit.edu.cn

Boren Hu
The Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, China
huboren99@gmail.com

Yun Zhu
Zhejiang University
Hangzhou, China
zhuyun_dcd@zju.edu.cn

Yanjia Zeng
Zhejiang University
Hangzhou, China
22151306@zju.edu.cn

Xuming Hu*
The Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, China
xuminghu@hkust-gz.edu.cn

Abstract

The e-commerce platform has evolved rapidly due to its widespread popularity and convenience. Developing an e-commerce shopping assistant for customers is crucial to aiding them in quickly finding desired products and recommending precisely what they need. However, most previous shopping assistants face two main problems: (1) *task-specificity*, which necessitates the development of different models for various tasks, thereby increasing development costs and limiting effectiveness; and (2) *poor generalization*, where the trained model performs inadequately on up-to-date products. To resolve these issues, we employ Large Language Models (LLMs) to construct an omnipotent assistant, leveraging their adeptness at handling multiple tasks and their superior generalization capability. Nonetheless, LLMs lack inherent knowledge of e-commerce concepts. To address this, we create an instruction dataset comprising 65,000 samples and diverse tasks, termed as **EshopInstruct**[1]. Through instruction tuning on our dataset, the assistant, named **LLaSA**, demonstrates the potential to function as an omnipotent assistant. Additionally, we propose various inference optimization strategies to enhance performance with limited inference resources. In the Amazon KDD Cup 2024 Challenge[2], our proposed method, LLaSA, achieved an overall ranking of 3rd place on ShopBench, including 57 tasks and approximately 20,000 questions, and we secured top-5 rankings in each track, especially in track4, where we achieved the best performance result among all student teams. Our extensive practices fully demonstrate that LLMs possess the great potential to be competent e-commerce shopping assistants[3].

CCS Concepts

• **Computing methodologies** → **Natural language processing**.

Keywords

Multi-Task Online Shopping, Large Language Models, Instruction Construction, Model Quantification, KDD Cup

ACM Reference Format:
Shuo Zhang, Boci Peng, Xinping Zhao, Boren Hu, Yun Zhu, Yanjia Zeng, and Xuming Hu. 2024. LLaSA: Large Language and E-Commerce Shopping Assistant. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Amazon KDD Cup 2024 Workshop).* ACM, New York, NY, USA, 6 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

1.1 Background

The rapid growth of e-commerce has transformed how we shop, offering unprecedented convenience and access to a vast array of products. However, this convenience comes with the challenge of navigating an overwhelming volume of information. When shopping online, users often face the daunting task of sifting through countless products, reading numerous reviews, comparing prices, and ultimately making a purchase decision. This process can be time-consuming and stressful, highlighting the complexities inherent in online shopping [8, 10, 12].

Large language models (LLMs) offer a promising solution to address these challenges [20]. Current techniques often struggle

*Xuming Hu is the corresponding author.
[1]Our instruction dataset can be found at https://github.com/suyan-liang/EshopInstruct.
[2]https://www.aicrowd.com/challenges/amazon-kdd-cup-2024-multi-task-online-shopping-challenge-for-llms
[3]All of the authors contributed equally to this work. Our team name is "shimmering_as_the_stars", whose expression in Chinese is 灿若星辰.

Amazon KDD Cup 2024 Workshop, Aug 28, 2024, Barcelona, Spain

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

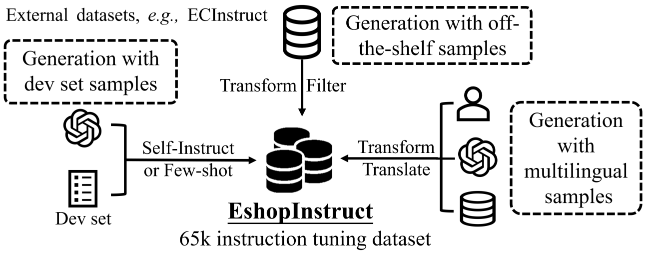

Figure 1: Construction pipeline of EshopInstruct. We design three strategies for building the EshopInstruct dataset: generating data from seed data, extracting data from publicly available ECInstruct, and designing new tasks to generate data. Based on these strategies, we obtained 65k data points.

to fully grasp the nuances of specific shopping terms, customer behaviors, and the diverse nature of products and languages. In contrast, LLMs, with their multi-task and few-shot learning capabilities, have the potential to enhance the online shopping experience significantly.

To encourage LLMs to meet the unique needs of online shopping, enhance user experience, and streamline decision-making, Amazon has introduced ShopBench and organized the Amazon KDD Cup 2024 challenge. This competition features five tracks, focusing on four key shopping skills: Shopping Concept Understanding, Shopping Knowledge Reasoning, User Behavior Alignment, and Multilingual Abilities.

1.2 Datasets Description

ShopBench is a multi-task dataset derived from real-world shopping data in the Amazon platform, designed for the Amazon KDD Cup 2024 challenge. The dataset is divided into a few-shot development set and a test set, designed to more accurately simulate the few-shot learning settings. It contains 57 tasks and approximately 20,000 questions, which are all reformulated into a unified text-to-text generation format to facilitate LLM-based solutions. The detailed statistics of the datasets are summarized in Table 1.

1.3 Task Description

In the ShopBench benchmark, five abilities, including Generation, Ranking, Retrieval, Multiple-Choice, and NER (Named Entity Recognition), are introduced to evaluate four important shopping skills:

- Track1 (Shopping Concept Understanding): Given the prevalence of domain-specific concepts in online shopping, the goal is to enhance LLMs' ability to effectively understand and respond to queries about these concepts.
- Track2 (Shopping Knowledge Reasoning): Considering the complex reasoning required for shopping decisions, the goal is to assess the model's capability in reasoning about products and their attributes using domain-specific implicit knowledge.
- Track3 (User Behavior Alignment): Given the diversity and implicit nature of user behaviors in online shopping, the goal is to align language models with these behaviors to improve their effectiveness in this domain.

Figure 2: The data distribution of development set, including four important shopping skills: (1) Shopping Concept Understanding; (2) Shopping Knowledge Reasoning; (3) User Behavior Alignment; (4) Multi-Lingual Abilities, and five abilities: (i) Generation; (ii) Ranking; (iii) Retrieval; (iv) Multiple-Choice; (v) NER.

- Track4 (Multi-Lingual Abilities): Recognizing the need for multi-lingual models in online shopping, the goal is to evaluate a single model's performance across different shopping locales without re-training, focusing on multi-lingual concept understanding and user behavior alignment.

2 TRAINING DATASET CONSTRUCTION

While LLMs exhibit strong generalization across multiple tasks, they often perform poorly in specific domains due to a lack of relevant knowledge. This competition involves many tasks related to online shopping, and general-purpose models lack knowledge in this area. Therefore, directly adapting a general-purpose model to the online shopping scenario is quite challenging. To improve the model's performance in this domain, we need to inject relevant knowledge into it.

In this challenge, the organizers did not provide a large-scale training dataset. As a result, we constructed our training dataset using publicly available data, our data construction pipeline is shown in Figure 1.

2.1 Development Set Analysis

We analyzed the provided development data to gain insights for constructing the training dataset. The development set comprises 96 data points across 18 different tasks, the distribution of task types is shown in Figure 2.

2.1.1 Shopping Concept Understanding. This track focuses on evaluating the model's ability to understand entities and concepts specific to the online shopping domain, which can be divided into the following sub-tasks:

Table 1: Statistics of Shopbench.

Track	# Tasks	# Questions	# Products	# Product Category	# Attributes	# Reviews	# Queries
All	57	20598	~13300	400	1032	~11200	~4500
Track1	27	11129	~1500	400	1032	~9600	361
Track2	8	3117	~1000	400	~10	/	552
Track3	15	3973	~4800	/	/	1600	~3600
Track4	7	2379	~6000	/	/	/	~520

- Concept Normalization: Given a product name, select the product that represents the same concept as the current product name.
- Elaboration: Given a concept, explain it in plain, understandable, and concise language.
- Extraction and Summarization: Extract and summarize the product names mentioned within the product description.
- Relation Inference: Given four options, select the product category that has a certain attribute.
- Concept Explanation: Describe the concept of the corresponding product.
- Sentiment Analysis: Select 3 snippets from a list that customers are most likely to write in their reviews.

2.1.2 Shopping Knowledge Reasoning. This track aims to evaluate the model's ability to understand complex implicit knowledge in the online shopping domain and to apply this knowledge to various types of reasoning:

- Numerical Reasoning: Extract related numeric information and perform numeric reasoning.
- Commonsense Reasoning: Recommend daily products that are most likely to be purchased based on the current product in the shopping list.
- Implicit, Multi-Hop Reasoning: Understand the implicit, domain-specific knowledge and infer multi-hop relations between shopping entities.

2.1.3 User Behavior Alignment. This track aims to assess a model's ability to understand implicit relationships in user behavior, thereby enabling its recommendation capabilities.

- Recommendation based on user queries: Given a list of product IDs, rank the products according to their relevance to the query.
- Behavior predictions: Given the user's previous actions, infer the next action.
- Recommendation based on purchase histories: Given the products a user has just purchased, predict what they might buy next.
- Sentiment Label Predictions: Given a comment, score it based on its sentiment.

2.1.4 Multi-Lingual Abilities. This track focuses on how models can extend their capabilities to multiple languages to simultaneously meet the needs of global markets.

- Multilingual shopping concept understanding: The tasks in Track1 are expanded to multiple languages.
- Multilingual user behavior alignment: The tasks in Track3 are expanded to multiple languages.

2.2 External Datasets

We collect several external datasets related to Amazon products. Leveraging these datasets, we can construct various tasks and corresponding data tailored for SFT. These external datasets are listed as follows:

- ECInstruct [12]: It is an open-source SFT dataset in the e-commerce domain, encompassing 10 different tasks and comprising 264K instances, including Sequential Recommendation, Query Product Rank, etc.
- Amazon-ESCI [14]: It is a large-scale multilingual query-product dataset, which was employed in the KDD Cup 2022 competition. It includes three sub-tasks: Query-Product Ranking, Multi-class Product Classification, and Product Substitute Identification.
- Amazon-M2 [7]: It is a multilingual session-based recommendation dataset designed for the KDD Cup 2023 competition.
- Amazon Reviews 2023 [4]: It is a comprehensive Amazon product dataset that not only includes user reviews for various products but also provides extensive information such as brand, description, category, features, and co-purchase relationships. It is an updated version of Amazon Reviews 2018.
- OA-Mine & AE-110K [1]: They are two NER datasets in the E-commerce domain, designed to extract categories, brands, target audiences, and other product characteristics from product names.
- Amazon-Category[4]: It provides various products along with their corresponding categories, encompassing items from multiple languages.

2.3 Training Data Construction Strategy

We utilized public datasets and OpenAI's ChatGPT and GPT4[5] [11] for the data construction of our EshopInstruct, whose detailed construction pipeline is shown in Figure 1. Our data construction strategy can be categorized into three main approaches:

- We analyzed 18 tasks and their corresponding data in the development set, using this analysis to generate more data

[4]https://huggingface.co/datasets/ikeno-ada/amazon_category
[5]https://openai.com/api/

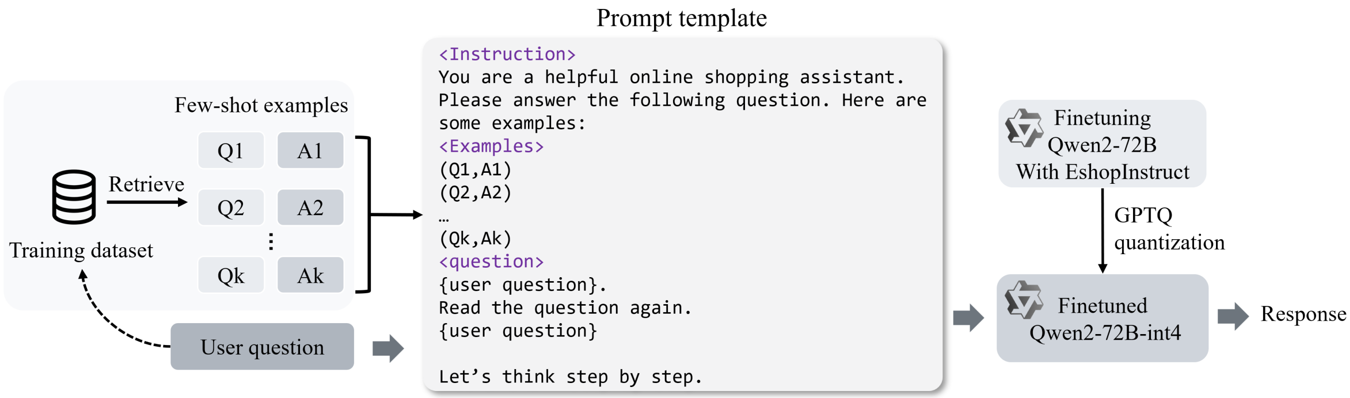

Figure 3: The overall inference framework of our solution. In our prompt construction, we enhanced the model's reasoning capabilities by incorporating few-shot examples retrieved using queries, the "read again" technique, and chain-of-thought reasoning. Additionally, for Qwen2-72B, we applied GPTQ quantization, enabling it to run efficiently on limited resources.

that aligns more closely with the types found in the development set.
- Based on the practical scenarios of each track, we developed additional task types and corresponding data beyond those in the development set.
- To increase the proportion of the real-world data in the SFT dataset and provide more knowledge to the LLMs, we also created a substantial amount of data based on external datasets.

To create data that aligns with the task types in the development set, we adopted two strategies. Firstly, for task types that can be directly constructed or transformed using existing datasets like ECInstruct, we generated the corresponding data directly from these datasets. For example, tasks such as Elaboration, Extraction and Summarization, Relation Inference, Sentiment Analysis in Track1, and Recommendation based on query in Track3, we identified similar data in ECInstruct [12]. For these tasks, we directly extracted data from ECInstruct and transformed them into the standard format. Secondly, for task types where it was challenging to extract data from existing datasets, we utilized LLMs, such as GPT-4, for data generation. For numeric reasoning, implicit and multi-hop reasoning in Track2, as well as user behavior prediction in Track3, we used GPT-4 for data construction. When using GPT-4 to generate this portion of the data, we provided the model with few-shot examples and employed the chain-of-thought method, enabling it to generate the reasoning process to ensure data quality.

Considering that the task types in the development set do not comprehensively cover all scenarios, we constructed additional tasks and corresponding data based on descriptions from various tracks. For instance, we observed that there is no relevant data about the Concept Normalization task in Track1 and the Daily Product Recommendation in Track2 in the development set. Therefore, we constructed corresponding data for them. These data constructions may involve transformations from external datasets or generation by LLMs. Additionally, we referred to the methods in Self-Instruct [16] to generate a portion of the data. Specifically, we used development data as seed data and then utilized GPT-3.5-turbo to generate instructions and corresponding responses based on this data. Subsequently, we employed GPT-4 as a judge to filter the data.

Furthermore, given that much of the data constructed through the first two methods is generated by LLMs, there may be a considerable amount of noise, and scalability could be limited due to cost constraints. Therefore, to introduce a substantial amount of real-world data to our models, we have also constructed additional tasks and corresponding data from external datasets. For example, leveraging the Amazon-ESCI dataset, we constructed tasks such as Query Generation, Related Product Retrieval, etc. It should be noted that in order to enhance our model's multilingual processing capabilities, we incorporated a significant amount of data related to products in various languages other than English during the dataset construction phase.

Following the above strategy, we ultimately obtained approximately 65,000 data entries in EshopInstruct. Moreover, to further augment our training dataset, we strategically sampled a subset of data from the ECInstruct dataset. We then used these data for instruction tuning.

3 INSTRUCTION TUNING

We use instruction tuning to incorporate online shopping-related knowledge into the LLMs and enhance their instruction-following capabilities. Given the size of our constructed dataset (65,000 entries) and our limited training resources, we adopted the LoRA (Low-Rank Adaptation) [5] fine-tuning method, following the standard approach of auto-regressive language modeling. During phases 1 and 2 of the challenge, we experimented with four models[6] of different sizes: Mistral-7B[7] [6], LLama3-8B[8] [2], and Qwen2-7B/72B[9] [19]. Some key training hyper-parameters are listed in Table 2. We used the standard AdamW optimizer [9] for supervised fine-tuning (SFT) optimization, with a cosine learning rate schedule, a peak learning rate of $4 \times e^{-5}$, and a 10% warmup ratio. All the models were trained with multiple NVIDIA A800 80G GPUs. For models with

[6]All models used are chat or instruct models
[7]https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2
[8]https://github.com/meta-llama/llama3
[9]https://github.com/QwenLM/Qwen2

Table 2: Hyperparameter for Instruction-tuning.

Configuration	Value
Model	Qwen2-72B
Number of epochs	2
Learning Rate	$4e$-5
Max Length	2048
Devices	8 NVIDIA A800 GPUs (80GB)
LR Scheduler	Cosine
Warmup Raion	0.1
Total Batch Size	256
Optimizer	AdamW [9]
Lora Rank	8
Lora Target	$q_{proj}, k_{proj}, v_{proj}$

fewer than 10 billion parameters, such as Mistral-7B, LLama3-8B, and Qwen2-7B, we trained on a single GPU without quantization. For the Qwen2-72B model, we used bf16 precision for LoRA fine-tuning, employed DeepSpeed's ZeRO Stage3 [13] for fine-tuning across four GPUs, and then used GPTQ to quantize the parameters to 4-bit precision.

4 INFERENCE

In this section, we will introduce our inference strategy, which comprises two key components: quantization and prompt engineering. Our overall inference strategy pipeline is shown in Figure 3.

4.1 Quantification

During the challenge, the submission will run on a T4 GPU with 16GB of memory. In Phase 2, four T4 GPUs will be provided, which means only 64GB of GPU memory will be available. To experiment with larger models (such as models with 72B parameters) and minimize the reduction in model capability while reducing the required GPU memory as much as possible, we leveraged quantization techniques. Specifically, we adopted GPTQ quantization [3], a post-training quantization method where each row of the weight matrix is independently quantized to int4 to reduce error but restored to fp16 during inference for better performance. In this challenge, we only applied quantization to the Qwen2-72B model. After training, we utilized 1,000 data samples from Alpaca [15] generated by GPT-4 for quantization calibration of the model.

4.2 Prompting Strategies

Through the analysis of the development set, we found that many tasks in the challenge involve reasoning. To improve the model's performance on these tasks, we introduced Chain of Thought (CoT) [17], a technique that can significantly enhance the complex reasoning abilities of large language models. we implemented a simple zero-shot Chain-of-Thought in our solution. Additionally, we retrieved the three most relevant samples from the constructed training dataset as few-shot examples, which yielded better results.

The test data can be roughly divided into multiple-choice and non-multiple-choice types. We adopt different processing measures and prompts for these two types of data. For questions that may involve reasoning, we encourage the model to think more deeply and use regular expressions to extract the final answer. For generation-related questions, we let the model directly output the final result.

Considering the importance of user input in online shopping scenarios, we have also implemented a simple and effective prompting method called Re-Reading [18] which entails re-reading the question to enhance reasoning capabilities in Large Language Models.

5 RESULTS

In this section, we will compare and analyze the performance of different models across the five tracks, as shown in Table 3. Overall, the size of the model parameters has a considerable impact on performance, with larger parameter models generally performing better. **(A)** shows relatively weaker performance across Track1 to Track4, especially on Track2, where it scored only 0.529. In comparison, **(B)** shows improved performance across all tracks, particularly on Track1 and Track5. Despite having the same 7B parameters as **(A)**, **(C)** performs well across all available tracks, especially on Track2 and Track4. Comparing **(B)** and **(C)**, we find that Qwen2-7B owns a greater potential than LLama3-8B in providing e-commerce shopping assistance. Therefore, we chose the Qwen2 series as our backbone model since it performs relatively well.

With 72B parameters, **(D)** demonstrates excellent performance across all tracks, particularly on Track1 to Track3, achieving high scores of 0.786, 0.716, and 0.706, respectively. However, its performance slightly drops on Track4 to 0.654, but it remains at a high level. To incorporate the domain knowledge of e-commerce, we fine-tuned Qwen2-72B on ECInstruct and our constructed EshopInstruct, respectively. Comparing **(E)** and **(F)**, we can see that **(F)** consistently and considerably outperforms **(E)** in all tracks, indicating the superiority of supplementing e-commerce shopping tasks with our EshopInstruct. It is worth mentioning that we find LLMs fine-tuned on ECInstruct perform badly on generation tasks. The performance of **(F)** on specific tasks is detailed in Table 4.

By utilizing our carefully constructed training dataset EshopInstruct for instruction fine-tuning and employing effective inference strategies, we ultimately secured 3rd place overall in the Amazon KDD Cup 2024 Challenge, 3rd place in Track1, 2nd place in Track4, and ranked within the top 5 for the remaining tracks.

6 CONCLUSION

In this paper, we present our solution for the Amazon KDD Cup 2024 Challenge. We constructed a multi-task instruction dataset called **EshopInstruct**, which contained 65,000 samples tailored to online shopping scenarios. In addition, we utilized **EshopInstruct** for instruction tuning on large language models, resulting in knowledgeable shopping assistants named **LLaSA**. To optimize inference performance with limited resources, we employed GPTQ quantization and prompting strategies such as Chain-of-Thought and Re-Reading. Evaluation results demonstrated the effectiveness of our approach, securing 3rd place on the overall leaderboard and ranking within the top 5 in each track. Especially in track4 (Multi-lingual Abilities), we obtained the best student team award.

Table 3: Overall performance on different tracks, where the best results are boldfaced and the second-best results are underlined. "-" denotes missing experimental data due to the unstable evaluation system.

Model	Quantization	Track1	Track2	Track3	Track4	Track5
(A) Mistral-7B+ECInstruct	none	0.702	0.529	0.602	0.635	-
(B) LLama3-8B+EshopInstruct	none	0.741	0.6	0.625	0.651	0.67
(C) Qwen2-7B+EshopInstruct	none	0.744	0.629	0.614	0.664	-
(D) Qwen2-72B	GPTQ-int4	0.786	0.716	0.706	0.654	0.722
(E) Qwen2-72B+ECInstruct	GPTQ-int4	0.801	0.719	0.703	0.686	0.747
(F) Qwen2-72B+EshopInstruct	GPTQ-int4	**0.824**	**0.747**	**0.713**	**0.735**	**0.763**

Table 4: Best Model's Detailed Performance on different task types. "-" means that this track does not evaluate this type of task.

Track	Generation	Multiple-Choice	NER	Retrieval	Ranking	Overall
Track1	0.732	0.860	0.789	0.858	-	0.824
Track2	-	0.770	-	0.588	-	0.747
Track3	0.618	0.695	-	0.813	0.845	0.713
Track4	0.482	0.838	-	-	0.83	0.735
Track5	0.763	0.793	0.802	0.765	0.659	0.844

References

[1] Alexander Brinkmann, Roee Shraga, and Christian Bizer. 2024. Product Attribute Value Extraction using Large Language Models. arXiv:2310.12537 [cs.CL] https://arxiv.org/abs/2310.12537

[2] Abhimanyu Dubey, Abhinav Jauhri, and et al. 2024. The Llama 3 Herd of Models. arXiv:2407.21783 [cs.AI] https://arxiv.org/abs/2407.21783

[3] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2022. GPTQ: Accurate Post-training Compression for Generative Pretrained Transformers. *arXiv preprint arXiv:2210.17323* (2022).

[4] Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. 2024. Bridging Language and Items for Retrieval and Recommendation. arXiv:2403.03952 [cs.IR] https://arxiv.org/abs/2403.03952

[5] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685* (2021).

[6] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825* (2023).

[7] Wei Jin, Haitao Mao, Zheng Li, Haoming Jiang, Chen Luo, Hongzhi Wen, Haoyu Han, Hanqing Lu, Zhengyang Wang, Ruirui Li, Zhen Li, Monica Xiao Cheng, Rahul Goutam, Haiyang Zhang, Karthik Subbian, Suhang Wang, Yizhou Sun, Jiliang Tang, Bing Yin, and Xianfeng Tang. 2023. Amazon-M2: A Multilingual Multi-locale Shopping Session Dataset for Recommendation and Text Generation. arXiv:2307.09688 [cs.IR] https://arxiv.org/abs/2307.09688

[8] Yangning Li, Shirong Ma, Xiaobin Wang, Shen Huang, Chengyue Jiang, Hai-Tao Zheng, Pengjun Xie, Fei Huang, and Yong Jiang. 2023. EcomGPT: Instruction-tuning Large Language Models with Chain-of-Task Tasks for E-commerce. arXiv:2308.06966 [cs.CL] https://arxiv.org/abs/2308.06966

[9] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=Bkg6RiCqY7

[10] Shirong Ma, Shen Huang, Shulin Huang, Xiaobin Wang, Yangning Li, Hai-Tao Zheng, Pengjun Xie, Fei Huang, and Yong Jiang. 2023. EcomGPT-CT: Continual Pre-training of E-commerce Large Language Models with Semi-structured Data. arXiv:2312.15696 [cs.CL] https://arxiv.org/abs/2312.15696

[11] OpenAI. 2023. GPT-4 Technical Report. *CoRR* abs/2303.08774 (2023). https://doi.org/10.48550/ARXIV.2303.08774 arXiv:2303.08774

[12] Bo Peng, Xinyi Ling, Ziru Chen, Huan Sun, and Xia Ning. 2024. eCeLLM: Generalizing Large Language Models for E-commerce from Large-scale, High-quality Instruction Data. In *Forty-first International Conference on Machine Learning*. https://openreview.net/forum?id=LWRI4uPG2X

[13] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. ZeRO: memory optimizations toward training trillion parameter models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2020, Virtual Event / Atlanta, Georgia, USA, November 9-19, 2020*, Christine Cuicchi, Irene Qualters, and William T. Kramer (Eds.). IEEE/ACM, 20. https://doi.org/10.1109/SC41405.2020.00024

[14] Chandan K. Reddy, Lluís Màrquez, Fran Valero, Nikhil Rao, Hugo Zaragoza, Sambaran Bandyopadhyay, Arnab Biswas, Anlu Xing, and Karthik Subbian. 2022. Shopping Queries Dataset: A Large-Scale ESCI Benchmark for Improving Product Search. arXiv:2206.06588 [cs.IR] https://arxiv.org/abs/2206.06588

[15] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html* 3, 6 (2023), 7.

[16] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-Instruct: Aligning Language Models with Self-Generated Instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 13484–13508. https://doi.org/10.18653/v1/2023.acl-long.754

[17] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

[18] Xiaohan Xu, Chongyang Tao, Tao Shen, Can Xu, Hongbo Xu, Guodong Long, and Jian-guang Lou. 2023. Re-reading improves reasoning in language models. *arXiv preprint arXiv:2309.06275* (2023).

[19] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671* (2024).

[20] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A Survey of Large Language Models. arXiv:2303.18223 [cs.CL] https://arxiv.org/abs/2303.18223