Efficient Detection of Toxic Content in Chinese Texts with Smaller-Scale Language Models: A Feasibility Study

Jiaxiang Gao
The Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, Guangdong, China
jgao329@connect.hkust-gz.edu.cn

ABSTRACT

Online social media platforms have emerged as vital conduits for communication, sharing insights, and forging connections between individuals and organizations. Identifying toxic content is crucial for online platforms to shield users from damaging and offensive material, thereby fostering a safer and more constructive environment. This study explores the application of smaller-scale language models for detecting toxic content in Chinese texts, aiming to provide a practical alternative to large language models (LLMs) that require substantial computational resources. By utilizing a series of experiments, the paper evaluates the effectiveness of these models in recognizing various forms of toxic content, including hate speech, biased content, and cyberbullying. The research focuses on fine-tuning these models with high-quality datasets specifically curated for this purpose, including ToxiCN and Chinese Offensive Language Dataset (COLD). The study demonstrates that smaller-scale models, when properly fine-tuned, can achieve competitive accuracy in toxic content detection, thereby offering a viable solution for real-time applications on platforms where resources are constrained.

KEYWORDS

Toxic Content Detection, Smaller-Scale Language Models, Machine Learning, LLM

1 INTRODUCTION

Online social media platforms have emerged as vital conduits for communication, sharing insights, and forging connections between individuals and organizations. Yet, they are frequently exploited to disseminate hateful or toxic content, and to facilitate bullying and intimidation. Identifying toxic content is crucial for online platforms to shield users from damaging and offensive material, thereby fostering a safer and more constructive environment. Common toxic content categories include hate speech, biased content, sexual content, violent content, bullying content, etc [1]. Given the vast quantities of data online, manually screening each item for toxicity is unfeasible. Consequently, there is a compelling need for automated systems to detect toxic content, which remains a practical challenge due to the evolving nature of hate speech. The characteristics of hate speech can change over time, often depend on the context, and are inherently subjective.

To address this, machine learning (ML) approaches, particularly those using supervised learning, have been extensively adopted. Language Models (LMs), when fine-tuned on specific datasets, have achieved state-of-the-art results in automating the detection of toxic content [2, 3]. However, these supervised ML solutions encounter several challenges. They require labeled training data, which can be difficult to procure, especially for subtle forms of toxic content, due to the absence of standardized definitions. Furthermore, LMs that have been fine-tuned may overfit the training datasets, limiting their transferability to other datasets.

The advent of Large Language Models (LLMs) has introduced potential solutions to these challenges through their robust zero-shot and few-shot learning capabilities, which allow for better generalization across various contexts [4, 5]. Despite the effectiveness of LLMs in enhancing toxic content detection, their deployment in real-world settings is often hindered by high computational demands, particularly for models of the scale of 7 billion parameters or more.

This paper primarily focuses on fine-tuning smaller-scale models, engaging in prompt engineering, and experimenting with secondary fine-tuning of models. The aim is to explore maintaining the advantages and accuracy of larger models in detecting toxic content, while reducing the model size. This approach attempts to balance performance with practicality, making it suitable for real-time applications across various online platforms.

2 LITERATURE REVIEW

2.1 Text Classification

Identifying toxic content is a key application of text classification within the field of natural language processing. This method involves organizing texts or documents, known as 'instances', into predefined categories or classes [6]. It typically employs a supervised learning approach, where a classifier is trained on data previously annotated by human experts. The process begins with data preprocessing to condense the text, utilizing techniques such as removing stopwords and applying stemming. The next step, feature extraction, converts instances into a vector space model that facilitates computational analysis. Finally, using these vectors, a classifier is trained to categorize new and similar text instances.

2.2 Toxic Content Classification

The field of toxic content detection encompasses two main research strategies. The initial strategy focuses on developing benchmark datasets. This can be achieved through methods such as crowdsourcing to annotate text data [7–9] or utilizing advanced machine learning methods to generate high-quality datasets efficiently [10]. The second strategy involves refining language models (LMs) to specifically target toxic content. For example, Caselli et al. [2] introduced HateBERT, an adaptation of the BERT model specially trained to detect abusive language, which demonstrates improved performance over the original BERT. Further advancements were made by Kim, Park, and Han [3], who employed contrastive learning to enhance HateBERT's performance across various datasets. Additionally, recent innovations have applied large language models (LLMs) for detecting toxic content; for instance, Wang and Chang [4] used a generative prompt-based approach with LLMs. Moreover, Zhang et al. [5] developed a novel, interpretable method called unified language checking (UniLC), which enhances the capability of LLMs in detecting misinformation, stereotypes, and hate speech through improved in-context learning.

3 DATASET AND MODEL

3.1 Dataset

3.1.1 ***ToxiCN Dataset****[11].* ToxiCN dataset is constructed by crawling posts from online platforms Zhihu and Tieba, focusing on sensitive topics such as gender, race, region, and LGBTQ. After data cleaning, the dataset consists of 12,011 comments, 9,600 from the training set and 2,411 from the testing set. The dataset includes various toxic categories such as hate speech against single and multi-class attacked groups, with about 15% of samples containing attacks and discrimination against multiple groups. Figure 1 illustrates the distribution of labels in the ToxiCN Test Dataset and COLD Test datasets. In the ToxiCN Test dataset, 47.16% of the content is non-toxic, while 52.84% is classified as toxic.

3.1.2 ***Chinese Offensive Language Dataset (COLD)****[12].* COLD contains 37,480 comments with binary offensive labels and covers diverse topics such as race, gender, and region. The dataset is crucial for offensive language detection research in Chinese, addressing the scarcity of reliable datasets in this area . The training data in COLD consists of 32,157 semi-automatically labeled comments, while the test set contains 5,323 manually labeled data with fine-grained categories including attacking individuals, attacking groups, anti-bias, and other non-offensive content. The data in COLD is collected from social platforms like Zhihu and Weibo, with users posting on these platforms considered as the speakers who generate the data. As shown in Figure 1, the COLD Test dataset has 60.42% non-toxic content and 39.58% toxic content.

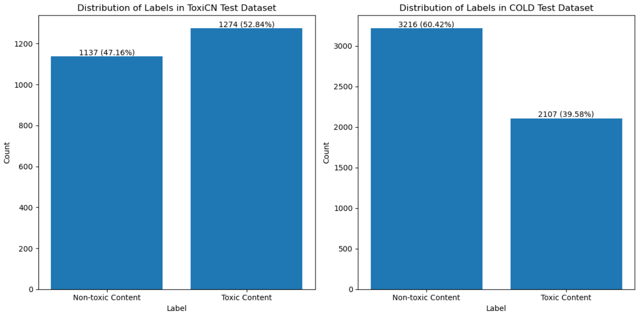

Figure 1: Distribution of Labels in ToxiCN Test Dataset and COLD Test Datasets

3.1.3 ***Chinese Open Instruction Generalist - Quality is All You Need(COIG-CQIA)****[13].* COIG-CQIA dataset is a high-quality Chinese instruction fine-tuning dataset containing 48,375 entries, designed to bridge the gap in Chinese instruction fine-tuning datasets. It is crafted by collecting human-written data from various online sources in China, which is then rigorously filtered and processed to align with human interactions. The dataset aims to enhance the proficiency of large language models (LLMs) in following Mandarin instructions by providing a diverse and comprehensive selection of real Mandarin language data from different fields. The dataset has undergone detailed manual cleaning to ensure its high quality, diversity, and relevance.

3.2 Model

3.2.1 ***GPT-4*** *[14].* GPT-4 is a large-scale multimodal model that can take image and text inputs and generate text outputs. It has shown performance comparable to human levels in various professional and academic benchmarks, including scoring significantly higher than previous GPT-3.5 models on internal factuality evaluations.

3.2.2 ***Qwen1.5*** *[15].* Qwen1.5 is the improved version of Qwen, the large language model series developed by Alibaba Cloud. It is based on the Transformer architecture with SwiGLU activation, attention QKV bias, group query attention, mixture of sliding window attention and full attention, etc. It has 8 model sizes, including 0.5B, 1.8B, 4B, 7B, 14B, 32B and 72B dense models, and an MoE model of 14B with 2.7B activated.

3.2.3 ***ChatGLM3*** *[16, 17].* ChatGLM3 is a generation of pre-trained dialogue models jointly released by Zhipu AI and Tsinghua KEG. ChatGLM3-6B is the open-source model in the ChatGLM3 series, maintaining many excellent features of the first two generations such as smooth dialogue and low deployment threshold.

4 METHODOLOGY

4.1 Fine-tuning

Fine-tuning refers to the process of adapting a pre-trained model to a specific downstream task by updating its parameters based on task-specific data. This process involves initializing the model with pre-trained weights and then updating these weights through gradient descent to optimize performance on the new task. Fine-tuning allows for leveraging the knowledge learned during pre-training while tailoring the model to perform well on a specific task, thereby improving its effectiveness and efficiency for that task.

LoRA[18], which stands for Low-Rank Adaptation, is a method proposed for adapting large language models to specific tasks or domains. It involves freezing the pretrained model weights and introducing trainable rank decomposition matrices into each layer of the Transformer architecture, significantly reducing the number of trainable parameters. This approach allows for more efficient training, lowers the hardware barrier to entry, reduces storage requirements, and introduces no additional inference latency compared to fully fine-tuned models. LoRA is compatible with various prior methods and can be combined with them, such as prefix-tuning. The method is particularly useful for language modeling tasks and has been shown to outperform several baselines with comparable or fewer trainable parameters in experiments. The principles of LoRA can be applied to any neural networks with dense layers, not limited to Transformer language models.

4.2 Prompt Engineering

Prompt engineering has become an essential strategy for optimizing the performance of pre-trained large language models (LLMs) [19]. This technique involves the careful creation of task-specific directions, known as prompts, that direct the outputs of the model without modifying its underlying parameters. Through the use of meticulously designed prompts, this approach allows for the refinement of model responses, thereby enabling these models to perform effectively across a variety of tasks and settings. This method stands in contrast to conventional models that typically require retraining or extensive parameter fine-tuning for task-specific enhancement. As the field continues to grow, recent studies frequently introduce new methods and uses for prompt engineering. The importance of this technique is highlighted by its ability to influence the responses of LLMs, significantly increasing their versatility and usefulness in different industries.

Zero-shot prompting [20] offers a paradigm shift in leveraging large LLMs. This technique removes the need for extensive training data, instead relying on carefully crafted prompts that guide the model toward novel tasks. Specifically, the model receives a task description in the prompt but lacks labeled data for training on specific input-output mappings. The model then leverages its pre-existing knowledge to generate predictions based on the given prompt for the new task

4.3 Multi-stage Fine-Tuning

To optimize the performance of our language model for detecting toxic content, we explores a multi-stage fine-tuning process. This methodology leverages the power of large, high-quality datasets followed by subsequent tuning on task-specific datasets, enabling the model to both generalize and specialize effectively.

- **Stage 1: Initial Fine-Tuning on a Large Dataset**
  In the first stage, the model undergoes fine-tuning using a comprehensive and diverse dataset. This dataset is curated from various sources to encompass a broad spectrum of language usage and contexts, which helps in building a robust foundational model. The aim of this stage is to adjust the model's parameters to better understand complex language patterns and nuances that are commonly present in large datasets.
- **Stage 2: Task-Specific Fine-Tuning**
  Following initial adjustments, the model is further refined using a dataset specifically tailored for toxic content detection. This dataset includes examples of toxic and non-toxic content that are closely representative of the actual data the model will encounter in application. By focusing on this narrower domain, the fine-tuning process enhances the model's ability to identify and classify toxic content accurately, adjusting its parameters to focus on the subtleties and specific features of toxic language.

4.4 vLLM

vLLM[21] refers to a distributed Large Language Model (LLM) serving engine built on top of PagedAttention, an attention algorithm inspired by virtual memory and paging techniques in operating systems. It is designed and implemented to efficiently manage memory in LLM serving systems, achieving near-zero waste in key-value cache memory and enabling flexible sharing of cache within and across requests. Through evaluations, vLLM has been shown to substantially outperform previous state-of-the-art solutions such as FasterTransformer and Orca.

5 EXPERIMENTAL DESIGN

To assess the capabilities of various language models in detecting inappropriate content within Chinese texts, a series of experiments has been designed as follows:

Experimental Setup: The experiments are conducted on a single A800 GPU, unless otherwise specified, on an Ubuntu system. For all models undergoing fine-tuning, the parameters are uniformly set with a learning rate of 10^{-5}, batch size of 8, and 10 training epochs.

Prompt Design: In this experiment, the prompt design is:
请仔细阅读下面提供的文本片段。在评估时，请特别注意识别任何可能被视为’不良内容’的元素。在此上下文中，’不良内容’被定义为包括但不限于以下几类：

(1) 仇恨言论：任何旨在贬低、嘲讽或激发针对特定群体（如基于种族、性别、性取向、宗教等）的敌意或暴力的言论。
(2) 性别或种族歧视：任何表现出对特定性别或种族固有不平等的观点或言论。
(3) 猥亵表达：包含过度色情、露骨性行为描述或其他淫秽内容的言论或表达。
(4) 不实信息：故意传播的虚假或误导性信息，可能导致公众误解或恐慌。
(5) 违反公共道德的内容：任何普遍被认为是不道德、不恰当或违反社会公序良俗的言论或行为。

在审查文本时，请使用上述标准作为参考。如果文本片段中不存在上述定义的不良内容，那么请返回’否’。相反，如果文本片段中包含上述定义的任何一种或多种不良内容，请返回’是’。请注意，仅当文本明确包含上述定义的内容时，才判断为包含不良内容

Datasets: The ToxiCN dataset's training set is utilized for model training, while its test set is used for model evaluation. The COLD dataset's test set is employed solely for assessing the generalization capabilities of the models.

6 RESULT ANALYSIS

The effectiveness of smaller-scale language models in detecting toxic content in Chinese texts is demonstrated in Table 1. By utilizing the ToxiCN and COLD datasets, we evaluated the performance of various models. The experimental results affirm that smaller models, when fine-tuned, perform exceptionally well in toxic content detection tasks, confirming their feasibility as practical tools.

ChatGLM3-6B identified toxic content in the ToxiCN dataset without training, with a recall rate of 99.24% but an accuracy rate of only 44.84%, indicating that this model tends to excessively label content as toxic without fine-tuning, resulting in a higher false positive rate. The GPT4 Turbo preview, used as a baseline in the study, achieved an accuracy of 75.07% on the ToxiCN test set and 76.48% on the COLD test set. Compared to the untrained results of the ChatGLM3-6B and Qwen1.5-1.8B-Chat models, larger models

Table 1: Performance Metrics of Models on Toxicity Detection

Model	Train Dataset	vLLM	Test Dataset	Accuracy	Precision	Recall	F1 Score	AUC	Pred. samp./s
GPT4-Turbo-preview	-	No	ToxiCN	75.07%	**84.92%**	64.52%	73.33%	75.89%	0.7216 (API)
			COLD	76.48%	69.97%	71.10%	70.53%	75.55%	0.8260 (API)
ChatGLM3-6B	Untrained	No	ToxiCN	44.84%	55.51%	**99.24%**	71.20%	53.95%	4.218
			COLD	36.54%	46.86%	99.23%	63.66%	60.89%	3.901
	ToxiCN	No	ToxiCN	78.43%	79.09%	80.46%	79.77%	78.31%	11.412
			COLD	70.20%	59.14%	79.97%	68.00%	71.89%	12.068
	ToxiCN	Yes	ToxiCN	71.42%	81.48%	61.70%	70.22%	73.08%	57.49
			COLD	73.57%	67.35%	65.27%	66.30%	72.16%	56.28
Qwen1.5-1.8B-Chat	Untrained	No	ToxiCN	62.88%	68.37%	55.65%	61.36%	63.44%	33.676
			COLD	67.35%	57.52%	67.01%	61.90%	67.29%	33.151
	ToxiCN	No	ToxiCN	**81.46%**	81.32%	84.38%	**82.82%**	**81.33%**	33.235
			COLD	72.70%	62.87%	75.79%	68.73%	73.24%	37.379
	ToxiCN	Yes	ToxiCN	77.31%	74.71%	86.26%	80.07%	76.77%	209.79
			COLD	64.87%	53.55%	84.77%	65.64%	68.30%	204.61
	COIG-CQIA & ToxiCN	Yes	ToxiCN	79.72%	80.17%	81.87%	81.01%	79.59%	**211.95**
			COLD	73.68%	63.08%	80.78%	70.84%	74.90%	204.83

demonstrated better performance in detecting toxic content in Chinese texts.

Both the ChatGLM3-6B and Qwen1.5-1.8B-Chat models showed significant improvements in accuracy after fine-tuning, highlighting the critical role of fine-tuning in enhancing model performance on specific tasks.

After implementing vLLM technology for inference acceleration, the accuracy of both models decreased slightly, but the increase in inference speed was significant. For the Qwen1.5-1.8B-Chat model, the accuracy on the ToxiCN test set only dropped by 4.15% after vLLM acceleration, while the inference speed increased from predicting 33.235 samples per second to 209.79 samples per second. This suggests that while the introduction of vLLM technology may slightly compromise accuracy, the substantial increase in speed is highly valuable for applications requiring quick responses. Moreover, this technology shows a significant performance advantage when handling large datasets, effectively reducing processing time and enhancing overall system efficiency.

Furthermore, in experiments where models were first fine-tuned using the COIG-CQIA dataset followed by the ToxiCN training set, there was a slight increase in accuracy without a significant drop in inference speed. Prioritizing fine-tuning with high-quality datasets before adjusting models to specific tasks appears to be an effective strategy for improving model performance.

7 CONCLUSION

In summary, our study underscores the potential of smaller-scale language models for effectively detecting toxic content in Chinese texts. While larger models typically offer better initial accuracy, our results demonstrate that smaller models, when appropriately fine-tuned, can achieve comparable performance, thus offering a more resource-efficient alternative. The vLLM technology, although it may slightly reduce accuracy, significantly enhances inference speed, making it a valuable tool for applications that demand fast processing times. Moreover, multi-stage fine-tuning has been proven to be a technique that enables these models to effectively generalize and specialize, thereby optimizing their performance in specific tasks.

8 FUTURE WORK

Following completion of the outlined experiments, several avenues remain open for further exploration to enhance the robustness and applicability of the models in detecting inappropriate content. These include:

(1) **Model Quantization:** To improve the deployment efficiency of the models on resource-constrained environments, quantization techniques can be applied. Model quantization reduces the precision of the numerical values used within the model, thus decreasing the computational demand and memory usage without significantly sacrificing performance. This process will be crucial for enabling the real-time analysis of text on mobile devices or in other bandwidth-limited settings.

(2) **Exploring Multilingual Model Generalization:** The models' ability to generalize across different languages and cultural contexts should be explored. Given that regions may vary significantly in their social norms and what is considered offensive, developing and testing models that can understand and adapt to these differences is essential. This involves training models on diverse datasets encompassing multiple languages and regional dialects to evaluate their cross-cultural effectiveness in detecting inappropriate content.

(3) **Adaptive Learning Systems:** Implementing adaptive learning mechanisms that allow models to continuously learn and adapt from new data inputs post-deployment could enhance their accuracy over time. This involves developing systems that can integrate feedback loops where the

model's predictions are regularly updated based on user feedback and real-world interaction.

(4) **Advanced Prompt Engineering:** The development of more sophisticated prompt engineering techniques to improve model performance on specific tasks without extensive retraining could be explored. This might include creating dynamic prompts that adjust based on the context of the interaction or the specific requirements of the task at hand.

(5) **Enhanced Interpretability and Explainability:** Increasing the interpretability and explainability of the models to make their decisions more transparent and understandable to users. This could involve integrating techniques that elucidate why certain outputs or decisions were made, particularly in borderline or controversial cases.

REFERENCES

[1] Jiang Zhang, Qiong Wu, Yiming Xu, Cheng Cao, Zheng Du, and Konstantinos Psounis. Efficient toxic content detection by bootstrapping and distilling large language models, 2023.

[2] Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. Hatebert: Retraining bert for abusive language detection in english, 2021.

[3] Youngwook Kim, Shinwoo Park, and Yo-Sub Han. Generalizable implicit hate speech detection using contrastive learning. In Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na, editors, *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6667–6679, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics.

[4] Yau-Shian Wang and Yingshan Chang. Toxicity detection with generative prompt-based inference, 2022.

[5] Tianhua Zhang, Hongyin Luo, Yung-Sung Chuang, Wei Fang, Luc Gaitskell, Thomas Hartvigsen, Xixin Wu, Danny Fox, Helen Meng, and James Glass. Interpretable unified language checking, 2023.

[6] Alper Kursat Uysal. An improved global feature selection scheme for text classification. *Expert Systems with Applications*, 43:82–92, 2016.

[7] Yiran Ye, Thai Le, and Dongwon Lee. Noisyhate: Benchmarking content moderation machine learning models with human-written perturbations online, 2023.

[8] Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A. Smith, and Yejin Choi. Social bias frames: Reasoning about social and power implications of language, 2020.

[9] Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. Learning from the worst: Dynamically generated datasets to improve online hate detection, 2021.

[10] Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection, 2022.

[11] Junyu Lu, Bo Xu, Xiaokun Zhang, Changrong Min, Liang Yang, and Hongfei Lin. Facilitating fine-grained detection of chinese toxic language: Hierarchical taxonomy, resources, and benchmarks, 2023.

[12] Jiawen Deng, Jingyan Zhou, Hao Sun, Chujie Zheng, Fei Mi, Helen Meng, and Minlie Huang. Cold: A benchmark for chinese offensive language detection, 2022.

[13] Yuelin Bai, Xinrun Du, Yiming Liang, Yonggang Jin, Ziqiang Liu, Junting Zhou, Tianyu Zheng, Xincheng Zhang, Nuo Ma, Zekun Wang, Ruibin Yuan, Haihong Wu, Hongquan Lin, Wenhao Huang, Jiajun Zhang, Wenhu Chen, Chenghua Lin, Jie Fu, Min Yang, Shiwen Ni, and Ge Zhang. Coig-cqia: Quality is all you need for chinese instruction fine-tuning, 2024.

[14] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024.

[15] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodeng Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[16] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335, 2022.

[17] Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*, 2022.

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

[19] Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. A systematic survey of prompt engineering in large language models: Techniques and applications, 2024.

[20] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2018.

[21] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention, 2023.