NLP Term Project Report

組別：第3組 組員：唐文蔚、張世諭、李倧陞

一、任務介紹

本組的研究主題是 LLM Classification Fine-tuning，這是一個專注於預測用戶偏好的任務，這項分類微調任務旨在透過機器學習來理解並預測：當用戶面對兩個不同 LLM 產生的回應時，他們會更偏好哪一個。

本題目的數據集來自 Chatbot Arena，其中包含了用戶對話的提示（prompt）以及兩個 LLM 所生成的回應。用戶需要從中選擇他們偏好的回應。

會有這個任務的原因是當直接使用現有的 LLM 進行偏好預測時，存在幾個問題：

1. 位置偏差（Position Bias）：對先出現的回應產生偏好
2. 冗長偏差（Verbosity Bias）：冗長的回應會獲得不當的高評價
3. 自我增強偏差（Self-enhancement Bias）：模型過度推崇自己生成的回應

這項研究的重要性體現在：

* 有助於優化聊天機器人的交互方式
* 確保生成的回應更符合人類的期望與偏好
* 為建構preference model 和 reward model奠定基礎
* 與基於人類反饋的強化學習（RLHF）理念密切相關

二、探索性數據分析 (EDA)

(2.1) Analysis of Model Performance

模型獲勝次數的統計結果顯示出顯著的遞減趨勢。其中，GPT-4 及其變體的獲勝次數穩居榜首，顯示出絕對的領先優勢。緊隨其後的是 Claude 系列模型，獲勝次數約在 2000 至 2500 次之間。這樣的分布充分反映了當前市場上主流大型語言模型的實際表現 —— GPT 和 Claude 展現的優異性能使其成為目前最多人使用的chatbot。值得一提的是，兩者之間的差距仍然相當明顯，GPT-4 的領先地位尤為突出。

一張含有 文字, 螢幕擷取畫面, 圖表, 繪圖 的圖片

自動產生的描述

(2.2) Winner Distribution of Training Data

從Training Data中Model\_A 和 Model\_B 獲勝統計可以發現，資料集在分布上是均勻的，但仍然可能會有positional bias的問題，因此我們有嘗試用shuffle data的方式確保資料分布不會影響結果。

一張含有 文字, 螢幕擷取畫面, 圖表, 鮮豔 的圖片

自動產生的描述

三、資料集和資料分析

(3.1) Dataset

train.csv

* id：編號
* model\_[a/b]：模型名稱
* prompt：模型輸入
* response\_[a/b]：模型A、B的回應
* winner\_model\_[a/b/tie]：Ground truth (target column)

test.csv

* id：編號
* prompt：模型輸入
* response\_[a/b]：模型A、B的回應

sample\_submission.csv

* id：編號
* winner\_model\_[a/b/tie]：預測結果

(3.2) Data Analysis

根據Data中的資料，我們可以發現平手和獲勝的原因：

1. Winner\_tie：問題沒有絕對答案，例如主觀的問題或藝術相關的問題；不明確的問題 (prompt)；prompt中有錯字等。

2. Win：回答有豐富的資料、建議、和字數；能夠修正錯誤的prompt；給予明確的答案而不是答非所問。

四、參考notebook

(4.1) LMSYS: KerasNLP Starter

用KerasNLP的Shared Weight strategy配合fine-tuning基於 Transformer 的模型 (DebertaV3)，並使用mixed precision 以加速訓練。

一張含有 螢幕擷取畫面, 文字, 圖表, 字型 的圖片

自動產生的描述模型架構：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，並經過linear pooling layers和softmax layer 分類。

(4.2) LLM Classification Finetuning with CNN

用CNN的特性來分析文本的特徵，並用神經網路模型做分類。

模型架構：採用文本向量化作前處理，使用 TextVectorization 層將文本轉換為整數序列，再拼接prompt 和 response，再搭配基於 CNN 的架構做Classification，CNN架構包含：多層 1D convolution layers、spatial dropout 和 max pooling、global average pooling和MLP層。

特性：多模型訓練，採用不同隨機種子的集成方法，集合每個模型採集到的特徵。

(4.3) LLM Classification Finetuning | ML LightGBM

用TF-IDF來處理文本，再用LightGBM這種基礎的機器學習模型做分類。

特性：稀疏向量、機器學習模型

五、實驗結果

(5.1) Shuffle Data：雖然從EDA可以發現兩邊模型的勝率並沒有很大的差異，但為了確保模型不會受到positional bias，我們嘗試隨機將資料的label和winner互換，達到shuffle data的效果。

(5.2) 嘗試不同的Embedding 模型：我們比較了多個不同的Embedding模型的表現。在保持其他模型架構不變的前提下，實驗結果顯示參數量較大的模型能夠取得越好的結果。這一發現與當前NLP的研究趨勢相符 - 模型參數量的增加確實能帶來性能的提升。雖然因為Kaggle的平台限制，我們無法進一步驗證更大規模模型的效能，但現有實驗數據已足以支持「參數規模與模型效能呈正相關」這一結論。

|  |  |  |  |
| --- | --- | --- | --- |
| Model Name | 參數數量 | 輸出維度 | 結果 |
| deberta\_v3\_large\_en | 430M | 1024 | 1.01798 |
| deberta\_v3\_base\_en | 180M | 768 | 1.03812 |
| bert-base-uncased | 110M | 768 | 1.09840 |
| e5-base | 109M | 768 | 1.09690 |

(5.3) 使用Sparse Vector和不同的分類器

除了使用 sentence transformer，我們也嘗試了 TF-IDF 和 BM25 等 Sparse vector 的Encoding 方式，但結果比使用Dense Vector的效果差，原因應該是因為這個任務很注重語意，因此注重語意和上下文的embedding模型可以取得較好的效果；除了encoding的部分，我們也嘗試了不同的classifier架構，嘗試將LLM Classification Finetuning with CNN的CNN架構加入模型中，但結果並沒有改進，代表簡單的分類器就可以很好的處理embedding的輸出。

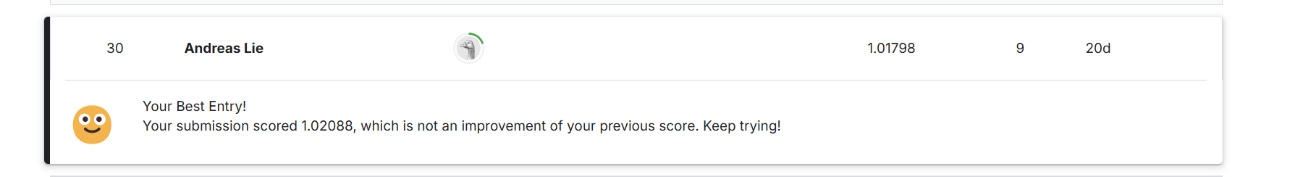
|  |  |  |
| --- | --- | --- |
| Model | | Score |
| Dense Vector | deberta\_v3\_large\_en + MLP | 1.01798 |
| Sparse Vector | TF-IDF + LightBGM | 1.05737 |

(5.4) Integrate model

我們嘗試透過 Hard Voting 的方法整合不同模型的結果，即通過對預測結果進行投票，期望藉由多模型的結合提升穩定性。然而，結果未達顯著改善，可能原因在於若單一模型表現欠佳，集成方法難以實現顯著效能提升。

六、結論

根據實驗結果，我們發現在目前使用的 encoder-classifier 架構中，透過 embedding 模型生成的 dense vector 表現優於透過 TF-IDF 或 BM25 生成的 sparse vector。而在選擇 embedding 模型時，參數更多且規模較大的模型能獲得更高的分數。另外，針對 classifier 的修改並未帶來顯著的成效，代表如果 encoder 已提供高品質的embedding，改進 classifier 的空間很有限。

我們小組在public score獲得了 1.01798 ，排行30。(deberta\_v3\_large\_en)

七、分工

|  |  |
| --- | --- |
| 唐文蔚 | EDA；資料分析；改進嵌入模型；稀疏向量比較；  CP1簡報製作；CP2 評分；CP4報告製作 |
| 張世諭 | EDA；改進嵌入模型；  CP2 評分；CP3 海報製作 |
| 李倧陞 | 改進嵌入模型（取得最高成績）；Shuffle Data；  CP2 評分 |

八、參考資料

<https://www.kaggle.com/code/addisonhoward/lmsys-kerasnlp-starter/notebook>

<https://www.kaggle.com/code/lonnieqin/llm-classification-finetuning-with-cnn>

[https://www.kaggle.com/code/gallo33henrique/llm-classification-finetuning-ml-lightgbm](https://www.kaggle.com/code/gallo33henrique/llm-classification-finetuning-ml-lightgbm#Feature-Engineering)