text_features

December 11, 2019

https://github.com/QuantCS109/TrumpTweets/blob/master/notebooks_features/text_features.ipynb

1 Overview

This notebook uses the 'TextFeaturesGenerator' class (from text_features) to convert textual data into qunatitaive data.

For now, it creates a bag-of-words representation and a tf-idf representation. We will also add SVD/PCA of these matrices and a Word2Vec representation in the next few days.

Will update the TextFeaturesGenerator class on an ongoing basis and update the usage here.

[47]:
```
import sys
sys.path.append('..') #to add top-level to path

from modules.text_features import TextFeaturesGenerator
from modules.project_helper import TweetData
import pandas as pd
import numpy as np
from datetime import timedelta
import datetime
import matplotlib.pyplot as plt
```

Reusing the TweetData class to get cleaned tweets.

[2]:
```
tweet_data = TweetData()
tweet_data.clean_tweets.head()
```

[2]:
```
                                                                tweets  \
timestamp
2019-11-17 19:57:12-06:00  tell jennifer williams whoever that is to read…
2019-11-17 19:56:02-06:00
2019-11-17 19:49:47-06:00  paul krugman of has been wrong about me from t…
2019-11-17 19:47:32-06:00                    schiff is a corrupt politician
2019-11-17 19:30:09-06:00  blew the nasty amp obnoxious chris wallace wil…

                                          timestamp after4_date
timestamp
2019-11-17 19:57:12-06:00 2019-11-17 19:57:12-06:00  2019-11-18
```

```
2019-11-17 19:56:02-06:00 2019-11-17 19:56:02-06:00  2019-11-18
2019-11-17 19:49:47-06:00 2019-11-17 19:49:47-06:00  2019-11-18
2019-11-17 19:47:32-06:00 2019-11-17 19:47:32-06:00  2019-11-18
2019-11-17 19:30:09-06:00 2019-11-17 19:30:09-06:00  2019-11-18
```

2 Daily Tweets

This does the following two things:

1) Change the date of the tweets after 3 PM Chicago time to the following day (as trading closes then)
2) Concatenate all tweets in a given day to one large document

[3]:
```
tweet_data.daily_tweets.head()
```

[3]:
```
                                                   tweets
date
2009-05-05  donald trump will be appearing on the view tom…
2009-05-08  donald trump reads top ten financial tips on l…
2009-05-09  new blog post celebrity apprentice finale and …
2009-05-12  my persona will never be that of a wallflower …
2009-05-13  miss usa tara conner will not be fired ive alw…
```

3 Feature Generator

Creating a 'TextFeaturesGenerator' instance which takes the tweets as an argument

[4]:
```
feature_generator = TextFeaturesGenerator(tweet_data.clean_tweets.tweets)
```

'get_bow_matrix' creates the bag-of-words matrix

[5]:
```
bow_mat = feature_generator.get_bow_matrix()
```

[6]:
```
bow_mat.shape
```

[6]:
```
(28813, 17035)
```

The shape of this matrix is 27.96K rows (same number as the tweets) and the columns are 16,781, which is equal to the unique number of words in the vocabulary.

[7]:
```
bow_mat[:10,:10].todense()
```

[7]:
```
matrix([[0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
```

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]], dtype=int64)
```

As you can see, most of the values are zero which is why it is stored as a 'sparse-matrix'

Bag-of-words is simply a count of words in the tweet. A better representation is 'tf-idf'. The 'get_tfidf_matrix' creates

[8]:
```
tfidf_mat = feature_generator.get_tfidf_matrix()
tfidf_mat.shape
```

[8]:
```
(28813, 17035)
```

The matrices can be saved using the matrices function. You can either specify a 'folder' which will be created and both matrices stored in it, else will store in the working directory.

[9]:
```
feature_generator.save_matrices()
```

The two matrices will be saved with the names "bow_mat.npz" and "tfidf_mat.npz"

You can also specify a folder and a suffix to the file names.

[10]:
```
#feature_generator.save_matrices(folder="../data/intermediate_data/matrices/
↪",suffix="_v2")
```

The files can be loaded using the following commands:

[10]:
```
from scipy import sparse
bow_loaded = sparse.load_npz("../data/intermediate_data/bow_mat.npz")
tfidf_loaded = sparse.load_npz("../data/intermediate_data/tfidf_mat.npz")
print(bow_loaded.shape)
print(tfidf_loaded.shape)
```

```
(28813, 17035)
(28813, 17035)
```

3.1 PCA (through SVD) of the matrices

You can get the SVD of the bow and tfidf matrices as well.

[11]:
```
svd_bow_mat = feature_generator.get_svd_bow_mat()
```

[12]:
```
svd_bow_mat.shape
```

[12]:
```
(28813, 2)
```

By default, it gives back two components. You can changet that using the n_components argument.

[13]:
```
svd_bow_mat = feature_generator.get_svd_bow_mat(n_components=100)
```

[14]:
```
svd_bow_mat.shape
```

[14]:
```
(28813, 100)
```

You can get the SVD of the tf-idf as well.

[15]:
```
svd_tfidf_mat = feature_generator.get_svd_bow_mat(n_components=100)
```

[16]:
```
svd_tfidf_mat.shape
```

[16]:
```
(28813, 100)
```

These matrices can be saved as well.

[17]:
```
feature_generator.save_matrices()
```

You can load them back using np.load

[18]:
```
svd_loaded_mat = np.load('../data/intermediate_data/svd_tfidf_mat.npy')
```

[19]:
```
svd_loaded_mat.shape
```

[19]:
```
(28813, 100)
```

4 Aggregagte SVD per day

[20]:
```
svd_df = pd.DataFrame(svd_loaded_mat)
```

[21]:
```
svd_df['timestamp'] = tweet_data.clean_tweets.index
svd_df['date'] = svd_df.timestamp.dt.date
```

[22]:
```
svd_df.head()
```

[22]:

	0	1	2	3	4	5	6	\
0	3.827242	1.058184	-0.753201	0.539504	0.672026	1.173379	-0.282925	
1	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
2	3.060190	0.759136	0.960683	-0.707494	1.130351	1.936883	-0.004514	
3	0.200777	-0.107046	0.113282	0.877040	-0.034224	0.142449	-0.058900	
4	2.915336	0.145921	0.789791	-0.586309	1.237927	-0.773927	-0.802348	

	7	8	9	…	92	93	94	95	\
0	0.095729	0.447752	-0.022804	…	-0.155376	0.021729	0.156688	0.113619	
1	0.000000	0.000000	0.000000	…	0.000000	0.000000	0.000000	0.000000	
2	-0.147008	-0.626272	-0.132273	…	0.192523	-0.159143	0.302086	-0.142012	
3	0.020884	-0.023058	-0.125923	…	-0.017129	0.015688	0.020679	-0.006425	

```
4 -0.924382 -0.588656 -0.114364  …  0.264748 -0.347075 -0.256552  0.116289

         96        97        98        99                 timestamp  \
0  0.277421 -0.008233 -0.655812 -0.421291 2019-11-17 19:57:12-06:00
1  0.000000  0.000000  0.000000  0.000000 2019-11-17 19:56:02-06:00
2  0.392160 -0.119812  0.029349  0.113289 2019-11-17 19:49:47-06:00
3  0.002942 -0.011792 -0.023175 -0.000544 2019-11-17 19:47:32-06:00
4  0.170073  0.082550 -0.148944 -0.027615 2019-11-17 19:30:09-06:00

         date
0  2019-11-17
1  2019-11-17
2  2019-11-17
3  2019-11-17
4  2019-11-17

[5 rows x 102 columns]
```

[23]:
```
svd_df_daily = svd_df.groupby('date').agg(np.mean)
```

[25]:
```
svd_df_daily.head()
```

[25]:
```
                 0         1         2         3         4         5   \
date
2009-05-04  1.914085 -0.744047 -0.003781 -0.297262  0.104558 -0.762712
2009-05-05  1.728747 -0.735490 -0.032372 -0.510345 -0.136988 -0.583485
2009-05-08  0.656670  0.017658  0.343568 -0.132163 -0.182062 -0.136581
2009-05-12  0.759489 -0.616653 -0.256694 -0.132355  0.892657 -0.322315
2009-05-13  0.549987 -0.714711 -0.650531 -0.033119 -0.053220  0.134759

                 6         7         8         9   …        90        91  \
date                                               …
2009-05-04  0.079807 -0.860134 -0.830335 -0.303307  … -0.165399 -0.067746
2009-05-05 -0.960771 -0.846735 -0.502048 -0.394207  … -0.089814  0.167276
2009-05-08 -0.153954 -0.149953 -0.287268 -0.003448  … -0.033192 -0.007058
2009-05-12 -0.301618 -0.619333  0.309500 -0.410879  … -0.000917  0.080225
2009-05-13  0.064928 -0.091967 -0.262836 -0.087195  … -0.056626  0.169364

                 92        93        94        95        96        97  \
date
2009-05-04  0.004524 -0.051457 -0.022368 -0.035046 -0.117022 -0.018562
2009-05-05 -0.051590 -0.018636  0.033445 -0.031762 -0.109483 -0.002220
2009-05-08 -0.009867 -0.046665 -0.035062  0.011595  0.068346  0.056166
2009-05-12  0.135558 -0.084527 -0.098101  0.224687 -0.164727 -0.113767
2009-05-13 -0.056050 -0.119903  0.075697 -0.110502 -0.074038  0.056684

                 98        99
```

```
date
2009-05-04 -0.053331 -0.008457
2009-05-05 -0.002091  0.053873
2009-05-08  0.000621 -0.068242
2009-05-12  0.053716 -0.041663
2009-05-13  0.024885  0.068076

[5 rows x 100 columns]
```

[26]:
```
svd_df_daily.to_csv('../data/intermediate_data/svd_df_daily.csv')
```

5 4 PM

[27]:
```
tweet_data.clean_tweets['timestamp'] = tweet_data.clean_tweets.index
after_4_tweets = tweet_data.clean_tweets.timestamp.dt.hour >= 15
tweet_data.clean_tweets['after4_date'] = tweet_data.clean_tweets.timestamp.dt.
 ↪date
tweet_data.clean_tweets.loc[after_4_tweets,'after4_date'] = tweet_data.
 ↪clean_tweets.timestamp[after_4_tweets].dt.date + timedelta(days=1)
```

[28]:
```
tweet_data.clean_tweets.head(100)
```

[28]:
```
                                                                tweets  \
timestamp
2019-11-17 19:57:12-06:00  tell jennifer williams whoever that is to read…
2019-11-17 19:56:02-06:00
2019-11-17 19:49:47-06:00  paul krugman of has been wrong about me from t…
2019-11-17 19:47:32-06:00                    schiff is a corrupt politician
2019-11-17 19:30:09-06:00  blew the nasty amp obnoxious chris wallace wil…
…                                                                        …
2019-11-12 11:25:11-06:00  why is such a focus put on nd and rd hand witn…
2019-11-12 03:07:37-06:00           a great try by we are all proud of you
2019-11-12 01:33:57-06:00  vote for sean spicer on dancing with the stars…
2019-11-12 00:57:13-06:00  this isn t about ukraine this isn t about impe…
2019-11-11 23:58:15-06:00  want that to be an impeachable offense good lu…

                                          timestamp after4_date
timestamp
2019-11-17 19:57:12-06:00 2019-11-17 19:57:12-06:00  2019-11-18
2019-11-17 19:56:02-06:00 2019-11-17 19:56:02-06:00  2019-11-18
2019-11-17 19:49:47-06:00 2019-11-17 19:49:47-06:00  2019-11-18
2019-11-17 19:47:32-06:00 2019-11-17 19:47:32-06:00  2019-11-18
2019-11-17 19:30:09-06:00 2019-11-17 19:30:09-06:00  2019-11-18
…                                                 …           …
2019-11-12 11:25:11-06:00 2019-11-12 11:25:11-06:00  2019-11-12
2019-11-12 03:07:37-06:00 2019-11-12 03:07:37-06:00  2019-11-12
```

```
2019-11-12 01:33:57-06:00 2019-11-12 01:33:57-06:00  2019-11-12
2019-11-12 00:57:13-06:00 2019-11-12 00:57:13-06:00  2019-11-12
2019-11-11 23:58:15-06:00 2019-11-11 23:58:15-06:00  2019-11-12

[100 rows x 3 columns]
```

[29]:
```
combined_daily_tweets = tweet_data.clean_tweets.
 ↪groupby('after4_date')['tweets'].apply(lambda x: ' '.join(x))
combined_daily_tweets.head()
```

[29]:
```
after4_date
2009-05-05    donald trump will be appearing on the view tom…
2009-05-08    donald trump reads top ten financial tips on l…
2009-05-09    new blog post celebrity apprentice finale and …
2009-05-12    my persona will never be that of a wallflower …
2009-05-13    miss usa tara conner will not be fired ive alw…
Name: tweets, dtype: object
```

[30]:
```
combined_daily_tweets.to_csv('../data/intermediate_data/combined_daily_tweets.
 ↪csv')
```

```
c:\users\gufra\.virtualenvs\trump_tweets-t_tuxmg9\lib\site-
packages\ipykernel_launcher.py:1: FutureWarning: The signature of
`Series.to_csv` was aligned to that of `DataFrame.to_csv`, and argument 'header'
will change its default value from False to True: please pass an explicit value
to suppress this warning.
  """Entry point for launching an IPython kernel.
```

6 Check if the concatenation is correct

[31]:
```
tweet_data.clean_tweets.tweets[tweet_data.clean_tweets.after4_date==pd.
 ↪to_datetime("2019-10-03")]
```

[31]:
```
timestamp
2019-10-03 13:40:19-05:00    fake news just like the snakes and gators in t…
2019-10-03 12:09:33-05:00      schiff is a lowlife who should resign at least
2019-10-03 11:36:23-05:00    schiff is a lying disaster for our country he …
2019-10-03 11:33:00-05:00     the republican party has never had such support
2019-10-03 11:31:53-05:00    book is doing really well a study in unfairnes…
2019-10-03 11:29:53-05:00                                       thank you hugh
2019-10-03 11:28:49-05:00       a great book by a brilliant author buy it now
2019-10-03 11:22:55-05:00                                   great job richard
2019-10-03 10:52:11-05:00                        keep up the great work kellie
2019-10-03 10:37:33-05:00    the ukraine controversy continues this morning…
2019-10-03 10:00:00-05:00    the u s won a billion award from the world tra…
2019-10-02 23:41:51-05:00                democrats want to steal the election
2019-10-02 23:27:52-05:00    mississippi there is a very important election…
```

```
2019-10-02 23:27:52-05:00    he loves our military and supports our vets de…
2019-10-02 21:06:36-05:00                              look at this photograph
2019-10-02 19:51:56-05:00    schiff house intel chairman got early account …
2019-10-02 15:48:47-05:00    the do nothing democrats should be focused on …
2019-10-02 15:39:07-05:00    adam schiff should only be so lucky to have th…
2019-10-02 15:31:53-05:00    democrats are trying to undo the election rega…
2019-10-02 15:31:03-05:00    nancy pelosi just said that she is interested …
2019-10-02 15:19:09-05:00    all of this impeachment nonsense which is goin…
2019-10-02 15:02:11-05:00    now the press is trying to sell the fact that …
Name: tweets, dtype: object
```

[32]:
```
combined_daily_tweets[combined_daily_tweets.index.values==pd.
↪to_datetime("2019-10-03")]
```

[32]:
```
after4_date
2019-10-03    fake news just like the snakes and gators in t…
Name: tweets, dtype: object
```

7 Create SVD matrix of the combiened 4 PM tweets

[33]:
```
combined_generator = TextFeaturesGenerator(combined_daily_tweets)
```

[34]:
```
n_components = 2
combined_svd_df = pd.DataFrame(combined_generator.
↪get_svd_tfidf_mat(n_components=n_components))
```

[35]:
```
combined_svd_df['after4_date'] = combined_daily_tweets.index.values
```

[49]:
```
combined_svd_df.head()
```

[49]:

	0	1	after4_date
0	0.229959	0.195915	2009-05-05
1	0.052085	0.062540	2009-05-08
2	0.079564	0.035554	2009-05-09
3	0.101352	0.043649	2009-05-12
4	0.068212	0.062037	2009-05-13

[52]:
```
combined_svd_df.to_csv('../data/features/combined_svd_df.csv')
```

8 Scoring Tweets

Use the below parts if you want to train on one set and score on another set (not used currently).

[38]:
```
tweet_data = TweetData()
tweet_data.clean_tweets.head()
```

[38]:
```
                                                                tweets  \
timestamp
2019-11-17 19:57:12-06:00  tell jennifer williams whoever that is to read…
2019-11-17 19:56:02-06:00
2019-11-17 19:49:47-06:00  paul krugman of has been wrong about me from t…
2019-11-17 19:47:32-06:00                    schiff is a corrupt politician
2019-11-17 19:30:09-06:00  blew the nasty amp obnoxious chris wallace wil…

                                          timestamp after4_date
timestamp
2019-11-17 19:57:12-06:00 2019-11-17 19:57:12-06:00  2019-11-18
2019-11-17 19:56:02-06:00 2019-11-17 19:56:02-06:00  2019-11-18
2019-11-17 19:49:47-06:00 2019-11-17 19:49:47-06:00  2019-11-18
2019-11-17 19:47:32-06:00 2019-11-17 19:47:32-06:00  2019-11-18
2019-11-17 19:30:09-06:00 2019-11-17 19:30:09-06:00  2019-11-18
```

[39]:
```
tweet_data.daily_tweets.head()
```

[39]:
```
                                                    tweets
date
2009-05-05  donald trump will be appearing on the view tom…
2009-05-08  donald trump reads top ten financial tips on l…
2009-05-09  new blog post celebrity apprentice finale and …
2009-05-12  my persona will never be that of a wallflower …
2009-05-13  miss usa tara conner will not be fired ive alw…
```

Split into train at test a certain date (in the example, 2018-01-01)

[40]:
```
train_tweets = tweet_data.daily_tweets[tweet_data.daily_tweets.index<=pd.
 ↪to_datetime("2018-01-01")]
score_tweets = tweet_data.daily_tweets[tweet_data.daily_tweets.index>pd.
 ↪to_datetime("2018-01-01")]
```

Create the feature generator class

[41]:
```
feature_generator_with_scores = TextFeaturesGenerator(train_tweets.
 ↪tweets,score_tweets.tweets)
```

[42]:
```
train_svd, test_svd = feature_generator_with_scores.
 ↪get_svd_tfidf_mat(n_components=10)
```

[43]:
```
print(train_svd.shape)
print(test_svd.shape)
```

```
(2395, 10)
(682, 10)
```

Convert to dataframe and add date

[44]:
```
train_svd_df = pd.DataFrame(train_svd)
train_svd_df['date'] = train_tweets.index

train_svd_df = pd.DataFrame(train_svd)
train_svd_df['date'] = train_tweets.index
train_svd_df.head()
```

[44]:

	0	1	2	3	4	5	6	\
0	0.255383	0.094552	0.166693	0.268462	0.084693	0.037153	0.004419	
1	0.060717	0.020085	0.073662	0.057923	0.091849	0.026812	0.024343	
2	0.081151	0.018288	0.060130	0.137244	-0.040755	0.028226	-0.140432	
3	0.108293	0.008944	0.051318	0.012111	0.093586	0.004212	0.044807	
4	0.076052	0.024311	0.064737	0.046367	0.075149	0.016460	0.013030	

	7	8	9	date
0	-0.012995	0.019448	-0.057069	2009-05-05
1	0.005461	0.046865	-0.023403	2009-05-08
2	-0.038887	0.006821	0.016878	2009-05-09
3	0.036510	0.011957	-0.038894	2009-05-12
4	0.021578	0.009604	-0.023408	2009-05-13

[45]:
```
test_svd_df = pd.DataFrame(test_svd)
test_svd_df['date'] = score_tweets.index
test_svd_df.head()
```

[45]:

	0	1	2	3	4	5	6	\
0	0.477176	-0.055520	-0.089123	-0.017706	-0.019953	-0.021424	-0.113065	
1	0.481053	-0.085727	-0.085393	-0.002136	-0.024705	-0.017996	0.014202	
2	0.397138	-0.071503	-0.070591	-0.022799	-0.015327	-0.044193	-0.042444	
3	0.442618	-0.027874	-0.130795	-0.002797	0.000490	-0.038139	-0.065656	
4	0.365602	-0.071861	-0.074386	0.006737	-0.016290	0.081641	-0.026068	

	7	8	9	date
0	0.080983	-0.000404	0.034550	2018-01-02
1	0.071317	-0.013739	0.016359	2018-01-03
2	0.040499	0.005976	0.005290	2018-01-04
3	0.050083	0.072613	0.050317	2018-01-05
4	0.053572	0.048362	0.031814	2018-01-06