EC-Guide: A Comprehensive E-Commerce Guide for Instruction Tuning and Quantization by ZJU-AI4H

Zhaopeng Feng*
Zijie Meng*
Zuozhu Liu†
Zhejiang University-University of Illinois at Urbana Champaign Institute, Zhejiang University
Jiaxing, Zhejiang, China
{zhaopeng.23,zijie.22,zuozhuliu}@intl.zju.edu.cn

Abstract

Large language models (LLMs) have attracted considerable attention in various fields for their cost-effective solutions to diverse challenges, especially with advancements in instruction tuning and quantization. E-commerce, with its complex tasks and extensive product-user interactions, presents a promising application area for LLMs. However, the domain-specific concepts and knowledge inherent in e-commerce pose significant challenges for adapting general LLMs. To address this issue, we developed EC-Guide[1], a comprehensive e-commerce guide for instruction tuning and quantization of LLMs. We also heuristically integrated Chain-of-Thought (CoT) during inference to enhance arithmetic performance. Our approach achieved the 2nd place in Track 2 and 5th place in Track 5 at the Amazon KDD Cup'24. Additionally, our solution is model-agnostic, enabling effective scalability across larger systems.

CCS Concepts

• **Computing methodologies** → **Natural language processing**; • **Applied computing** → **Electronic commerce**.

Keywords

Online shopping, large language models

ACM Reference Format:
Zhaopeng Feng, Zijie Meng, and Zuozhu Liu. 2024. EC-Guide: A Comprehensive E-Commerce Guide for Instruction Tuning and Quantization by ZJU-AI4H. In *Proceedings of the 30th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (KDDCup'24).* ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Current techniques struggle to understand the nuances of specific shopping terms, customer behaviors, preferences, and the diverse array of products and languages [13]. With the advent of large language models (LLMs), there is a growing belief in their capability to tackle these challenges. To this end, the organizers of Amazon KDD Cup'24 introduced ShopBench, a benchmark designed to simulate the complexities of online shopping. It includes 57 tasks and approximately 20,000 questions sourced from real-world Amazon shopping data. The competition includes 5 Tracks:

- Shopping Concept Understanding: Decoding complex shopping concepts and terminologies.
- Shopping Knowledge Reasoning: Making informed decisions based on shopping knowledge.
- User Behavior Alignment: Understanding dynamic customer behavior.
- Multilingual Abilities: Shopping across languages.
- All-Around: Solving all questions with a single solution.

Our team "ZJU-AI4H" finally achieved 2nd place in Track 2 and 5th place in Track 5. Our solution for both Tracks can be summarized as three key steps: dataset construction, instruction tuning, and post training quantization. Especially, we noticed that Chain-of-Thought (CoT) [9] can significantly boost the arithmetic performance.

2 Related Works

As a data-driven technology, LLMs exhibit exceptional performance, promoting the development of numerous datasets. GSM8K [1] focuses on grade school mathematics requiring multi-step reasoning. ECInstruct [13] introduces diverse e-commerce subtasks to guide the instruction tuning of general LLMs. Amazon-M2 [8] is a multilingual product recommendation dataset enhancing the understanding of user preferences. Additionally, other datasets [3, 11, 12, 14–17] provide varied formats for evaluating LLMs. To further explore e-commerce applications, we developed EC-Guide, a comprehensive guide for instruction tuning and quantization.

Apart from the rapid development of datasets, researchers are increasingly focusing on training and inference techniques. LoRA [7] significantly reduces fine-tuning costs by freezing LLM's weights and injecting a learnable matrix. QLoRA [2] further reduces costs by introducing quantization techniques. GPTQ [4] is a one-shot weight quantization for LLMs by leveraging approximate second-order information to compress models down to lower bits (like 4-bit) per weight with minimal impact on performance. CoT [9] enhances the reasoning ability of LLMs through appropriate prompts without additional training. Based on these methods, we adapted general LLM to specific domains through instruction tuning and deployed by quantization and CoT within limited resources.

*Equal contribution.
†Corresponding author.
[1]https://github.com/fzp0424/EC-Guide-KDDUP-2024

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
KDDCup'24, Aug 25–29, 2024, Barcelona, Spain
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

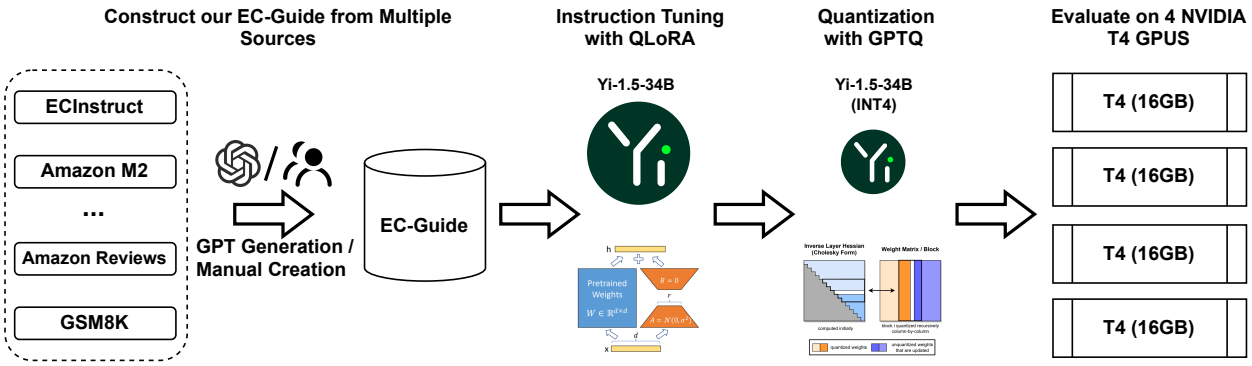

Figure 1: Illustration of our solution. We first construct our EC-Guide (74k examples for 24 sub-tasks across 5 types) dataset from multiple sources [8, 13]. Then we finetune Yi-1.5-34B [19] with QLoRA [2] and quantize it with GPTQ [4].

3 Methodology

We first constructed our instruction tuning dataset, EC-Guide, by mining existing datasets [8, 13], and utilizing ChatGPT[2] for data generation. We also expanded the official development dataset to enhance evaluation. Through instruction tuning, we embedded knowledge specifically relevant to e-commerce tasks into existing LLMs. However, in Round 2, solutions had access to 4×NVIDIA T4 (16GB) GPUs[3], making it impractical to directly deploy powerful LLMs like Yi-1.5-34B [19] or Qwen2-72B [18] without quantization. During inference, we applied CoT to further boost LLM calculation performance. Details are as follows.

3.1 Dataset Construction

Due to the lack of large-scale training datasets for the Amazon KDD Cup'24, we employed paraphrasing and extending existing datasets [8, 13] to develop EC-Guide, encompassing 74k examples across five task types: Generation, Multiple Choice Question, Retrieval, Ranking, and Named Entity Recognition. Statistic details are shown in Table 1. Furthermore, we expanded the official development dataset from 96 to 506 examples by strategies such as option reordering and segment sampling from our EC-Guide, which significantly enhances the robustness of the development dataset to effectively evaluate our model.

3.1.1 Generation. In the generation task, each question is paired with a specific instruction, and the model is to generate text that precisely follows these instructions. There are multiple types of generation questions, including elaboration, extraction, summarization and translation.

For the elaboration set, we introduced the following subtasks: 1. Product Elaboration (PE): We utilized the "Attribute Value Extraction" from ECInstruct [13] to collect type-related attributes, then filtered to obtain 479 entities, and used ChatGPT to generate detailed descriptions for each entity. 2. Product Question and Answer (PQA): We selected samples from McAuley and Yang [11] with only one answer, filtered them based on answer length, and finally obtained 6,834 QA pairs. 3. Category Recognition (CR): We extracted product-category relationships from McAuley et al. [10], and sampled 1,000 entries to prompt the model to recognize categories from product titles. 4. Explaining Pairwise Fashion (EPF): We modified prompts from the PFE dataset [17] and filtered based on response length, yielding 3,000 instances to evaluate whether the mentioned clothes matched. 5. Explaining Bought Together (EBT): We directly integrated 2,315 entries from IntentionQA [3] to assess the reasons for buying two products together.

For the extraction set, we introduced two subtasks: extracting review keyphrases (ERK) and extracting product keyphrases (EPK), which focus on extracting keyphrases related to aspects or features from reviews and products, respectively. ERK is based on the "Sentiment Analysis" of ECInstruct [13], where we used ChatGPT to extract aspect-keyphrase pairs from reviews and obtained 1,000

Table 1: Statistics of EC-Guide.

Task type	Sub-tasks	# of examples	Source
Generation	Product Elaboration (PE)	479	[13]
	Product Question and Answer (PQA)	6,834	[11]
	Category Recognition (CR)	1,000	[10]
	Explaining Pair Fashion (EPF)	3,000	[17]
	Explaining Bought Together (EBT)	2,315	[3]
	Extract Review Keyphrase (ERK)	1,000	[13]
	Extract Product Keyphrase (EPK)	3,000	[17]
	Product Keyword Summarization (PKS)	1,296	[6, 13, 14]
	Review Title Summarization (RTS)	1,455	[6, 12]
	Multilingual Translation (MT)	2,997	[5, 8]
Multiple Choice Question	Select Product based on Attribute (SPA)	520	[13]
	Select Attribute based on Product (SAP)	1,385	[6]
	Product Relation Prediction (PRP)	1,499	[13]
	Query Product Relation Prediction (QPRP)	2,150	[14]
	Query Product Relation Judgement (QPRJ)	501	[13]
	Sentiment Analysis (SA)	3,500	[12, 13]
	Product Keyword Summarization (PKS)	271	[14]
	Multilingual Description Matching (MDM)	500	[6]
	Arithmetic and Commonsense Reasoning (ACR)	7,184	[1, 16]
Retrieval	Inferring Potential Purchases (IPP)	10,774	[8, 13]
	Retrieving Review Snippets (RRS)	810	[6]
	Retrieving Review Aspects (RRA)	1,000	[6]
	Category Recognition (CR)	7,500	[10]
	Product Recognition (PR)	2,297	[10]
Ranking	Query Product Ranking (QPR)	4,008	[14]
Named Entity Recognition	Named Entity Recognition (NER)	7,429	[6, 13, 15]
ALL	-	74,704	-

[2]https://openai.com/chatgpt/
[3]https://www.nvidia.cn/data-center/tesla-t4/

samples. EPK is sourced from the PFE dataset [17], where we filtered out data items with overly short descriptions and extracted feature-keyphrase pairs, yielding 3,000 cases.

The summarization set includes two subtasks: 1. Product Keyword Summarization (PKS) involves summarizing keywords list to encapsulate product information. We sampled product information from existing datasets [6, 13, 14], and used ChatGPT for annotation to produce 1,296 entries. 2. Review Title Summarization (RTS) aims to create concise titles for reviews. We filtered previous dataset [6, 12] to obtain 1,455 cases.

For our translation set, we employed ChatGPT to translate product titles between English and several other languages, including Spanish, German, Italian, Japanese, and French. We also translated from other languages into English, resulting in a total of 2,000 translation pairs. Additionally, we utilized samples from Flores [5] to enhance our translation tasks, resulting in 997 additional pairs.

3.1.2 Multiple Choice Question. Multiple choice question, widely used for their objective assessment, require selecting the correct answer from the choices list identified by Arabic numerals. Specifically, we include the following subtasks: 1. Select Product based on Attribute (SPA): We sampled attributes from the "Attribute Value Extraction" of ECInstruct [13], and generated product options, resulting in 520 data entries. 2. Select Attribute based on Product (SAP): Using product titles extracted from Amazon Reviews [6], we generated multiple choice questions about attributes with ChatGPT, yielding 1,385 data entries. 3. Product Relation Prediction (PRP): We extracted 1,499 cases from "Product Relation Prediction" and "Product Matching" in ECInstruct [13]. The former predicts potential purchase or browsing intentions between two products, while the latter judges whether two products are the same. 4. Query Product Relation Prediction (QPRP): Based on relationships between queries or products in the Shopping Queries Dataset [14], we constructed 2,150 cases. 5. Query Product Relation Judgement (QPRJ): Sampled from "Product Substitute Indentification" of ECInstruct [13], QPRJ includes 501 examples. 6. Sentiment Analysis (SA): We directly extracted 3,500 data entries from "Sentiment Analysis" of ECInstruct [13] and Women's Clothing E-Commerce Reviews [12]. 7. Product Keyword Summarization (PKS): Using a method similar to PKS in Section 3.1.1, we obtained 271 cases. 8. Multilingual Description Matching (MDM): MDM comprises 300 cases, aiming to match product titles with the correct features in multiple languages. We filtered products from Amazon Reviews [6] and translated their features into various languages (English, Spanish, German, Italian, Japanese, and French) with ChatGPT. And then we sampled three features from other products to form the options for each final question. 9. Arithmetic and Commonsense Reasoning (ACR): We obtained 7,184 items from GSM8K [1] and CommonsenseQA [16]. And we annotated the rationale with ChatGPT for CommonsenseQA.

3.1.3 Retrieval. In this task, the model's objective is to retrieve answers from a list of candidate items to meet specific requirements. The subtasks are as follows: 1. Inferring Potential Purchases (IPP): There are two main categories multi-to-one and one-to-multi in IPP. For the multi-to-one, which predicts the next purchase item based on multiple items in the purchase history, we filtered cases from the "Sequential Recommendation" in ECInstruct [13] to obtain 3,950 cases. For the one-to-multi, which predicts multiple potential purchase items based on a single purchased item, we mined the data from Amazon-M2 [8], resulting in a total of 6,824 cases. 2. Retrieving Review Snippets (RRS): We sampled Amazon Reviews [6] to obtain 3,000 products and their corresponding reviews. Then we generated multiple aspect-snippet pairs from the reviews with ChatGPT. Finally, considering the proportion of positive and negative reviews, we created a total of 810 data entries. 3. Retrieving Review Aspects (RRA): Using the same data source as RRS, we randomly combined reviews and aspects to obtain 1,000 data entries. 4. Category Recognition (CR): Similar to the CR in Section 3.1.1, we constructed retrieval lists by randomly sampling categories, resulting in 7,500 data items. 5. Product Recognition (PR): Using the data from CR, we reversed the product-category table to create a category-product dictionary, resulting in 2,297 data entries.

3.1.4 Ranking. In this task, the model's goal is to reorganize items in the candidate list based on how well they meet the requirements. We utilized the Shopping Queries Dataset [14], which assesses the relevance between the queries and products using ESCI judgments (Exact, Substitute, Complement, Irrelevant). Furthermore, leveraging the multilingual source dataset, our ranking instances comprises 2,064 queries in English, 790 in Japanese, and 1,184 in Spanish.

3.1.5 Named Entity Recognition. Named Entity Recognition (NER) is an important benchmark for evaluating LLMs and is prevalent across various domains. In this task, we extracted 1,446 entries from "Attribute Value Extraction" of ECInstruct [13], 1,099 entries from Amazon Reviews [6], and 4,884 entries from Rif'at et al. [15].

3.2 Instruction Tuning

Instruction tuning enhances the model's ability to generate contextually appropriate answers based on the given instruction, which allows the model to understand and execute tasks specified in the instructions. During training, the objective is to optimize the model π_θ using a curated dataset $\mathcal{D}_{IT} = \{x^{(i)}, y^{(i)}\}_{i=1}^{N}$ by minimizing the negative log-likelihood of the output y:

$$\mathcal{L}(\theta; \mathcal{D}_{IT}) = -\mathbb{E}_{(x,y)\sim\mathcal{D}_{IT}} \left[\log \pi_\theta(y \mid x)\right] \tag{1}$$

Before tuning, we organized raw data in [*Instruction*, *Input*, *Output*], concatenated them, and applied chat template to form $\mathcal{D}_{IT}$.

Deploying models with larger parameters necessitates effective compression methods, such as model quantization, to accelerate inference and save memory. Quantization involves converting high-precision parameters to lower-precision formats without altering the model's parameters or architecture.

3.3 Post Training Quantization

It is widely accepted that larger scale generally leads to greater model capabilities [18, 19]. However, LLMs typically store parameters in high-precision floating-point formats, demanding significant computational resources for inference. Deploying larger models necessitates effective compression methods, such as quantization, to accelerate inference and save memory. Therefore, we utilize GPTQ [4], which is a training-free quantization for LLMs to achieve high accuracy and efficiency. We detailed the memory of weights loaded in Table 2. In particular, we equally sample training examples based on task types from our training set, ultimately allocate

Table 2: Results of various models under the same training settings. The loading weights are averaged across 4 GPU.

Model	PTQ (dataset)	Loading Weights	Track 2		Track 5	
			Dev	Round 2	Dev	Round 2
Qwen2-7B	\	3.55 GB	0.6562	0.6706	0.7092	0.7224
Yi-1.5-9B	\	4.45 GB	0.7031	0.6628	0.7153	0.7048
GLM-4-9B	\	6.65 GB	0.7187	0.6875	0.7299	0.7163
Yi-1.5-34B	INT4 (C4)	4.55 GB	0.6562	-	0.7017	-
	INT4 (Ours)	4.55 GB	0.7500	0.7452	0.7403	0.7323
	\	16.04 GB	0.7968	-	0.7583	-

Table 3: Results of submission with Yi-1.5-34B.

Model	Training Set (ratio, # of cases)	Track 2	Track 5
Yi-1.5-34B (INT4)	$\lambda 1$: $\lambda 2$: $\lambda 3$: $\lambda 4$: $\lambda 5$ = 0.31: 0.23: 0.30: 0.05: 0.10, 74704	0.7487	**0.7462**
	$\lambda 1$: $\lambda 2$: $\lambda 3$: $\lambda 4$: $\lambda 5$ = 0.33: 0.40: 0.17: 0.02: 0.08, 15764	**0.7834**	0.7386
	$\lambda 1$: $\lambda 2$: $\lambda 3$: $\lambda 4$: $\lambda 5$ = 0.46: 0.24: 0.14: 0.04: 0.12, 15043	0.7440	0.7402

1,846 examples for quantization. Additionally, we also compare our quantization with samples from the external data source like C4[4] as shown in Table 2.

3.4 Chain-of-Thought (CoT) Reasoning

After instruction tuning and quantization on the elaborately designed EC-Guide, we introduced CoT [9] in the inference only for multiple choice questions involving calculations. Specifically, we employed a heuristic strategy to determine if a question belong to arithmetic-based by counting the number of digits. We then prefixed the prompt with "Let's think step by step.", stimulating the model to generate a rationale that leads to the correct answer. Notably, we observed that applying CoT to Track 2 in Round 1 increased the score from 0.7417 to 0.7908 with the same model.

4 Experiments

We finetuned all models on 4×A40 GPUs or 8×RTX3090 GPUs with QLoRA. We deployed LLMs by vllm[5], which utilizes PagedAttention to manage attention keys and values, to accelerate inference.

Table 2 demonstrates the performance of different models with the same training setting. Notably, Yi-1.5-34B achieved the highest scores across both Track 2 and 5 in both development and official test set. We also observed that models quantized using out-of-domain datasets C4 exhibited significant performance drops compared to those using in-domain sampled data. Table 3 presents our ablation study, which highlights the influence of different ratios of task types in training set, and suggests that smaller training sets sometimes outperform larger ones in specific scenarios. This observation leads us to hypothesize about the existence of a performance trade-off among different tasks.

5 Conclusion

The Amazon KDD Cup'24 competition presents a unique challenge by focusing on the application of LLMs in E-commerce across multiple tasks. Our solution for addressing Tracks 2 and 5 involves a comprehensive pipeline encompassing dataset construction, instruction tuning and post-training quantization. The core of our strategy is EC-Guide specifically tailored for E-commerce scenarios. Notably, we heuristically integrated CoT reasoning to enhance the arithmetic capabilities of LLMs, resulting in improved performance in both Tracks.

[4]https://huggingface.co/datasets/allenai/c4
[5]https://github.com/vllm-project/vllm

References

[1] 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 arXiv:2110.14168* (2021).

[2] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems* 36 (2024).

[3] Wenxuan Ding, Weiqi Wang, Sze Heng Douglas Kwok, Minghao Liu, Tianqing Fang, Jiaxin Bai, Junxian He, and Yangqiu Song. 2024. IntentionQA: A Benchmark for Evaluating Purchase Intention Comprehension Abilities of Language Models in E-commerce. *arXiv preprint arXiv:2406.10173* (2024).

[4] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2022. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323* (2022).

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

[6] Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. 2024. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952* (2024).

[7] 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).

[8] Wei Jin, Haitao Mao, Zheng Li, Haoming Jiang, Chen Luo, Hongzhi Wen, Haoyu Han, Hanqing Lu, Zhengyang Wang, Ruirui Li, et al. 2024. Amazon-m2: A multilingual multi-locale shopping session dataset for recommendation and text generation. *Advances in Neural Information Processing Systems* 36 (2024).

[9] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems* 35 (2022), 22199–22213.

[10] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. 43–52.

[11] Julian McAuley and Alex Yang. 2016. Addressing complex and subjective product-related queries with customer reviews. In *Proceedings of the 25th International Conference on World Wide Web*. 625–635.

[12] nicapotato. 2018. Women's E-Commerce Clothing Reviews. https://www.kaggle.com/datasets/nicapotato/womens-ecommerce-clothing-reviews

[13] 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. *arXiv preprint arXiv:2402.08831* (2024).

[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 preprint arXiv:2206.06588* (2022).

[15] Muhammad Rif'at, Rahmad Mahendra, Indra Budi, and Haryo Akbarianto Wibowo. 2018. Towards product attributes extraction in Indonesian e-commerce platform. *Computación y Sistemas* 22, 4 (2018), 1367–1375.

[16] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937* (2018).

[17] Yu Wang, Zexue He, Zhankui He, Hao Xu, and Julian McAuley. 2024. Deciphering Compatibility Relationships with Textual Descriptions via Extraction and Explanation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 9133–9141.

[18] 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).

[19] Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, et al. 2024. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652* (2024).