Kaggle - LLM Prompt Recovery竞赛银牌算法概览

竞赛概览：

竞赛要求参赛者利用算法恢复用于改写给定文本的LLM提示。竞赛使用1300多个原始文本及其经Gemma（Google新开放的大语言模型）改写的版本组成的数据集进行测试。评估标准基于sentence-t5-base模型生成的嵌入向量，通过“锐化余弦相似度”（Sharpened Cosine Similarity）计算得分。

所用算法：

1. 生成训练样本数据：
   1. 生成prompt：利用ChatGPT大量生成改写文本的prompt；
   2. 获取原始文本：从https://huggingface.co/datasets/Skylion007/openwebtext获取开源网络文本，过滤较长文本；
   3. 重写原始文本：使用步骤a与b的文本数据输入到谷歌开源LLM模型gemma-7b-it，生成新文本数据，用此方法大量生成训练样本。
2. Seq2Seq模型：
   1. 预处理：将生成样本的prompt转化为embedding向量化存储，以便加速训练；
   2. 构建训练pipeline：将原始文本和重写文本一起输入到deberta-v3-large模型，拼接两者的特征输出，与实际prompt的embedding（即步骤1的向量数据）进行相似度训练；
   3. 检索库：构建一个拥有大量prompt的检索库，以便推理检索；
   4. 推理：将步骤b训练完成的模型线上推理，预测得到prompt的embedding向量，从步骤c的检索库中寻找最相似的prompt文本，作为seq2seq模型的预测输出。
3. Phi2-微调模型：使用开源phi2微调模型推理预测，截取关键文本作为模型预测输出。
4. zero-shot的LLM模型：使用开源模型Mistral-7B-Instruct-v0.2，输入数条范例，直接使用相关指令让模型预测。对模型预测结果修剪，去除多余的符号或文本。
5. 集成预测：将三种模型的预测文本进行字符串拼接，作为最终的预测结果。

数据说明：

prompts\_df.csv：改写文本的prompt

train\_clean.parquet：训练样本数据

validation826.csv：验证集

Silver Medal Algorithm Overview for

**LLM Prompt Recovery** Competition

**Competition Overview:**

In recent years, the development of Large Language Models (LLMs) is becoming matured, making the text they generate increasingly difficult to distinguish from human writing. The competition required participants to develop a machine learning model capable of accurately detecting whether an essay was written by a student or an LLM. The competition dataset included essays written by students and articles generated by various LLMs. This competition was a typical binary classification problem, with the evaluation metric being AUC.

**Algorithm Descriptions:**

1. **Training Sample Generation**:
   1. **Prompts Creation**: Extensively generate rewriting prompts using ChatGPT.
   2. **Original Texts**: Source open web texts from Hugging Face, filtering longer texts.
   3. **Text Rewriting**: Input texts from steps above into Google's open LLM, gemma-7b-it, to create rewritten text data for training samples.
2. **Seq2Seq Model**:
   1. **Preprocessing**: Convert prompts from generated samples into embedding vectors for faster training.
   2. **Training Pipeline**: Input original and rewritten texts into deberta-v3-large model, concatenate feature outputs, and train on similarity with actual prompt embeddings.
   3. **Retrieval Database**: Create a large database of prompt embeddings for inference retrieval.
   4. **Inference**: Use the trained model for online inference to predict prompt embeddings and retrieve the most similar prompt text from the database.
3. **Phi2 Fine-Tuning Model**: Employ an open-source Phi2 fine-tuning model for prediction, focusing on key text segments.
4. **Zero-Shot LLM Model**: Use the open-source model Mistral-7B-Instruct-v0.2, inputting examples to predict directly.
5. **Ensemble Prediction**: Combine predictions from the three models by string concatenation for the final result.

**Data Files:**

* **prompts\_df.csv**: Prompts for rewritten texts.
* **train\_clean.parquet**: Training data samples.
* **validation826.csv**: Validation set.