Benchmark MMLU

Ce billet évalue un modèle de langage sur le benchmark MMLU (Massive Multitask Language Understanding).

Le benchmark MMLU est un test complet de la capacité d'un modèle à effectuer diverses tâches dans un large éventail de sujets. Il consiste en des questions à choix multiples couvrant des domaines variés tels que les mathématiques, l'histoire, le droit et la médecine.

Liens des jeux de données :

- Papers with Code
- Hugging Face Datasets

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

# Charger le jeu de données MMLU
subject = "abstract_algebra"  # Choisissez votre sujet
dataset = load_dataset("cais/mmlu", subject, split="test")

# Formater le prompt avec des exemples few-shot
def format_mmlu_prompt(example, few_shot_examples=5):
    prompt = "Les questions suivantes sont des questions à choix multiples (avec réponses) sur {}.\n\n".format

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

    # Ajouter la question actuelle
    prompt += f"Question: {example['question']}\n"
    prompt += "Choix:\nA. {}\nB. {}\nC. {}\nD. {}\n".format(*example['choices'])
    prompt += "Réponse:"
    return prompt

# Boucle d'évaluation
correct = 0
```

```
total = 0

for example in dataset:
    prompt = format_mmlu_prompt(example)

    # Envoyer une requête au serveur llama
    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"Erreur: {response.status_code} - {response.text}")

    # Comparer avec la vérité terrain
    if predicted_answer.upper() == example["answer"]:
        correct += 1
    total += 1

# Calculer la précision
accuracy = correct / total
print(f"Sujet: {subject}")
print(f"Précision: {accuracy:.2%} ({correct}/{total})")
```