بسمه تعالی

Vowpal Wabbit

ساختار و پوشه بندی کل پروژه :

پوشه src : شامل تمامی کد های اجرایی است.
پوشه data :شامل تمامی داده های جمع آوری شده یا ساخته شده است .
پوشه tools : شامل ابزار های استفاده شده در کد برنامه است.
پوشه analyze : شامل نمودار ها و نتایج بدست آمده از در هر فاز و نیز داکیومنت ها است.

نحوه تمیز کردن داده :

فایل preprocessor.sh شامل دستورات tr زیر است
برای مشاهدا بهتر مراجعه شود به فایل

```
tr 'ی' 'ئ' < ../data/raw_celebrity_comments | tr '' ' ' | tr -sc "A-Za-z ضصثقفغعهخحجچشسیبلاتنمکگظطزرذدپوائ'.,-" " " | tr -s [:space:] " " | tr [:upper:] [:lower:] | tr -s "A-Za-z ضصثقفغعهخحجچشسیبلاتنمکگظطزرذدپوائ'.,-" > ../data/celebrity_comments
```

```
tr 'ی' 'ئ' < ../data/raw_friendly_comments | tr '' ' ' | tr -sc "A-Za-z ضصثقفغعهخحجچشسیبلاتنمکگظطزرذدپوائ'.,-" "" | tr -s [:space:] " " | tr [:upper:] [:lower:] | tr -s "A-Za-z ضصثقفغعهخحجچشسیبلاتنمکگظطزرذدپوائ'.,-" > ../data/friendly_comments
```

روش کار :
۱ : ی عربی را با فارسی جایگزین میکنیم
۲: نیم فاصله را با فاصله جایگزین میکنیم
۳: همه کاراکتر ها به غیر از فارسی و ابگلیسی را حذف میکنیم.
۴: فاصله های زیاد را با یکی جایگزین میکنیم (لزومی ندارد)
۵:حروف بزرگ انگلیسی را با کوچک جایگزین میکنیم .
۶:حروف تکراری را با یک حرف جایگزین میکنیم (بعضی کلمات انگلیسی شامل ۲ حرف پشت هم با معنی است اما به دلیل اینکه حالتی که هم خودشان و هم تک حرفشان معنی بدهد کم هستند از این حالت صرف نظر میکنیم و تکرار کم آنها انها را با نسخه یک حرفشان یکی میگیریم) مانند good , god

۷ : داده تمیز شده را در خروجی میریزیم

بسمه تعالی

فایل های مربوط به این فاز :

داده ها را برچسب گذاری و آماده آموزش و تست میکند src/p2_vowpal_labeler.py
دستور ووپل را برای آموزش و تست اجرا میکند src/p2_vowpal_trainer.py
کامنت های آموزش زا دارد data/p2_all_labeled_comments_vowpal.txt
data/p2_test_comments_vowpal.txt

مکانیزم جدا سازی داده ی تست و ترین :

از هر sample_rate = 10 نمونه یکی را به عنوان تست کیس در نظر میگیریم و از دیتای اصلی خارج میکنیم.

نحوه اجرای کد :

وارد پوشه src میشویم.

```
python3 p2_vowpal_wabbit_labeler.py
Sudo ./p2_vowpal_wabbit_trainer
```

توضیح :

ابتدا باید دیتا ها را به شکل نمونه ی زیر در دو کلاس ۱ (friendly) و -۱ (celebrity) را برچسب گذاری میکنیم تا آماده ی یادگیری شود

-1 | ادکلن اورجینال میخوای با قیمت مناسب
-1 | چطور ممبر تلگرامی خودمون رو افزایش بدیم بیا پیج جوابش رو بگم بهت
-1 | لایک

1 | بسیار سپاسگزارم زیبا و دیدنی ست
1 | محمد جان تشکر از بابت عکسای عالیت. اقا نزدیک یزدی یه سر بیا پیشمون

سپس باید دستورات vowpal زیر را وارد کنیم :
دستور آموزش :

```
  vw -d ../data/p2_all_labeled_comments_vowpal.txt  --passes 10 -f ../
data/predictor.vw  --ngram 1 -p ../data/p2_train_prediction_vowpal.txt
                                --loss_function quantile -c
```

پارامتر f- خروجی مدل لرن شده است
پارامتر ngram تعداد n-gram ها را مشخص میکند .
پارامتر passes تعداد بارهای ورودی دادن را مشخص میکند.

بسمه تعالی

پارامتر p- خروجی پردیکت را نشان میدهد
پارامتر c- داده را کش میکند.
پارامتر loss_function نشان میدهد میزان خطا با چه تابعی باید اندازه گیری شود
https://github.com/JohnLangford/vowpal_wabbit/wiki/Loss-functions
نکته : اگر p- را موقع آموزش استفاده کنیم . دیتای آموزشی را تست و predict میکند

دستور تست :

```
vw  ../data/p2_test_comments_vowpal.txt -t -i ../data/predictor.vw -
p ../data/p2_test_prediction_vowpal.txt
```

پارامتر i- مدل ورودی(لرن شده) برای تست را نشان میدهد
پارامتر (test only) t- نشان میدهد فقط برای تست است و آموزش ندارد
پارامتر p- برای خروجی تست است

خروجی نمونه:

برای 1-gram

```
Generating 1-grams for all namespaces.
final_regressor = ../data/predictor.vw
predictions = ../data/p2_train_prediction_vowpal.txt
Num weight bits = 18
learning rate = 0.5
initial_t = 0
power_t = 0.5
decay_learning_rate = 1
using cache_file = ../data/p2_all_labeled_comments_vowpal.txt.cache
ignoring text input in favor of cache input
num sources = 1
average  since         example        example  current  current
current
loss     last          counter         weight    label  predict
features
0.500000 0.500000            1            1.0   1.0000   0.0000
3
0.427849 0.355698            2            2.0   1.0000   0.2886
3
0.321700 0.215551            4            4.0   1.0000   0.6634
2
0.203983 0.086267            8            8.0   1.0000   0.8795
7
0.108205 0.012427           16           16.0   1.0000   0.9878
3
0.055539 0.002873           32           32.0   1.0000   0.9953
8
```

بسمه تعالی

```
0.028142 0.000745           64           64.0   1.0000   1.0000
6
0.014071 0.000000          128          128.0   1.0000   1.0000
7
0.007036 0.000000          256          256.0   1.0000   1.0000
17
0.003518 0.000000          512          512.0   1.0000   1.0000
3
0.001759 0.000000         1024         1024.0   1.0000   1.0000
2
0.022727 0.043696         2048         2048.0  -1.0000  -1.0000
25
0.029168 0.029168         4096         4096.0   1.0000   1.0000
4 h
0.051132 0.073095         8192         8192.0   1.0000   1.0000
7 h

finished run
number of examples per pass = 3365
passes used = 4
weighted example sum = 13460.000000
weighted label sum = 4.000000
average loss = 0.013299 h
best constant = 1.000000
best constant's loss = 0.499851
total feature number = 102716
Generating 1-grams for all namespaces.
only testing
predictions = ../data/p2_test_prediction_vowpal.txt
Num weight bits = 18
learning rate = 0.5
initial_t = 0
power_t = 0.5
using no cache
Reading datafile = ../data/p2_test_comments_vowpal.txt
num sources = 1
average  since         example        example  current  current
current
loss     last          counter         weight    label  predict
features
0.000000 0.000000            1            1.0  -1.0000  -1.0000
5
2.000000 4.000000            2            2.0   1.0000  -1.0000
7
2.000000 2.000000            4            4.0   1.0000  -1.0000
2
2.000000 2.000000            8            8.0   1.0000  -1.0000
2
```

بسمه تعالی

```
2.000000 2.000000           16           16.0   1.0000  -1.0000
4
2.000000 2.000000           32           32.0   1.0000  -1.0000
1
2.000000 2.000000           64           64.0   1.0000  -1.0000
1
1.999992 1.999985          128          128.0   1.0000  -1.0000
3
1.999996 2.000000          256          256.0   1.0000  -1.0000
4

finished run
number of examples per pass = 416
passes used = 1
weighted example sum = 416.000000
weighted label sum = 0.000000
average loss = 1.999992
best constant = 0.000000
best constant's loss = 1.000000
total feature number = 3056
```

برای 2-gram

```
Generating 2-grams for all namespaces.
final_regressor = ../data/predictor.vw
predictions = ../data/p2_train_prediction_vowpal.txt
Num weight bits = 18
learning rate = 0.5
initial_t = 0
power_t = 0.5
decay_learning_rate = 1
using cache_file = ../data/p2_all_labeled_comments_vowpal.txt.cache
ignoring text input in favor of cache input
num sources = 1
average  since         example        example  current  current
current
loss     last          counter         weight    label  predict
features
0.500000 0.500000            1            1.0   1.0000   0.0000
4
0.437515 0.375031            2            2.0   1.0000   0.2499
4
0.349676 0.261837            4            4.0   1.0000   0.5596
2
0.233554 0.117431            8            8.0   1.0000   0.8260
12
0.136471 0.039388           16           16.0   1.0000   0.9391
4
```

بسمه تعالی

```
0.077122 0.017774           32           32.0   1.0000   0.9691
14
0.041570 0.006017           64           64.0   1.0000   1.0000
10
0.020785 0.000000          128          128.0   1.0000   1.0000
12
0.010392 0.000000          256          256.0   1.0000   1.0000
32
0.005196 0.000000          512          512.0   1.0000   1.0000
4
0.002598 0.000000         1024         1024.0   1.0000   1.0000
2
0.035694 0.068790         2048         2048.0  -1.0000  -1.0000
48
0.051135 0.051135         4096         4096.0   1.0000   1.0000
6 h
0.091343 0.131552         8192         8192.0   1.0000   1.0000
12 h

finished run
number of examples per pass = 3365
passes used = 4
weighted example sum = 13460.000000
weighted label sum = 4.000000
average loss = 0.021561 h
best constant = 1.000000
best constant's loss = 0.499851
total feature number = 180012
Generating 2-grams for all namespaces.
only testing
predictions = ../data/p2_test_prediction_vowpal.txt
Num weight bits = 18
learning rate = 0.5
initial_t = 0
power_t = 0.5
using no cache
Reading datafile = ../data/p2_test_comments_vowpal.txt
num sources = 1
average  since         example        example  current  current
current
loss     last          counter         weight    label  predict
features
0.000000 0.000000            1            1.0  -1.0000  -1.0000
8
2.000000 4.000000            2            2.0   1.0000  -1.0000
12
2.000000 2.000000            4            4.0   1.0000  -1.0000
2
```

بسمه تعالی

```
2.000000 2.000000            8            8.0   1.0000  -1.0000
2
2.000000 2.000000           16           16.0   1.0000  -1.0000
6
2.000000 2.000000           32           32.0   1.0000  -1.0000
1
2.000000 2.000000           64           64.0   1.0000  -1.0000
1
2.000000 2.000000          128          128.0   1.0000  -1.0000
4
2.000000 2.000000          256          256.0   1.0000  -1.0000
6

finished run
number of examples per pass = 416
passes used = 1
weighted example sum = 416.000000
weighted label sum = 0.000000
average loss = 1.999893
best constant = 0.000000
best constant's loss = 1.000000
total feature number = 5334
```

برای 3-gram

```
Generating 3-grams for all namespaces.
final_regressor = ../data/predictor.vw
predictions = ../data/p2_train_prediction_vowpal.txt
Num weight bits = 18
learning rate = 0.5
initial_t = 0
power_t = 0.5
decay_learning_rate = 1
using cache_file = ../data/p2_all_labeled_comments_vowpal.txt.cache
ignoring text input in favor of cache input
num sources = 1
average  since         example        example  current  current
current
loss     last          counter         weight    label  predict
features
0.500000 0.500000            1            1.0   1.0000   0.0000
4
0.437515 0.375031            2            2.0   1.0000   0.2499
4
0.349676 0.261837            4            4.0   1.0000   0.5596
2
0.234100 0.118524            8            8.0   1.0000   0.8236
16
```

بسمه تعالی

```
0.138668 0.043237           16           16.0   1.0000   0.9343
4
0.079530 0.020391           32           32.0   1.0000   0.9652
19
0.043566 0.007603           64           64.0   1.0000   1.0000
13
0.021783 0.000000          128          128.0   1.0000   1.0000
16
0.010892 0.000000          256          256.0   1.0000   1.0000
46
0.005446 0.000000          512          512.0   1.0000   1.0000
4
0.002723 0.000000         1024         1024.0   1.0000   1.0000
2
0.050116 0.097509         2048         2048.0  -1.0000  -1.0000
70
0.073299 0.073299         4096         4096.0   1.0000   0.9731
7 h
0.134481 0.195663         8192         8192.0   1.0000   1.0000
16 h

finished run
number of examples per pass = 3365
passes used = 4
weighted example sum = 13460.000000
weighted label sum = 4.000000
average loss = 0.029213 h
best constant = 1.000000
best constant's loss = 0.499851
total feature number = 247752
Generating 3-grams for all namespaces.
only testing
predictions = ../data/p2_test_prediction_vowpal.txt
Num weight bits = 18
learning rate = 0.5
initial_t = 0
power_t = 0.5
using no cache
Reading datafile = ../data/p2_test_comments_vowpal.txt
num sources = 1
average  since         example        example  current  current
current
loss     last          counter         weight    label  predict
features
0.000000 0.000000            1            1.0  -1.0000  -1.0000
10
2.000000 4.000000            2            2.0   1.0000  -1.0000
16
```

بسمه تعالی

```
2.000000 2.000000            4            4.0   1.0000  -1.0000
2
2.000000 2.000000            8            8.0   1.0000  -1.0000
2
2.000000 2.000000           16           16.0   1.0000  -1.0000
7
2.000000 2.000000           32           32.0   1.0000  -1.0000
1
2.000000 2.000000           64           64.0   1.0000  -1.0000
1
2.000000 2.000000          128          128.0   1.0000  -1.0000
4
2.000000 2.000000          256          256.0   1.0000  -1.0000
7

finished run
number of examples per pass = 416
passes used = 1
weighted example sum = 416.000000
weighted label sum = 0.000000
average loss = 1.999860
best constant = 0.000000
best constant's loss = 1.000000
total feature number = 7317
```