Report B10705054

1. 標註執行環境

Google colab

2. 程式語言

Python 3 , 模組**BERT-TINY**

3. 執行方式：

非原生套件:

-From google .colab import drive:讓 Google Drive 中的文件可以在 google.colab 中使用。

-from sklearn.model_selection import train_test_split: 使用這個函數，可以將數據集劃分為訓練集和測試集，以便在機器學習模型的訓練和評估過程中使用

BERT:

from keras_bert import extract_embeddings

extract_embeddings 函數：

這個函數用於從 BERT 模型中提取嵌入向量（embedding vectors)。這些嵌入向量包含了輸入文本中每個詞的表示，這是 BERT 模型根據上下文學到的。這些向量通常用於後續的任務，例如文本分類、情感分析等。

SVM:

-from scikit-learn import SVM：SVM 是一種監督式機器學習演算法，用於分類和回歸任務。

-from sklearn import metrics:該模組提供了各種用於評估機器學習模型性能的函數。

-import numpy as np:-

原生套件:

-引入 python 標準庫 os，用於操作檔案和資料夾，方便讀取或寫入檔案。

4. 作業處理邏輯說明：

導入相關模組和設定 Google Colab 掛載：

載入文檔標籤：

-classes = [list(map(int, line.split())) for line in file]：

最終，classes 是一個包含多個子列表的列表，每個子列表代表一個標籤，第一個元素是類別標籤，其餘的元素是相應的文檔編號。

載入文本數據和標籤：

with open(os.path.join(document_folder, f"{i}.txt"), 'r', encoding='utf-8') as file:

content = file.read() 載入 PA1-data folder

BERT:

model_path = '/content/drive/My Drive/uncased_L-2_H-128_A-2'

使用 **BERT-TINY** model

```
dict_path = '/content/drive/My Drive/uncased_L-2_H-128_A-2/vocab.txt'
embeddings = extract_embeddings(model_path, texts)
```

這個函數用於從 BERT-Tiny 模型中提取嵌入向量（embedding vectors）。從 embeddings 中建構訓練集（x_train 和 y_train）和測試集（x_test 和 y_test）。這些資料集通常用於機器學習模型的訓練和測試。

構建訓練集和測試集:

```
for i in classes:
    cls = i[0]
    for doc_id in i[1:]:
```

將文檔的嵌入向量和標籤添加到訓練集：

建立測試集 x_test：

```
all_docs = set(range(1, 1096))
docs_not_in_train = all_docs - set(docs_in_train)
```

創建測試集的過程中，首先建立了一個包含所有文檔 ID 的集合 all_docs。然後，計算出不在訓練集中的文檔 ID 集合 docs_not_in_train。接著，對於每個不在訓練集中的文檔 ID，從 embeddings 中獲取對應的嵌入向量，然後將其添加到 x_test。

SVM：

```
SVM(Linear) : SVM_model = SVC(kernel='linear', C=1.0, probability=True)
SVM_model. fit(x_train, y_train)
```

進行預測和評估模型：

```
predicted_results = model.predict(x_test)
expected_results = y_test
print(metrics.classification_report(expected_results,predicted_results))
```

寫入 CSV 檔案:

具體步驟包括打開檔案（在 "/content/drive/My Drive/op.csv" 位置）、建立 CSV 寫入器、寫入表頭（"Id" 和 "Value" 兩個欄位），接著將集合 docs_not_in_train 轉換為列表 docs_not_in_train_list，最後使用迴圈將每組資料寫入檔案，其中每一行包含兩個欄位，分別是文檔 ID (doc_id) 和對應的預測結果 (predicted_results[i])。

5.結果:

由於使用 BERT-TINY，所以 kaggle 的 score 並不是很高。來到 0.86944.