FreeEval: A Modular Framework for Trustworthy and Efficient Evaluation of Large Language Models

Zhuohao Yu[1], Chang Gao[1], Wenjin Yao[1], Yidong Wang[1], Zhengran Zeng[1], Wei Ye[1*], Jindong Wang[2], Yue Zhang[3], Shikun Zhang[1]

[1]Peking University. [2]Microsoft Research. [3]Westlake University.

zyu@stu.pku.edu.cn, wye@pku.edu.cn

Abstract

The rapid growth of evaluation methodologies and datasets for large language models (LLMs) has created a pressing need for their unified integration. Meanwhile, concerns about data contamination and bias compromise the trustworthiness of evaluation findings, while the efficiency of evaluation processes remains a bottleneck due to the significant computational costs associated with LLM inference. In response to these challenges, we introduce FreeEval, a modular framework not only for conducting trustworthy and efficient automatic evaluations of LLMs but also serving as a platform to develop and validate new evaluation methodologies. FreeEval addresses key challenges through: (1) unified abstractions that simplify the integration of diverse evaluation methods, including dynamic evaluations requiring complex LLM interactions; (2) built-in meta-evaluation techniques such as data contamination detection and human evaluation to enhance result fairness; (3) a high-performance infrastructure with distributed computation and caching strategies for efficient large-scale evaluations; and (4) an interactive *Visualizer* for result analysis and interpretation to support innovation of evaluation techniques. We open-source all our code at `https://github.com/WisdomShell/FreeEval`[1].

1 Introduction

Large Language Models (LLMs) have revolutionized Natural Language Processing (NLP) with their impressive performance across various tasks (Brown et al., 2020; Zhang et al., 2022; Bubeck et al., 2023; OpenAI, 2023). As LLMs play a critical role in academia and industry, evaluating their capabilities has become essential (Guo et al., 2023). Consequently, researchers have proposed automatic evaluation methodologies using benchmark datasets (Clark et al., 2018; Zellers et al., 2019; Cobbe et al., 2021; Bang et al., 2023) for objective assessments, and LLM-based subjective evaluation tools (Wang et al., 2023c; Zheng et al., 2023b; Li et al., 2023b; Chan et al., 2023).

The rapid emergence of evaluation data and methods has intensified the challenge of incorporating cutting-edge techniques cost-effectively and conducting reliable evaluations. In response, several open-source evaluation platforms for LLMs have been proposed, each with unique features. Table 1 provides a comprehensive comparison. Specifically, Eval-Harness (Gao et al., 2021) evaluates LLMs using various benchmark datasets. HELM (Liang et al., 2022) offers metrics beyond accuracy on custom datasets and models. OpenAI Evals (Contributors, 2023) implements interfaces for LLM-based judges and their meta-evaluation. OpenCompass (Contributors, 2023b) introduces distributed inference with SLURM (Yoo et al., 2003) on clusters. PromptBench (Zhu et al., 2023b) incorporates prompt attacks and DyVal (Zhu et al., 2023a) in its framework.

Despite these promising efforts, current evaluation platforms still face three bottlenecks.

First, *a unified and extensible framework is required to integrate evaluation methods seamlessly*. This consequently affects the flexibility, transparency, and interpretability of the evaluation. The evaluation results are highly dependent on the deployment settings and prompting techniques, as LLMs are not robust enough for these intricate settings (Zheng et al., 2023a). For example, Table 2 shows that these settings can significantly influence results, confirming the need for standardized implementation of evaluation methods to assure consistent assessment.

Second, *the reliability of results from these platforms cannot always be guaranteed.* Automatic evaluation of LLMs remains a challenging

[*] Corresponding author.

[1] Our demonstration video, live demo, and installation guides are available at: `https://freeeval.zhuohao.me/`

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing System Demonstrations, pages 1–13
November 12-16, 2024 ©2024 Association for Computational Linguistics

Table 1: Comparison of popular evaluation toolkits on features.

Toolkit	Custom Datasets	Custom Models	Custom Prompting	LLM Judges	Dynamic Evaluation	Distributed Inference	Contamination Detection	Meta Evaluation	Visual Analysis
Eval-Harness (Gao et al., 2021)	✓	✓	✓	✗	✗	✗	✗	✗	✗
HELM (Liang et al., 2022)	✓	✓	✓	✗	✗	✗	✗	✗	✗
OpenAI Evals (Contributors, 2023)	✓	✓	✓	✓	✗	✗	✗	✓	✗
BIG-Bench (Contributors, 2023)	✓	✓	✓	✗	✗	✗	✗	✗	✗
OpenCompass (Contributors, 2023b)	✓	✓	✓	✓	✗	✓	✗	✗	✗
PromptBench (Zhu et al., 2023b)	✓	✓	✓	✗	✓	✗	✗	✗	✗
UltraEval (He et al., 2024)	✓	✓	✓	✗	✗	✓	✗	✗	✗
FreeEval (Ours)	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 2: Comparison of different inference implementations. We report 25-shot accuracy of `llama-2-7b-chat` on ARC-Challenge (Clark et al., 2018), 5-shot accuracy on MMLU (Hendrycks et al., 2020) and HellaSwag (Zellers et al., 2019). 'CP' and 'MCP' denote Cloze Prompt and Multiple Choice Prompt from Robinson et al. (2022).

Method	ARC-C	MMLU	HellaSwag
CP+PromptA	51.11%	40.65%	50.07%
CP+PromptB	47.53%	38.72%	50.19%
MCP+PromptA	54.18%	42.73%	30.61%
MCP+PromptB	54.10%	41.28%	30.96%

task (Chang et al., 2023) due to their open-ended nature and the presence of data contamination, which lead to inflated performance metrics (Schaeffer, 2023; Sainz et al., 2023; Yu et al., 2024). Moreover, the lack of tools for in-depth analysis and visualization of evaluation results makes it difficult for researchers to interpret the performance of LLMs across different tasks and scenarios.

Third, *the efficiency of previous evaluation toolkits has significant room for improvement*. LLM inference could be a substantial challenge for both industry and researchers, since it requires strong GPUs or paid APIs, especially for large-scale evaluations (Wang et al., 2023c). Optimizing inference computation is crucial for reducing the costs of LLM evaluation and supporting rapid iteration in both evaluation and development.

To address these challenges, we propose FreeEval, a modular and extensible framework for trustworthy and efficient automatic evaluation of LLMs, as well as a platform for developing new evaluation methodologies. The main features of FreeEval are:

Unified abstraction and modular implementation of various evaluation methods. We introduce concepts of `step`, `dataset`, and `config` to uniformly describe dataset-based, classic reference-based, and LLM-based evaluators. Dataset-based evaluators include task-specific datasets along with dataset operations such as custom prompting, data augmenting and generation. LLM-based evaluators, such as MT-Bench (Zheng et al., 2023b), AlpacaEval (Li et al., 2023b), PandaLM (Wang et al., 2023c) and KIEval (Yu et al., 2024), are also integrated to provide subjective assessment. Classic Judges, which utilize reference-based evaluation metrics like ROUGE (Lin, 2004) and BERTScore (Zhang et al., 2019) to examine model output. FreeEval's modular design allows for easy implementation of new evaluation protocols and supports evaluating both open-source and proprietary models. The abstractions also bring transparency to the evaluation process since all the evaluation settings are open to users.

Practical meta-evaluation modules for trustworthiness. FreeEval incorporates contamination detection, human judgment, case analysis, and bias evaluation. These features mitigate overfitting risks, enhance interpretability, and support the development and validation of new evaluation methods. A user-friendly interface for human annotation further improves explainability and reliability of results.

Optimized distributed and concurrent inference with load balancing and caching mechanisms. Leveraging cutting-edge inference engines with concurrency and caching strategies, FreeEval efficiently handles large-scale evaluations on multi-node multi-GPU clusters. This infrastructure supports both open-source models and proprietary APIs, ensuring scalability and cost-effectiveness.

Intuitive Visualizer for result analysis and interpretation. This component provides interactive tools for exploring results, conducting case studies, and identifying patterns. It enhances interpretability and supports the development of new evaluation methods through visual feedback.

By combining these features, FreeEval addresses key challenges in LLM evaluation while serving as a powerful platform for researchers to build new evaluation methods.

2 Background

In this section, we provide an overview of the current landscape of LLM evaluation methods, the challenges posed by data contamination, and the importance of meta-evaluation in assessing the reliability and validity of evaluation protocols.

2.1 Automatic Evaluation Methods for LLMs

The rapid development of Large Language Models (LLMs) has led to the emergence of various evaluation methods, each aiming to assess different aspects of model performance. These methods can be broadly categorized into three groups: classic reference-based evaluation, dataset-based benchmarks, and LLM-based evaluators.

Reference-Based Evaluation methods, such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and BERTScore (Zhang et al., 2019), assess the quality of generated text by comparing it against human-written references. While straightforward, they may not fully capture the open-ended nature of LLM-generated outputs and can be sensitive to reference quality and diversity (Wang et al., 2023c).

Dataset-Based Benchmarks, such as ARC (Clark et al., 2018), HellaSwag (Zellers et al., 2019), MMLU (Hendrycks et al., 2020), and C-Eval (Huang et al., 2023), evaluate LLMs using carefully curated datasets that test specific skills or knowledge. However, they may not fully capture the open-ended nature of LLMs and can be vulnerable to data contamination (Schaeffer, 2023; Wei et al., 2023).

LLM-Based Evaluators leverage strong LLMs, such as GPT-4 (OpenAI, 2023), to assess the performance of other models. Examples include PandaLM (Wang et al., 2023c), MT-Bench (Zheng et al., 2023b), GPTScore (Fu et al., 2023), PRD (Li et al., 2023a), and KIEval (Yu et al., 2024). These evaluators can capture nuanced aspects of language understanding and generation, but their performance is influenced by the evaluator LLM and prompting strategies. Biases present in the evaluator LLM may propagate to the evaluation process (Zeng et al., 2023; Wang et al., 2023b), requiring careful meta-evaluation. Additionally, the inference cost of LLMs necessitates optimization for large-scale evaluation.

2.2 Meta-Evaluation of LLMs

Meta-evaluation refers to the process of evaluating the fairness, reliability, and validity of evaluation protocols themselves. We incorporate several meta-evaluation methods into FreeEval.

Data Contamination occurs when an LLM is exposed to test data during training, leading to inflated performance scores and an inaccurate assessment of the model's true capabilities (Schaeffer, 2023; Sainz et al., 2023; Zhu et al., 2023a). This issue is particularly important due to its impact on evaluation fairness, and should be considered. We implement data contamination detection methods like Min-K prob (Shi et al., 2023) and average loss (Wei et al., 2023) in FreeEval as modules, to make contamination detection a fundamental process in evaluating LLMs or creating a new evaluation protocol.

Human Evaluation is the gold standard for meta-evaluation (Chang et al., 2023), as it directly reflects human preferences on generated texts. This is particularly important for LLM-based evaluators, which subjectively evaluate output quality like human experts. However, the lack of standardized platforms or guidelines for human annotation can lead to biased, inconsistent, and unfair judgments. To address this, we incorporate meta-evaluation protocols from Wang et al. (2023c); Zeng et al. (2023); Zheng et al. (2023b), as they reflect preferences from human experts in different scenarios. Additionally, we create a user-friendly interface for human experts to create new preference datasets, facilitating the collection of high-quality human evaluations for meta-evaluation purposes.

3 Design and Implementation

In this section, we present the design and implementation of FreeEval, we discuss the framework's architecture, its key components, and how they address the challenges identified previously.

3.1 Design Principles

To build a flexible, efficient research tool for LLM evaluation we make sure the architecture of FreeEval follows the following principles:

- **Modular**: Enables easy integration of new evaluation methods, datasets, and protocols. Ensures transparency by making all evaluation settings and details openly accessible to users.
- **Trustworthy**: Promotes fair and effective evaluation processes. Supports meta-evaluation for validating evaluation methods and ensures result interpretability.

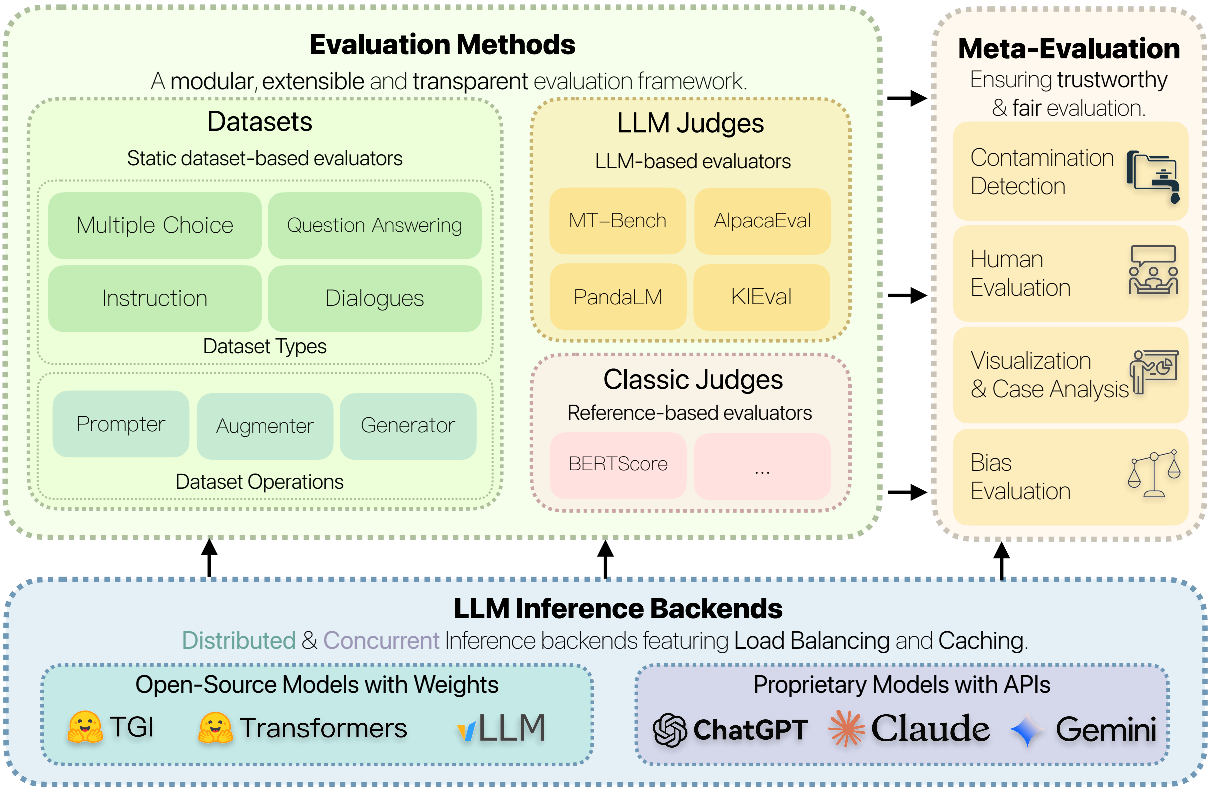


Figure 1: Overall architecture of FreeEval.

- **Efficient**: Minimizes computational costs for LLM inference, enabling large-scale evaluations and rapid prototyping of new methodologies.

3.2 FreeEval Architecture Overview

FreeEval's architecture, illustrated in Figure 1, features a modular design that could be separated into Evaluation Methods, Meta-Evaluation and LLM Inference Backends. Evaluation Methods contain different datasets and implementation for evaluation methods. The Meta-Evaluation module ensures the integrity and fairness of assessments by providing data contamination detection methods and popular meta-evaluation method implementation. LLM Inference Backends form the computational backbone, as it provide distributed and concurrent inference of LLMs featuring performance optimization techniques.

3.3 Extensible Modular Design

FreeEval's modular architecture is designed to accommodate the rapidly evolving landscape of LLM evaluation. To help users implement evaluation methods without complexity, FreeEval is implemented around the concept of `step`, `dataset` and `config`, which serve as the building blocks for creating flexible and extensible evaluation pipelines:

- **`step`**: A `step` encapsulates a specific evaluation method, data augmentation technique, or metric calculation. Each `step` contain three phases: `preprocess` handles initializing the required `dataset` or models; `run` handles the execution; `postprocess` parse the outputs, collects evaluation results and free up the resources.
- **`dataset`**: Data used by the evaluators are defined as `dataset`. Each `dataset` handles the preprocessing required to load data, few-shot settings, prompting, augmentation of instances, and postprocessing of inference results.
- **`config`**: A `config` file is used to compose evaluation pipelines with `steps` and `datasets`. The config file contains all the details and settings. `steps` defined in the `config` are executed sequentially, and they share the same context which stores intermediate results.

These abstractions improve transparency in evaluations by providing users with full access to the configuration details for each evaluation pipeline. The `config` file also serves as a complete record of the evaluation process, including all necessary hyperparameters and settings. The modular design also allow data to be re-used in different scenarios without effort. For example, GSM8K (Cobbe et al., 2021) is a evaluation dataset, we could simply calculate perplexity of models on this dataset, or we could use a data generation `step` to generate new data with GPT-4 in the same distribution to detect data contamination following Wei et al. (2023). The modular approach allows researchers to easily add new evaluation methods or modify existing ones without disrupting the overall structure of the framework. By defining each evaluator as a self-contained unit, FreeEval promotes code reusability and maintainability.

This configuration-driven approach eliminates the need for users to write Python code when defining and running an evaluation pipeline. All settings and parameters for each `step` and `dataset` are specified within the `config`, making the evaluation process highly customizable and accessible to researchers with varying levels of programming expertise. Figure 2 shows an example config for a pipeline evaluating LLaMA-2 70B (Touvron et al., 2023b) on ARC-Challenge (Clark et al., 2018) dataset with a fixed seed for sampling 25-shot examples and custom prompt. The model can be deployed locally or on a remote machine. The pipeline also include detecting data contamination with Min-K% Prob (Shi et al., 2023).

```
{
  "steps": [
    {
      "step_name": "ARC-Challenge 25-shot MCP",
      "step_type": "simple_multiple_choice",
      "dataset_config": {
        "type": "arc_challenge",
        "dataset_kwargs": {
          "seed": 2,
          "fewshot_split": "train",
          "fewshot_num": 25,
          "multiple_choice_template_name": "prompt1"}
      },
      "inference_config": {
        "type": "remote_hf",
        "inference_kwargs": {
          "model_name": "llama2-70b",
          "base_url": ...,
          "generation_config": ... }
      },
      "eval_config": {"aggregate_mode": "mean"}
    },
    {
      "step_name": "Contamination Detection",
      "step_type": "min_k_prob",
      "dataset_config": ...,
      "inference_config": ...
    }
  ]
}
```

Figure 2: Config for an example pipeline, evaluating LLaMA-2 70B (Touvron et al., 2023b) on ARC-Challenge (Clark et al., 2018) dataset and then detecting data contamination with Min-K% Prob (Shi et al., 2023).

3.4 Trustworthy Evaluation

FreeEval prioritizes trustworthiness and fairness in evaluations by incorporating a range of meta-evaluation modules that validates the evaluation results and processes. As human preference remain the gold standard for measuring the effectiveness of evaluation protocols, FreeEval model human preference into two types: *pairwise comparison* and *direct scoring*. We incorporate existing meta-evaluation datasets from PandaLM (Wang et al., 2023c), MT-Bench (Zheng et al., 2023b), LLMBar (Guo et al., 2023), AlpacaEval (Li et al., 2023b), and provide a user-friendly interface for annotating and curating human evaluation datasets.

To ensure the trustworthiness of the evaluation results, we also implement data contamination detection methods, as introduced in subsection 2.2, into our toolkit as `steps`. Understanding whether the tested dataset appear in the training phase of the evaluated models would help users assess the validity and reliability of evaluation results. We also provide bias evaluation modules and visualization tools specifically for LLM-based evaluators, as previous studies have reported they exhibit position bias and length bias (Zheng et al., 2023b; Wang et al., 2023c). These meta-evaluation modules can be easily integrated into existing evaluation pipelines, allowing researchers to understand the effectiveness of their results, the fairness of the evaluation process, and study bad cases that lead to unexpected evaluation results.

3.5 Efficient Inference Backends

FreeEval's high-performance inference backends are designed to efficiently handle the computational demands of large-scale LLM evaluations.

The inference backends in FreeEval support both open-source models and proprietary models with APIs. For all models, FreeEval support concurrent inference given a fixed number of workers. We implement a caching mechanism for queries based on hash values of the request. We hash the request prompt and inference config, and store locally the request content and response for each individual request. By checking the cache before making a query, FreeEval skips cached requests, enabling quick recovery from exceptions and saving inference costs. This is particularly beneficial when implementing and debugging new evaluation methods. Caching also ensures reproducibility, as all requests, settings, and responses are saved and can

```python
from freeeval.models import load_inference_function

# Initialize inference backends
openai_inference = load_inference_function("openai")
huggingface_inference = load_inference_function("remote_hf")

# Parallel inference with load balancing and caching
huggingface_inference(
    requests,
    output_path,
    max_concurrency = 128,
    num_workers = 8
)
openai_inference(
    requests,
    output_path,
    openai_model,
    api_key,
    num_workers = 4,
    request_per_min = 100
)
```

Figure 3: Example code for running FreeEval's inference backends. We rely on these backends for efficient inference and provide a simple abstraction.

be inspected using FreeEval's visualization tools.

For open-source models, we leverage Huggingface's `text-generation-inference` (TGI, Contributors (2023a)) package which is a production-ready high-performance inference toolkit. We implement a load-balancing technique in conjunction with the continuous batching feature provided by TGI to maximize GPU utilization on multi-node multi-GPU clusters. For proprietary models, we have a rate-limiting mechanism to avoid causing too much stress on API providers.

We evaluate FreeEval's performance by comparing the execution times (excluding downloading times) for `llama-2-7b-chat-hf` model on 3 common datasets using different toolkits. Our experiments are done on the same Ubuntu machine with a single NVIDIA A800 80GB PCIe GPU. As shown in Table 3, even on a single GPU, FreeEval exhibit significant advantage on all benchmark datasets.

The inference backends in FreeEval are designed to seamlessly integrate with the evaluation methods of the framework. As illustrated in Figure 3, initializing the inference backends and running parallel inference is straightforward and user-friendly. This simplicity allows developers of new evaluation methods to focus on prompting or interactions between models, using the backends sequentially. As a result, implementing interactive evaluation methods, such as those proposed by Li et al. (2023a); Chan et al. (2023); Yu et al. (2024), becomes much easier and more accessible to researchers.

Table 3: Comparison of execution time (in hours) of different toolkits. All experiments are done on the same machine with a single NVIDIA A800 80GB PCIe GPU.

Toolkit	ARC-C	MMLU	HellaSwag
Eval-Harness	0.160	0.510	1.080
OpenCompass	0.084	1.431	1.716
FreeEval (Sequential)	0.211	0.949	0.966
FreeEval (Concurrent)	**0.067**	**0.233**	**0.357**

3.6 FreeEval Visualizer

Unlike traditional evaluation toolkits that provide only accuracy or performance scores, FreeEval automatically converts and saves evaluation results for comprehensive visualization. Users can launch the Visualizer with a simple command for an intuitive web interface for detailed analysis.

The FreeEval Visualizer offers a dashboard overview of evaluation results and settings, in-depth analysis tools, a case browser for examining individual cases and a human evaluation toolkit. These features enable researchers to explore outcomes, identify patterns, and study potential biases or anomalies. By providing immediate visual feedback, the Visualizer aids in rapid prototyping and refinement of new evaluation methodologies, contributing to the trustworthiness and interpretability of the evaluation process.

For detailed screenshots and a comprehensive introduction to the Visualizer's functionalities, please refer to Appendix B. A demonstration video and live demo are also available on our project website.

4 Conclusion

We introduce FreeEval, a modular and extensible framework for trustworthy and efficient automatic evaluation of LLMs. FreeEval innovatively addresses key challenges in LLM evaluation by providing a unified implementation of various evaluation methods, incorporating meta-evaluation modules, and leveraging high-performance inference backends. The framework's modular design facilitates easy integration of new evaluation protocols and improves transparency. The integrated Visualizer enhances result interpretation and analysis, supporting comprehensive evaluation and the development of new methodologies. We will continue to maintain and expand the FreeEval toolkit, striving to provide deeper insights into the capabilities and limitations of LLMs and contribute to the development of more robust and trustworthy language models.

A Limitations and Ethical Considerations

In this Appendix section, we discuss the limitations and ethical considerations of FreeEval. While FreeEval addresses several challenges in LLM evaluation, it has limitations and raises ethical considerations:

- **Bias and Discrimination:** FreeEval includes bias evaluation modules but cannot eliminate biases inherent in training data or models. Researchers should strive for more inclusive and equitable LLMs.
- **Environmental Impact:** Despite efficient inference backends, the overall environmental impact of LLM development remains a concern requiring further innovation.
- **Human Evaluation Subjectivity:** The human evaluation component may introduce subjective biases, necessitating careful design of evaluation protocols.
- **Accountability and Misuse:** While FreeEval enhances transparency in evaluation, ethical deployment and appropriate safeguards in real-world applications remain the responsibility of researchers and developers.

These points highlight the need for ongoing research in LLM evaluation methodologies and responsible AI development practices.

B FreeEval Visualizer

The FreeEval Visualizer is a web-based interface designed to enhance the interpretability and analysis of LLM evaluation results. It provides an intuitive platform for researchers to explore evaluation data, conduct case studies, and perform human evaluations.

The Visualizer consists of six main components:

- **Dashboard**: Offers an overview of evaluation results, including distribution charts and summary statistics.
- **Analysis Tools**: Provides detailed visualizations and statistical analyses of evaluation data.
- **Case Browser**: Allows users to search, filter, and examine individual evaluation cases.
- **Human Evaluation Creator**: Enables researchers to set up new human evaluation sessions.
- **Human Evaluation Session**: Manages ongoing human evaluation tasks.
- **Case Annotation Interface**: Facilitates detailed annotation of individual cases.

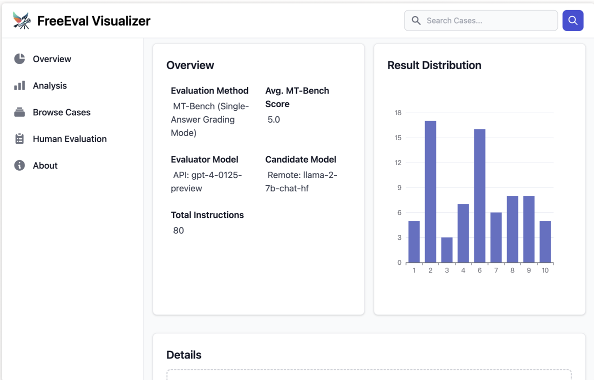
The Visualizer is built using Flask, a lightweight Python web framework, and incorporates modern front-end technologies for responsive design. It integrates seamlessly with FreeEval's core evaluation modules, providing a unified workflow for LLM assessment.

Key features of the Visualizer include interactive data exploration, customizable visualizations, and support for various evaluation types (e.g., pairwise comparisons, direct scoring). The human evaluation interfaces facilitate the creation, management, and execution of expert judgment collection, which can be used for meta-evaluation or to create new evaluation datasets.

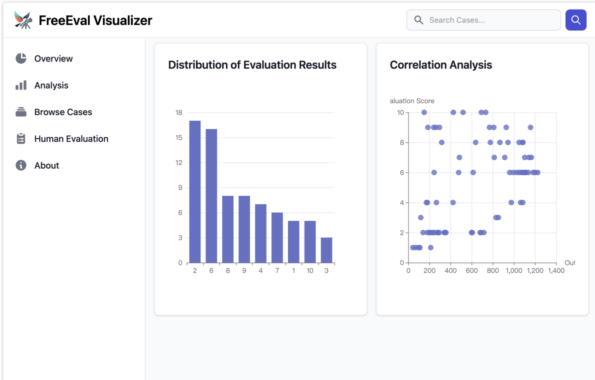
Figure 4 showcases the main interfaces of the FreeEval Visualizer. The dashboard (Figure 4a) provides an overview of evaluation results, while the analysis page (Figure 4b) offers more detailed statistical insights. The case browser (Figure 4c) allows for detailed exploration of individual cases.

The human evaluation workflow is supported by three interfaces: the creation page for setting up new evaluation sessions (Figure 4d), the session management page (Figure 4e) for overseeing ongoing evaluations, and the case annotation interface (Figure 4f) for collecting detailed judgments on specific outputs.

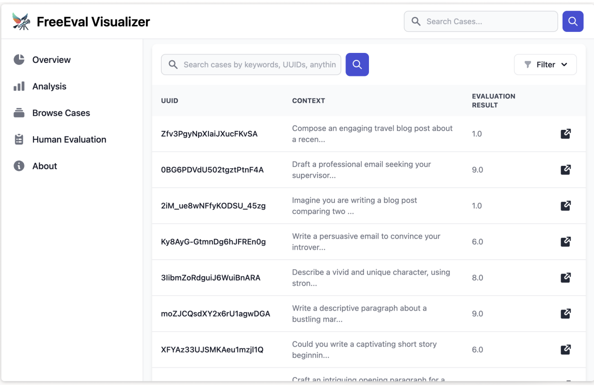
By providing these visual and interactive tools, the FreeEval Visualizer aims to streamline the process of analyzing LLM evaluation results, enabling researchers to gain deeper insights and make more informed decisions in their work with large language models. The comprehensive set of features supports the entire evaluation lifecycle, from initial data exploration to in-depth analysis and human-in-the-loop assessment.



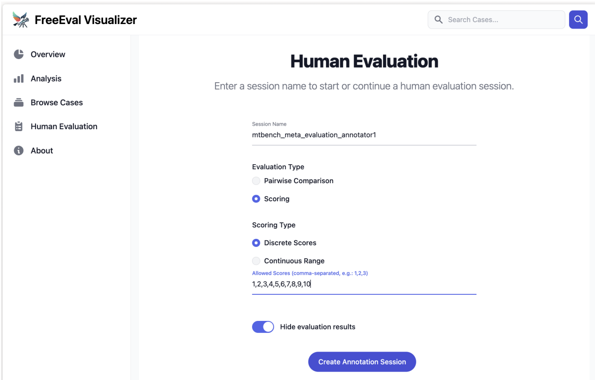
(a) Dashboard overview



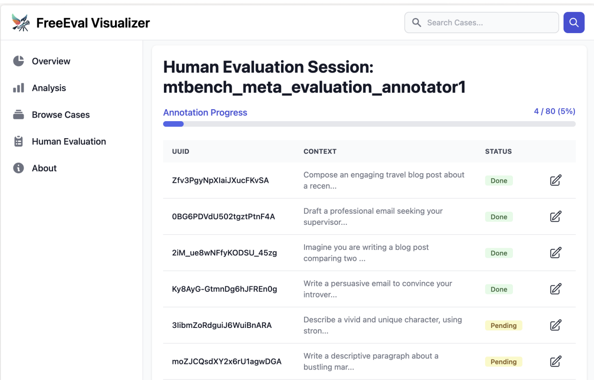
(b) Overall analysis



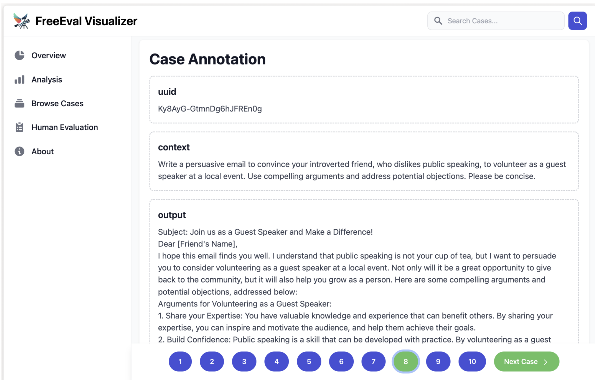
(c) Case Browser



(d) Creating human evaluation session



(e) Human evaluation session



(f) Case Annotation

Figure 4: Screenshots of the FreeEval Visualizer web application

References

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

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. 2017. A closer look at memorization in deep networks. In *International conference on machine learning*, pages 233–242. PMLR.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. 2023. Open llm leaderboard. `https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard`.

Yoshua Bengio and Yann LeCun. 2007. Scaling learning algorithms towards AI. In *Large Scale Kernel Machines*. MIT Press.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. 2023. The reversal curse: Llms trained on" a is b" fail to learn" b is a". *arXiv preprint arXiv:2309.12288*.

Stella Biderman, USVSN Sai Prashanth, Lintang Sutawika, Hailey Schoelkopf, Quentin Anthony, Shivanshu Purohit, and Edward Raf. 2023a. Emergent and predictable memorization in large language models. *arXiv preprint arXiv:2304.11158*.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023b. Pythia: A suite for analyzing large language models across training and scaling. *arXiv preprint arXiv:2304.01373*.

Rishi Bommasani, Percy Liang, and Tony Lee. 2023. Holistic evaluation of language models. *Annals of the New York Academy of Sciences*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2023. Chateval: Towards better llm-based evaluators through multi-agent debate. *arXiv preprint arXiv:2308.07201*.

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

Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? *arXiv preprint arXiv:2305.01937*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Contributors. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Contributors. 2023. Openai evals. `https://github.com/openai/evals`.

Contributors. 2023a. Text generation inference: A rust, python and grpc server for text generation inference. `https://github.com/huggingface/text-generation-inference`.

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

Luigi Daniele and Suphavadeeprasit. 2023. Amplify-instruct: Synthetically generated diverse multi-turn conversations for effecient llm training. *arXiv preprint arXiv:(comming soon)*.

Nolan Dey, Gurpreet Gosal, Hemant Khachane, William Marshall, Ribhu Pathria, Marvin Tom, Joel Hestness, et al. 2023. Cerebras-gpt: Open compute-optimal language models trained on the cerebras wafer-scale cluster. *arXiv preprint arXiv:2304.03208*.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.

Mengnan Du, Subhabrata Mukherjee, Yu Cheng, Milad Shokouhi, Xia Hu, and Ahmed Hassan. 2023. Robustness challenges in model distillation and pruning for natural language understanding. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1758–1770.

Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. 2024. Do membership inference attacks work on large language models? *arXiv preprint arXiv:2402.07841*.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpacafarm: A simulation framework for methods that learn from human feedback. *arXiv preprint arXiv:2305.14387*.

Dom Eccleston. 2023. Sharegpt dataset. `https://sharegpt.com/`.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021. A framework for few-shot language model evaluation.

Varun Godbole, George E. Dahl, Justin Gilmer, Christopher J. Shallue, and Zachary Nado. 2023. Deep learning tuning playbook. Version 1.0.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep learning*, volume 1. MIT Press.

Google. 2023. Bard.

Zishan Guo, Renren Jin, Chuang Liu, Yufei Huang, Dan Shi, Linhao Yu, Yan Liu, Jiaxuan Li, Bojian Xiong, Deyi Xiong, et al. 2023. Evaluating large language models: A comprehensive survey. *arXiv preprint arXiv:2310.19736*.

Chaoqun He, Renjie Luo, Shengding Hu, Yuanqian Zhao, Jie Zhou, Hanghao Wu, Jiajie Zhang, Xu Han, Zhiyuan Liu, and Maosong Sun. 2024. Ultraeval: A lightweight platform for flexible and comprehensive evaluation for llms.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh. 2006. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554.

Lynette Hirschman and Robert Gaizauskas. 2001. Natural language question answering: the view from here. *natural language engineering*, 7(4):275–300.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

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

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

Marzena Karpinska, Nader Akoury, and Mohit Iyyer. 2021. The perils of using mechanical turk to evaluate open-ended text generation. *arXiv preprint arXiv:2109.06835*.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Ruosen Li, Teerth Patel, and Xinya Du. 2023a. Prd: Peer rank and discussion improve large language model based evaluations. *arXiv preprint arXiv:2307.02762*.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023b. AlpacaEval: An Automatic Evaluator of Instruction-following Models.

Yucheng Li. 2023. An open source data contamination report for llama series models. *arXiv preprint arXiv:2310.17589*.

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

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Yen-Ting Lin and Yun-Nung Chen. 2023. Llm-eval: Unified multi-dimensional automatic evaluation for open-domain conversations with large language models. *arXiv preprint arXiv:2305.13711*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. Gpteval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.

Carlo A Mallio, Andrea C Sertorio, Caterina Bernetti, and Bruno Beomonte Zobel. 2023. Large language models for structured reporting in radiology: performance of gpt-4, chatgpt-3.5, perplexity and bing. *La radiologia medica*, pages 1–5.

MosaicML. 2023. Introducing mpt-7b: A new standard for open-source, commercially usable llms.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for nlg. *arXiv preprint arXiv:1707.06875*.

OpenAI. 2023. Gpt-4 technical report.

Yonatan Oren, Nicole Meister, Niladri Chatterji, Faisal Ladhak, and Tatsunori B Hashimoto. 2023. Proving test set contamination in black box language models. *arXiv preprint arXiv:2310.17623*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

Kaiping Peng, Richard E Nisbett, and Nancy YC Wong. 1997. Validity problems comparing values across cultures and possible solutions. *Psychological methods*, 2(4):329.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems*, 34:4816–4828.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Samyam Rajbhandari, Olatunji Ruwase, Jeff Rasley, Shaden Smith, and Yuxiong He. 2021. Zero-infinity: Breaking the gpu memory wall for extreme scale deep learning. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–14.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506.

Joshua Robinson, Christopher Rytting, and David Wingate. 2022. Leveraging large language models for multiple choice question answering. *ArXiv*, abs/2210.12353.

Oscar Sainz, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023. Nlp evaluation in trouble: On the need to measure llm data contamination for each benchmark. *arXiv preprint arXiv:2310.18018*.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Rylan Schaeffer. 2023. Pretraining on the test set is all you need. *arXiv preprint arXiv:2309.08632*.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2023. Detecting pretraining data from large language models. *arXiv preprint arXiv:2310.16789*.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings 18*, pages 194–206. Springer.

Ekaterina Svikhnushina, Anastasiia Filippova, and Pearl Pu. 2022. iEval: Interactive evaluation framework for open-domain empathetic chatbots. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 419–431, Edinburgh, UK. Association for Computational Linguistics.

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

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

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

Lewis Tunstall, Leandro Von Werra, and Thomas Wolf. 2022. *Natural language processing with transformers*. " O'Reilly Media, Inc.".

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

Cunxiang Wang, Xiaoze Liu, Yuanhao Yue, Xiangru Tang, Tianhang Zhang, Cheng Jiayang, Yunzhi Yao, Wenyang Gao, Xuming Hu, Zehan Qi, Yidong Wang, Linyi Yang, Jindong Wang, Xing Xie, Zheng Zhang, and Yue Zhang. 2023a. Survey on factuality in large language models: Knowledge, retrieval and domain-specificity.

Cunxiang Wang, Ruoxi Ning, Boqi Pan, Tonghui Wu, Qipeng Guo, Cheng Deng, Guangsheng Bao, Qian Wang, and Yue Zhang. 2024. Novelqa: A benchmark for long-range novel question answering.

Peiyi Wang, Lei Li, Liang Chen, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023b. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*.

Yidong Wang, Zhuohao Yu, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, Xing Xie, et al. 2023c. Pandalm: An automatic evaluation benchmark for llm instruction tuning optimization. *arXiv preprint arXiv:2306.05087*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*.

Tianwen Wei, Liang Zhao, Lichang Zhang, Bo Zhu, Lijie Wang, Haihua Yang, Biye Li, Cheng Cheng, Weiwei Lü, Rui Hu, et al. 2023. Skywork: A more open bilingual foundation model. *arXiv preprint arXiv:2310.19341*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, et al. 2020. Clue: A chinese language understanding evaluation benchmark. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4762–4772.

Linyi Yang, Shuibai Zhang, Libo Qin, Yafu Li, Yidong Wang, Hanmeng Liu, Jindong Wang, Xing Xie, and Yue Zhang. 2022. Glue-x: Evaluating natural language understanding models from an out-of-distribution generalization perspective. *arXiv preprint arXiv:2211.08073*.

Linyi Yang, Shuibai Zhang, Zhuohao Yu, Guangsheng Bao, Yidong Wang, Jindong Wang, Ruochen Xu, Wei Ye, Xing Xie, Weizhu Chen, and Yue Zhang. 2023. Supervised knowledge makes large language models better in-context learners.

Andy B Yoo, Morris A Jette, and Mark Grondona. 2003. Slurm: Simple linux utility for resource management. In *Workshop on job scheduling strategies for parallel processing*, pages 44–60. Springer.

Zhuohao Yu, Chang Gao, Wenjin Yao, Yidong Wang, Wei Ye, Jindong Wang, Xing Xie, Yue Zhang, and Shikun Zhang. 2024. Kieval: A knowledge-grounded interactive evaluation framework for large language models. *arXiv preprint arXiv:2402.15043*.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34:27263–27277.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

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

Zhengran Zeng, Yidong Wang, Rui Xie, Wei Ye, and Shikun Zhang. 2024. Coderujb: An executable and unified java benchmark for practical programming scenarios.

Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. 2023. Evaluating large language models at evaluating instruction following. *arXiv preprint arXiv:2310.07641*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2023a. Large language models are not robust multiple choice selectors. In *The Twelfth International Conference on Learning Representations*.

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

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

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

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