Twitter Writing Style Analysis

Paris Diderot University
Computer Science Master 2
Data Mining
FAN Yi-Zhe
HSIEH Yung-Kun

Introduction

Final Goal

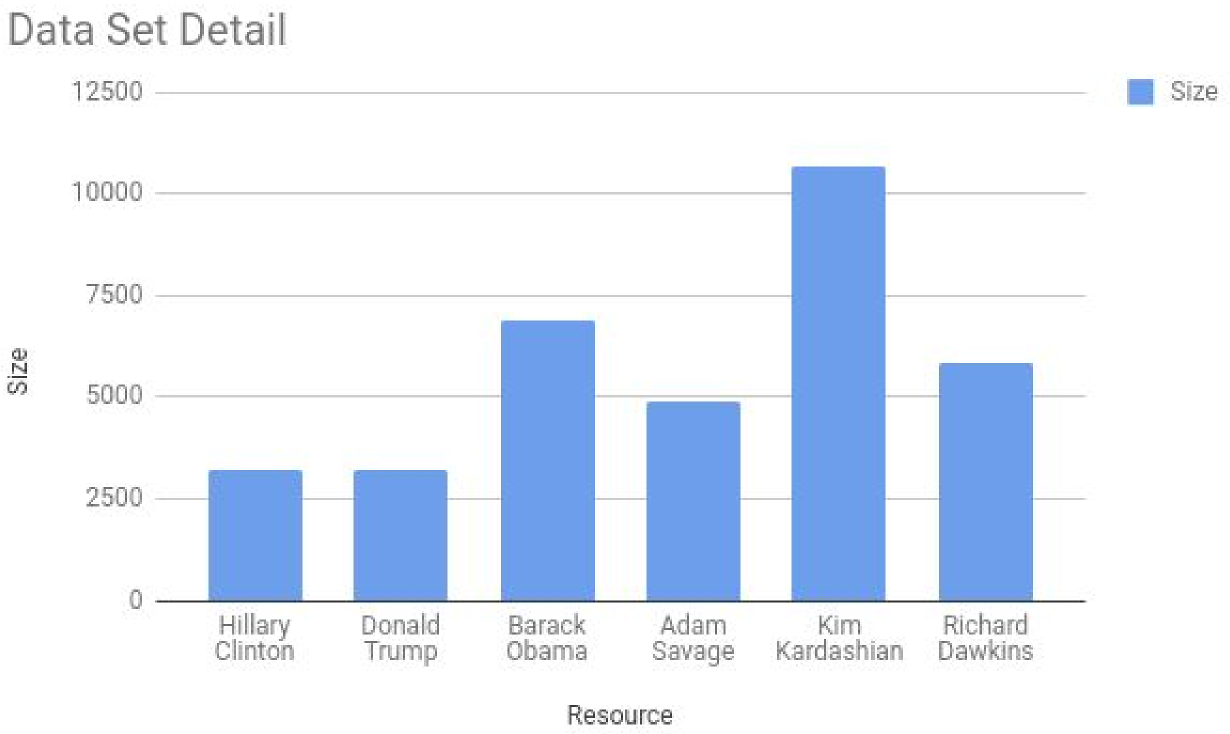
To detect the **plagiarism** of text on the internet from intellectuals/copywriters.

Current Status

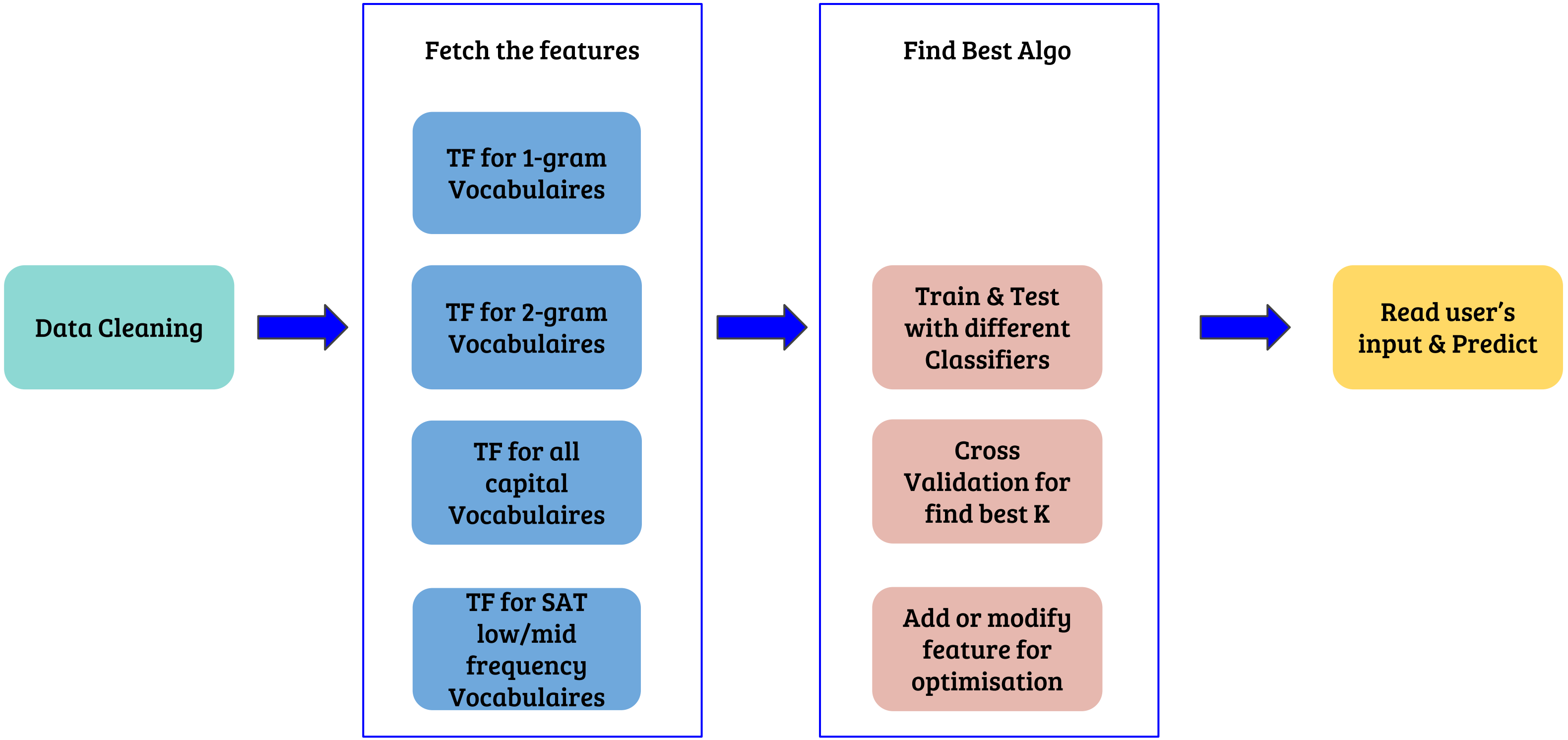
Twitter analysis in order to correctly recognize the tweets from different authors.

6 Datasets of Tweets

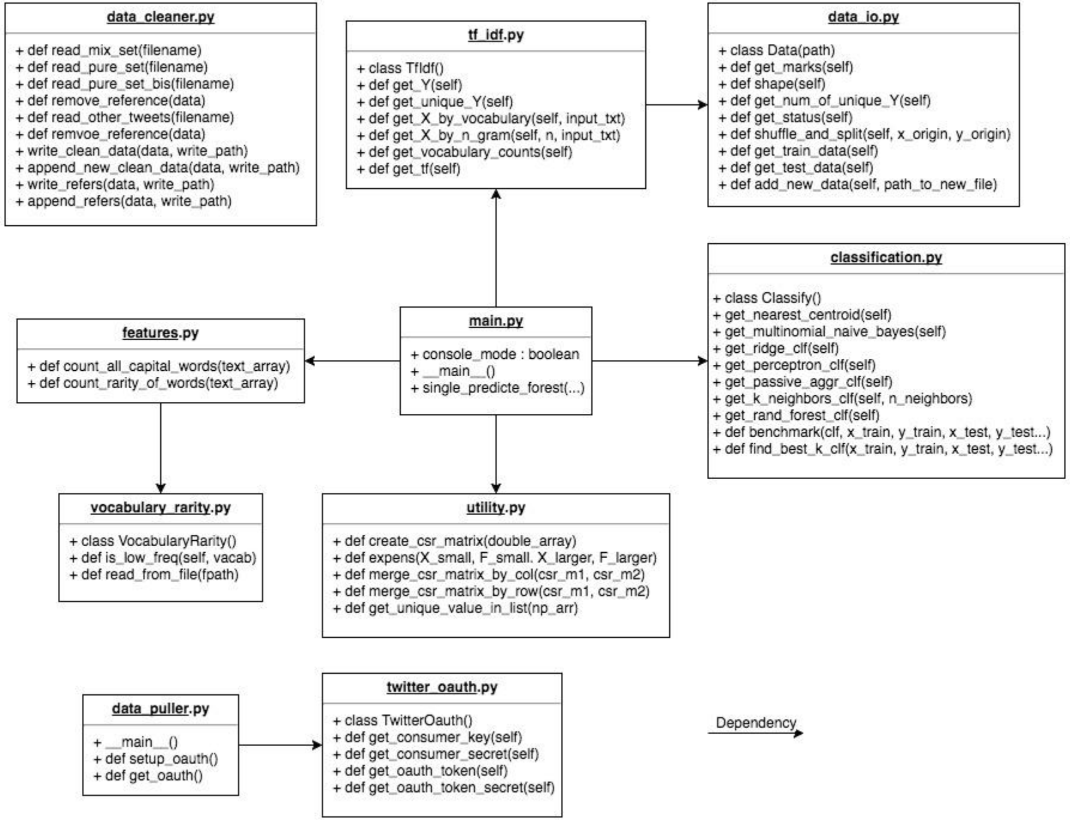
kaggle



Implementations



Modularized Python Code



Data Cleaning

1. Replace the retweets or reference by “ __QUOTE__ ”
2. Replace the links “https:\\” by “ __URL__ ”
3. Replace the breakline “\n” with blank.
4. Save the reference content to the other file for future analysis.

Format of the clean data
[Name Text]

Some exemples
DonaldTrump New national Bloomberg poll just released - thank you! Join the MOVEMENT: __URL__ #TrumpTrain… __URL__
DonaldTrump TONIGHT! NORTH CAROLINA: __URL__ WEDNESDAY! GEORGIA: __URL__ SATURDAY! NEVADA: __URL__
HillaryClinton .@realdonaldtrump: Attacking Muslim Americans is wrong, and it makes it harder for us to defeat terrorism. __URL__
KimKardashian East coast turn to E NOW!!!!! Keeping Up With The Kardashian's is on!!!!!! who's excited?

Parameters Optimisations

Cross validation for parameter n_estimators / neighbors

- Number of partition : 5
- Random Forest from 50 ~350 (jump by 50) K=200 : 80.71%
- K-Nearest-Neighbors from 5~ 55 (jump by 10) K=5: 64.72%

Parameters for other classifiers (Mostly Defaults)

- NearestCentroid()
- MultinomialNB(alpha=.01)
- RidgeClassifier(tol=1e-2, solver="auto")
- Perceptron(max_iter=50, tol=None)
- PassiveAggressiveClassifier(max_iter=50, tol=None)
- KNeighborsClassifier(n_neighbors=5)
- RandomForestClassifier(n_estimators=200)

3003 Features

1. Term frequency of 1500 (from *total: 15488*) 1-gram vocabularies. (Tf-idf + ANOVA variant)
2. Term frequency of 1500 (from *total: 50485*) 2-gram vocabularies. (Tf-idf + ANOVA variant)
3. Term frequency of all capital vocabularies.
4. Term frequency of very difficult 5348 SAT vocabularies. (+ stemmer.stem)
5. Term frequency of medium difficult 2698 SAT vocabularies. (+ stemmer.stem)

pythoning -> python
vocabularies -> vocabulary

Top 100 Important Features Sorted (By Random Forest)

[**Capital Words**] [__url__] [__quote__] [bo] [president obama] [http ofa] [ofa bo] [**obama**] [ofa] [**president**] [**Difficult Vocabs**] [my] [**mythbusters**] [hillary] [trump] [twitter] [twitter com] [__quote__ president] [rt] [ly] [and] [net] [so] [bit] [bit ly] [**Med-difficult Vocabs**] [by] [**richarddawkins**] [is] [love] [net http] [richarddawkins net] [it] [pic] [that] [for] [this] [www] [pic twitter] [http bit] [you] [we] [instagr] [not] [me] [thank you] [**kimkardashian**] [thank] [http instagr] [at] [http www] [http instagram] [great] [am] [but] [lol] [instagram com] [instagram] [trump2016] [will] [if] [religion] [instagr am] [from] [with] [just] [**jamienotweet**] [donald] [be] [up] [guys] [can] [twitpic] [today] [our] [twitpic com] [are] [tonight] [khloekardashian] [was] [all] [as] [jamie] [**makeamericagreatagain**] [awesome] [amp] [donald trump] [http twitpic] [health] [cruz] [what] [now] [here] [romney] [of the] [uk] [an] [__quote__ hillary] [out] [new]

Classifiers & Accuracies

The Worst

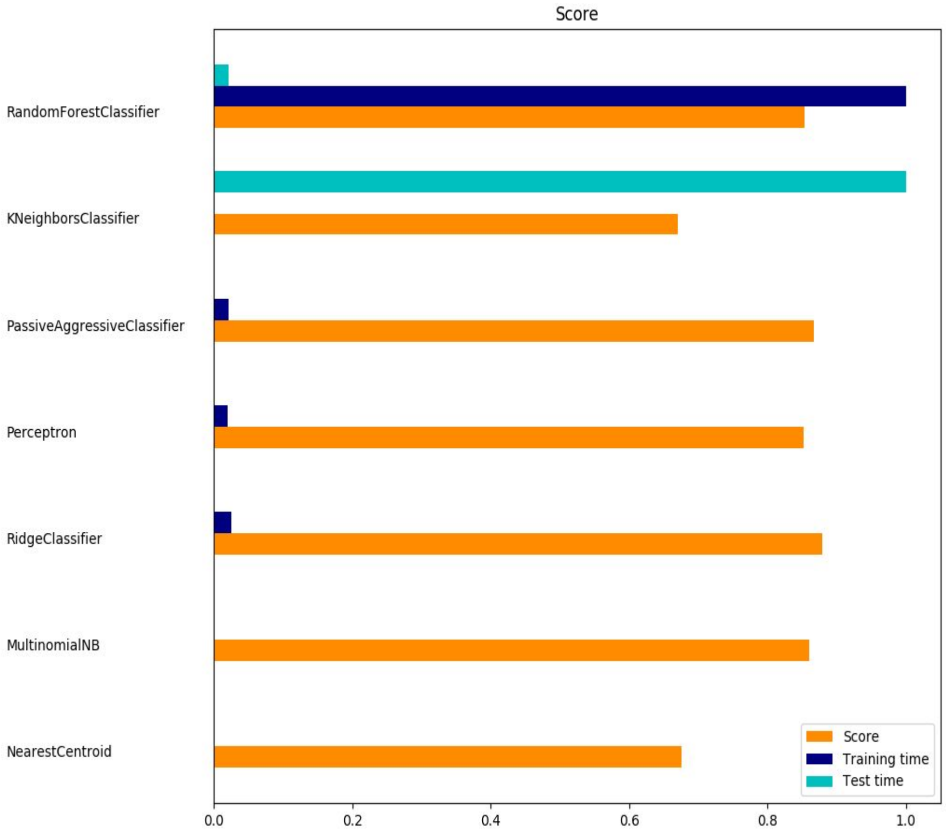
Nearest Centroid Classifier - 68.3%

Top 3 Best

Ridge Classifier - 87.9%

Passive Aggressive - 86.9%

Multinomial NB - 86.4%



Prediction Accuracy Improvement

Change	Improved ?
Add another 1000 3-gram features.	0%
Increase 1-gram and 2-gram features from 1000 to 1500.	+1-2%
From “sum of counts” to “sum of counts/total words count” for all capital words and SAT vocabularies.	+2-3% (+22% for Nearest Centroid)
Use sklearn.feature_selection.f_classif to select the most variant Tf-idf of 1-gram & 2-gram features.	+2%
	+6% (88% Best)

Demo!

Difficulties

1. Transformer the feautres fetched from one user-input tweet to 3003 features from training data.
2. Manipulate the sparse matrices correctly to add new features manually (to merge our new matrices into Tf-idf matrices).
3. To find and understand the best parameters for each classifier.
4. To increase prediction accuracy.

Generalization ?

~/git/ParisVII_ProjectMachineLearning — -bash

```
 [ 203 1082  153]
 [ 217  259  992]]
================================================================================
Training by: MultinomialNB(alpha=0.01, class_prior=None, fit_prior=True)
Training time: 0.020s
Predict time:  0.002s
Accuracy:        0.825
Classification report:
             precision    recall  f1-score   support

        EAP       0.82      0.81      0.82      1989
        HPL       0.84      0.83      0.84      1438
        MWS       0.81      0.84      0.82      1468

avg / total       0.83      0.83      0.83      4895

Confusion matrix:
[[1616  155  218]
 [ 173 1197   68]
 [ 175   67 1226]]
Dimensionality: 3004
Density: 1.000000
================================================================================
Training by: RidgeClassifier(alpha=1.0, class_weight=None, copy_X=True, fit_intercept=True,
        max_iter=None, normalize=False, random_state=None, solver='auto',
        tol=0.01)
Training time: 0.679s
Predict time:  0.002s
Accuracy:        0.827
Classification report:
             precision    recall  f1-score   support

        EAP       0.80      0.86      0.83      1989
        HPL       0.88      0.80      0.84      1438
        MWS       0.83      0.81      0.82      1468

avg / total       0.83      0.83      0.83      4895

Confusion matrix:
[[1707  101  181]
 [ 219 1150   69]
 [ 217   60 1191]]
Dimensionality: 3004
Density: 1.000000
```

Overview Data Kernels Discussion Leaderboard Rules Team My Submissions **Late Submission**

835	▾ 70	**Neural Nut**	0.52629	6	19d
836	▾ 23	**Fabia**	0.52763	2	14d
837	▾ 3	**BoBdec**	0.52990	1	12d
838	▾ 7	**Fisher**	0.53290	7	1mo
839	▴ 107	**Miles Hill**	0.53354	1	2mo
840	▴ 5	**Suresh**	0.53672	1	1mo
841	▾ 13	**EnriqueSantos**	0.54228	8	2mo
842	▴ 11	**Aleksandr Shatilov**	0.54325	7	19d
843	▴ 20	**Quincy**	0.54499	2	4d
844	▴ 20	**haoeric**	0.54559	1	5d
845	▾ 2	**ShahMuzaffarBashir**	0.54620	1	4d
846	▾ 5	**Robert Sobolewski**	0.54784	2	2mo

References

Clinton-trump-tweets.csv
https://www.kaggle.com/benhamner/clinton-trump-tweets/data

Trump-tweets.csv
https://www.kaggle.com/doughersak/donald-trump-tweet-statistics/data

Trump-tweets-bis.csv
https://www.kaggle.com/austinvernsonger/donaldtrumptweets/data

Others' tweets:
https://www.kaggle.com/adhok93/president-obama/data

Tutorial:
http://scikit-learn.org/stable/auto_examples/text/document_classification_20newsgroups.html#sphx-glr-auto-examples-text-document-classification-20newsgroups-py

Spooky Author Identification
https://www.kaggle.com/c/spooky-author-identification

Thanks for your attention !