

Sentiment Analysis using Deep Learning technique

(使用深度學習技術實作情感分析)

姓名(陳宇杰) & 學號(111511281)

完成日期 (2025/11/26)

Introduction

本次作業的主要 **Objective** 是利用 Deep Learning techniques 針對 **Social Media Dataset** 進行 **Sentiment Analysis**。由於 Social media posts 中的文本經常包含非正式語法 (潮流語句)、Slang 以及各類 Noise, 如何提取 Semantic meaning 並分類為 Positive, Neutral 或 Negative 是一大挑戰。

為了達到最佳的 performance, 以及資源限制。這次作業會採用 Transfer Learning, 在 Model Architecture 的設計上, 使用了 **Microsoft/deberta-v3-large** 作為 **Backbone**, 利用 Disentangled Attention 機制來捕捉上下文資訊。

Implementation Details

● Data processing

在這一次的作業中, 為了減少資料的 noise (@user, URL, #hashtag), 因此我會使用 re library, 並定義 basic_clean function, 去清除 noise。可以有效減少對 Sentiment 無關的字符干擾。設定 padding="max_length" 與 truncation=True, 並將 max_length 設定為 90, 以覆蓋大部分 Social Media 句子的長度, 同時節省記憶體。然後使用 sklearn library 的 train_test_split 進行切割。設定 val_ratio=0.05, 並開啟 stratify 選項。Stratified sampling 確保了 Training set 與 Validation set 中的 Label distribution 一致, 避免因類別不平衡導致評估失準。

● Model architecture

1. **Backbone:** 使用 Deberta-V3-Large 作為本次作業的 backbone
2. **Pooling Strategy:** 採用 cls 進行 pooling, 取最後一層 Hidden State 的第一個 token 為整句的 Semantic Representation。
3. **Customs_Block:** 以 MLP 為主體, 首先為 linear layer 層, 進行 linear transform, 第二步進行 LayerNorm, 把每一筆資料在同一層裡做正規化, 讓輸出特徵有平均值 0、標準差 1, 有助於模型訓練穩定。然後加入 ReLU, 進行非線性轉換, 把負值設為零, 只保留正值。引入 dropout, 把非必要的神經元 delete, 防止 overfitting, 最後再做一次 linear transformation。其中, 在本次作業, 我設定 dropout=0.3, layer=2, hidden=128, 因 backbone 效能太好, 而且 dataset token 太小, 因此我不設定過大 layer 數量。防止 overfit, 從而降低 accuracy。

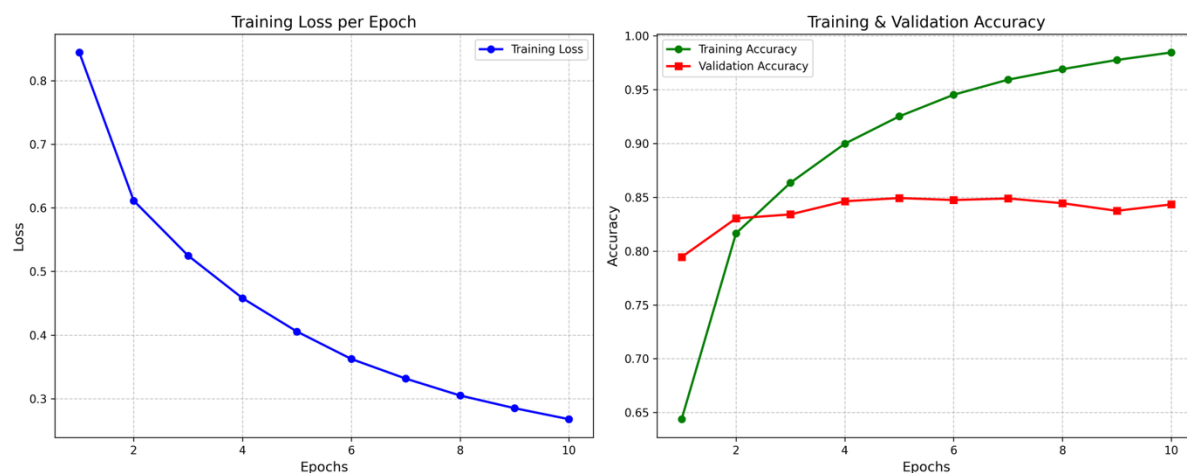
然後，在 customs block 的實作上，原本是使用 Cross entropy，但在觀察 confusion Matrix 後發現分類問題，因此選擇性使用 Focal Loss，盡量解決分類中性詞的任務（中性詞缺乏強烈情感關鍵詞，使分類出現問題）透過降低那些已經分類良好的簡單樣本權重 γ ，並將訓練重心聚焦於 **Hard Negatives**，模型被迫學習區分中性語氣與微弱情感之間的細微差異，進而學習到更具 **Robustness** 的 Decision Boundaries。此外，我們採用了 **Differential Learning Rate** 策略（針對 Backbone 與 Head 分配不同的學習率），這讓 pre-trained 的特徵能被溫和地 **Fine-tune**，同時讓客製化的 Head 能夠快速收斂。

● Training pipeline

使用 Focal Loss，強化模型對 Hard Samples 的學習能力。設定 Focusing parameter 與 Balancing parameter。這能降低簡單樣本的權重，迫使模型專注於那些預測機率較低（較難分類）的樣本。另外，採用 **Differential Learning Rate** 策略，並設定 warmup-ratio，在訓練初期逐步增加 LR 以穩定 Gradient。並且，為了加速訓練並減少記憶體佔用。使用 mixed precision training。

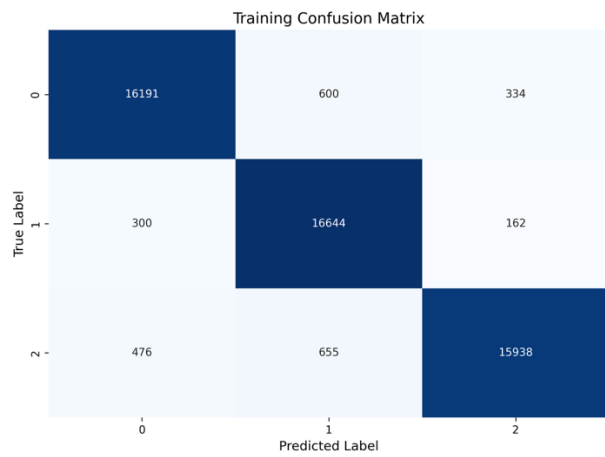
Experiment Result

Loss & Accuracy

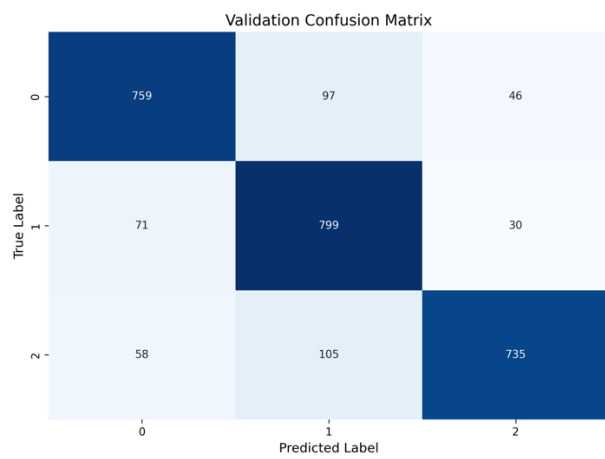


Confusion Matrix

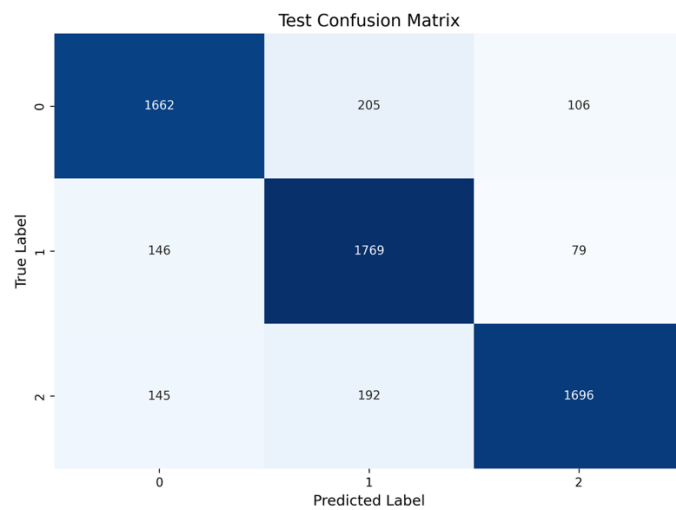
Train



validation



Test



Discussion

Analysis of Results

在這一次的訓練中，我把 epoch 調整為十

1. Training Dynamics & Convergence:

根據 Training History 圖表，模型的訓練過程展現了良好的收斂性。Loss 有一直收斂。從最初的 ~ 0.85 穩定下降至第 10 個 Epoch 的 ~ 0.27 。Validation Accuracy 在前 3 個 Epoch 快速爬升，隨後在第 5 至第 7 個 Epoch 左右，峰值約落在 85% 上下。雖然 Training Accuracy 最終接近 98%，與 Validation Accuracy 存在約 13% 落差，但 Validation Curve 並未出現劇烈下降，顯示模型雖有 Overfitting 的傾向，但有設定 dropout，使模型保持了穩定的泛化能力。

2. Confusion Matrix:

三個類別的正確預測數分別為 759 (Positive)、799 (Neutral) 與 735 (Negative)，顯示模型並無嚴重 Bias，這種 Balanced Prediction 的表現，主要歸功於 Stratified Split 成功確保了各類別資料分佈的一致性。然而，分析 Test Set 的錯誤分佈，發現主要的誤判集中在 Neutral (Class 1)，共有 205 筆 Positive 與 192 筆 Negative 樣本被錯誤歸類。這表明當資料缺乏強烈的 Keywords 時，模型會語氣隱晦的句子判斷為 Neutral。相對地，Positive 與 Negative 之間的直接互誤較少，例如 Test Set 中僅有 106 筆 Positive 被誤判為 Negative。這證實模型已能區分極端情感特徵，目前的挑戰主要在於判定情感的「強度」，而非情感的「方向」。

Observation

Focal Loss 有效引導模型，使 Neutral 類別在 Validation Set 中取得最高的 799 筆正確預測。雖然訓練集分類近乎完美，但 test_data 中，Class 0 與 Class 2 常被誤判為 Class 1 的情形，有可能是因為資料標記上，或語言表達上使用詞語的問題，使模型適別中性詞出現困難。但是，測試集與驗證集高度相似，證明了包含 Differential Learning Rate 與 Warmup 的策略使模型有較好的泛化能力。針對目前的誤判，理論上若能使用資料增強 (Data Augmentation)，應該能提升模型對模糊語意的辨識度與正確率。然而，受限於本次實作的規範限制，無法運用此類方法來進一步擴充訓練資料的多樣性。未來若能排除此限制並結合增強策略，模型效能應該會得到提升，達到更佳的分類效果。