MMLU-Benchmark

Dieser Beitrag bewertet ein Sprachmodell anhand des MMLU (Massive Multitask Language Understanding) Benchmarks.

Der MMLU-Benchmark ist ein umfassender Test der Fähigkeit eines Modells, verschiedene Aufgaben in einer Vielzahl von Fachgebieten zu bewältigen. Er besteht aus Multiple-Choice-Fragen, die unterschiedliche Bereiche wie Mathematik, Geschichte, Recht und Medizin abdecken.

Dataset-Links:

- Papers with Code
- Hugging Face Datasets

```
import torch
from datasets import load_dataset
import requests
import json

# MMLU-Dataset laden
subject = "abstract_algebra"  # Wählen Sie Ihr Fach
dataset = load_dataset("cais/mmlu", subject, split="test")

# Prompt mit Few-Shot-Beispielen formatieren
def format_mmlu_prompt(example, few_shot_examples=5):
    prompt = "Die folgenden sind Multiple-Choice-Fragen (mit Antworten) über {}.\n\n".format(subject.replace('

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

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

# Evaluationsschleife
```

```
correct = 0
total = 0

for example in dataset:
    prompt = format_mmlu_prompt(example)

    # Anfrage an llama-server senden
    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"Fehler: {response.status_code} - {response.text}")

    # Mit der richtigen Antwort vergleichen
    if predicted_answer.upper() == example["answer"]:
        correct += 1
    total += 1

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