



NLP TERM PROJECT



Group 3

唐文蔚、張世諭、李侗陞

任務介紹

LLM Classification Fine-tuning 是一項專注於預測用戶偏好的任務，其目的是判斷用戶更喜歡哪一個大型語言模型（LLM）的回應。

數據集由提示（prompt）及兩個模型生成的回應（A 和 B）組成，模型需要學習評估哪一個回應更佳。這一過程有助於改進聊天機器人與人類的互動方式，確保它們更能反映人類的偏好。



任務介紹

該任務與基於人類反饋的強化學習（RLHF）中的“獎勵模型”或“偏好模型”概念相似。研究表明，直接利用現有 LLM 預測偏好可能存在一定的局限性，例如：對先出現的回應產生偏袒，或偏向於冗長的回應等。

這些偏見影響了模型對人類真實偏好的準確捕捉，因此需要其他方法來克服這些挑戰。



資料集

train.csv

- id - A unique identifier for the row.
- model_[a/b] - The identity of model_[a/b]. Included in train.csv but not test.csv.
- prompt - The prompt that was given as an input (to both models).
- response_[a/b] - The response from model_[a/b] to the given prompt.
- winner_model_[a/b/tie] - Binary columns marking the judge's selection.
The ground truth target column.

資料集


test.csv

- id
- prompt
- response_[a/b]

sample_submission.csv

- id
- winner_model_[a/b/tie] - This is what is predicted from the test set.





現有模型比較

我們研究了LLM Classification Finetuning with CNN 和 LMSYS: KerasNLP Starter兩種模型，以下是兩者的比較：



BASE APPROACH

LMSYS : KERAS NLP STARTER

採用基於 Transformer 的方法，使用預訓練模型，例如 BERT、Multilingual E5 和 ROBERTA。

LLM CLASSIFICATION FINETUNING WITH CNN

採用基於 CNN 的架構，搭配文本向量化。

PRE-PROCESSING

LMSYS : KERAS NLP STARTER

使用分詞器（tokenizers）將文本轉換為 Token ID，並生成特定的注意力掩碼（attention masks），最後將 prompt 和 response 的 embedding vector 結合成為模型 Input

LLM CLASSIFICATION FINETUNING WITH CNN

使用 TextVectorization 層將文本轉換為整數序列，再拼接 prompt 和 response。

EMBEDDING APPROACH

LMSYS : KERAS NLP STARTER

- 使用預訓練的 Transformer 嵌入向量。

LLM CLASSIFICATION FINETUNING WITH CNN

- 固定詞彙表大小的學習嵌入層
(learned embedding layer)。

MODEL STRUCTURE

LMSYS : KERAS NLP STARTER

- 使用線性池化層(linear pooling layers)
- 最終使用 softmax layer 分類

LLM CLASSIFICATION FINETUNING WITH CNN

- 多層 1D convolution layers
- spatial dropout 和 max pooling
- global average pooling
- 最終使用Fully connected layers

TRAINING STRATEGY

LMSYS : KERAS NLP STARTER

- 單模型訓練
- Loss : CrossEntropy
- Optimizer : AdamW

LLM CLASSIFICATION FINETUNING WITH CNN

- 多模型訓練，採用不同隨機種子的集成方法 (ensemble approach)
- Loss : sparse_categorical_crossentropy
- Optimizer : Adam

COMBINATION

TRANSFORMER

使用第一個方法
Transformer 強大的文
本處理能力生成
Embedding Vector

AGGREGATION

結合第二個方法的集成
模型架構，使用不同的
Pre-Trained Model 進
行 soft voting

CNN LAYER

嘗試加入第二個方法的
CNN Layers 深入分析
文本特徵。

OTHER ATTEMPT

Shuffle Data：交換 response_a 跟 response_b 的 value，也就是將label互換，以避免 Positional Bias

使用不同模型：嘗試了 "bert-based-uncase", "roberta-base", "multilingual-e5-small", "deberta_v3_extra_small_en" 等 pre-trained 模型來比較準確率以挑選用來集成的模型。

PROS AND CONS

PROS

- 集成多個模型的數據，有助於減少誤差並提升結果的穩定性
- 複雜的模型架構能深入挖掘和分析文本特徵，提升預測精度。

CONS

- 複雜的模型架構可能導致 Overfitting，降低模型的泛化能力
- 如果單個模型的效能較差，集成模型可能無法見效
- 訓練耗時且需要大量運算資源

PROGRESS

COMPLETE

- shuffle data
- 將架構改為Torch以方便操作
- 嘗試多種pre-trained model

INCOMPLETE

- 集成模型，平均不同模型的輸出
- 嘗試使用不同的 model 架構
- ...

CURRENT SCORE

LLM Classification Finetuning

[Overview](#) [Data](#) [Code](#) [Models](#) [Discussion](#) [Leaderboard](#) [Rules](#) [Team](#) [Submissions](#)

53

Group 3



1.03812

THANK YOU

