Project 3 – Text Classification

1. Overview

In this project, you are asked to design an algorithm that correctly predicts whether the text expresses positive or negative emotions. You are given a data set which is a text binary classification data set. Its source data is comment data from various platforms. There are only label 1 and label 0, which are used to indicate whether the source texts express positive or negative emotions.

<table>
<tr><th></th><th>Scoring rules</th><th>Dead Line</th></tr>
<tr><td>Lab6</td><td>The scores you get in this test depend on your prediction accuracy.

<table><tr><th>Accuracy level</th><th>Score level</th></tr><tr><td>89.8%</td><td>90</td></tr><tr><td>89%</td><td>85</td></tr><tr><td>85%</td><td>80</td></tr><tr><td>80%</td><td>70</td></tr><tr><td>Submitted and the output satisfies our format requirement.</td><td>60</td></tr></table>
The score for the accuracy that lies in the interval between two accuracy levels is given based on a linear function.</td><td>Date: Dec. 23th

Time: 23:55

If later than DDL, 20% points will be taken off. (The latest time is Dec.26th, 23:55. We will not accept submissions later than this time)</td></tr>
</table>

Table 1

After Lab6, report is not required for this project.

Project3_score = Lab6_Score*100%

2. Data Description

a) What we have done to the dataset

All text data in the original data set is encrypted. The original data set is English data set. After data cleaning, all words are one-to-one mapped to ciphertexts generated randomly. In the mapping process, all words are one-to-one mapped to ciphertexts without many-to-one or one-to-many mapping relationship. In the process of data cleaning, punctuation marks and other meaningless parts in the text are all removed. The overall data can be processed as sentences written in English grammar using a completely new language.

b) Training data and test data

The training set and test set are redivided according to the label after we have shuffled the original data, to ensure the ratio of the label 0 and the label 1 in the training set and the test set is 1:1(i.e. no data imbalance).

There are 25,000 samples in the training set released to you. The size of the test set will not exceed that of the training set.

There are about 90,000-110,000 words in the training set and the test set, among which 76,025 words are in the training set. The maximum string length of the document in the training set is 2,497, while the minimum length is 8. There is a similar string length distribution in the test set, will not greatly exceed the maximum length of the training set.

c) How to load the training data

The training data is saved in **json** format. After being loaded in Python, it will be an array contains 25,000 elements. Each element in the array is a dictionary which has two fields, 'data' and 'label'. The value of 'data' is a string, the value of 'label' indicates the emotional tendency of the string.

3. Submission requirement

You should submit all your code in **zip** format on a website(will release later). The main executable should be named **train.py(The time of training should not exceed 24 hours)** , **test.py(The time of test should not exceed 15min.)** and model(not larger than 500M) put in **root directory** or named **train_test.py(The time of training and test should not exceed 20min.)** Multiple files and subdirectories are allowed. Your directory structure may look like the following:

```
|-- train.py
|-- test.py
|-- model
`-- submodule
            |-- submoduleA.py
            |-- submoduleB.py
```

or

```
|-- train_test.py
`-- submodule
            |-- submoduleA.py
            |-- submoduleB.py
```

4. Evaluation

After your submission, we will test your training program and test program.

a) Test program requirement

The test program must be named by **test.py. The time of test should not exceed 15min.**

i. **Input**: **test.py** will be executed with the following command:

python test.py –i <test data file>

or

python test.py –i <test data file> -m <model file>

< test data file> is the absolute path of the test data file. The test data file only include an array contains only values of “data” fields of the test

data set, you should output the "label" of each sample satisfying the output format requirement.

An example is provided in the package, please see **testdataexample**.

< model file > is the absolute path of the model file which is the output of your train.py.

ii. **Output**: the predicted label of your test.py according to the test data. You can output one label per line for each test sample. The content of your output **should be written into a file** named as **output.txt** in your root folder (In the same **directory** as file **test.py**).

Output format:

1

0

.....

1

The scores you get in this project depend on the prediction accuracy of your model on the test data set，please see Table 1 for details.

b) Training program requirement

The training program must be named by **train.py. The time of training should not exceed 24 hours.**

i. **Input**: **train.py** will be executed with the following command::

python train.py -t < training data file >

< training data file > is the absolute path of the training data file.

ii. **Output**: you should save your model to a file named **model** in the **same directory as train.py.**

Notice: We do not define the format of **model** file, but you should make sure that your test.py can correctly load the parameters saved in your own **model** file, and can use the parameters of **model** file to predict and output the labels of the test data. This is to check the consistency between your training program and test program.

c) Train and Test program requirement

If your training and test time is no longer than 20 minutes, you can put train and test together, the program must be named by **train_test.py**.

i. **Input**: **train_test.py** will be executed with the following command:
python train_test.py -t < training data file > –i <test data file>

ii. **Output**: the same as a.ii.

5. Test Environment

1) Python version: 3.6.9
   Scikit-learn version: 0.23.2
   Pytorch:1.7.0
   Tensorflow:1.14.0
   Joblib:0.17.0
   h5py:3.1.0
   torchtext:0.8.1
2) CPU: E5-2690 v4*2
   GPU: TITAN V*8
3) OS: Ubuntu 18.04.3 LTS