Natural Language Processing and Text Mining:

HW#3

小組成員和負責工作：

109590041 范遠皓 撰寫程式和測試

109590043 柯瑞霖 撰寫程式和報告

環境

使用的語言：Python

所需套件：

- gensim==4.3.1
- joblib==1.2.0
- numpy==1.24.3
- pandas==2.0.1
- python-dateutil==2.8.2
- pytz==2023.3
- scikit-learn==1.2.2
- scipy==1.10.1
- six==1.16.0
- sklearn==0.0.post5
- smart-open==6.3.0
- threadpoolctl==3.1.0
- tzdata==2023.3

安裝辦法：

Zip 檔內有 requirements.txt

終端機輸入以下指令安裝

pip3 install -r requirements.txt

載入預訓練的詞向量模型，使用 gensim word2vec model

下載連結

```
word_vectors = KeyedVectors.load_word2vec_format('GoogleNews-vectors-negative300.bin', binary=True)
```

讀取 WordSim-353 資料集

```
# 讀取WordSim-353資料集
Humans_mean = []
def read_wordsim_dataset(file_path):
    pairs = []
    with open(file_path, 'r') as file:
        for line in file:
            word1, word2, score = line.strip().split('\t')
            if (score == "Human (mean)"):
                continue
            pairs.append((word1, word2, float(score)))
            Humans_mean.append(float(score))
    return pairs

wordsim_pairs = read_wordsim_dataset('wordsim353/combined.txt')
```

計算 WordSim-353 資料集各個單詞的相似度

```
# 計算單詞相似度
similarities = []
for pair in wordsim_pairs:
    word1, word2, _ = pair
    similarity = word_vectors.similarity(word1, word2)
    similarities.append(similarity)
```

輸出 correlation

```
# 輸出 correlation
correlation, p_value = pearsonr(Humans_mean, similarities)
print('Word similarity')
print("Correlation:", correlation)
```

讀取 BATS_3.0 資料並轉成 features 的形式

```
# 讀取 BATS_3.0 資料並轉成 features 的形式
def read_BATS_dataset():
    folder1_path = 'BATS_3.0'
    folder2_paths = []
    datas = pd.DataFrame(columns=['features', 'label'])

    if os.path.exists(folder1_path) and os.path.isdir(folder1_path):
        for file_name in os.listdir(folder1_path):
            folder2_paths.append(os.path.join(folder1_path, file_name))
    else:
        print('No such folder')

    for folder_name in folder2_paths:
        if not(os.path.exists(folder_name) and os.path.isdir(folder_name)):
            continue
        for file_name in os.listdir(folder_name):
            label = file_name.split(']')[0].split('[', 1)[1]
            temp = []
            data = pd.read_csv(os.path.join(folder_name, file_name), sep='\t', header=None, names=['word1', 'word2'])
            for index, row in data.iterrows():
                if '/' in row['word2']:
                    for word2 in row['word2'].split('/'):
                        try:
                            features = word_vectors[row['word1']] - word_vectors[word2]
                            temp.append(pd.Series({'features' : features, 'label' : label}))
                        except:
                            pass
                else:
                    try:
                        features = word_vectors[row['word1']] - word_vectors[row['word2']]
                        temp.append(pd.Series({'features' : features, 'label' : label}))
                    except:
                        pass
            datas = pd.concat([datas, pd.DataFrame(temp)], ignore_index=True)
    return datas
```

前段部分是在進入 BATS_3.0 資料夾內層中的各個資料夾，之後將資料夾內的兩個或多個單字做向量的相減，並用此數值作為特徵值和資料夾名稱作為 Label。

分類和輸出

```
# 分類
X_train, X_test, y_train, y_test = train_test_split(list(datas['features']), list(datas['label']), test_size=0.2, random_state=42)
classifier = LogisticRegression()
classifier.fit(X_train, y_train)
predictions = classifier.predict(X_test)
print()
print('Analogy prediction')
print(classification_report(y_test, predictions))
```

用 LogisticRegression 進行訓練，前 80%訓練，後 20%測試。

執行結果：

```
Word similarity
Correlation: 0.6525349618875615

Analogy prediction
                      precision    recall  f1-score   support

     UK_city - county       1.00      0.38      0.55         8
    adj - comparative       0.92      1.00      0.96        11
    adj - superlative       1.00      1.00      1.00        11
           adj+ly_reg       1.00      1.00      1.00        10
         adj+ness_reg       1.00      0.82      0.90        11
     animal - shelter       1.00      0.90      0.95        20
       animal - sound       0.95      0.95      0.95        21
       animal - young       0.93      0.81      0.87        16
    antonyms - binary       0.40      0.32      0.36        37
  antonyms - gradable       0.79      0.87      0.83       145
    country - capital       1.00      0.50      0.67        10
   country - language       1.00      0.57      0.73        14
  hypernyms - animals       0.89      0.99      0.93        94
     hypernyms - misc       0.88      0.94      0.91       122
      hyponyms - misc       0.80      0.88      0.84       226
        male - female       1.00      0.87      0.93        15
    meronyms - member       0.53      0.73      0.62        11
      meronyms - part       0.73      0.79      0.76       119
 meronyms - substance       0.77      0.64      0.70        36
   name - nationality       0.50      0.80      0.62         5
    name - occupation       1.00      0.86      0.92        21
  noun - plural_irreg       0.50      0.25      0.33        12
    noun - plural_reg       0.50      0.55      0.52        11
        noun+less_reg       1.00      0.38      0.55         8
         over+adj_reg       1.00      0.62      0.77         8
          re+verb_reg       0.83      0.45      0.59        11
     synonyms - exact       0.60      0.47      0.53        32
 synonyms - intensity       0.61      0.61      0.61        44
       things - color       0.97      0.97      0.97        30
           un+adj_reg       0.67      0.50      0.57        12
        verb+able_reg       1.00      0.88      0.93         8
        verb+er_irreg       1.00      0.80      0.89         5
      verb+ment_irreg       0.40      0.67      0.50         6
      verb+tion_irreg       0.50      0.33      0.40         6
      verb_3pSg - Ved       1.00      1.00      1.00        12
     verb_Ving - 3pSg       1.00      1.00      1.00         9
      verb_Ving - Ved       1.00      0.80      0.89        15
      verb_inf - 3pSg       1.00      1.00      1.00        13
       verb_inf - Ved       1.00      1.00      1.00        13
      verb_inf - Ving       0.92      0.86      0.89        14

             accuracy                           0.81      1232
            macro avg       0.84      0.74      0.77      1232
         weighted avg       0.81      0.81      0.81      1232
```