

एमएमएलयू बेंचमार्क

यह पोस्ट [LLM \(मासिव मल्टीटास्क लैंग्वेज अंडरस्टैंडिंग\) बेंचमार्क](#) पर एक भाषा मॉडल का मूल्यांकन करती है।

[LLM](#) बेंचमार्क एक व्यापक परीक्षण है जो मॉडल की विभिन्न विषयों पर कार्य करने की क्षमता का मूल्यांकन करता है। इसमें गणित, इतिहास, कानून, और चिकित्सा जैसे विविध क्षेत्रों को कवर करने वाले बहुविकल्पीय प्रश्न शामिल हैं।

डेटासेट लिंक:

```

□ □□□□□□ □□□□ □□□□
□ □□□□□□□ □□□□ □□□□□□□□

import torch
from datasets import load_dataset
import requests
import json

# MMLU
subject = "abstract_algebra" #
dataset = load_dataset("cais/mmlu", subject, split="test")

#
def format_mmlu_prompt(example, few_shot_examples=5):
    prompt = " {} ( ) \n\n".format(subject.replace("_", " "))

    #
    few_shot_dataset = load_dataset("cais/mmlu", subject, split="validation")
    for i in range(few_shot_examples):
        ex = few_shot_dataset[i]
        prompt += f" : {ex['question']}\n"
        prompt += " : \nA. {} \nB. {} \nC. {} \nD. {} \n".format(*ex['choices'])
        prompt += f" : {ex['answer']}\n\n"

    #
    prompt += f" : {example['question']}\n"
    prompt += " : \nA. {} \nB. {} \nC. {} \nD. {} \n".format(*example['choices'])
    prompt += " : "

    return prompt

#
correct = 0
total = 0
```

```

for example in dataset:
    prompt = format_mmlu_prompt(example)

    # llama-server
    url = "http://localhost:8080/v1/chat/completions"
    headers = {"Content-Type": "application/json"}
    data = {
        "messages": [{"role": "user", "content": prompt}],
        "max_tokens": 5,
        "temperature": 0,
    }

    response = requests.post(url, headers=headers, data=json.dumps(data))

    if response.status_code == 200:
        output_text = response.json()["choices"][0]["message"]["content"]
        predicted_answer = output_text.strip()[0] if len(output_text.strip()) > 0 else ""
    else:
        predicted_answer = ""
        print(f" : {response.status_code} - {response.text}")

    #
    if predicted_answer.upper() == example["answer"]:
        correct += 1
    total += 1

#
accuracy = correct / total
print(f" : {subject}")
print(f" : {accuracy:.2%} ({correct}/{total})")

```