電子專題第一組期末報告

題目：基於 BERT 的社群惡意言論偵測與管理系統

指導教授：蔡宗漢

組長：110501521 電機 3A 徐松廷

組員：110501007 電機 3A 黃宇晟、110501506 電機 3A 薛善庭

一、 分工表

徐松廷

1. 觀看李弘毅教授 Machine learning 影片(圖像辨識卷積神經網路與 training tips)，統整重點後和組員分享。
   https://www.youtube.com/@HungyiLeeNTU
2. 跑圖像辨識(分辨快樂與悲傷的照片)和組員分享。
   https://drive.google.com/file/d/1tYCn6QLFyMVERXtP3DOsHoAClU2mwfQ0/view?usp=sharing
3. 觀看陳縕儂教授自然語言處理的教學(RNN 循環式神經網路、LSTM 長短記憶體單元、Transformer、Bert 等)，並向組員解說。
   https://www.youtube.com/@VivianMiuLab/videos
4. 在 Kaggle 上尋找文字辨識相關的資料集，使用 LSTM 模型訓練 model (Accuracy 接近 90%，但 Precision、Recall 和 F1 score 都只有 70%)，看懂 code 後和組員解說。
   https://www.kaggle.com/code/sungting/2023-9-4-toxicity-recognition
5. 決定使用 Bert model 提升效能，對 BERT model 做指標驗證(true negative rate)以及資料視覺化，Accuracy 接近 90%，Precision、Recall 和 F1 score 有 80%。
6. 決定加入 wordnet 的同義詞替換技術，擴增數據集並使類別平衡，成功使 Precision、Recall 和 F1 score 來到 85%以上。
   https://www.kaggle.com/code/sungting/12-7-bert-model-with-wordnet-merged-ver
7. 建構使用者介面的 prototype(使用 Dyjango HTML 與 Flask)。

黃宇晟

1. 觀看李弘毅教授 Machine learning 影片(圖像辨識卷積神經網路與 training tips 的部分)，並統整重點後和組員分享。
2. 嘗試圖像辨識模型(分辨快樂與悲傷的照片)後和組員分享。
3. 觀看陳縕儂教授自然語言處理方面相關的教學(RNN 循環式神經網路、LSTM 長短記憶體單元、Transformer、BERT 等)。
4. 研讀論文 *Text Analysis and Recognition of Emotional Content Using Deep Learning Methods and BERT*，並確認使用 BERT model 的優點。
   https://ieeexplore.ieee.org/document/10210232
5. 在 Kaggle 上尋找文字辨識相關的資料集，使用 LSTM 模型訓練 model，調整參數以優化結果。
6. 決定使用 BERT model 提升效能，對 BERT model 做指標驗證(Hamming loss)以及資料前處理。
7. 分析指標結果的意義與原因，並嘗試平衡數據集。

薛善庭

1. 觀看李弘毅教授 Machine learning 影片(圖像辨識卷積神經網路與 training tips 的部分)，並統整重點後和組員分享
2. 嘗試圖像辨識模型(分辨快樂與悲傷的照片)後和組員分享
3. 觀看陳縕儂教授自然語言處理方面相關的教學(RNN 循環式神經網路、LSTM 長短記憶體單元、Transformer、BERT 等)
4. 研讀論文 *How to Fine-Tune BERT for Text Classification?*並蒐集解決資料不平衡問題的方法
   https://link.springer.com/chapter/10.1007/978-3-030-32381-3_16
5. 在 Kaggle 上尋找文字辨識相關的資料集，使用 LSTM 模型訓練 model，調整參數以優化結果
6. 決定使用 BERT model 提升效能，對 BERT model 做指標驗證(ROC、AUC)以及調整 Train model 的參數(epoch、batch size、learning rate)
7. 分析指標結果的意義與原因，並嘗試平衡數據集

二、 教授與助教的建議紀錄

10 月

- ✓ **教授指導建議**：因為我們是做和文字辨識相關的自然語言處理，所以可以多上網查找和文字辨識相關的論文別人大概都是如何執行，除了了解更細部的技術也可以理解別人的進展以及我們能解決的問題。
- ✓ **學長建議**：有初步的成果相當不錯，建議我們可以閱讀更多論文，來了解別人如何提升 BERT model 的 performance。
- ✓ **結果**：我們在 11 月閱讀了兩篇和 BERT model 相關的論文(*How to Fine-Tune BERT for Text Classification*、*Text Analysis and Recognition of Emotional Content Using Deep Learning Methods and BERT*)，並且瞭解了多任務訓練、文本截斷、逐層遞減學習率、降低學習率以克服災難性遺忘等技巧。

11 月

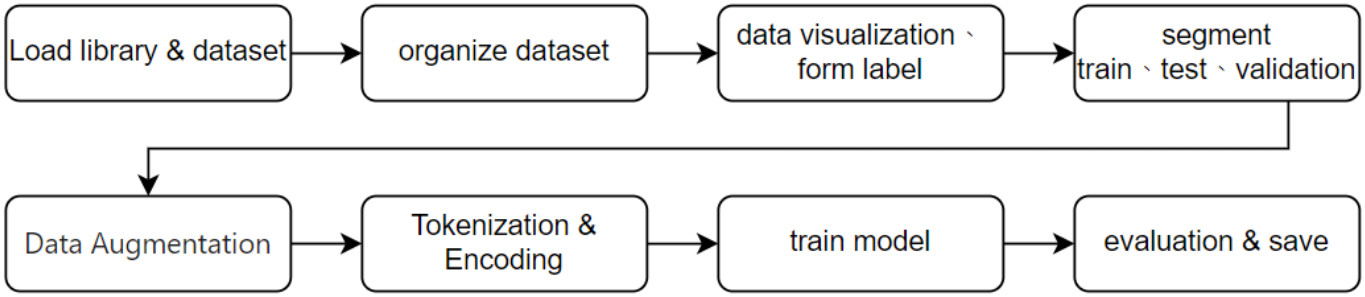
- ✓ **教授建議**：可以把我們的進展與給我們的建議記錄下來，這樣清楚明瞭，也能給出更合適的建議。
- ✓ **學長建議**：可以閱讀更多論文或是參考別人的 code，來了解別人 fine-tuning bert model 的技巧。
- ✓ **結果**：完成，感謝教授與學長的建議。

12 月

- ✓ **教授建議**：IEEE 的論文相當重要，是寶貴的學術資源，要好好珍惜。另外也可以多看看別人相關的論文怎麼呈現，讓你們的作品增加說服力。
- ✓ **學長建議**：寒假時可以嘗試用 warm-up 的技巧，讓我們的 Model 有更好的表現。
- ✓ **結果**：在寒假時我們會專注在使用 warm-up 等技巧提升 model 的表現，並且完成我們的使用者介面。

☆★感謝教授與學長姊這學期的幫助!!!!☆★

三、 流程圖



四、 專題的功能和目標

在 Facebook、YouTube 等社群媒體的言論管理環境中，留言區的觀眾與內容創作者常常無法了解中心化平台(例如 meta、google 等)的管理政策與規則，有時留言的觀眾甚至不知道自己的留言為何會被刪除或是被禁止，所以我們的專題致力於使用網頁與深度學習的技術來解決這樣的問題。我們利用 fine-tuned BERT model 自動為每一則觀眾的留言上標籤，來判定是否有攻擊性、毒性、威脅性、恐嚇、性騷擾意味的存在，有了自動判定留言毒性的功能後，影片或貼文的創作者可以自主決定什麼性質的留言需要在自己的留言區被屏蔽，也可以封鎖特定不良紀錄留言紀錄的觀眾，而留言的觀眾則可以知道每個留言區的規定是什麼，以及如果有被屏蔽或是停權的狀況也可以知道原因，我們的專題利用網頁與深度學習技術，建構出一個完整的系統，讓創作者對於自己的留言區有自主管理性、讓留言審核機制在 Bert model 的幫助下公開透明。

整體而言，我們是利用 BERT model tuning 的技術以及網頁與資料庫的技術建構出一個社群媒體言論管理工具的完整網頁系統。

五、 選用 BERT 的原因

1. 論文探討-*Text Analysis and Recognition of Emotional Content Using Deep Learning Methods and BERT*

   這篇論文主要比較了各個 NLP 模型(包含 CNN、LSTM、Bi-LSTM、BERT)在針對各個數據集進行情感分析時的表現，並且以 F1 分數做為評估方式，其原因是為了減低數據類別不平衡時對評估結果產生的誤差。

$$F1\ score = \frac{1}{\frac{1}{2}\left(\frac{1}{recall} + \frac{1}{precision}\right)}$$

Dataset	*CNN*	*LSTM*	*BI-LSTM*	*BI-LSTM ATTENTION*	*BERT*
Twitter Dataset	0.65	0.72	0.75	0.83	0.86
ISEAR	0.46	0.50	0.51	0.61	0.63
CBET	0.47	0.51	0.48	0.55	0.56
GoEMO	0.42	0.52	0.49	0.56	0.60
AMAN	0.71	0.77	0.80	0.82	0.84
Semeval	0.54	0.49	0.50	0.59	0.64

從上圖可看出 BERT 在各個模型間的整體性能上表現最佳，在 Twitter 數據集上獲得了最高的 0.86 的 F1 分數，也因此最終我們選擇 BERT 作為我們應用於辨識留言攻擊性辨識工作的模型。

2. 論文探討-*How to Fine-Tune BERT for Text Classification?*

此論文嘗試許多微調 BERT 模型預訓練的方式，文本截斷、逐層遞減學習率、降低學習率以克服災難性遺忘，以及更進一步的 In-Domain 和 Cross-Domain 預訓練、多任務微調。此論文中運用八個擁有不同特色的資料集進行實驗。如下圖所示，結果顯示了幾乎所有模型在經過 In-Domain、Cross-Domain 預訓練後，表現接有提升。此種微調方式可以幫助不同目標的模型理解廣泛的語意和結構，也能幫助相同目標的模型適應特定領域的語言和內容，提供我們資料集分布不均問題的解決方法。目前預計在下學期開始嘗試。

Domain	sentiment			question		topic	
Dataset	IMDb	Yelp P.	Yelp F.	TREC	Yah. A.	AG's News	DBPedia
IMDb	**4.37**	2.18	29.60	2.60	22.39	5.24	0.68
Yelp P.	5.24	1.92	29.37	2.00	22.38	5.14	**0.65**
Yelp F.	5.18	1.94	29.42	2.40	22.33	5.43	**0.65**
all sentiment	4.88	**1.87**	29.25	3.00	22.35	5.34	0.67
TREC	5.65	2.09	29.35	3.20	22.17	5.12	0.66
Yah. A.	5.52	2.08	29.31	**1.80**	22.38	5.16	0.67
all question	5.68	2.14	29.52	2.20	**21.86**	5.21	0.68
AG's News	5.97	2.15	29.38	2.00	22.32	**4.80**	0.68
DBPedia	5.80	2.13	29.47	2.60	22.30	5.13	0.68
all topic	5.85	2.20	29.68	2.60	22.28	4.88	**0.65**
all	5.18	1.97	**29.20**	2.80	21.94	5.08	0.67
w/o pretrain	5.40	2.28	30.06	2.80	22.42	5.25	0.71

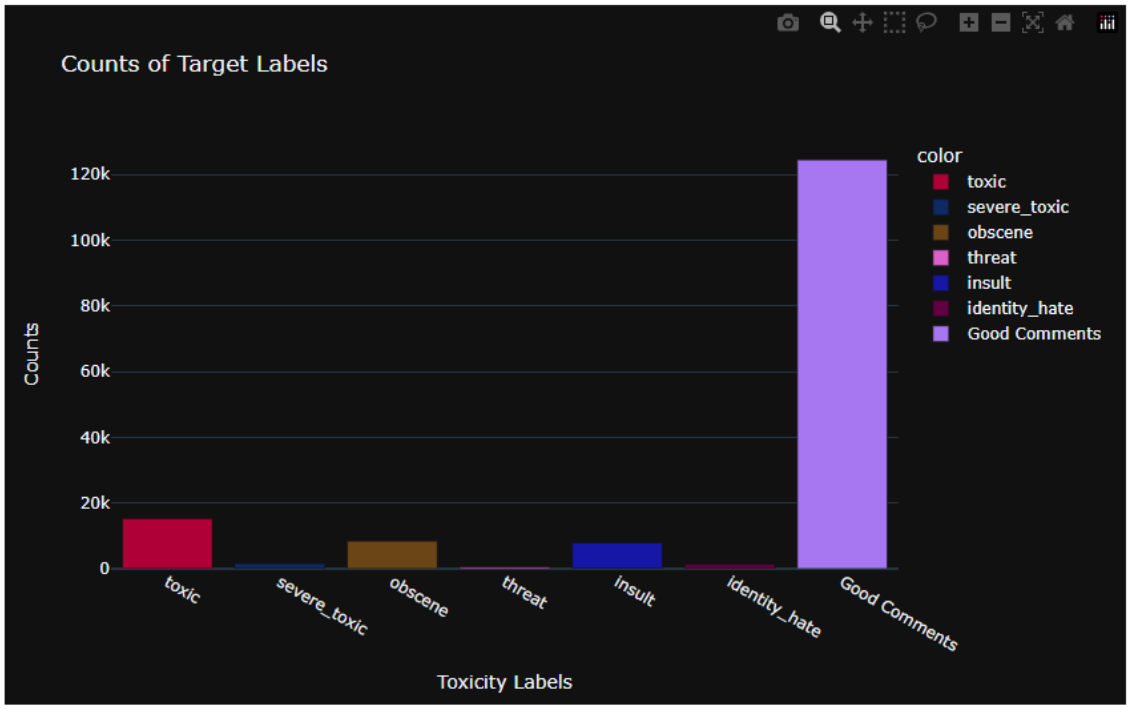
六、 Dataset 介紹與資料前處理

1. **Dataset**：Jigsaw Toxic Comment Classification Challenge

   是由 Jigsaw 所提供的資料集。這個資料集的目標是幫助建立模型來辨識網路上的惡意、具攻擊性或令人不安的評論。這些評論可能包含種族歧視、仇恨言論、性別歧視、威脅性言論等。這個 dataset 的目的是希望幫助建立一個能夠自動識別和過濾出這類評論的模型，以維護網路上的安全和友善環境。

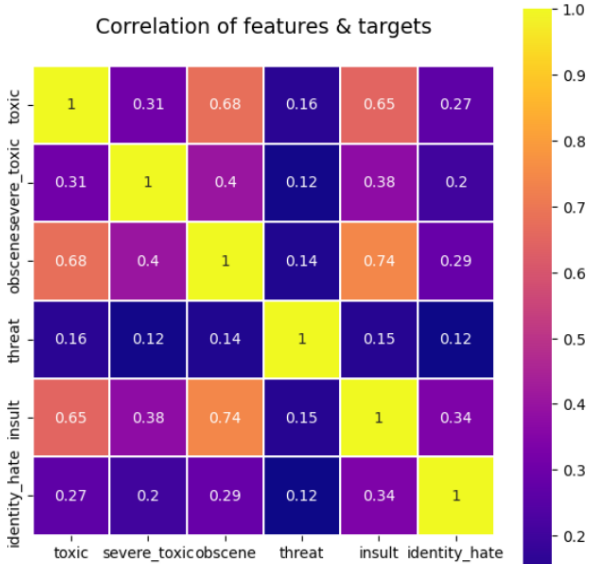
2. **Dataset category**

   我們的資料集的屬性分布如下，可以看出 good comment 的資料佔大多數，約為整體的 75%(分布嚴重不均!!)，可能會導致模型偏向預測 good comments 來取得高的 accuracy，因此要特別注意用於評估模型的指標，應使用 F1-score、Hamming loss 等較適合評估數據不平衡的指標。



3. **Training Dataset visualization**：經由 word cloud 統計出每個類別最常出現的單字

4. **Correlation of the category**：Dataset 中，每個類別之間的關聯性，即兩者同時出現的機率。



5. **Data preprocessing**
   I. 去除數字、HTML tag、特殊字元
   II. 把超過兩格的空格與特殊字符換成一格空格

III. 把換行變成一個空格

IV. 全部轉成小寫

V. Wordnet

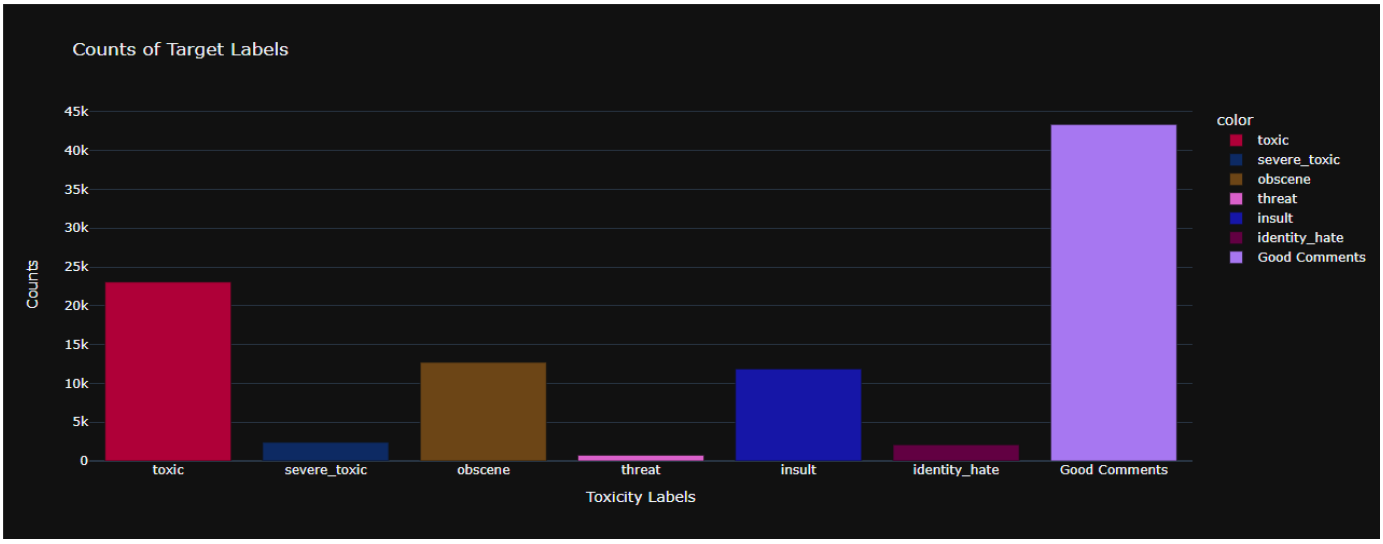
toxic	15294
severe_toxic	1595
obscene	8449
threat	478
insult	7877
identity_hate	1405

經過 wordnet 前的標籤數量

toxic	23057
severe_toxic	2404
obscene	12709
threat	739
insult	11863
identity_hate	2093

經過 wordnet 後的標籤數量

經過 wordnet 後，具有毒性標籤的數量都有所提升，約增加至原本的 1.5 倍，而 good comments 的部分也減少為整體的 45%，也稍微降低了我們使用的數據集中數據不平衡的問題，然而 threat 的部分因為本來數據數量就不多，經過增加後仍然占少數，預測結果可能還是會因此受影響。




```python
# 停用词列表，可以根据需要进行扩展
stopwords = ["the", "and", "is", "on", "in", "if", "for", "a", "an", "of", "or", "to", "it", "you", "your"]

def clean_text(text):
    # Remove HTML tags
    text = re.sub(r'<[^>]+>', '', text)

    # Remove web links
    text = re.sub(r'http\S+|www\S+|https\S+', '', text)

    # Remove special characters, punctuation marks, and newlines
    text = re.sub(r'[^a-zA-Z\s]', '', text)

    # Remove extra white spaces
    text = re.sub(r'\s+', ' ', text)

    # Remove stopwords
    text = ' '.join(word for word in text.split() if word.lower() not in stopwords)

    return text.lower()

# Example usage with the provided text
texts = [
    "\nMore\nI can't make any       real sugges%%%%tions on improvement - I wondered if the section statistics sho
]

cleaned_texts = [clean_text(text) for text in texts]
print(cleaned_texts)
```

七、 Model training process

1. **Loading the bert-uncased model**：匯入所用模型 BERT

```
# Token Initialization
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased', do_lower_case=True)

# Model Initialization
model = BertForSequenceClassification.from_pretrained('bert-base-uncased', num_labels=6)
```

2. **The setting of hyperparameter**
   I. Learning rate=0.0002

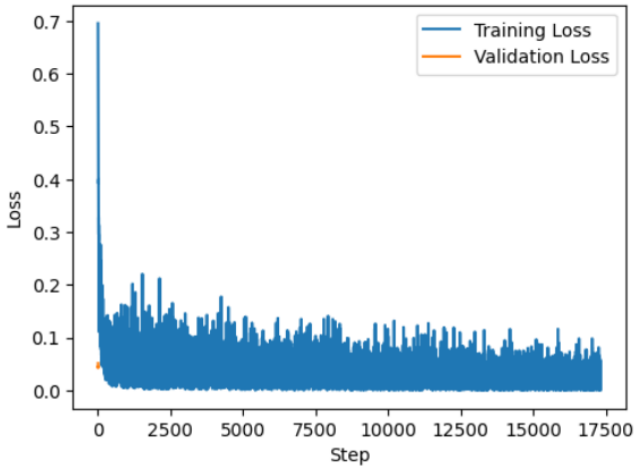
```
# Optimizer setup
optimizer = AdamW(model.parameters(), lr=0.00002)
```

   II. Batch size=32

```
Batch Size : 32
Each Input ids shape : torch.Size([32, 128])
```

   III. Epoch=5(4280 steps for each epoch)
   因為 validation loss 大概在 epoch=4 之後就沒有再下降了，故 epoch 設為 5。

3. **The change of Training loss during the training process**
   可以由圖中得到，當 step 越來越多，Loss 會有明顯的下降，直到最後小於 0.1。



八、 Performance analysis

測試結果：

```
Accuracy: 0.8949
Precision: 0.8579
Recall: 0.8976
True Negative Rate: 0.8976
Hamming Loss: 0.0225
```

I. Accuracy
定義如下

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Samples}}$$

我們的 accuracy 落在 89.5%，代表我們正確區分六種流言毒性的能力相當卓越

II. Precision
定義如下

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

我們的 accuracy 落在 85.8%，代表我們判定有毒性的留言中，有 89.5%是真正有毒性的留言，表現卓越。

III. Recall

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

我們的 accuracy 落在 89.76%，代表我們在所有有毒性的留言中，我們正確區分出了 89.76%的樣本，表現卓越。

IV. Hamming loss

Hamming loss 是用來評估多標籤分類問題中模型預測錯誤的指標，衡量的是模型在預測多標籤分類中的每個標籤時的錯誤率。對於每個樣本，Hamming loss 會計算預測的標籤和實際的標籤之間不同的比特數（二元分類的錯誤）除以所有標籤的總數，計算模型在預測每個標籤時錯誤的平均比例，計算方式如下

$$\text{Hamming Loss} = \frac{1}{N}\sum_{i=1}^{N}\frac{XOR(\text{True Labels}_i, \text{Predicted Labels}_i)}{L}$$

Hamming loss 的數值範圍在 0 到 1 之間，代表著模型的錯誤率，Hamming loss 越低表示模型在多標籤分類問題上的表現越好，而我們的 Hamming loss 為 0.0225，表示在我們這個多標籤分類問題中，模型表現卓越

V. True negative rate

$$TNR = \frac{TN}{Actual\ Negative} = \frac{TN}{TN + FP}$$

我們的 True nagative rate 落在 89.76%，代表在所有不具有毒性的留言中，我們正確將 89.76%的留言判定為沒有毒性。

VI. ROC 曲線下面積(ROC-AUC)

ROC 曲線下面積（ROC-AUC）是衡量二元分類模型性能的指標之一，它表示的是 Receiver Operating Characteristic（ROC）曲線下的面積。ROC 曲線是以真正類率（True Positive Rate，也稱為召回率）為縱軸，假正類率（False Positive Rate）為橫軸所繪製的曲線，用來衡量二元分類器在不同閾值下的性能表現。

ROC 曲線下的面積（AUC）表示模型在所有可能的分類閾值下對於正樣本和負樣本的分類能力，AUC 的值介於 0 和 1 之間，越接近 1 表示模型的性能越好，即模型在不同閾值下能更好地區分正負樣本。ROC 曲線下面積（AUC）的物理意義可以解釋為：

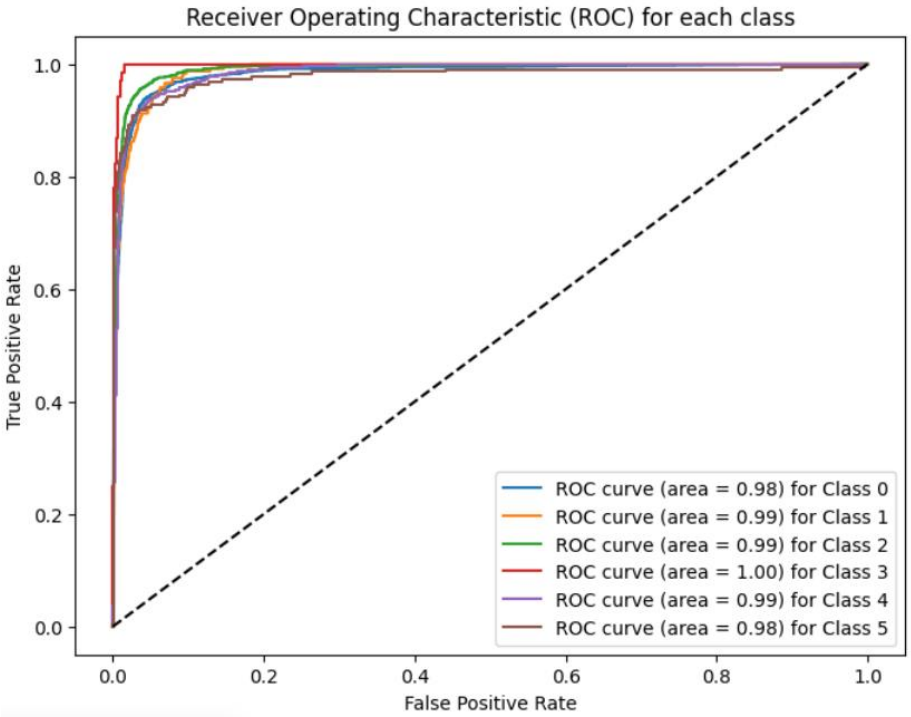
- ROC-AUC = 1：表示模型完美區分正負樣本，沒有誤分類。
- ROC-AUC = 0.5：表示模型的分類能力等同於隨機猜測，也就是說，模型無法區分正負樣本。
- ROC-AUC < 0.5：表示模型分類能力比隨機猜測還差，預測效果比隨機猜測還糟糕。

ROC-AUC 的計算方法一般使用積分曲線下的面積，ROC 曲線是一條連續曲線，可以使用數值積分的方法來計算:

$$\text{AUC} = \int_0^1 \text{TPR}(fpr)\, dfpr$$

我們的 ROC 曲線下面積如下：

Label 0 至 label 5 分別表示(toxic, severe toxic, obscene, threat, insult, identity hate)的六個類別，我們六個類別的 ROC 曲線下面積分別為 ROC-AUC(toxic, severe toxic, obscene, threat, insult, identity hate)=[0.98,0.99,0.99,1.00,0.99,0.98]，表示我們在六個類別區分正負樣本的能力都相當卓越。



九、 未來規劃

目前模型架構已有大致雛形，雖然眾多指數(第八點所述)表現皆良好並穩定，我們仍然想對資料分布不均的問題多做補強。故未來預定計畫如下：

1. **多任務**
   我們將嘗試使用第五點第 2 小點的論文 *How to Fine-Tune BERT for Text Classification?* 中的 In-Domain、Cross-Domain 預訓練及多任務微調，對我們當前的模型增加同為情感分析或用於其他目標的資料集，在預訓練時改善 good comment 所占比例。
2. **使用 Wordnet 的函式**
   預計將深入了解 Wordnet，使用更多函式做文字增強。
3. **Warm up 技術**
   此為學長建議。因為我們的目標為情感分析，故可以了解看看像 Chatgpt 使用的 Warm up 技術，或許可以增強模型特性。
4. **使用者介面**
   如同一開始我們立定的目標，我們接下來將架設一個簡單的使用者介面，內容預計包含留言功能、創作者管理介面、使用者登入、留言統計以及封鎖使用者功能。這將會是與訓練模型完全不同的領域，這部分預計也會需要花時間琢磨。