Information Retrieval and Data Mining

Stock Market and Twitter

Team Members

This project had been realized by Maud Picq, Xia Cui and Pawat Supaongprapa.

Code Structure

Our project is based on three different pieces of codes in Python:

1. In the first part, we have the codes corresponding to the tweets parsing, some preprocessing, two different sentiment analysis methods (Profile of Mood States and SentiWordNet) as well as a quick code to predict changes using results from Profile of Mood States.
2. In the second part, we have the codes for the Subjectivity-Polarity sentiment analysis technique.
3. In the third part, we have the codes for evaluation using granger causality tests.

Yahoo! Finance API

See https://code.google.com/p/yahoo-finance-managed/wiki/csvHistQuotesDownload for details

To get Boeing's stock data:
http://ichart.yahoo.com/table.csv?s=BA&a=0&b=13&c=2014&d=2&e=6&f=2014&g=d&ignore=.csv
To get Intel's stock data:
http://ichart.yahoo.com/table.csv?s=INTC&a=0&b=13&c=2014&d=2&e=6&f=2014&g=d&ignore=.csv
To get Coke's stock data:
http://ichart.yahoo.com/table.csv?s=COKE&a=0&b=13&c=2014&d=2&e=6&f=2014&g=d&ignore=.csv

Python Code (named Python Code in the GitHub Repository)

Requirements

This piece of code has the following prerequisite in order to be run:
- Python 2.7
- NLTK
- Scikit Learn
- SentiWordNet data file (http://sentiwordnet.isti.cnr.it/download.php), that has to be named SentiWordNet.txt and should be in the same repository.

Configuration

The config files included the paths to all the data files. If the data has already been parsed, it is possible to set REIMPORT to False (in run_projectpyin order to use the previously imported file, as this operation is time consuming.

Two sentiments analysis are implemented. The first one is named POMS, whereas the second one is named Lexicon. It should be notices that Lexicon was not used in this report as it takes too much time to run.

The name of the output file can be defined in the run_project.py, as well as the name of the input file if you have already been importing and parsing some data.

Running the code

In order to run the code to perform the sentiment analysis, just launch the python file run_project.py.

Subjectivity-Polarity (named Subjectivity-Polarity in the GitHub Repository)

Requirements

This piece of code has the following prerequisite in order to be run:
- Python 2.7
- NLTK
- pandas
- IPython
- nltk-trainer (https://github.com/japerk/nltk-trainer)

- Corpora
    - Movie Review Subjectivity Dataset Version 1.0 by Bo Pang and Lillian Lee
    - Movie Review Sentiment Polarity Dataset Version 2.0 by Bo Pang and Lillian Lee
    - Tweet corpus by Sentiment140

Configuration
Make sure that the preprocessed tweets file(s) are in the same directory as the codes.

Running the code

Classification

To use **unigram** with **Movie** Review corpus to classify **subjectivity**
1. Edit line 6 of read_csv_subjectivity.py to point to the preprocessed tweets file.
2. Edit line 44 to specify the output file name.
3. Run the file

To use **unigram** with **Tweet** corpus to classify **subjectivity**
1. Edit line 6 of read_csv_subjectivity - twitter_corpus.py to specify input file name.
2. Edit line 44 to specify output file name
3. Run file.

To use **unigram** with **Movie** Review corpus to classify **polarity**
1. Edit line 6 of read_csv_polarity.py to point to the input file.
2. Edit line 37 to specify the output file name.
4. Run the file

To use **bigram** with **Movie** Review corpus to classify **polarity**

1. Run in the extracted nltk-trainer directory “python train_classifier.py --algorithm NaiveBayes --instances files --fraction 0.75 --filename bigram_movie.pickle --min_score 3 --ngrams 2 --show-most-informative 10 movie_reviews”
2. Edit line 9 of read_csv_polarityBigram.py to point to the input file.
3. Edit line 16 to point to bigram_movie.pickle file outputted from part 1.
4. Edit line 39 to specify the output file name.
5. Run the file

To use **unigram** with **Tweet** corpus to classify **polarity**

1. Edit line 6 of read_csv_polarity - twitter-corpus.py to specify the input file name.
2. Edit line 37 to specify the output file name.
3. Run the file

To use **bigram** with **Tweet** corpus to classify **polarity**

1. Edit line 6 of read_csv_polarity_bigram - twitter-corpus.py to specify the input file name.
2. Edit line 37 to specify the output file name.
3. Run the file

Visualise and calculate daily average sentiments

To visualise and output daily average sentiments

1. **Remove bad rows.** Having performed classification, modify line 7 in remove_badrows.py to the input data file name. Then modify line 33 to specify the output file name and run the script. This step ensures that there is no empty cell in the input data.
2. **Sort dates.** Modify line 5 in sort_dates.py as step 1’s output file name. Modify line 23 to specify the output file name.
3. Copy the output file from step 2 to Python Notebook’s directory.
4. Start IPython Notebook.
5. Import Sentiment Analysis Visualization!.ipynb in Visualise folder.
6. Modify code under Read stock data heading to specify location of file containing a company’s stock prices obtained using Yahoo! Finance API
7. Modify code under Read sentiment data heading to specify file name of the copied output file in step 3.
8. Modify code under Calculate daily average for the sentiments heading (last line) to specify the output file name of the daily average sentiments.
9. Run all by going to Cell > Run all

Evaluation (Named evaluation in Github Repository)

Requirements

This piece of code has the following prerequisite in order to be run:
- Python 2.7
- statsmodel
- numpy
- pandas

Configuration

The input file should be in the same directory of the source code, and the name can be modified in the code or you can rename the filename to dailytweets.csv, sentiment.csv and stockprices.csv. The eva_gct_numoftweets.py takes first two csvs and eva_gct_stockpriceschange.py takes the last two.

Running the Code

To run the code, just run each of two methods, it takes two csvs as an input and will generate a structured dictionary result for the tests.