

# HW2 Mini-LIMA

CS2916 大语言模型

饮水思源 愛國榮校

<https://plms.ai/teaching/index.html>



# Overview

---

- You should learn and finish the following 3 sub tasks:
  - Instruction data construction (指令数据集构建)
  - Aligning via supervised fine-tuning (微调)
  - Model performance evaluation (模型表现评估)
  
- For reference resource for HW2, please check:
  - <https://github.com/GAIR-NLP/cs2916/tree/main/hw2>



# LIMA: Less is More for Alignment

---

- Alignment:
  - Turn your base model better for end tasks and user preferences
- LIMA: <https://arxiv.org/pdf/2305.11206.pdf>
  - 1K instruction data can strongly boost model's alignment ability
  - Quality > Quantity !



# How to Build Your Own Instruction Data?

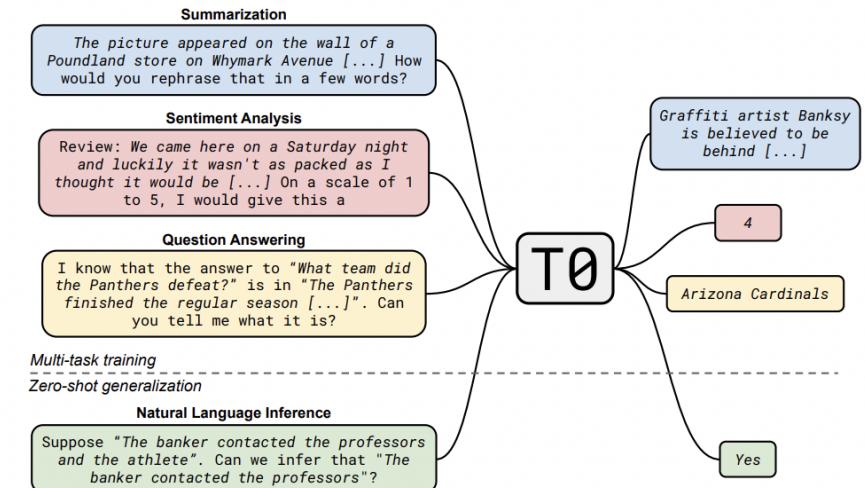
---

- Self-instruct:
  - <https://arxiv.org/pdf/2212.10560.pdf>
- Utilize LLMs for instruction data construction.
  - You can try: GPT-3.5-Turbo (powerful, fast and relatively “cheap”)
  - Of course, you can also try other LLM API services like:
    - **Mistral:** <https://openrouter.ai/models/mistralai/mistral-7b-instruct:free>
    - **Groq:** <https://console.groq.com/docs/text-chat>
  - We also recommend you can try “smaller” models for “high” quality data generation



# Training

- Alignment Methods:
  - **Supervised fine-tuning**
  - Reinforcement Learning from Human Feedback
  - Reinforcement Learning from AI Feedback
  
- Due to computation limit:
  - Recommend: <https://qwenlm.github.io/blog/qwen1.5> (Qwen-1.5-0.5B / 1.8B)





# Evaluation

- Two Mainstream Evaluation Methods:
  - Compare Mode: Compare the responses of the models pairwise to calculate the win rate against each other.
  - Scoring Method: Evaluate responses for individual models and assign scores (e.g., [Chatbot Arena](#))
- Alpaca Eval:
  - [https://tatsu-lab.github.io/alpaca\\_eval](https://tatsu-lab.github.io/alpaca_eval)

AlpacaEval  Leaderboard

An Automatic Evaluator for Instruction-following Language Models  
Length-controlled (LC) win rates alleviate length biases of GPT-4, but it may favor models finetuned on its outputs.

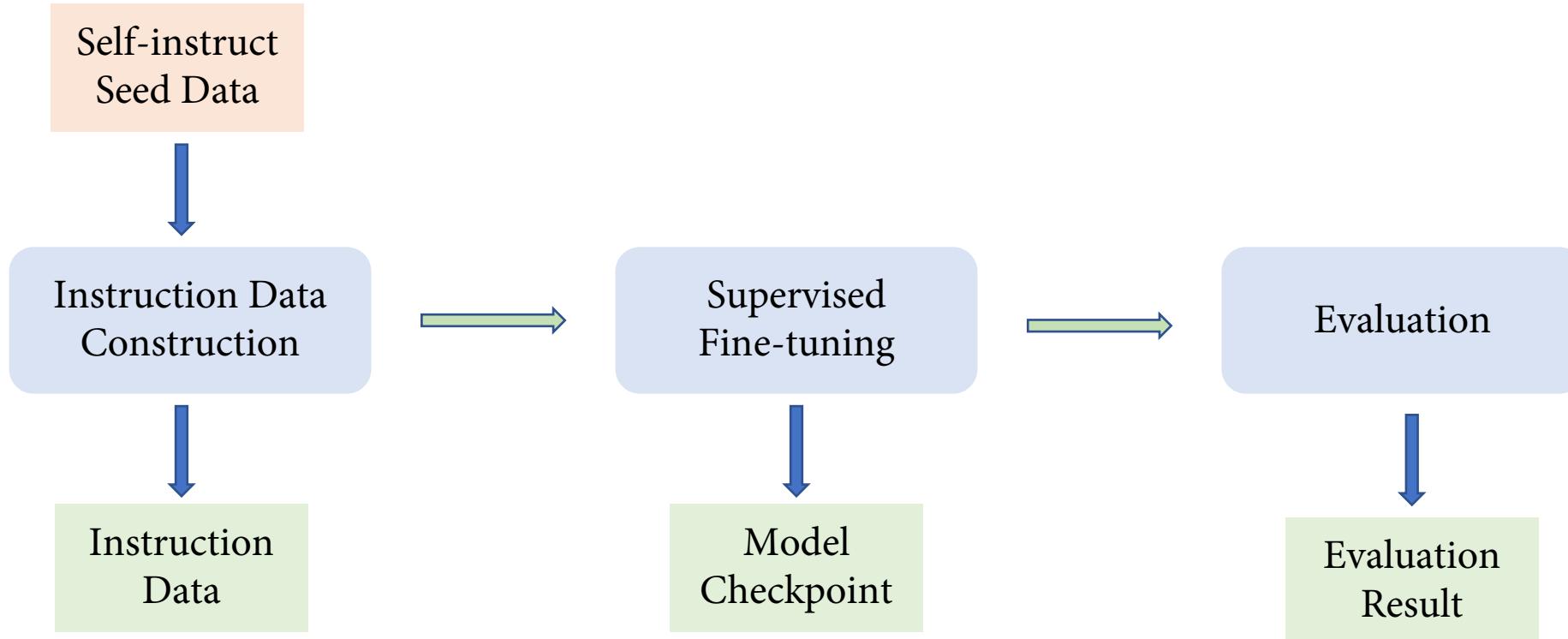
Version: AlpacaEval AlpacaEval 2.0 Filter: Community Verified

Baseline: GPT-4 Preview | Auto-annotator: GPT-4 Preview

| Model Name                  | LC Win Rate | Win Rate |
|-----------------------------|-------------|----------|
| GPT-4 Preview               | 50.0%       | 50.0%    |
| Aligner 2B+Claude 3 Opus    | 41.8%       | 34.5%    |
| Claude 3 Opus (02/29)       | 40.4%       | 29.0%    |
| GPT-4                       | 38.1%       | 23.6%    |
| Aligner 2B+Qwen1.5 72B Chat | 36.7%       | 31.8%    |
| Qwen1.5 72B Chat            | 36.6%       | 26.5%    |
| GPT-4 0314                  | 35.3%       | 22.1%    |
| Claude 3 Sonnet (02/29)     | 34.9%       | 25.6%    |
| Mistral Large (24/02)       | 32.7%       | 21.4%    |
| Stable Diffusion 1.4        | 21.5%       | 27.0%    |



# Recap: the whole pipeline





# Experimental Resources

## FREE GPUs!

- Register “ModelScope”, You can get about 100 GPU hours
- <https://www.modelscope.cn/my/mynotebook/preset>

## Notes:

- 打开实例后即**启动计时**, 因此请及时下载所需文件(**默认只保留ipynb file**), 并关闭实例 !
- Huggingface too slow? Use Modelscope instead.
  - <https://github.com/hiyouga/LLaMA-Factory?tab=readme-ov-file#use-modelscope-hub-optional>
- GPT API Calling : 建议在自己电脑上进行, 并开启VPN等, 确保能够连接OpenAI
- Also, you can generate the predictions first, and then perform evaluation on **CPU instance**.





# Experimental Resources

Seed Data:

[https://github.com/yizhongw/self-instruct/blob/main/data/seed\\_tasks.jsonl](https://github.com/yizhongw/self-instruct/blob/main/data/seed_tasks.jsonl)

Code    Blame    175 lines (175 loc) · 108 KB

Raw

```
1   {"id": "seed_task_0", "name": "breakfast_suggestion", "instruction": "Is there anything I can eat for a breakfast"},  
2   {"id": "seed_task_1", "name": "antonym_relation", "instruction": "What is the relation between the given pairs?"},  
3   {"id": "seed_task_2", "name": "one_sentence_description", "instruction": "Generate a one-sentence description for"},  
4   {"id": "seed_task_3", "name": "harmful_stereotype_example", "instruction": "Describe a situation in which the giv"},  
5   {"id": "seed_task_4", "name": "email_subject_generation", "instruction": "Generate an appropriate subjective titl"},  
6   {"id": "seed_task_5", "name": "interview_question_answering", "instruction": "How do you answer this question in"},  
7   {"id": "seed_task_6", "name": "new_year's_resolutions", "instruction": "Brainstorm a list of possible New Year's"},  
8   {"id": "seed_task_7", "name": "idiom_meaning", "instruction": "Explain the following idiom to me, and try to give"},  
9   {"id": "seed_task_8", "name": "english_haiku_generation", "instruction": "Generate a haiku using the following wo"},  
10  {"id": "seed_task_9", "name": "movie_recommendation", "instruction": "Recommend a movie for me to watch during th"},  
11  {"id": "seed_task_10", "name": "game_suggestion", "instruction": "Suggest some games that can be played by a grou"},  
12  {"id": "seed_task_11", "name": "grocery_list", "instruction": "Make a grocery list for a healthy meal.", "instanc"},  
13  {"id": "seed_task_12", "name": "explain_behavior", "instruction": "Explain human's behavior.", "instances": [{"in"},  
14  {"id": "seed_task_13", "name": "dependency_parsing", "instruction": "Dependency parsing for the sentence. Output"},  
15  {"id": "seed_task_14", "name": "question_answering_generation_from_facts", "instruction": "Using the facts given,"},  
16  {"id": "seed_task_15", "name": "analogy_completion", "instruction": "To make the pairs have the same analogy, wri"},  
17  {"id": "seed_task_16", "name": "medicine_suggestion", "instruction": "Given a description of the symptom, identif"},  
18  {"id": "seed_task_17", "name": "birthday_planning_checklist", "instruction": "Create a birthday planning checklis"},  
19  {"id": "seed_task_18", "name": "refuse_invitation_email", "instruction": "Reply to the email and refuse the invit"},  
20  {"id": "seed_task_19", "name": "explain_without-background", "instruction": "In your opinion, how would you expla"},  
21  {"id": "seed_task_20", "name": "horror_movie_opening", "instruction": "You need to write a creative opening scene"},  
22  {"id": "seed_task_21", "name": "fun_math_question", "instruction": "Create a fun math question for children.", "in}
```



# Experimental Resources

Code for fine-tuning: <https://github.com/hiyouga/LLaMA-Factory>

Fine-tune your model with just one click!

The screenshot shows the GitHub repository page for `https://github.com/hiyouga/LLaMA-Factory`. The page features a dark theme with a navigation bar at the top containing links for README, Code of conduct, Apache-2.0 license, and Security. Below the navigation is a "Table of Contents" section with a list of links to various project documentation pages. The main content area is titled "Features" and lists several bullet points about the repository's capabilities.

Table of Contents

- [Features](#)
- [Benchmark](#)
- [Changelog](#)
- [Supported Models](#)
- [Supported Training Approaches](#)
- [Provided Datasets](#)
- [Requirement](#)
- [Getting Started](#)
- [Projects using LLaMA Factory](#)
- [License](#)
- [Citation](#)
- [Acknowledgement](#)

Features

- Various models: LLaMA, Mistral, Mixtral-MoE, Qwen, Yi, Gemma, Baichuan, ChatGLM, Phi, etc.
- Integrated methods: (Continuous) pre-training, supervised fine-tuning, reward modeling, PPO and DPO.
- Scalable resources: 32-bit full-tuning, 16-bit freeze-tuning, 16-bit LoRA and 2/4/8-bit QLoRA via AQLM/AWQ/GPTQ/LLM.int8.
- Advanced algorithms: GaLore, DoRA, LongLoRA, LLaMA Pro, LoRA+, LoftQ and Agent tuning.
- Practical tricks: FlashAttention-2, Unslot, RoPE scaling, NEFTune and rsLoRA.
- Experiment monitors: LlamaBoard, TensorBoard, Wandb, MLflow, etc.
- Faster inference: OpenAI-style API, Gradio UI and CLI with vLLM worker.



# Experimental Resources

## Evaluation Codebase

Reference: [https://github.com/tatsu-lab/alpaca\\_eval](https://github.com/tatsu-lab/alpaca_eval)

AlpacaEval 🦙 Leaderboard

An Automatic Evaluator for Instruction-following Language Models  
Length-controlled (LC) win rates alleviate length biases of GPT-4, but it may favor models finetuned on its outputs.

Version: AlpacaEval AlpacaEval 2.0 Filter: Community Verified

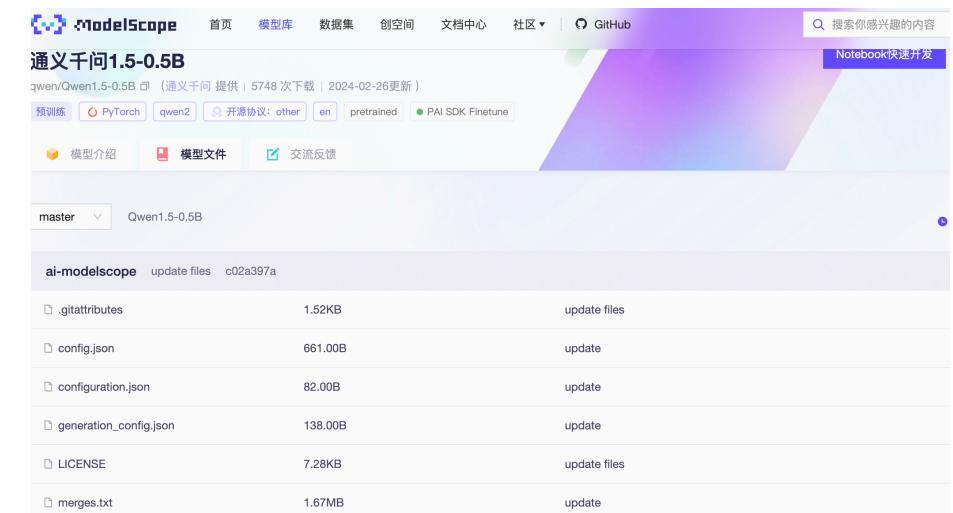
Baseline: GPT-4 Preview | Auto-annotator: GPT-4 Preview

| Model Name                    | LC Win Rate | Win Rate |
|-------------------------------|-------------|----------|
| GPT-4 Preview ↗               | 50.0%       | 50.0%    |
| Aligner 2B+Claude 3 Opus ↗    | 41.8%       | 34.5%    |
| Claude 3 Opus (02/29) ↗       | 40.4%       | 29.0%    |
| GPT-4 ↗                       | 38.1%       | 23.6%    |
| Aligner 2B+Qwen1.5 72B Chat ↗ | 36.7%       | 31.8%    |
| Qwen1.5 72B Chat ↗            | 36.6%       | 26.5%    |
| GPT-4 0314 ↗                  | 35.3%       | 22.1%    |
| Claude 3 Sonnet (02/29) ↗     | 34.9%       | 25.6%    |
| Mistral Large (24/02) ↗       | 32.7%       | 21.4%    |
| Santa Cruz 1.0.2 (last 640) ↗ | 31.5%       | 27.0%    |



# 作业提交

- Canvas提交zip文件，包含：
  - 一份英文实验报告，说明整个实验如何构建(页数<=5页)
    - 推荐Latex: <https://www.overleaf.com/project/5f64f1fb97c4c50001b60549>
    - 内容包含：
      - 数据集构建和处理
      - 微调的各项超参数设置、loss等指标的变化
      - 最终的alpaca eval 2.0的评测结果
  - 完整代码和简单README说明如何运行这个project
    - **模型预测输出: Model Predictions as Json file**
    - 如何生成Instruction data
    - 如何进行微调
    - .....
  - Optional
    - Github link
    - Downloadable Huggingface / ModelScope model checkpoint





# 作业提交

---

## 给分标准

- < 85 : 未完整完成 (1)符合格式规范的数据集、(2)进行指令微调模型、(3)最后在 benchmark上的测试结果。
- 85~90 : 完成(1) (2) (3)的整个pipeline，并提交了代码和report。
- 90~95 : 完成了(1) (2) (3)，有完整清晰的代码和内容充实的report。
- 95~100 : 在95分的基础上，在数据集构建、微调方面有创新的设计，有完整清晰的代码、checkpoint和report。
  - 自动筛选高质量数据
  - 更高效的LLM调用（鼓励使用更小的模型生成更好的数据）



# 作业提交

---

- 正常提交的截止日期：北京时间 **2024.04.25 23:59**
- 每个小组需要在Report中写明小组组员，最后由一位同学提交即可(若无特别说明分工，默认每个组员相同得分)

分组文档如下：

