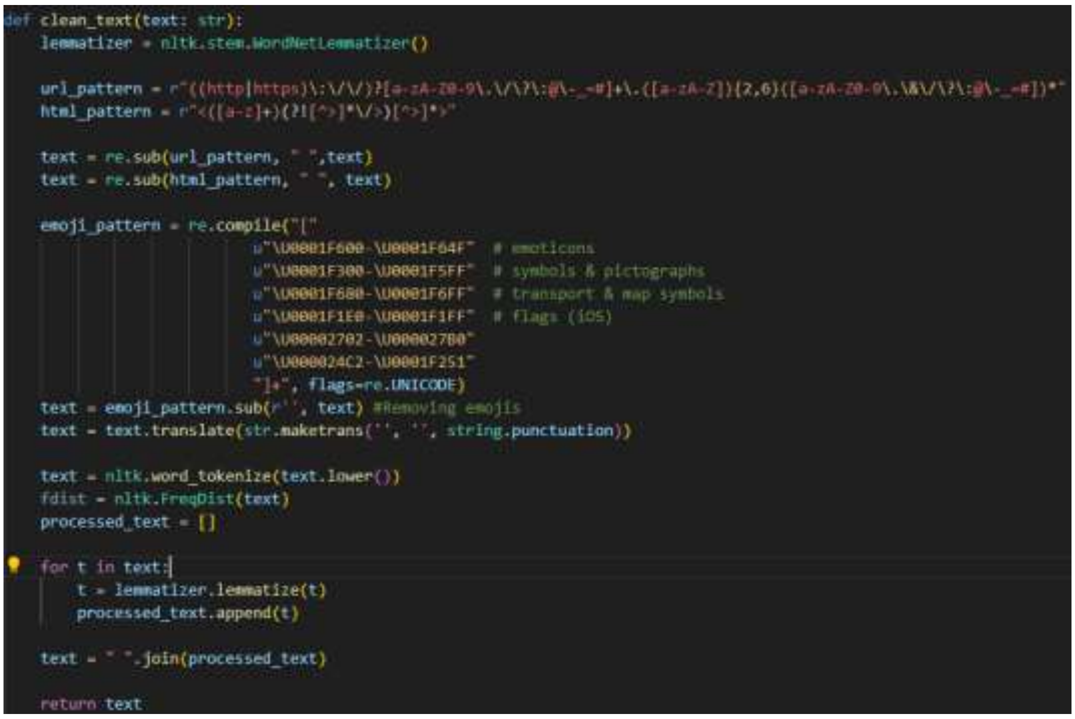
Data Mining HW2 Report

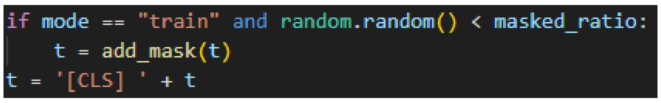
Student ID: 110550014 Name: 吳權祐

1. How do you select features for your model input, and what preprocessing did you perform to review text?

I used the following code to preprocess the data. I removed the URL, emoji, html labels and all punctuations in the sequence, then I transformed it into lowercase and lemmatized it. Additionally, I added a classify token in front of the sequence and randomly add the mask.



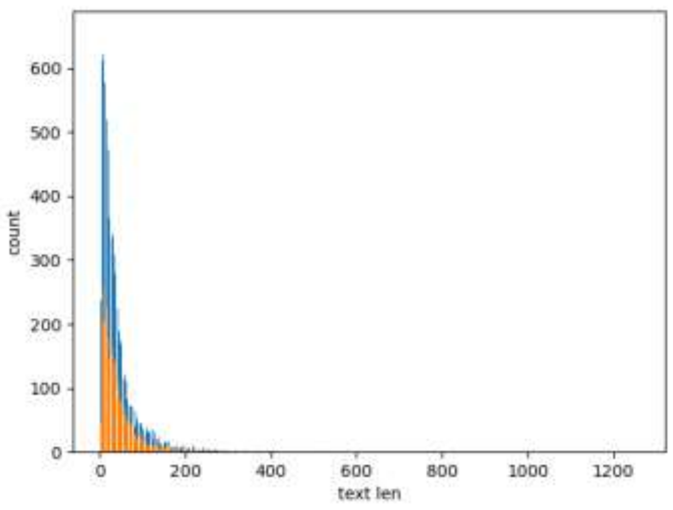
```python
def clean_text(text: str):
    lemmatizer = nltk.stem.WordNetLemmatizer()

    url_pattern = r"((http|https)\:\/\/)?[a-zA-Z0-9\.\/\?\:@\-_=#]+\.([a-zA-Z]){2,6}([a-zA-Z0-9\.\&\/\?\:@\-_=#])*"
    html_pattern = r"<([a-z]+)(?![^>]*\/>)[^>]*>"

    text = re.sub(url_pattern, " ",text)
    text = re.sub(html_pattern, " ", text)

    emoji_pattern = re.compile("["
                           u"\U0001F600-\U0001F64F"  # emoticons
                           u"\U0001F300-\U0001F5FF"  # symbols & pictographs
                           u"\U0001F680-\U0001F6FF"  # transport & map symbols
                           u"\U0001F1E0-\U0001F1FF"  # flags (iOS)
                           u"\U00002702-\U000027B0"
                           u"\U000024C2-\U0001F251"
                           "]+", flags=re.UNICODE)
    text = emoji_pattern.sub(r'', text) #Removing emojis
    text = text.translate(str.maketrans('', '', string.punctuation))

    text = nltk.word_tokenize(text.lower())
    fdist = nltk.FreqDist(text)
    processed_text = []

    for t in text:
        t = lemmatizer.lemmatize(t)
        processed_text.append(t)

    text = " ".join(processed_text)

    return text
```



```python
if mode == "train" and random.random() < masked_ratio:
    t = add_mask(t)
t = '[CLS] ' + t
```

2. Please describe how you tokenize your data, calculate the distribution of tokenized sequence length of the dataset and explain how you determine the padding size.

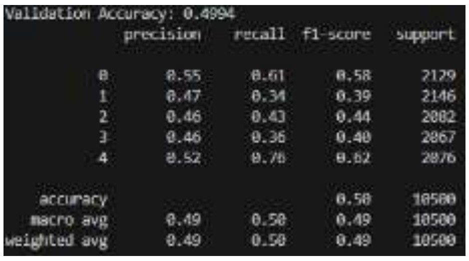
The figure below illustrates the distribution of tokenized sequence lengths. The blue curve represents the distribution for the training data, with an average length 27, while the orange curve depicts the distribution for the validation data, with an average length 31. Most sequences have a length of less than 200, so I set the padding size at 256.



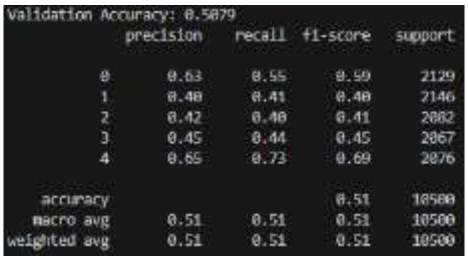
3. **Please compare the impact of using different methods to prepare data for different rating categories.**

   I used three different methods to prepare the data. The results indicate that the model trained on both the title and text data achieves higher accuracy across all rating categories. Additionally, the model trained separately on title data and text data demonstrates comparable performance. Notably, the title-trained model shows increased accuracy for rating 3, while the text-trained model excels in ratings 4 and 5.

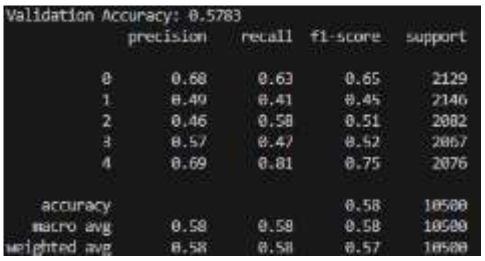
   Only title　　　　　　Only text　　　　　　Title + text



```
Validation Accuracy: 0.4994
              precision    recall  f1-score   support

           0       0.55      0.61      0.58      2129
           1       0.47      0.34      0.39      2146
           2       0.46      0.43      0.44      2082
           3       0.46      0.36      0.40      2067
           4       0.52      0.78      0.62      2076

    accuracy                           0.50     10500
   macro avg       0.49      0.50      0.49     10500
weighted avg       0.49      0.50      0.49     10500
```



```
Validation Accuracy: 0.5079
              precision    recall  f1-score   support

           0       0.63      0.55      0.59      2129
           1       0.40      0.41      0.40      2146
           2       0.42      0.40      0.41      2082
           3       0.45      0.44      0.45      2067
           4       0.65      0.73      0.69      2076

    accuracy                           0.51     10500
   macro avg       0.51      0.51      0.51     10500
weighted avg       0.51      0.51      0.51     10500
```



```
Validation Accuracy: 0.5783
              precision    recall  f1-score   support

           0       0.68      0.63      0.65      2129
           1       0.49      0.41      0.45      2146
           2       0.46      0.58      0.51      2082
           3       0.57      0.47      0.52      2067
           4       0.69      0.81      0.75      2076

    accuracy                           0.58     10500
   macro avg       0.58      0.58      0.58     10500
weighted avg       0.58      0.58      0.57     10500
```