**DM LAB02 report**

**Student information**

* Name: 潘怡璇
* Student ID: 112034575
* GitHub ID: c24668343
* Kaggle name: yeexuan pan
* Kaggle private scoreboard snapshot

**Overview**

My project implements a BERT-based text classification pipeline to predict emotions from textual data. The emotions include eight classes: anger, anticipation, disgust, fear, joy, sadness, surprise, and trust. I will explain these process I adopt involves data preprocessing, feature engineering, model training, evaluation, and prediction.

1. **Preprocessing Steps**
   1. **Data Loading**

First, I load the JSON file into a dataframe and export it to CSV files. Then, I load the training (train.csv) and test (test.csv) datasets. Each dataset contains a column for textual input and its corresponding emotion label.一張含有 文字, 螢幕擷取畫面, 軟體, 多媒體軟體 的圖片

自動產生的描述

* 1. **Label Encoding**

The categorical emotion labels in the training data are converted into numerical values using LabelEncoder. And I create a dictionary, mylabels, that maps emotion names to integers to facilitate interpretation.

* 1. **Dataset Splitting**

To evaluate model performance during training, I split the training data into 80% training and 20% validation sets.

1. **Feature Engineering**
   1. **Tokenization**  
      The textual data is tokenized using BertTokenizer. This step transforms the input text into tokens that are compatible with the BERT model.

* Parameters：  
  max\_length=40: Ensures all tokenized sequences are of equal length (padded or truncated as needed).  
  return\_tensors='pt': Returns data in PyTorch tensor format for compatibility with the model.
  1. **Dataset Preparation**

A custom TextClassificationDataset class extends torch.utils.data.Dataset. It stores tokenized texts, labels, and handles data retrieval via indexing. And

it ensures consistent formatting of inputs (input\_ids, attention\_mask) for BERT.

一張含有 文字, 螢幕擷取畫面, 軟體, 多媒體軟體 的圖片

自動產生的描述

* 1. **Dataloader**

DataLoader is used to create batches for both training and validation datasets, ensuring efficient data handling during model training.

1. **Model Explanation**
   1. **BERTClassifier**

I utilize PyTorch-based classifier that leverages a pre-trained BERT model (bert-base-uncased) for feature extraction.

* Architecture:

1. BERT Backbone: Extracts contextual embeddings from input tokens.
2. Dropout Layer: Reduces overfitting by randomly setting some activations to zero during training.
3. Fully Connected Layer: Maps the embeddings to eight output classes corresponding to emotion labels.

* Outputs:

1. Logits for each class, which are converted to probabilities using CrossEntropyLoss.
   1. **Loss Function**

I use CrossEntropyLoss to compute the loss by comparing predicted logits with true labels.

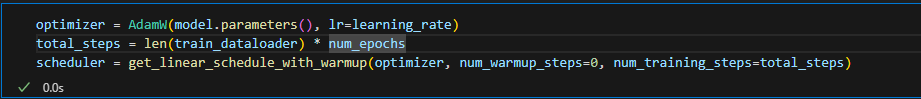
* 1. **Optimizer and Scheduler**

Optimizer: AdamW is used for efficient weight updates with a learning rate of 2e-5.

Scheduler: get\_linear\_schedule\_with\_warmup adjusts the learning rate during training, helping the model converge smoothly.

一張含有 文字, 螢幕擷取畫面, 軟體, 陳列 的圖片

自動產生的描述



1. **Training and Evaluation**
   1. **Training Loop**

Each epoch involves:

* + Forward pass: The model processes input batches and computes predictions.
  + Backward pass: Gradients are calculated and parameters are updated.
  + Learning rate adjustment: The scheduler fine-tunes the learning rate.
  1. **Evaluation**

After each epoch, the model's performance is evaluated on the validation set.

Metrics:

* Accuracy: Proportion of correct predictions.
* Classification Report: Includes precision, recall, and F1-score for each class.

一張含有 文字, 螢幕擷取畫面, 軟體, 作業系統 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 軟體, 多媒體軟體 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 字型, 陳列 的圖片

自動產生的描述

1. **Prediction and Output**
   1. **Prediction Function**

The predict\_sentiment function processes individual text samples:

* Tokenizes input text.
* Computes class logits using the trained model.
* Maps the predicted label index back to the corresponding emotion using mylabels.

一張含有 文字, 螢幕擷取畫面, 軟體, 多媒體軟體 的圖片

自動產生的描述

* 1. **Test Set Predictions**

The test dataset is processed row-wise, and predicted emotions are stored in a new column, predicted\_emotion.

一張含有 文字, 螢幕擷取畫面, 字型, 行 的圖片

自動產生的描述

* 1. **Output File**

The predictions are saved in a CSV file (bert\_classifier\_v2.csv) with two columns:

* id: Tweet ID from the test dataset.
* emotion: Predicted emotion.