Journal of Machine Learning Research 25 (2024) 1-22 Submitted 1/24; Revised 6/24; Published 8/24

# PromptBench: A Unified Library for Evaluation of Large Language Models

**Kaijie Zhu**[1,2]*, **Qinlin Zhao**[1,3]*, **Hao Chen**[4], **Jindong Wang**[1]†, **Xing Xie**[1]
[1]Microsoft Research Asia [2]Institute of Automation, Chinese Academy of Sciences
[3]University of Science and Technology of China [4]Carnegie Mellon University

**Editor:** Zeyi Wen

## Abstract

The evaluation of large language models (LLMs) is crucial to assess their performance and mitigate potential security risks. In this paper, we introduce PromptBench, a unified library to evaluate LLMs. It consists of several key components that can be easily used and extended by researchers: prompt construction, prompt engineering, dataset and model loading, adversarial prompt attack, dynamic evaluation protocols, and analysis tools. PromptBench is designed as an open, general, and flexible codebase for research purpose. It aims to facilitate original study in creating new benchmarks, deploying downstream applications, and designing new evaluation protocols. The code is available at: `https://github.com/microsoft/promptbench` and will be continuously supported.

**Keywords:** Evaluation, large language models, framework

## 1. Introduction

Large language models (LLMs) are revolutionizing aspects of human life and society, such as medical diagnostics (McDuff et al., 2023; Thirunavukarasu et al., 2024), and educational tools (Ho et al., 2023). Evaluation is of paramount importance to understand the true capabilities of LLMs, mitigate potential risks, and eventually, benefit society further (Eisenstein, 2023; Chang et al., 2023). Recent efforts have evaluated LLMs from diverse aspects (Hendrycks et al., 2021; Liang et al., 2022; Zheng et al., 2023; Li et al., 2023c; Huang et al., 2023; HuggingFace, 2023). Among the findings, one of the most important is that current LLMs are sensitive to prompts (Wang et al., 2023b), vulnerable to adversarial prompt attacks (Zhu et al., 2023b), and exposed to testset data contamination (Willig et al., 2023; Zhou et al., 2023b; Zhu et al., 2023a), which pose severe security and privacy issues (Wang et al., 2023a; Simmons, 2022). On top of that, there have been various prompt learning algorithms developed based on different evaluation metrics, such as BDPL (Diao et al., 2022), GrIPS (Prasad et al., 2022) and Plum (Pan et al., 2023). Given the increasing popularity of LLMs, it is indispensable to develop a unified codebase to enable easy, fast, and flexible evaluation of large foundation models.

There are existing libraries such as LlamaIndex (Liu, 2022), semantic kernel (Microsoft, 2023), and LangChain (Chase, 2022). LlamaIndex and LangChain enhance LLM applications by incorporating databases and various data sources. Semantic Kernel aims to

*. The first two authors contributed equally. Work done at MSRA.
†. Corresponding author: Jindong Wang (jindong.wang@microsoft.com).

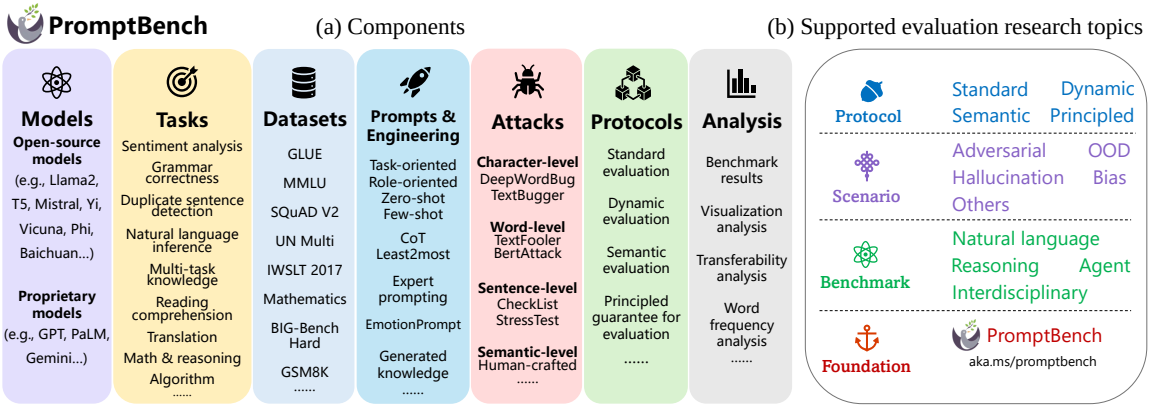


Figure 1: The components and supported research areas of PromptBench.

merge AI services with programming languages for versatile AI app development. Evalharness (Gao et al., 2023) offers a comprehensive framework for evaluating generative language models. Zeno (Zeno, 2023) is an AI evaluation platform supporting interaction and visualization, but it is not easy to customize. LiteLLM (BerriAI, 2023) implements a unified API call for different LLM service prodiders.

This paper introduces **PromptBench**, a unified Python library to evaluate LLMs from comprehensive dimensions. It is designed to fill the gaps current libraries have, offering comprehensive support not only for standard model evaluations but also for advanced scenarios including adversarial prompt attacks and dynamic evaluations. Its extensible architecture allows for the incorporation of new evaluation protocols, addressing the limitations found in other tools. The detailed comparisons are shown in Appendix A, highlighting how PromptBench provides a more complete toolkit for the nuanced evaluation of language models, especially in research contexts where adaptability and thoroughness are paramount. It consists of a wide range of LLMs and evaluation datasets, covering diverse tasks, evaluation protocols, adversarial prompt attacks, and prompt engineering techniques. As a holistic library, it also supports several analysis tools for interpreting the results. Our library is designed in a modular fashion, allowing researchers to easily build evaluation pipelines for their own projects. We open-source PromptBench with comprehensive documents and tutorials[1] to support easy, flexible, and collaborative evaluation. We believe PromptBench could enhance our understanding of LLMs and spur new research within the community.

## 2. PromptBench

PromptBench can be easily installed either via `pip install promptbench` or `git clone`. In this section, we briefly introduce the components of PromptBench and how to use it to build an evaluation pipeline for LLMs. An overview of PromptBench is shown in Figure 1.

### 2.1 Components

**Models.** PromptBench supports both open-source and proprietary LLMs and VLMs and it is open to add more. Currently, it supports a diverse range of LLMs and VLMs, ranging from

1. `https://promptbench.readthedocs.io/en/latest/`

Llama2 series (Touvron et al., 2023b), Mixtral series (Jiang et al., 2024), LlaVa series (Liu et al., 2023a) to GPT series (OpenAI, 2023a,b). PromptBench provides unified `LLMModel` and `VLMModel` interfaces to allow easy construction and inference of a model with specified max generating tokens and generating temperature. The interfaces also support customized models, including those that have been fine-tuned for specific applications. More details of the supported models are shown in Appendix B.1.

**Datasets and tasks.** PromptBench comprises a wide array of tasks, currently supporting diverse challenges across 12 tasks and 22 public datasets, with the capacity for additional expansions. The supported tasks include fundamental NLP tasks such as sentiment analysis, grammar correctness, and duplicate sentence detection, as well as complex challenges involving natural language inference, multi-task knowledge, and reading comprehension. It also covers specialized areas like translation, mathematical problem-solving, and various forms of reasoning—logical, commonsense, symbolic, and algorithmic. For detailed descriptions of each dataset and specific task configurations (see Appendix B.2). The unified `DatasetLoader` interface facilitates easy and customizable loading and processing of these datasets, enhancing the usability and flexibility of PromptBench.

**Prompts and prompt engineering.** PromptBench offers a suite of 4 distinct prompt types, and additionally, users have the flexibility to craft custom prompts using the `Prompt` interface. Task-oriented prompts are structured to clearly delineate the specific task expected of the model, whereas role-oriented prompts position the model in a defined role, such as an expert, advisor, or translator. These prompt categories are adaptable for both zero-shot and few-shot learning contexts, offering diverse application possibilities. Moreover, PromptBench currently includes 6 prominent prompt engineering methods, details can be found in Appendix B.4.2. Our framework is not only equipped for the easy integration of these existing techniques through the `prompt_engineering` module but is also actively evolving to encompass a broader spectrum of prompt engineering methods, enhancing its adaptability in varied evaluation scenarios.

**Adversarial prompt attacks.** To facilitate the investigation of LLMs' robustness on prompts, PromptBench integrates 4 types of attacks: (1) character-level attacks (Li et al., 2019; Gao et al., 2018), which manipulate texts by introducing typos or errors to words; (2) word-level attacks (Jin et al., 2019; Li et al., 2020), which aim to replace words with synonyms or contextually similar words to deceive LLMs; (3) sentence-level attacks (Ribeiro et al., 2020; Naik et al., 2018), which append irrelevant or extraneous sentences to the end of prompts, intending to distract LLMs; (4) semantic-level attacks (Zhu et al., 2023b), which simulate the linguistic behavior of people from different countries. Details can be found in Appendix B.4.3 These attacks can be easily called via the `prompt_attack` interface. It also supports the usage of curated adversarial prompts to efficiently evaluate robustness.

**Different evaluation protocols.** By default, PromptBench supports the standard protocol, i.e., the direct inference. PromptBench further supports dynamic (Zhu et al., 2023a) and semantic (Liu et al., 2023b) evaluation protocols by dynamically generating testing data. It is open to integrate more new protocols to avoid data contamination.

**Analysis tools.** Finally, PromptBench offers a series of analysis tools to help researchers analyze their results. Particularly, it support sweep running to get the benchmark results. Then, attention visualization analysis can be done through the `utils` interface.

PromptBench also supports word frequency analysis to analyze the words used in attacks as well as defense analysis by integrating word correction tools.

### 2.2 Evaluation pipeline

PromptBench allows easy construction of an evaluation pipeline via four steps. Firstly, specify task and then load dataset via `pb.DatasetLoader`. PromptBench offers a streamlined one-line API for loading the desired dataset. Secondly, users can customize LLMs using the `pb.LLMModel`, which provides integrated inference pipelines compatible with most LLMs implemented in Huggingface. Thirdly, the prompt for the specified dataset and task is defined via `pb.Prompt`. Users have the option to input a list of prompts for evaluation and performance comparison. In cases where no prompts are supplied, our default prompts for the dataset are utilized. Finally, the pipeline requires the definition of input and output processing functions via `class InputProcess` and `class OutputProcess` defined in `pb.utils.dataprocess`, as well as the evaluation function via `pb.metrics`. The detailed introduction of the components are shown in Appendix B.

### 2.3 Supported research topics

Compared to current evaluation libraries, PromptBench is designed mainly for research purpose, thus it is easy to customize for different topics. As shown in Figure 1(b), it supports different evaluation topics from the research community including benchmarks, scenarios, and protocols. In benchmarks research, it supports standard natural language understanding, natural language generation, and reasoning tasks. It can also be extended to support research on AI agent and interdisciplinary study. In scenario research, it supports adversarial and out-of-distribution evaluation, and can also support other topics such as hallucination and bias by changing the `metrics` and `DatasetLoader` interface. In protocol, it naturally supports standard and dynamic evaluation, and can further be verified by including measurement theory. PromptBench offers three leaderboards to allow easy comparison: adversarial prompt attack, prompt engineering, and dynamic evaluation, as shown in Appendix C. Researchers are welcome to submit new results to our platform. Extensibility is shown in Appendix D that allows convenient extension of the framework.

## 3. Conclusion and Discussion

We presented PromptBench, a unified framework for LLMs evaluation, designed in a modular fashion to build evaluation pipelines with various models, tasks, and prompts. It supports research in prompt engineering, adversarial attacks, and dynamic evaluation. PromptBench is the first step in assessing and exploring the capabilities of current LLMs. We believe our benchmark and analysis will inform the design of more robust, human-aligned models. In the future, new datasets, evaluation protocols, prompt variations, and analytical tools will be added into PromptBench. We also welcome any contributions for PromptBench.

Despite its versatility, PromptBench has limitations. It may not cover all evaluation scenarios, and some metrics might miss nuanced performance differences. The framework's effectiveness depends on the quality and diversity of datasets and prompts. Addressing these limitations is a priority for our ongoing and future work.

## References

Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. Nocaps: Novel object captioning at scale. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8948–8957, 2019.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023.

BIG bench authors. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=uyTL5Bvosj.

BerriAI. https://docs.litellm.ai/, 2023.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. Gpt-neox-20b: An open-source autoregressive language model, 2022. URL https://arxiv.org/abs/2204.06745.

Google Brain. A new open source flan 20b with ul2, 2023. URL https://www.yitay.net/blog/flan-ul2-20b.

Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Jan Niehues, Sebastian Stüker, Katsuhito Sudoh, Koichiro Yoshino, and Christian Federmann. Overview of the IWSLT 2017 evaluation campaign. In *Proceedings of the 14th International Conference on Spoken Language Translation*, pages 2–14, Tokyo, Japan, December 14-15 2017. International Workshop on Spoken Language Translation. URL https://aclanthology.org/2017.iwslt-1.1.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Kaijie Zhu, Hao Chen, Linyi Yang, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. A survey on evaluation of large language models. *arXiv preprint arXiv:2307.03109*, 2023.

Harrison Chase. Langchain. https://github.com/langchain-ai/langchain, 2022. Date released: 2022-10-17.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022. URL https://arxiv.org/abs/2210.11416.

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. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Contributors. promptfoo: test your llm app, 2023a. URL https://github.com/promptfoo/promptfoo.

Evals Contributors. Openai evals, 2023b. URL https://github.com/openai/evals.

OpenCompass Contributors. Opencompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass, 2023c.

Databricks. Hello dolly: Democratizing the magic of chatgpt with open models, 2023. URL https://www.databricks.com/blog/2023/03/24/hello-dolly-democratizing-magic-chatgpt-open-models.html.

Nolan Dey, Gurpreet Gosal, Zhiming, Chen, Hemant Khachane, William Marshall, Ribhu Pathria, Marvin Tom, and Joel Hestness. Cerebras-gpt: Open compute-optimal language models trained on the cerebras wafer-scale cluster, 2023.

Shizhe Diao, Zhichao Huang, Ruijia Xu, Xuechun Li, Yong Lin, Xiao Zhou, and Tong Zhang. Black-box prompt learning for pre-trained language models. *arXiv preprint arXiv:2201.08531*, 2022.

William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005. URL https://aclanthology.org/I05-5002.

Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Xilin Wei, Songyang Zhang, Haodong Duan, Maosong Cao, Wenwei Zhang, Yining Li, Hang Yan, Yang Gao, Xinyue Zhang, Wei Li, Jingwen Li, Kai Chen, Conghui He, Xingcheng Zhang, Yu Qiao, Dahua Lin, and Jiaqi Wang. Internlm-xcomposer2: Mastering free-form text-image composition and comprehension in vision-language large model. *arXiv preprint arXiv:2401.16420*, 2024.

Andreas Eisele and Yu Chen. MultiUN: A multilingual corpus from united nation documents. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2010/pdf/686_Paper.pdf.

Michael Eisenstein. A test of artificial intelligence. *Nature Outlook: Robotics and artificial intelligence*, 2023.

J. Gao, J. Lanchantin, M. L. Soffa, and Y. Qi. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56, May 2018. doi: 10.1109/SPW.2018.00016.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 12 2023. URL `https://zenodo.org/records/10256836`.

Google. `https://deepmind.google/technologies/gemini/#introduction`, 2023.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=d7KBjmI3GmQ`.

Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 14852–14882. Association for Computational Linguistics, 2023. URL `https://aclanthology.org/2023.acl-long.830`.

Jordan Hoffmann et al. Training compute-optimal large language models, 2022.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, et al. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *arXiv preprint arXiv:2305.08322*, 2023.

HuggingFace. Open-source large language models leaderboard. `https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard`, 2023.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian

Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? natural language attack on text classification and entailment. *arXiv preprint arXiv:1907.11932*, 2019.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 235–251. Springer, 2016.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. Qasc: A dataset for question answering via sentence composition, 2020.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=e2TBb5y0yFf`.

Hector Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*, 2012.

Cheng Li, Jindong Wang, Yixuan Zhang, Kaijie Zhu, Wenxin Hou, Jianxun Lian, Fang Luo, Qiang Yang, and Xing Xie. Large language models understand and can be enhanced by emotional stimuli, 2023a.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. TextBugger: Generating adversarial text against real-world applications. In *Proceedings 2019 Network and Distributed System Security Symposium*. Internet Society, 2019. doi: 10.14722/ndss.2019.23138. URL `https://doi.org/10.14722%2Fndss.2019.23138`.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023b.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.500. URL `https://aclanthology.org/2020.emnlp-main.500`.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models. *Github repository*, 2023c.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. Textbooks are all you need ii: **phi-1.5** technical report. *arXiv preprint arXiv:2309.05463*, 2023d.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

Bill Yuchen Lin, Seyeon Lee, Rahul Khanna, and Xiang Ren. Birds have four legs?! numersense: Probing numerical commonsense knowledge of pre-trained language models, 2020.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023a.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

Jerry Liu. LlamaIndex, 11 2022. URL `https://github.com/jerryjliu/llama_index`.

Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. Generated knowledge prompting for commonsense reasoning, 2022.

Yachuan Liu, Liang Chen, Jindong Wang, Qiaozhu Mei, and Xing Xie. Meta semantic template for evaluation of large language models. *arXiv preprint arXiv:2310.01448*, 2023b.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023.

Ahmed Masry, Do Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2263–2279, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.177. URL `https://aclanthology.org/2022.findings-acl.177`.

Daniel McDuff, Mike Schaekermann, Tao Tu, Anil Palepu, Amy Wang, Jake Garrison, Karan Singhal, Yash Sharma, Shekoofeh Azizi, Kavita Kulkarni, Le Hou, Yong Cheng, Yun Liu, S Sara Mahdavi, Sushant Prakash, Anupam Pathak, Christopher Semturs, Shwetak Patel, Dale R Webster, Ewa Dominowska, Juraj Gottweis, Joelle Barral, Katherine Chou, Greg S Corrado, Yossi Matias, Jake Sunshine, Alan Karthikesalingam, and Vivek Natarajan. Towards accurate differential diagnosis with large language models, 2023.

Microsoft. Semantic kernel. https://github.com/microsoft/semantic-kernel, 2023.

Mixtral. Mixtral, 2023. URL https://mistral.ai/news/mixtral-of-experts/.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. Stress test evaluation for natural language inference. In *ACL*, pages 2340–2353, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://aclanthology.org/C18-1198.

OpenAI. https://chat.openai.com.chat, 2023a.

OpenAI. Gpt-4 technical report, 2023b.

Rui Pan, Shuo Xing, Shizhe Diao, Xiang Liu, Kashun Shum, Jipeng Zhang, and Tong Zhang. Plum: Prompt learning using metaheuristic. *arXiv preprint arXiv:2311.08364*, 2023.

Archiki Prasad, Peter Hase, Xiang Zhou, and Mohit Bansal. Grips: Gradient-free, edit-based instruction search for prompting large language models. *arXiv preprint arXiv:2203.07281*, 2022.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *ACL*, pages 784–789, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2124. URL https://aclanthology.org/P18-2124.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *ACL*, pages 4902–4912, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.442. URL https://aclanthology.org/2020.acl-main.442.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. Analysing mathematical reasoning abilities of neural models. In *ICLR*, 2019. URL https://openreview.net/forum?id=H1gR5iR5FX.

Gabriel Simmons. Moral mimicry: Large language models produce moral rationalizations tailored to political identity. *arXiv preprint arXiv:2209.12106*, 2022.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D13-1170.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge, 2019.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

AJ Thirunavukarasu, S Mahmood, A Malem, WP Foster, R Sanghera, and R Hassan. Large language models approach expert-level clinical knowledge and reasoning in ophthalmology: A head-to-head cross-sectional study. *PLOS Digital Health*, 3(4):e0000341, 2024. doi: 10.1371/journal.pdig.0000341.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023a.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. 2019. In the Proceedings of ICLR.

Ben Wang and Aran Komatsuzaki. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`, May 2021.

Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, et al. Decodingtrust: A comprehensive assessment of trustworthiness in gpt models. *arXiv preprint arXiv:2306.11698*, 2023a.

Jindong Wang, Xixu Hu, Wenxin Hou, Hao Chen, Runkai Zheng, Yidong Wang, Linyi Yang, Haojun Huang, Wei Ye, Xiubo Geng, et al. On the robustness of chatgpt: An adversarial and out-of-distribution perspective. In *International conference on learning representations (ICLR) workshop on Trustworthy and Reliable Large-Scale Machine Learning Models*, 2023b.

Zhiguo Wang, Wael Hamza, and Radu Florian. Bilateral multi-perspective matching for natural language sentences, 2017.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*, 2018.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL HLT*, pages 1112–1122. Association for Computational Linguistics, 2018. URL `http://aclweb.org/anthology/N18-1101`.

Moritz Willig, Matej Zecevic, Devendra Singh Dhami, and Kristian Kersting. Causal parrots: Large language models may talk causality but are not causal. *Transactions on machine learning research (TMLR)*, 8, 2023.

Benfeng Xu, An Yang, Junyang Lin, Quan Wang, Chang Zhou, Yongdong Zhang, and Zhendong Mao. Expertprompting: Instructing large language models to be distinguished experts, 2023.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, Fei Deng, Feng Wang, Feng Liu, Guangwei Ai, Guosheng Dong, Haizhou Zhao, Hang Xu, Haoze Sun, Hongda Zhang, Hui Liu, Jiaming Ji, Jian Xie, JunTao Dai, Kun Fang, Lei Su, Liang Song, Lifeng Liu, Liyun Ru, Luyao Ma, Mang Wang, Mickel Liu, MingAn Lin, Nuolan Nie, Peidong Guo, Ruiyang Sun, Tao Zhang, Tianpeng Li, Tianyu Li, Wei Cheng, Weipeng Chen, Xiangrong Zeng, Xiaochuan Wang, Xiaoxi Chen, Xin Men, Xin Yu, Xuehai Pan, Yanjun Shen, Yiding Wang, Yiyu Li, Youxin Jiang, Yuchen Gao, Yupeng Zhang, Zenan Zhou, and Zhiying Wu. Baichuan 2: Open large-scale language models, 2023.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*, 2023.

Yi. Yi, 2023. URL `https://github.com/01-ai/Yi`.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. *arXiv preprint arXiv:2311.16502*, 2023.

Zeno. `https://zenoml.com/`, 2023.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. Least-to-most prompting enables complex reasoning in large language models, 2023a.

Kun Zhou, Yutao Zhu, Zhipeng Chen, Wentong Chen, Wayne Xin Zhao, Xu Chen, Yankai Lin, Ji-Rong Wen, and Jiawei Han. Don't make your llm an evaluation benchmark cheater. *arXiv preprint arXiv:2311.01964*, 2023b.

Kaijie Zhu, Jiaao Chen, Jindong Wang, Neil Zhenqiang Gong, Diyi Yang, and Xing Xie. Dyval: Graph-informed dynamic evaluation of large language models. *arXiv preprint arXiv:2309.17167*, 2023a.

Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Neil Zhenqiang Gong, Yue Zhang, et al. Promptbench: Towards evaluating the robustness of large language models on adversarial prompts. *arXiv preprint arXiv:2306.04528*, 2023b.

## Appendix A. Comparison with Related Code Libraries

Table 1: Comparison with related code libraries.

| Library | Purpose | Customization | Functions |
|---|---|---|---|
| **OpenAI Evals (Contributors, 2023b)** | A framework for evaluating LLMs or systems built using LLMs. | model, dataset, prompt, eval | Evaluation pipelines<br>Benchmarks |
| **OpenCompass (Contributors, 2023c)** | One-stop platform for large model evaluation, aiming to provide a transparent benchmark for foundation models. | model, dataset | Evaluation pipelines<br>Benchmarks<br>Leaderboards |
| **promptfoo (Contributors, 2023a)** | A framework for evaluating prompts and large language models. | model, dataset, prompt, eval | Evaluation pipelines |
| **LM Evaluation Harness (Gao et al., 2023)** | A framework for evaluation of autoregressive LMs. | model, dataset, prompt, eval | Evaluation pipelines<br>Benchmarks<br>Leaderboards |
| **HELM (Liang et al., 2022)** | Holistic Evaluation of Language Models. | model | Evaluation pipelines<br>Benchmark<br>Leaderboard |
| **PromptBench (Ours)** | Research-focused evaluation toolkit. | model, dataset, prompt, prompt engineering, eval | Evaluation pipelines<br>Prompt Engineering<br>Prompt attacks<br>Dynamic evaluation<br>Leaderboard |

## Appendix B. Details of PromptBench

### B.1 Models

In this section, we list the LLMs and VLMS implemented in PromptBench.

**Open-source LLMs:**

- **Flan-T5-large (Chung et al., 2022)**: Google's Flan-T5-large, a variation of the Text-to-Text Transfer Transformer (T5).
- **Dolly-6B (Databricks, 2023)**: The Dolly-6B model, developed by Databricks, is a 6-billion parameter causal language model. It is an extension of EleutherAI's GPT-J (Wang and Komatsuzaki, 2021), further refined with Stanford's Alpaca (Taori et al., 2023) dataset comprising 52K question/answer pairs.
- **Vicuna series (Chiang et al., 2023)**: Developed from the LLaMA-13B base model, Vicuna-13B integrates over 70K user-shared conversations from ShareGPT.com, leveraging public APIs for data acquisition.
- **Cerebras series (Dey et al., 2023)**: Modeled on the GPT-3 architecture, Cerebras-13B is part of the Cerebras-GPT series, trained according to Chinchilla scaling laws (Hoffmann et al., 2022) to optimize computational efficiency.
- **Llama2 series (Touvron et al., 2023a)**: Engineered by Meta AI's FAIR team, the Llama2 model is an autoregressive language model adopting the transformer architecture.

- **GPT-NEOX-20B (Black et al., 2022)**: This variant, part of the extensive GPT model series, features 20 billion parameters, exemplifying large-scale language model implementation.
- **Flan-UL2 (Brain, 2023)**: Flan-UL2, an encoder-decoder model, is grounded in the T5 architecture and enhanced with Flan prompt tuning and dataset techniques.
- **phi-1.5 and phi-2 (Li et al., 2023d)**: phi-1.5 is an LLM with 1.3 billion parameters, builds upon the dataset used for phi-1 with the addition of diverse NLP synthetic texts.
- **Mistral 7B (Jiang et al., 2023)**: Mistral 7B is trained by Mistral AI team. It excels in tasks like reasoning, mathematics, and code generation. It uses grouped-query attention for faster inference and sliding window attention for efficient handling of sequences. There's also an instruction-following version, Mistral 7B-Instruct.
- **Mixtral8x7B (Mixtral, 2023)**: Engineering by Mistral AI team, this model is a high-quality sparse mixture of experts model (SMoE) with open weights. There's also an instruction-following version, Mixtral 8x7B Instruct.
- **Baichuan2 series (Yang et al., 2023)**: Baichuan2 is developed by Baichuan Intelligent. Trained on 2.6 trillion high-quality tokens, it achieves the best results in its size class on multiple authoritative benchmarks in Chinese, English, and multilingual general and domain-specific tasks.
- **Yi series (Yi, 2023)**: Developed by 01.AI, the Yi series are next-generation open-source large language models. Trained on a 3T multilingual corpus, they excel in language understanding, commonsense reasoning, and reading comprehension.
- **BLIP2 (Li et al., 2023b)**: This visual-language model is proposed by Junnan Li, Dongxu Li, Silvio Savarese, Steven Hoi. BLIP-2 leverages frozen pre-trained image encoders and large language models (LLMs) by training a lightweight, 12-layer Transformer encoder in between them, achieving excellent performance in various vision-language tasks
- **LLaVA (Liu et al., 2024)**: LlaVA (Language-Image LLaMA) is a multimodal model combining language and image data. It extends the LLaMA architecture to handle both modalities, enabling tasks like image captioning, visual question answering, and image-based text generation.
- **Qwen-VL series(Bai et al., 2023)**: Qwen-VL (Qwen Large Vision Language Model) is the multimodal version of the large model series, Qwen (abbr. Tongyi Qianwen), proposed by Alibaba Cloud. Qwen-VL accepts image, text, and bounding box as inputs, outputs text, and bounding box.
- **InternLM-XComposer2-VL (Dong et al., 2024)**: InternLM-XComposer2 is a cutting-edge vision-language model excelling in free-form text-image composition and comprehension, crafting content from diverse inputs like outlines, detailed specs, and reference images. Using a Partial LoRA (PLoRA) approach, it balances vision understanding and text composition.

**Proprietary LLMs:**

- **ChatGPT (OpenAI, 2023a) and GPT-4 (OpenAI, 2023b)**: OpenAI's ChatGPT and GPT-4 are advanced iterations of the GPT series. ChatGPT is tailored for interactive tasks, while GPT-4 is the most proficient in the series and supports image input.

- **PaLM 2 (Anil et al., 2023)**: PaLM 2 is an advanced language model that excels in multilingual and reasoning capabilities, offering greater computational efficiency than its predecessor, PaLM. This Transformer-based model enhances performance across various model sizes in English, multilingual tasks, and reasoning challenges.
- **Gemini (Google, 2023)**: The Gemini model is a multimodal language model developed by Google AI, capable of extracting insights from a diverse array of data formats, including images, and video.

### B.2 Tasks and Datasets

- **GLUE (Wang et al., 2019)**: The GLUE benchmark (General Language Understanding Evaluation) offers a suite of tasks to evaluate the capability of NLP models in understanding language. For this research, we employed 8 specific tasks: Sentiment Analysis (SST-2 (Socher et al., 2013)), Grammar Correctness (CoLA (Warstadt et al., 2018)), Identifying Duplicate Sentences (QQP (Wang et al., 2017), MRPC (Dolan and Brockett, 2005)), and various Natural Language Inference tasks (MNLI (Williams et al., 2018), QNLI (Wang et al., 2019), RTE (Wang et al., 2019), WNLI (Levesque et al., 2012)).
- **MMLU (Hendrycks et al., 2021)**: The MMLU dataset tests the broad knowledge and problem-solving skills of large language models through 57 tasks with multiple-choice questions in fields like mathematics, history, and computer science. It is a comprehensive multitask benchmark.
- **SQuAD V2 (Rajpurkar et al., 2018)**: The SQuAD v2 dataset is pivotal in training and assessing NLP models for reading comprehension. It builds upon the original SQuAD by adding unanswerable questions, making it more challenging. Models must either identify the correct answer in the text or recognize questions as unanswerable.
- **UN Multi (Eisele and Chen, 2010)**: Comprising texts in the six official United Nations languages, the Multi UN dataset is a vast parallel corpus from UN documents. However, its focus on formal texts may restrict its use in informal or conversational language contexts.
- **IWSLT 2017 (Cettolo et al., 2017)**: Designed for spoken language translation system evaluation, the IWSLT 2017 dataset includes multilingual, multi-domain text data, primarily from the TED Talks Open Translation Project. It encompasses numerous language pairs, providing a rich resource for translation tasks.
- **Math (Saxton et al., 2019)**: The DeepMind Mathematics Dataset assesses AI models' mathematical reasoning by posing a wide array of math problems, from algebra to calculus. It tests the models' understanding and logical reasoning in mathematics.
- **BIG-Bench (bench authors, 2023)**: BIG-bench is a collaborative benchmark designed to evaluate the capabilities of large language models and predict their future potential. It consists of over 200 tasks, contributed by 444 authors from 132 institutions, covering a wide range of topics like linguistics, math, common-sense reasoning, and more. These tasks are intended to probe areas believed to be beyond the current capabilities of LMs.
- **GSM8K (Cobbe et al., 2021)**: The GSM8K dataset is a collection of 8.5K high-quality, linguistically diverse grade school math word problems. It was created by human

problem writers and is divided into 7.5K training problems and 1K test problems. These problems, which require 2 to 8 steps to solve, primarily involve basic arithmetic operations and are designed to be solvable by a bright middle school student.

- **CommonsenseQA (Talmor et al., 2019)**: The CommonsenseQA dataset is a challenging commonsense question-answering dataset. It comprises 12,247 questions with 5 multiple-choice answers each.
- **QASC (Khot et al., 2020)**: QASC (Question Answering via Sentence Composition) is a specialized collection designed for question-answering tasks with a focus on sentence composition. It comprises 9,980 eight-way multiple-choice questions about grade school science, divided into 8,134 for training, 926 for development, and 920 for testing .(In our evaluation, we use development part.) The dataset is notable for its emphasis on multi-hop reasoning, requiring the retrieval and composition of facts from a broad corpus to answer each question.
- **NummerSense (Lin et al., 2020)**: NumerSense is a unique numerical commonsense reasoning probing task, featuring a diagnostic dataset with 3,145 masked-word-prediction probes. This dataset has applications in tasks such as knowledge base completion and open-domain question answering.
- **VQAv2 (Goyal et al., 2017)**: Visual Question Answering (VQA) v2.0 is a dataset containing open-ended questions about images. These questions require an understanding of vision, language and commonsense knowledge to answer. It is the second version of the VQA dataset.
- **NoCaps (Agrawal et al., 2019)**: NoCaps is a benchmark dataset for image captioning models that can describe images containing novel objects from object detection datasets. It consists of 166,100 human-generated captions describing 15,100 images from the Open Images validation and test sets.
- **MMMU (Yue et al., 2023)**: MMMU is a comprehensive benchmark designed to evaluate multimodal models on massive multi-discipline tasks demanding college-level subject knowledge and deliberate reasoning. MMMU includes 11.5K meticulously collected multimodal questions from college exams, quizzes, and textbooks. These questions span 30 subjects and 183 subfields, comprising 32 highly heterogeneous image types, such as charts, diagrams, maps, tables, music sheets, and chemical structures.
- **MathVista (Lu et al., 2023)**: MathVista is a comprehensive benchmark for mathematical reasoning in visual contexts. It includes three new datasets: IQTest, FunctionQA, and PaperQA. These datasets cover various visual domains and are designed to evaluate logical reasoning on puzzle test figures, algebraic reasoning using functional plots, and scientific reasoning with academic paper figures, respectively.
- **AI2D (Kembhavi et al., 2016)**: AI2D is a dataset of illustrative diagrams for research on diagram understanding and associated question answering. It contains 5000 grade-school science diagrams with over 150,000 rich annotations, their ground truth syntactic parses, and more than 15,000 corresponding multiple-choice questions.
- **ChartQA (Masry et al., 2022)**: ChartQA is a large-scale dataset of complex reasoning questions over charts that involve visual and logical operations, covering 9.6K

human-written questions as well as 23.1K questions generated from human-written chart summaries.

- **ScienceQA (Lu et al., 2022)**: ScienceQA is a large-scale dataset for multimodal reasoning with diverse science topics and annotations of answers, lectures and explanations. It covers 26 topics, 127 categories and 379 skills from natural science, language science, and social science and so on.

### B.3 Evaluation protocols

DyVal (Zhu et al., 2023a) is an approach for dynamic evaluation of LLMs by creating complexity-tailored evaluation samples on-the-fly, as opposed to relying on static benchmarks. DyVal synthesized seven distinct reasoning tasks, including: (1) Mathematics, focusing on arithmetic calculations and linear equation solving; (2) Logical Reasoning, involving boolean, deductive, and abductive logic; and (3) Algorithmic Analysis, covering reachability and the maximum sum path problem. MSTemp (Liu et al., 2023b) stands for the semantic evalaution protocol which generate out-of-distribution samples by relying on evlauator LLMs and word replacement.

### B.4 Prompts

#### B.4.1 Prompts

Our study examines four prompt categories, differentiated by their intended function and the required number of labeled samples. Task-oriented prompts are designed to clearly define the model's task, prompting it to generate outputs relevant to the task using its inherent pre-training knowledge. In contrast, role-oriented prompts position the model as a particular entity, like an expert, advisor, or translator, thereby subtly guiding the expected output format and behavior through the assumed role. Both categories can be adapted for zero-shot and few-shot learning contexts. We randomly choose three training set examples for each task to form the few shot examples. Examples of various prompt types are illustrated in Table 2.

#### B.4.2 Prompt Engineering

Prompt engineering is a process of structuring and optimizing prompts to efficiently use AI models. Methods in prompt engineering , such as chain-of-thought (Wei et al., 2023), generated knowledge prompting (Liu et al., 2022) and so on, help improve reasoning ability and task performance of AI models. We implement 6 prominent prompt engineering methods:

- **Chain-of-Thought (Wei et al., 2023)**: This method involves breaking down complex, multi-step problems into smaller, intermediate steps, enabling Models to tackle more intricate reasoning tasks. Chain-of-Thought differs from standard few-shot prompting by not just providing questions and answers but prompting the model to produce intermediate reasoning steps before arriving at the final answer.
- **Zero-Shot Chain-of-Thought (Kojima et al., 2022)**: Zero-Shot Chain of Thought improves Chain of Thought by simplifying the prompting process. The key innovation in

Table 2: Examples of 4 types of prompts.

| | | |
|---|---|---|
| Zero shot | Task oriented | `Determine if the given pair of statements can be considered the same by responding with 'equivalent' or 'not_equivalent'.` |
| | Role oriented | `As an instrument for question comparison evaluation, consider the questions and determine if their meaning is the same, responding with 'equivalent' for similar questions or 'not_equivalent' for different questions.` |
| Few shot | Task oriented | `Review the sentence below and identify whether its grammar is 'Acceptable' or 'Unacceptable': Here are three examples. Sentence: Our friends won't buy this analysis, let alone the next one we propose. Answer: acceptable. Sentence: One more pseudo generalization and I'm giving up. Answer: acceptable. Sentence: They drank the pub. Answer: unacceptable.` |
| | Role oriented | `Functioning as a grammar evaluation tool, analyze the given sentence and decide if it is grammatically correct, responding with 'acceptable' or 'unacceptable': Here are three examples. Sentence: Our friends won't buy this analysis, let alone the next one we propose. Answer: acceptable. Sentence: One more pseudo generalization and I'm giving up. Answer: acceptable. Sentence: They drank the pub. Answer: unacceptable.` |

Zero-Shot Chain-of-Thought is appending the phrase "`Let's think step by step`" to the end of a question.

- **EmotionPrompt (Li et al., 2023a)**: Drawing inspiration from psychology and social science theories about human emotional intelligence, this method adds emotional stimuli to origin prompts. For example: "`This is very important to my career`."
- **Expert Prompting (Xu et al., 2023)**: The key idea is to let model be an expert in role playing. To generate the expert identity, we first provide several instruction-expert pair exemplars, then the model generates an expert identity of this question. Finally, we ask the model to answer the instruction conditioned on expert identity.
- **Generated Knowledge (Liu et al., 2022)**: Generated Knowledge Prompting is a method where a model first generates knowledge and then uses this generated information as additional input to answer questions. It enhances commonsense reasoning in AI without requiring task-specific knowledge integration or a structured knowledge base.
- **Least to Most (Zhou et al., 2023a)**: Least to Most breaks down a complex problem into a series of simpler subproblems and then solves them in sequence. The key idea is to solve each subproblem by using the answers to previously solved subproblems. This method is particularly useful for tasks that require solving problems harder than the exemplars shown in the prompts.

Note that there are plenty of prompt engineering techniques and we tried our best to include those general techniques instead of specific prompt engineering techniques such as Tree of Thoughts (Yao et al., 2023) that requires specific prompt design and decomposition of each problem.

#### B.4.3 Adversarial Prompt Attacks

Adversarial prompt attacks, as proposed by Zhu et al. (2023b), aims to *simulate* potential disturbances that could naturally arise in practical scenarios. The proposed prompt attacks

are intended to resemble common user errors or expressions, as users often make various mistakes when inputting prompts, such as typos, diverse word choices, and different sentence constructions. The prompt attacks encompass four distinct levels:

- **Character-level:** Techniques such as TextBugger (Li et al., 2019) and DeepWordBug (Gao et al., 2018) are employed. These methods introduce errors or typos into words by altering characters.
- **Word-level:** Attacks like BertAttack (Li et al., 2020) and TextFooler (Jin et al., 2019) are utilized. They focus on substituting words with their synonyms or contextually similar alternatives.
- **Sentence-level:** StressTest (Naik et al., 2018) and CheckList (Ribeiro et al., 2020) are applied. These attacks add irrelevant or redundant sentences to prompts.
- **Semantic-level:** To simulate the linguistic styles of different global regions.

### B.5 Pipeline

The full pipeline of using PromptBench for evaluation is shown in Figure 2.

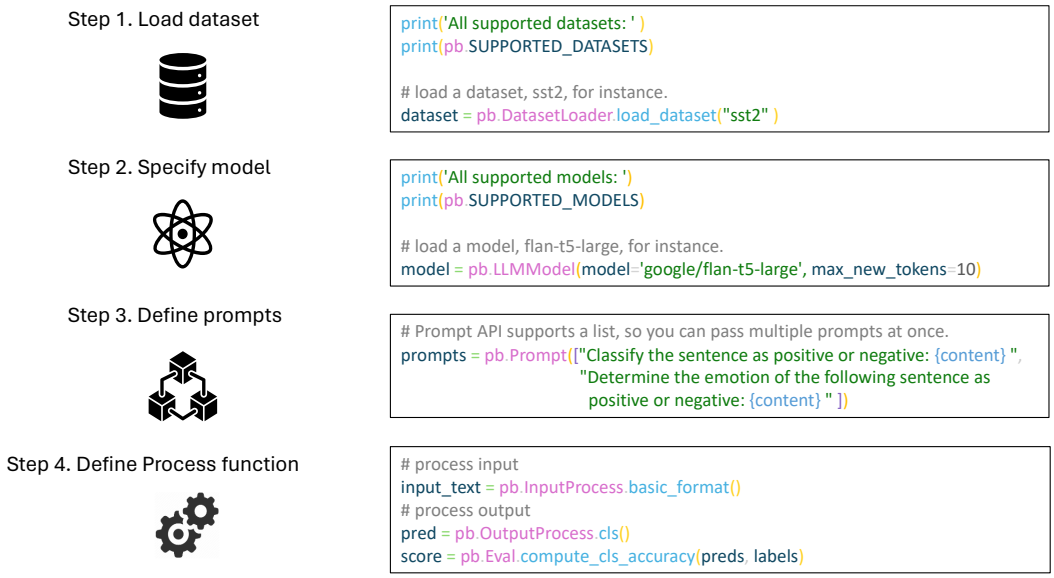


Figure 2: A pipeline for evaluation of LLMs.

## Appendix C. Benchmark Results

### C.1 Adversarial prompt robustness

The partial results of the robustness of different models on a range of tasks are presented in Figure 3. All models exhibit vulnerability to adversarial prompts, with ChatGPT and GPT-4 demonstrating the strongest robustness.

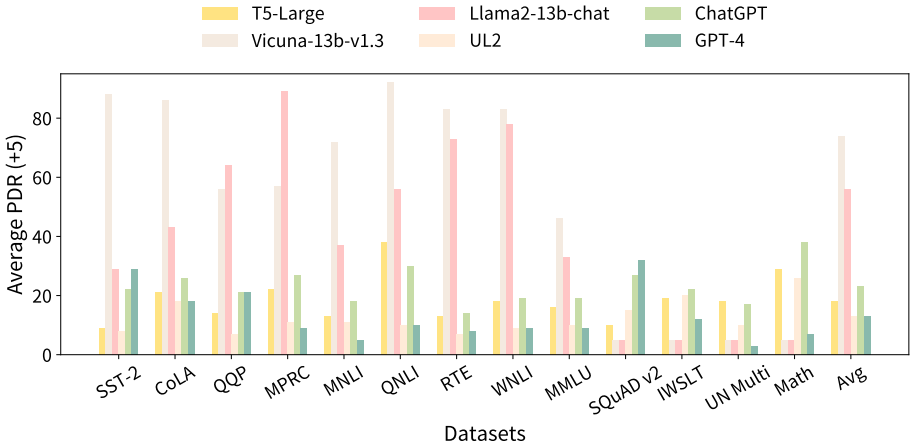


Figure 3: Adversarial prompt robustness results.


Figure 4: Comparison among different prompt engineering techniques.

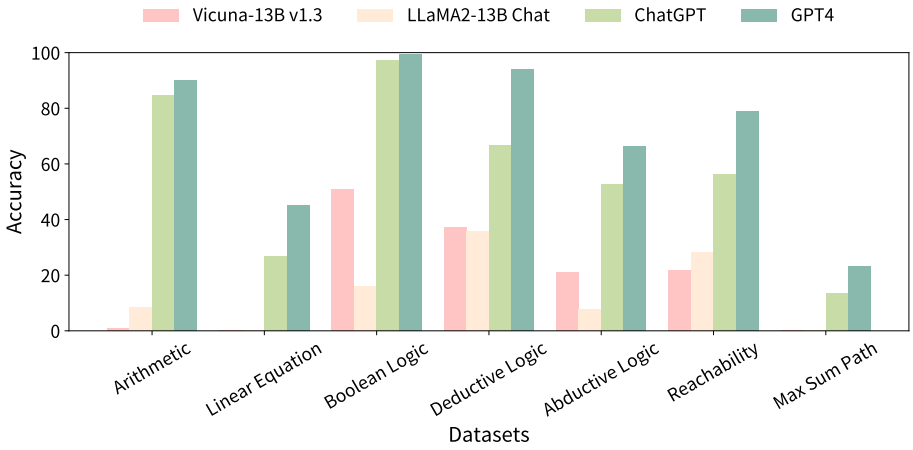


Figure 5: DyVal results.

### C.2 Prompt engineering

Prompt engineering results are shown in Figure 4. Most methods are effective for special fields, so these methods can not surpass the baseline in every dataset.

### C.3 Dynamic evaluation

Figure 5 illustrates the outcomes of dynamic evaluations across various models and tasks. GPT-4 outperforms its counterparts significantly, yet there remains potential for enhancement in the performance of linear equation, abductive logic, and max sum path task.

## Appendix D. Extensibility

Each module in PromptBench can be easily extended. In the following, we provide basic guidelines for customizing your own datasets, models, prompt engineering methods, and evaluation metrics. The sample code for adding new datasets, models can be found in `https://github.com/microsoft/promptbench/blob/main/examples/add_new_modules.md`

### D.1 Add new datasets

Adding new datasets involves two steps:

1. Implementing a New Dataset Class: Datasets are supposed to be implemented in `dataload/dataset.py` and inherit from the `Dataset` class. For your custom dataset, implement the `__init__` method to load your dataset. We recommend organizing your data samples as dictionaries to facilitate the input process.

2. Adding an Interface: After customizing the dataset class, register it in the `DataLoader` class within `dataload.py`.

### D.2 Add new models

Similar to adding new datasets, the addition of new models also consists of two steps.

1. Implementing a New Model Class: Models should be implemented in `models/model.py`, inheriting from the `LLMModel` class. In your customized model, you should implement `self.tokenizer` and `self.model`. You may also customize your own `predict` function for inference. If the `predict` function is not customized, the default `predict` function inherited from `LLMModel` will be used.

2. Adding an Interface: After customizing the model class, register it in the `_create_model` function within the `class LLMModel` and `MODEL_LIST` dictionary in `__init__.py`.

### D.3 Add new prompt engineering methods

Adding new methods in prompt engineering is similar to steps of C.1 and C.2.

1. Implementing a New Methods Class: Methods should be implemented in `prompt_engineering` Module. Firstly, create a new `.py` file for your methods. Then

implement two functions: `__init__` and `query`. For unified management, two points need be noticed: 1. all methods should inherits from `Base` class that has common code for prompt engineering methods. 2. prompts used in methods should be stored in `prompts/method_oriented.py`.

2. Adding an Interface: After implementing a new methods, register it in the `METHOD_MAP` that is used to map method names to their corresponding class.

### D.4 Add new metrics and input/output process functions

New evaluation metrics should be implemented as static functions in `class Eval` within the `metrics` module. Similarly, new input/output process functions should be implemented as static functions in `class InputProcess` and `class OutputProcess` in the `utils` module.