MMLU ベンチマーク

この記事では、言語モデルを MMLU（Massive Multitask Language Understanding）ベンチマークで評価します。

MMLU ベンチマークは、モデルがさまざまな科目にわたる多様なタスクを実行する能力を包括的にテストするものです。数学、歴史、法律、医学などの多岐にわたる分野の多肢選択問題で構成されています。

データセットリンク:

- Papers with Code
- Hugging Face Datasets

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

# MMLU データセットをロード
subject = "abstract_algebra"  # 科目を選択
dataset = load_dataset("cais/mmlu", subject, split="test")

# プロンプトをフォーマットし、few-shot の例を追加
def format_mmlu_prompt(example, few_shot_examples=5):
    prompt = " 以下は{}に関する多肢選択問題（解答付き）です。\n\n".format(subject.replace("_", " "))

    # few-shot の例を追加
    few_shot_dataset = load_dataset("cais/mmlu", subject, split="validation")
    for i in range(few_shot_examples):
        ex = few_shot_dataset[i]
        prompt += f" 質問: {ex['question']}\n"
        prompt += " 選択肢:\nA. {}\nB. {}\nC. {}\nD. {}\n".format(*ex['choices'])
        prompt += f" 答え: {ex['answer']}\n\n"

    # 現在の質問を追加
    prompt += f" 質問: {example['question']}\n"
    prompt += " 選択肢:\nA. {}\nB. {}\nC. {}\nD. {}\n".format(*example['choices'])
    prompt += " 答え:"
    return prompt

# 評価ループ
correct = 0
total = 0
```

```
for example in dataset:
    prompt = format_mmlu_prompt(example)

    # llama-server にリクエストを送信
    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" エラー: {response.status_code} - {response.text}")

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

# 精度を計算
accuracy = correct / total
print(f" 科目: {subject}")
print(f" 精度: {accuracy:.2%} ({correct}/{total})")
```