Web Prototype with sentiment analysis

P&G Test Code Report

Rong Peng

I. Introduction

The project trial aims at building a simple web prototype, allowing the user to upload the training and testing files for conducting simple sentiment analysis. They should be in **'.alt'** format, you could modify the text file suffix as '.alt'.

The web front-end was coded in Python Flask v1.1.1, and saving the upload data in the local folders, the backend used TF-IDF to extract text features, then return back the related information and display results in the Flask.

II. Problem Selection

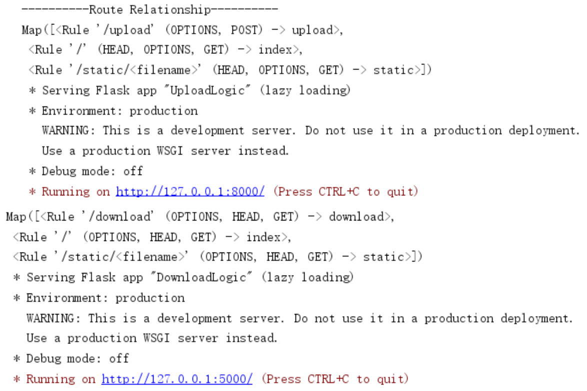
A. Premise

To run the prototype on your computer, you should install Flask v1.1.1 and Python interpreter I used Python 3.7 which included in Anaconda3. In addition, make sure you have installed sklearn, matplotlib, numpy packages. Recommended IDE is PyCharm.

After preparing the premise well, open the project, you can see three Python codes, three folders under the 'static' folder. The 'train' folder will save the training files which you uploading from the website. Same with it, the 'test' folder will save the testing files which you uploaded. The 'result' folder will save the final conclusion report and figure.

To execute the project, the codes should be run in order. First one run the 'UploadLogic.py', then run 'SentAnalysis.py', the last one should run the 'DownloadLogic.py'.

If you successfully execute, you will see the Fig1 as below, click the URL will direct to the upload page.

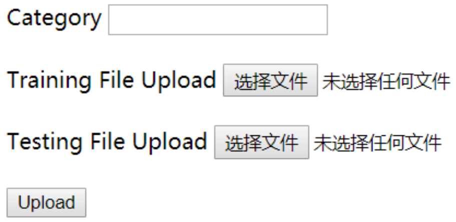


```
----------Route Relationship----------
 Map([<Rule '/upload' (OPTIONS, POST) -> upload>,
  <Rule '/' (HEAD, OPTIONS, GET) -> index>,
  <Rule '/static/<filename>' (HEAD, OPTIONS, GET) -> static>])
  * Serving Flask app "UploadLogic" (lazy loading)
  * Environment: production
    WARNING: This is a development server. Do not use it in a production deployment.
    Use a production WSGI server instead.
  * Debug mode: off
  * Running on http://127.0.0.1:8000/ (Press CTRL+C to quit)
Map([<Rule '/download' (OPTIONS, HEAD, GET) -> download>,
 <Rule '/' (OPTIONS, HEAD, GET) -> index>,
 <Rule '/static/<filename>' (OPTIONS, HEAD, GET) -> static>])
 * Serving Flask app "DownloadLogic" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Fig1. UploadLogic.py Flask running

B. Usage

The uploading page you have to define a category for the files you are going to upload. As Fig2 showing, fill in your Category, browse your local files and select to upload.



Category

Training File Upload 选择文件 未选择任何文件

Testing File Upload 选择文件 未选择任何文件

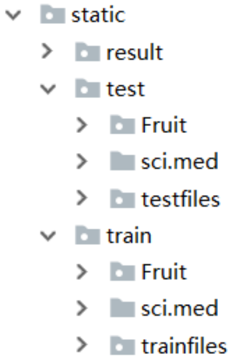
Upload

Fig2. Uploading Page

When you have done, it will prompt "Done the Uploading", you will find the newly created folder, named as your inputted category.

It is recommended to upload **more than one category** and more training and testing data, will be better for model training and fitting.

Take my uploading file as an example, randomly input three categories, as Fig3

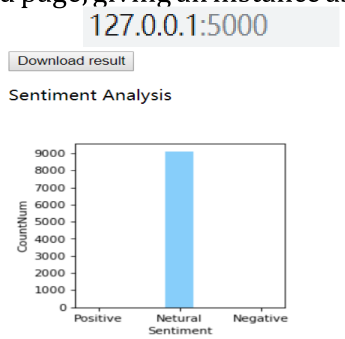


```
static
  result
  test
    Fruit
    sci.med
    testfiles
  train
    Fruit
    sci.med
    trainfiles
```

Fig3. Categories

After running SentAnalysis.py, results will be analyzed and generated in report.txt, and will be displayed the result Figure in download page.

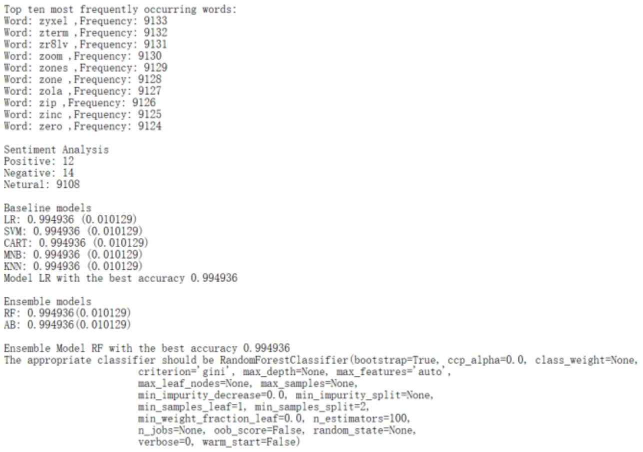
Manually change the PORT as '5000', will see the download page, giving an instance as Fig4.



127.0.0.1:5000

Fig4.Download Page

When clicking the ‘Download Result’ button, you can view the report of the analysis, as Fig5.



```
Top ten most frequently occurring words:
Word: zyxel ,Frequency: 9133
Word: zterm ,Frequency: 9132
Word: zr8lv ,Frequency: 9131
Word: zoom ,Frequency: 9130
Word: zones ,Frequency: 9129
Word: zone ,Frequency: 9128
Word: zola ,Frequency: 9127
Word: zip ,Frequency: 9126
Word: zinc ,Frequency: 9125
Word: zero ,Frequency: 9124

Sentiment Analysis
Positive: 12
Negative: 14
Netural: 9108

Baseline models
LR: 0.994936 (0.010129)
SVM: 0.994936 (0.010129)
CART: 0.994936 (0.010129)
MNB: 0.994936 (0.010129)
KNN: 0.994936 (0.010129)
Model LR with the best accuracy 0.994936

Ensemble models
RF: 0.994936(0.010129)
AB: 0.994936(0.010129)

Ensemble Model RF with the best accuracy 0.994936
The appropriate classifier should be RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=None, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=100,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

Fig5. Example Report.txt

III.Future Work

In future work, the sentiment lexicon should be improved and completed. Another aspect, the final suitable model should be utilized, and benefit for classification. And the website UI should also need to be upgraded.