1. Introduction/Objective (10%)

Objective

今天要做的是利用網友的留言，預測他的情緒，能夠將文本數據精確分類為三個類別：**Positive**、**Neutral** 和 **Negative**。Sentiment Analysis 是 NLP 中的基礎任務，在社群媒體監控、客戶反饋分析等領域有特殊應用。

本次實驗的具體目標如下：

1. 找出適合的模型
2. Fine-tune
3. 調整不同的參數
4. 清理訓練資料
5. 透過 Confusion Matrix 來分析模型，量化模型的好壞。
6. 在 500M parameters 下，希望模型 Accuracy 能達到 0.84 以上。

2. Implementation Details (30%)

Model Architecture

基於本篇論文[1]，讓我選擇正確率較高的模型--**Microsoft's DeBERTa-v3-large** 作為本任務的 Backbone，參數量為 **434.017283M**。

Architecture Overview: DeBERTa 透過以下兩項技術改進 BERT 和 RoBERTa：

i. **Disentangled Attention:**不同於 BERT 將 Position Embeddings 直接加到 Word Embeddings，DeBERTa 使用兩個向量來表示每個詞：一個代表內容，一個代表位置。Attention Weights 是基於內容和關係位置的 Decoupling 矩陣計算而得 [2]。

ii. **Enhanced Mask Decoder:** DeBERTa 在 Decoding layer 引入了 Absolute Positions 來預測 Masked tokens，幫助 Pre-training [3]。

iii. **Replaced Token Detection (RTD):** v3 版本特別採用了類似 ELECTRA 的 Pre-training 任務(RTD)來取代傳統的 MLM [1]，顯著提升了 Sample Efficiency 和效能。

Data Preprocessing

為了將資料集準備好以輸入 DeBERTa 模型，我設計以下預處理：

i. Tokenization: 使用與 microsoft/deberta-v3-large 的 DebertaV2TokenizerFast。Tokenizer 負責將原始文本轉換為 Input IDs 和 Attention Masks。

ii. **Sequence Length Management:** 設定 max_length 為 **128**。由於觀察 dataset.csv 裡留言不會有大量的詞彙，所以設置 128，同時加速訓練。

iii. **Label Encoding:** 將情感標籤 (Negative, Neutral, Positive) 映射為整數 ID (0, 1, 2) ，以符合 Cross-Entropy Loss function 的輸入要求。

iv. **Data Splitting:** 將 dataset.csv 的資料先分成 54000 筆給 training 使用(分成 9 成 train data、1 成 validation data) ，6000 給 test 進行測試。

Unique Configurations & Connectivity:

為了使 Backbone 適應 3-class classification：

i. **Pooling Layer:** 提取最後一層 Hidden Layer 的[CLS] token 作為句子的特徵。

ii. **Classifier Head:** 在 Encoder 上連接了一個客製化的 Linear Layer (Linear (hidden_size, 3))。

iii. **Dropout:** 在 Classifier 之前使用 **0.15** 的 Dropout rate 以防止 Overfitting。

What makes this model unique?

Differential Learning Rates 與 dropout rate，其他參數、Model 架構調整有點無能為力 (幾乎反效果)。考量到 DeBERTa Encoder 已經在大量資料庫上進行過 Pre-training，只需微幅調整，而 Classification Head 是隨機初始化的，需要較大幅度的訓練。

i. Encoder Learning Rate: 設定為 **1e-5**，保留 Pre-trained 好的語言特徵。

ii.Head Learning Rate: 設定為 **1e-4**，讓 Classifier 能快速學習針對這三個情感類別的 Decision Boundaries。

Training Pipeline

訓練過程透過以下 Hyperparameters 和技術進行嚴格控制：

i. **Optimizer: AdamW**，並包含 Weight Decay 處理。

ii. **Batch Size:8**。選擇 **Size=8** 而非 16，看似會導致梯度估計產生較大的雜訊，但也會讓模型跳出尖銳的局部最小值。

iii. **Epochs:4**。

iv. **Learning Rate Scheduler:** 使用帶有 **Warmup** 的 **Linear Scheduler**。

v. **Warmup Ratio:0.1**。這允許 Learning Rate 在前 10%的 Training Steps 中從 0 逐漸增加到目標值，有助於在訓練初期穩定 Gradients。

vi. **Seed:** 固定為 **42**。

vii. **Environment:** 訓練是在 Colab 的 NVIDIA A100/T4 GPU 環境下執行，使用 PyTorch 和 Hugging Face Accelerate/Transformers 框架。

3. Experiment Result (30%)

Loss and Accuracy History

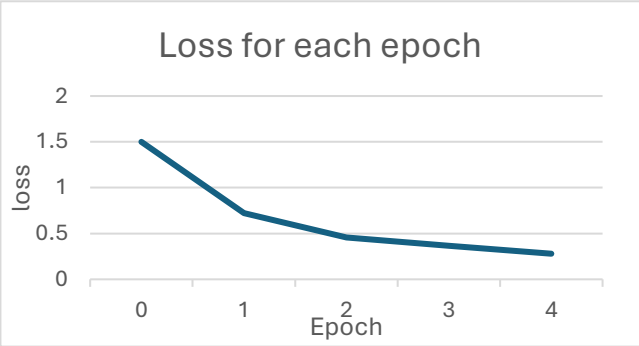


Fig. 1.

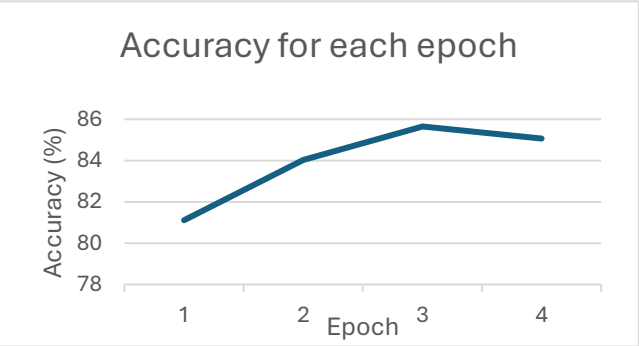


Fig. 2.

觀察 Fig. 1.，loss 在 **Epoch 0~1** 期間 Loss 有**劇烈且快速的下降**；到了 **Epoch 3**，下降速率顯著變緩；在 **Epoch 4**，Loss 開始**起伏不定(代表訓練到瓶頸)。觀察 Fig. 2.，在準確度方面，Epoch1~3 時逐漸上升，到 Epoch=4 時反而下降，代表出現過擬合現象。**

Confusion Matrix & Per-Class Performance

Confusion Matrix

		Actual		
		0	1	2
Predict	0	1684	167	122
	1	176	1713	105
	2	125	154	1754

Per-Class Performance

	precision	recall	f1-score	support
0	0.8484	0.8535	0.8509	1973
1	0.8422	0.8591	0.8505	1994
2	0.8854	0.8628	0.8739	2033
accuracy			0.8585	6000
macro avg	0.8587	0.8585	0.8585	6000
weighted avg	0.8589	0.8585	0.8586	6000

Accuracy

Accuracy	Value
Training	0.9390
Validation	0.8565
Testing	0.8585
Online test	0.8535

觀察 Confusion Matrix，能發現模型在 Label=0 (Negative)、1(Neutral)間最容易搞混，Label=2(Positive)正確率最高；觀察 Per-Class Performance，能發現 Label=2 的 recall、f1 score 都是最高，代表真的該類別準確率較高。觀察 Accuracy，能發現 Training 的準確率極高於其他三類，所以可以考慮剪枝，避免過擬合。

4. Discussion (30%)

Analysis of Results

本模型取得了 **Accuracy=0.8535** 的成果，達到目標 0.84。

Loss Analysis: Training Loss 展現了明顯的趨勢。我們觀察到在 **Epoch 0~1** 期間 Loss 有**劇烈且快速的下降**，顯示模型迅速適應了任務並學會了主要特徵。到了 **Epoch 3**，下降速率顯著變緩，進入 Fine-tuning 階段。在 **Epoch 4**，Loss 開始**起伏不定**，這暗示模型可能已達到高原期，這可以分析 Learning Rate 對於最後的收斂階段來說可能稍嫌過高。

DeBERTa-v3-large 模型展現了強大的效能。各類別的 **F1-score 差異極小(0.8509 vs0.8505 vs 0.8739)**，證明了模型**沒有嚴重的 Bias 問題**，即便在 Class 1 稍微遇到困難，整體表現依然穩健。

Differential Learning Rates 的影響: 將 Encoder 設為 1e-5、Head 設為 1e-4 是關鍵。Epoch 1 Loss 的驟降驗證 Head 較高的 Learning Rate 使 Classifier 能快速對齊，而 Encoder 較低的 Rate 保留 Pre-trained Knowledge。

Additional Observations

關於 Class 0、1 的模糊性: 數據顯示 Class 0/1 的 **Precision (0.8484/0.8422)** 顯著低於 Class 2(0.8854)。Class0 代表 Negative，有時候**反諷的句子較難察覺**；Class 1 代表 **Neutral**，**中性評論通常包含模稜兩可的語氣**，或者同時包含輕微的正負面詞彙，導致模型容易將其他類別誤判進來。相比之下，Class 2 Positive 特徵明顯，因此 Precision 最高。

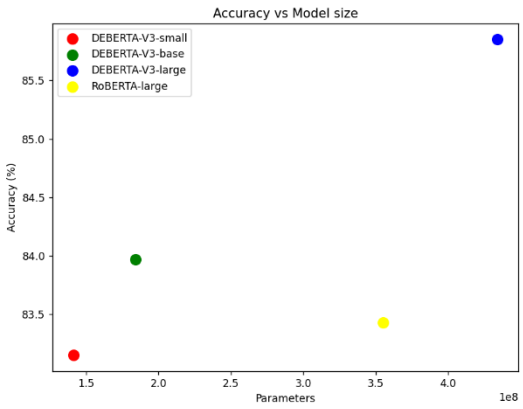
Training Stability: Epoch 4 觀察到的震盪提供了一個有價值的徵象。雖然模型達到了高準確率，但這種波動性表明，在**目前的 Learning Rate Schedule 下，上限是 4 Epochs**。

均衡的資料集: 從 Support (0: 1973, 1: 1994, 2: 2033) 可以看出資料集非常平衡。這解釋了為什麼 Macro Avg 和 Weighted Avg 的數值幾乎完全相同。在這種情況下，Accuracy 是個非常可信的指標

5. Extra (10%) (optional)

Model Comparisons:

除了 DEBERTA-v3-large，我也測試了 base、small，還有 RoBERTA-large。準確率與模型大小 比較如右圖，有明顯的差異。



Additional Experimentation:

除了更換模型，也可以利用許多方式來嘗試提升表現，雖然結果都不好，但更能體現 DEBERTA-V3-Large 的厲害。

i. **資料預處理**: (Using Small model)

為了減少資料中的冗詞，我寫了處理程序：將所有字母轉成小寫，並限制詞中連續字母的最大出現次數（例如 x=2，assshhhhh → asshh）。原意是希望詞更泛化，但效果不明顯：small 版 x=2 略好於原版 0.0001，但 large 版 x=2 反而輸了 0.1，顯示 DeBERTa-v3 模型本身能拆解字詞並強化語意理解。

x	Accuracy
No	0.8315
2	0.8316
1	0.8158

ii. **Freeze/Unfreeze 程序**: (Using Small model)

因資料集僅有 6 萬筆，我嘗試分批 freeze/unfreeze 訓練以充分更新各部分參數，但結果不理想，可能是因為缺乏參數間的連動關係。

Freeze	Accuracy
No	0.83
Yes	0.81

iii. **利用隨機刪除字詞，創造訓練測資:** (Using Large model)

我想說這樣會增加測資，所以利用這個方法把 6 萬筆變成 30 萬筆，但最後準確率超低，這跟過擬合有異曲同工。

Replicate	Accuracy
No	0.85
Yes	0.81

Reference

[1] Pengcheng He, Jianfeng Gao, Weizhu Chen, “ DEBERTAV3: IMPROVING DEBERTA USING ELECTRA-STYLE PRE-TRAINING WITH GRADIENTDISENTANGLED EMBEDDING SHARING,” ICLR 2023, Mar 2023

[2] Ritvik Rastogi, “Papers Explained 08: DeBERTa,” Feb 2023

[3] Pengcheng He, Xiaodong Liu, Jianfeng Gao,Wei Chen, “DeBERTa: Decoding-Enhanced BERT with Disentangled Attention,” May 2021