Sentiment Analysis For Stock Prediction using Tweets

- Aayush Jain

Sentiment Analysis

The process of drawing conceptual meaning and interpret hidden information in a text such as subjective information.

It is the task of determining the emotional value of a given expression in natural language.

It is essentially a multiclass text classification task where the given input text is classified into positive, neutral, or negative sentiment. The number of classes can vary according to the nature of the training dataset.

Why ?

On 6 April 2015, a piece of news about a potential deal that Intel acquiring Altera was published on Twitter. Around this time, 3158 call options in the derivative market were suddenly swept. People believed that a computer program that monitored Twitter bought them within a few seconds. Later that day when the stock price went up,the person behind this computer program exercised the options and made $2.4 million US dollars in 28 minutes.

Elon Musk uses Twitter (Not a real reason lol). But Elon Musk, wrote on Twitter that he was considering taking his company private at $420 per share. He also confirmed that the funding was ”secured”. Since the market price back then was much lower than $420 per share, the Tesla stock went up more than 10% intraday. It was thereafter suspended and U.S. Securities and Exchange Commission started an investigation of manipulating the stockprice against Musk. The market reacted to this news and $23 billion was wiped off share prices for Tesla

Tweets are real time !! News is generally something that has already happened

Behavioral Economics and the Hypothesis we intend to test …

Behavioral Economics tells us that emotions can affect individual behaviour and decision-making.

But can we generalize this notion at large?

Can we consider public sentiment to be a economic indicator ?

Hypothesis: Emotions and moods of individuals affect their decision making process thus, leading to a direct correlation between “public sentiment” and “market sentiment”.

How do you quantify public “mood/sentiment” ?

How do you get the tweets ?

4 Ways :

1. Retrieve from Twitter Public API
2. Find an existing Twitter Dataset
3. Purchase from Twitter
4. Access or purchase from a twitter service provider

I’ll cover the first two in the following slides

A few technical terms

API - (Application Programming Interface)

It allows two systems to communicate with one another. An API essentially provides the language and contract for how two systems interact. Each API has documentation and specifications which determine how information can be transferred.

They use HTTPS requests to get information from a web application/server

Mainly Categorized as 2 Types SOAP and REST

SOAP - XML REST- Use URLs and HTTP verbs (GET, POST, PUT, DELETE)

Continued ….

API endpoint : One end of a communication channel.

For APIs, an endpoint can include a URL of a server or service. Each endpoint is the location from which APIs can access the resources they need to carry out their function.

APIs work using ‘requests’ and ‘responses’. When an API requests information from a web application or web server, it will receive a response. The place that APIs send requests and where the resource lives, is called an endpoint.

Twitter API v2

Preliminary Steps:

1. Sign up for the developer account (it’s free)
2. Create a new project using your developer account and get the keys
3. We’ll use the bearer token to access the API(for authorization)
4. We’ll get the headers which we’ll use to access the API
5. Next we’ll build a request with the endpoint we want to use and any parameters we want to pass

NOTE:- The endpoints will be different for different levels of access. Twitter allows extracting only past 7 days of data for “Essential” access i.e for general purposes but for academic researchers they have allowed access to the entire archive(from the first tweet)

Building A Request

Endpoint : `tweets/search/recent`

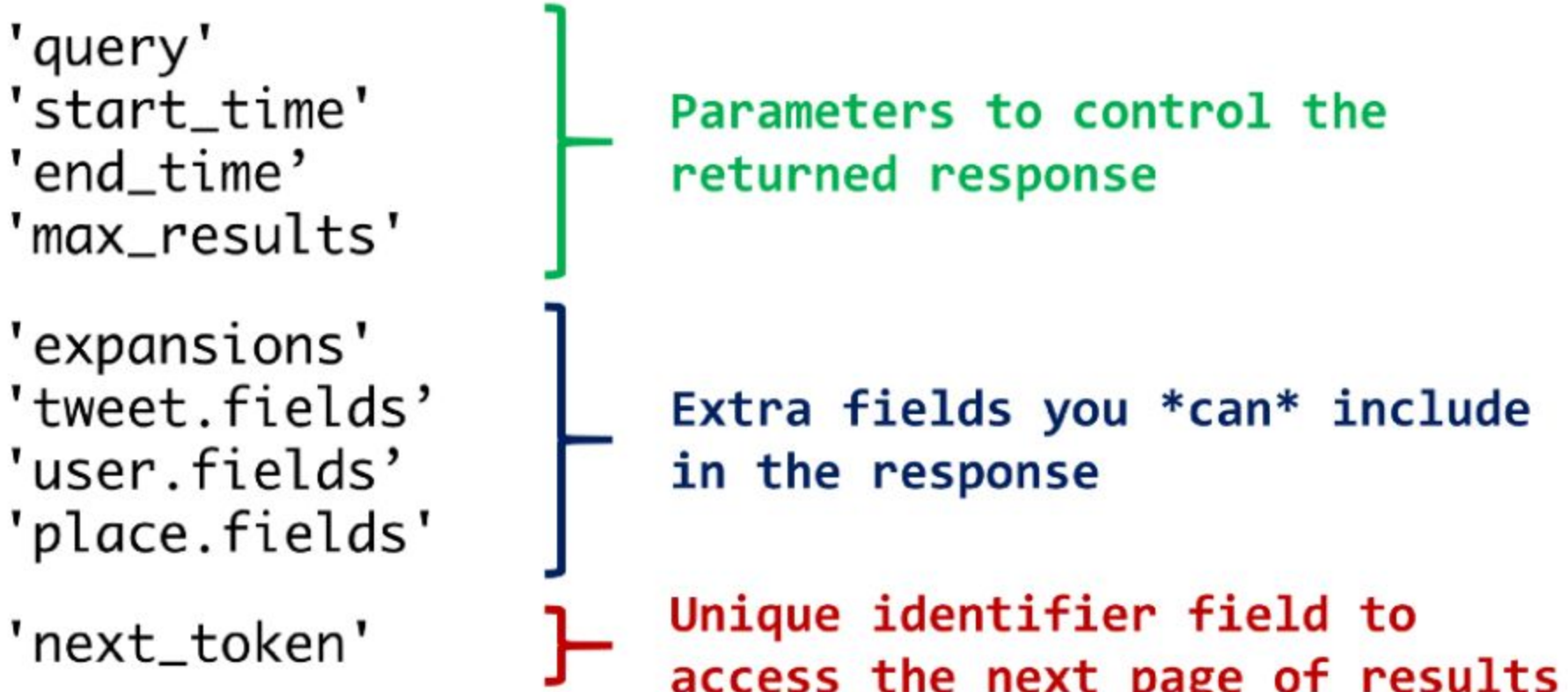
Search_URL: "https://api.twitter.com/2/tweets/search/recent" (Link of the endpoint we want to access)

Query_Params: Parameters offered by the endpoint used to customize our search request (https://developer.twitter.com/en/docs/twitter-api/tweets/search/api-reference/get-tweets-search-recent)

Ex:- You can use query operators to specify keywords to look for in tweets, the language of the tweet, the interval between which you want the tweets to be, from some specific location, user , the max results you want to be returned etc.

next_token() : Currently search results are restricted to a certain number per request. But what if you want more results for your search ? Twitter's got you covered

They return a unique token that you can use in your next request which will give you new results. You can keep looping until there is no next_token()


'query'
'start_time'
'end_time'
'max_results'
Parameters to control the returned response
'expansions'
'tweet.fields'
'user.fields'
'place.fields'
Extra fields you *can* include in the response
'next_token'
Unique identifier field to access the next page of results

Requests Library

Now that we have the Bearer token, Search URL, Query_Params we can put this all together and send a “GET” request and get a response using the requests library

response = requests.request("GET", url, headers = headers, params = params)

(Requests Module)

We can convert the response object i.e in the “JSON” format to CSV format using

```
df = pd.DataFrame(response['data'])
df.to_csv('data.csv')
```

Entire Process : Implementation

NOTE : If you prefer GUI you can use Postman

Existing Datasets

There are some limitations to using the API. If you don't have Academic Research access you can only get tweets from past 7 days (which is often not enough)

To overcome this you can access the public datasets created by individuals and organizations like these :

1) https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/PDI7IN
2) https://github.com/crisiscomputing/tbcov

But these only offer the tweet_ids (acc to Twitter Developer's Policy you can't share the real tweets only the ids) so you need to perform "hydrating" to extract the tweets

You can also use web applications like TweetSets to extract a particular subset from an existing dataset using keywords, hashtags etc.

Hydrator : https://github.com/DocNow/hydrator

DJIA (Dow Jones Industrial Average)

A index that tracks 30 large publicly owned blue-chip companies trading on the NYSE and NASDAQ

Blue-Chip ? (Derived from Poker - Of Most Value)

Some of the Blue Chip Companies

Bollen et al - Twitter Mood Predicts the Stock Market

- Analyze daily twitter feeds using two mood tracking tools Opinion Finder and GPOMS measuring mood of the public.
- Cross validate the resulting mood time series by ability to detect public's response towards presidential election and thanksgiving day in 2008
- Use Granger Causality analysis to validate the hypothesis
- Self Organizing Fuzzy Neural Network to predict the stock prices
- Achieve 87.6 % accuracy
- Show that accommodating certain mood states significantly increases prediction accuracy of DJIA values and reduces MAPE by > 6%

Pre- Processing the Tweets

Steps:

1) Remove acronyms, emoticons, unnecessary data like pictures, URL's
2) Tokenization : Form a list of words for each tweet
3) Stopword Removal : Words not expressing emotion are called stopwords ex :- a, is, the
4) Regex Matching for special characters : Often tweets contain (#)hashtags, @(mentioning other people) . They are replaced subtly.

Ex:- #Microsoft -> Microsoft, @BillGates -> USER, hiiiiiiiiii ! -> hi !

Pre Processing the DJIA Data

Steps:

1. Since Twitter Data is available daily and DJIA data is not (obviously the market is closed on Weekends and Holidays) we need to approximate the values for the missing days. (Ex:- Use Concave functions)
2. Adjust for sudden jumps and falls in the data caused due to unpredictable events by shifting up/down for steep falls/jumps, respectively; making sure that we do not disturb the daily directional trend
3. Prune periods of high volatility that are difficult to predict

Tools used to measure change in Public mood from tweets

1. Opinion Finder

Analyses the text content of the tweets to provide a positive vs negative daily time series of the public mood. It uses a lexicon containing positive and negative words and a tweet is classified on the basis of the ratio.

2. GPOMS

Analyses text content of tweets to generate 6-dimensional daily time series of public mood providing a more detailed view of the public mood.The 6 dimensions are Calm, Alert, Sure, Kind, Vital ,Happy.

GPOMS - continued

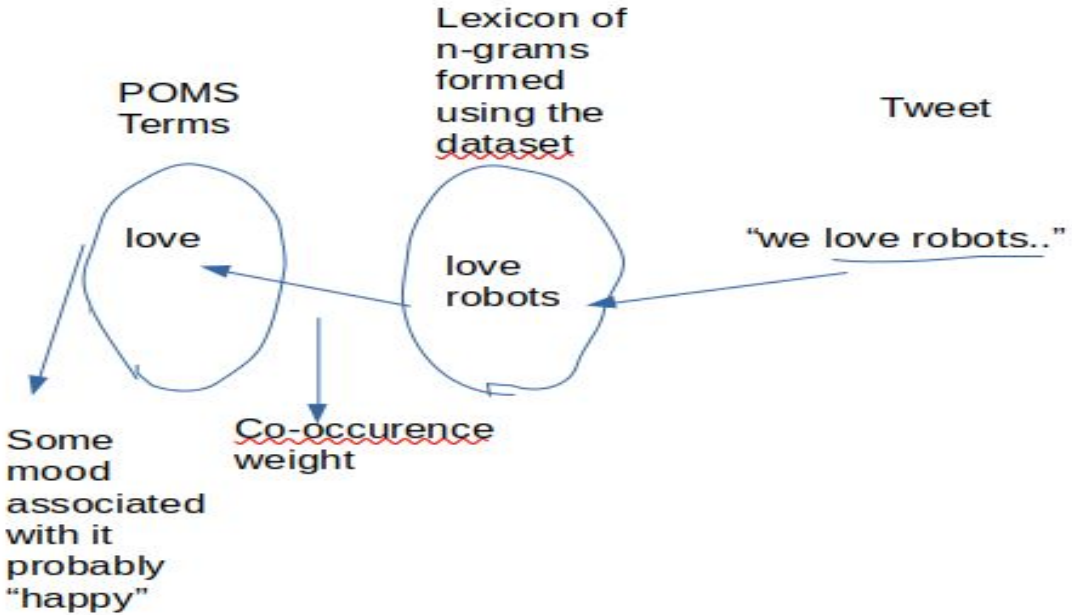
It uses the POMS questionnaire that contains 37-65 questions and evaluates the mood by the degree of responses given. Different degrees have different scores.

Google created datasets containing counts for n-grams extracted from about 1 Trillion words of English Web Text.

Expanded the original 72 terms of the questionnaire to form a lexicon of 964 associated terms by analyzing word co-occurrences.

We match terms in tweets with this lexicon of n-grams which then maps back to the original POMS terms with some weight

https://www.topendsports.com/psychology/poms.htm



So like this we match the different terms with n-grams that map back to some original word having a mood wrt to it. We then take a weighted sum of these moods (based on the co-occurrence weights) to classify the tweet.

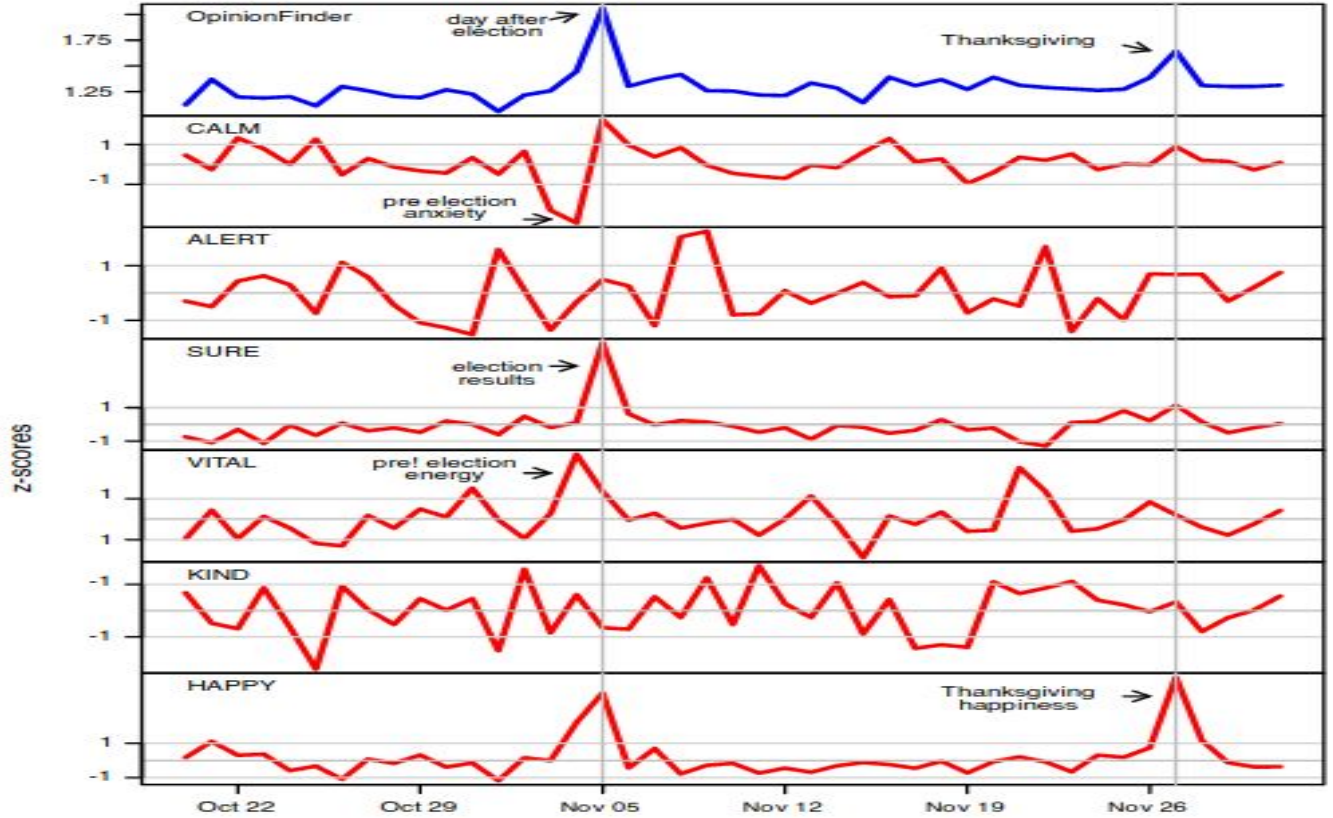


Fig. 2. Tracking public mood states from tweets posted between October 2008 to December 2008 shows public responses to presidential election and thanksgiving.

Granger Causality

If a variable X causes Y then changes in X will systematically occur before changes in Y i.e if X granger-causes Y then it should contain information that helps predict Y

Test 2 linear models of DJIA time series : With and without OF+GPOMS

Null Hypothesis : Mood time series do not predict DJIA values with high level of confidence

Results : Null hypothesis is rejected for “Calm” mood dimension which had highest granger causality with DJIA values

Other dimensions do not have significant causal relations with the changes in stock market and neither does Opinion Finder

How do you determine “significance” ?

The null hypothesis is said to be rejected if p-value < significance level

In practice significance level is taken to be = 0.05

So if, p < 0.05 -> Reject Null Hypothesis -> Alternative Hypothesis is true

Lag	OF	Calm	Alert	Sure	Vital	Kind	Happy
1 day	**0.085***	0.272	0.952	0.648	0.120	0.848	0.388
2 days	0.268	**0.013****	0.973	0.811	0.369	0.991	0.7061
3 days	0.436	**0.022****	0.981	0.349	0.418	0.991	0.723
4 days	0.218	**0.030****	0.998	0.415	0.475	0.989	0.750
5 days	0.300	**0.036****	0.989	0.544	0.553	0.996	0.173
6 days	0.446	**0.065***	0.996	0.691	0.682	0.994	**0.081***
7 days	0.620	0.157	0.999	0.381	0.713	0.999	0.150

(p-value < 0.05: $\star\star$, p-value < 0.1: $\star$)

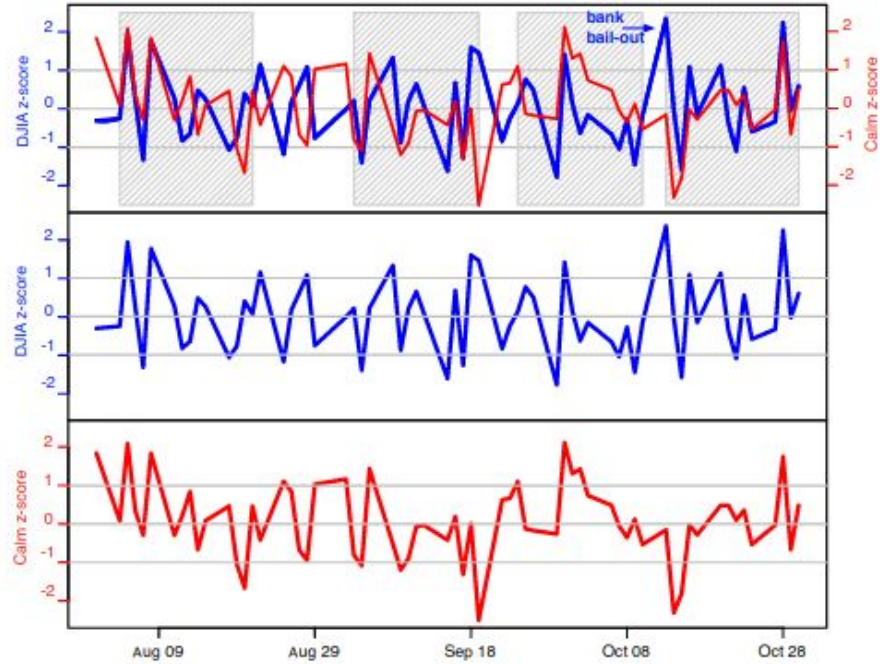


Fig. 3. A panel of three graphs. The top graph shows the overlap of the day-to-day difference of DJIA values (blue: $\mathbb{Z}_{D_t}$) with the GPOMS' Calm time series (red: $\mathbb{Z}_{X_t}$) that has been lagged by 3 days. Where the two graphs overlap the Calm time series predict changes in the DJIA closing values that occur 3 days later. Areas of significant congruence are marked by gray areas. The middle and bottom graphs show the separate DJIA and GPOMS' Calm time series.

Linear Fit ? How to model non-linearity -> SOFNN

Granger Causality suggests a predictive relation between certain mood dimension and DJIA values but it’s based on linear regression.

To model the nonlinear relationship between public mood and stock market values use a Self-Organizing Fuzzy Neural Network

Inputs

1) Past 3 days DJIA values
2) 1) + Mood time series

Null Hypothesis : Public mood measurement does not improve predictive models of DJIA values

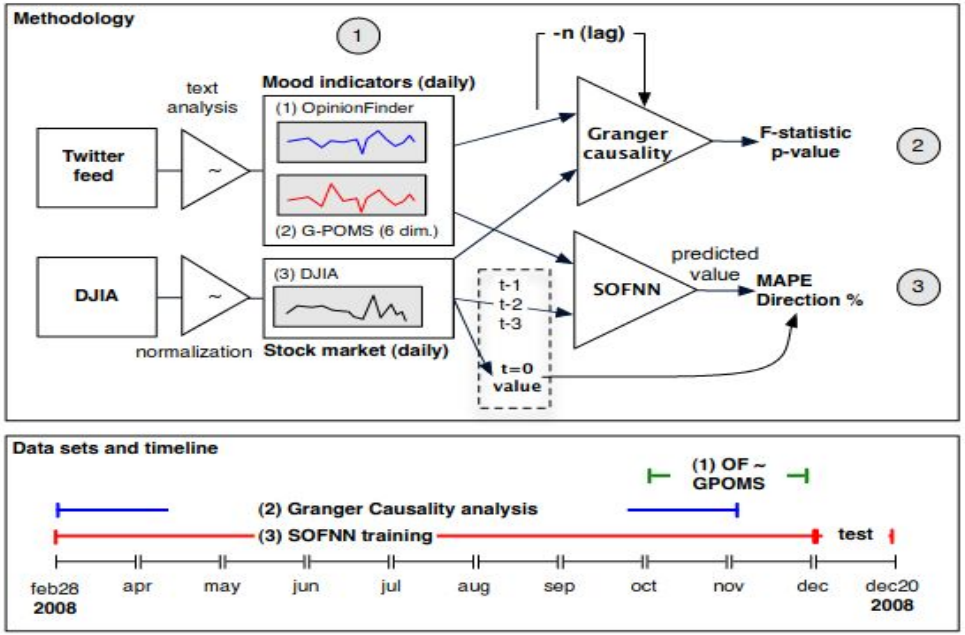


Fig. 1. Diagram outlining 3 phases of methodology and corresponding data sets: (1) creation and validation of OpinionFinder and GPOMS public mood time series from October 2008 to December 2008 (Presidential Election and Thanksgiving), (2) use of Granger causality analysis to determine correlation between DJIA, OpinionFinder and GPOMS public mood from August 2008 to December 2008, and (3) training of a Self-Organizing Fuzzy Neural Network to predict DJIA values on the basis of various combinations of past DJIA values and OF and GPOMS public mood data from March 2008 to December 2008.

SOFNN

Fuzzy neural network (FNN) systems possess the advantages of both fuzzy systems and neural networks.

They bring the low-level learning and computational power of neural networks into fuzzy systems and provide the high-level human-like IF-THEN rule thinking and reasoning of fuzzy systems into neural networks.

https://sci-hub.se/10.1109/codit.2016.7593551

https://www.google.com/search?channel=fs&client=ubuntu&q=fuzzy+neural+networks

https://sci-hub.se/10.1016/j.neucom.2014.09.079

The Self Organizing Fuzzy Neural Network (SOFNN) is a five layer fuzzy neural network which uses ellipsoidal basis function (EBF) neurons consisting of a center vector and a width vector. We implemented the online algorithm for creating SOFNNs in which neurons are added or pruned from the existing network as new samples arrive.

Conclusion

Using SOFNN try different Combinations of Moods to determine if moods other than Calm (predicted by GC) provide some predictive information

-> Happy + Calm improves predictions Even though Granger causality suggests otherwise. This means that there is a non-linear relationship between diff mood states

MAPE ~ 1.79% Direction Acc -> 80%

87.6% accuracy in predicting the daily up and down changes in the closing values of the DJIA and a reduction of the Mean Average Percentage Error by more than 6%.

Mittal et al

- Tested different ML models like SVMs, Logistic Re, Linear Re but SOFNN with Calm + Happy + DJIA is best
- Introduce Technique for cross validation
- k-fold sequential cross validation(k-SCV). In this method, we train on all days upto a specific day and test for the next k days. The direct k-fold cross validation method is not applicable in this context as the stock data is actually a time series unlike other scenarios where the data is available as a set. Therefore, it is meaningless to analyze past stock data after training on future values
- Portfolio Management using naive greedy-based strategy.
- https://cs229.stanford.edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis.pdf

Scope of improvement

- Better Feature Engineering rather than just analyzing texts. Many other tweet attributes like Tweet Volumes, Likes, Retweets, Followers, Favourite count etc. can be also utilized apart from text

https://www.researchgate.net/publication/262314292_Trading_on_Twitter_The_Financial_Information_Content_of_Emotion_in_Social_Media

This is a benchmark paper that analyzes the impact of the no.of followers of users and correlates it to how fast is the visible change in the stock market.

Results are quite intuitive: They find that tweets of more influential people with more followers have a strong impact on Intraday returns but people with lesser followers have a stronger impact on 10-Day returns

- Better Sentiment Analysis using ML models

https://arxiv.org/pdf/1610.09225.pdf

They develop a Sentiment analyzer. Use Word2Vec and n-gram representation for better feature extraction and train a classifier using Random Forest Algorithm to classify as positive, neutral , negative. Address the issue of Corpus based analyzers trained on not related data Ex:- An analyzer trained on psychological data being used to predict sentiment of stock related text data

Recent Developments

- Use Tools like Tweepy to extract tweets using pagination
- Textblob and Vader for sentiment analysis
- Recent DL models like RNN and LSTMs
- Combine News, Tweets and Technical indicators

https://arxiv.org/pdf/2105.01402.pdf

This thesis reports an 11% improvement in MSE on combining different tweet attributes and technical analysis.

VADER and TextBlob (Rule Based Analyzers)

- Prev methods only consider text but there are a lot of ways to express sentiment apart from text.
- Emoticons, Abbreviations, Slang etc.

TextBlob

1) Polarity -> A value between [-1, 1]

2) Subjectivity -> A value between [0, 1]

(How personal the tweet is)

Vader

1) Positive ([0, 1])

2) Negative([0, 1])

3) Neutral([0, 1])

4) Compound(Normalized Sum)

Vader : http://comp.social.gatech.edu/papers/icwsm14.vader.hutto.pdf
TextBlob : https://textblob.readthedocs.io/en/dev/

Implementation

- Used Tweepy v 4.4.0 i.e compatible with Twitter API v2
- Pagination to extract past 7 days tweets about Microsoft
- Downloaded stock data from Yahoo Finance
- Missing values of stock are replaced with next working day values
- Preprocessing Tweets as earlier
- Sentiment of all the tweets per day are averaged and combined with the daily stock data
- Lookback of 60 days
- Used 2 LSTM models stacked on one another
- Evaluated Sentiment using both Vader and TextBlob
- Vader : MAPE = 6.8440
- TextBlob : MAPE = 11.1401

LSTM + Tweet Extraction + Vader

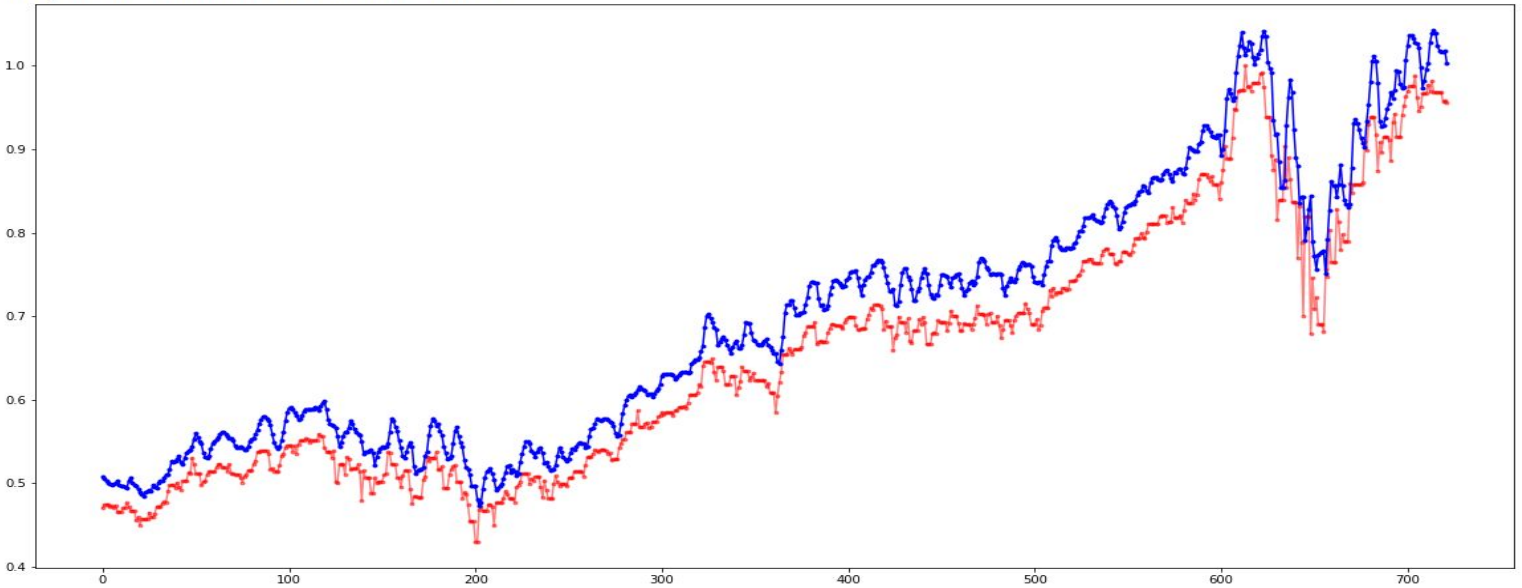
LSTM + Tweet Extraction + TextBlob

Vader

```
[24] y_pred = model.predict(x_test)

     plt.figure(figsize=(20,10))
     plt.plot( y_test, '.-', color='red', label='Real values', alpha=0.5)
     plt.plot( y_pred, '.-', color='blue', label='Predicted values', alpha=1)
```

```
[<matplotlib.lines.Line2D at 0x7f0b8e33c6d0>]
```



TextBlob

```
[84] y_pred = model.predict(x_test)

     plt.figure(figsize=(20,10))
     plt.plot( y_test, '.-', color='red', label='Real values', alpha=0.5)
     plt.plot( y_pred, '.-', color='blue', label='Predicted values', alpha=1)
```

```
[<matplotlib.lines.Line2D at 0x7fd935f93f90>]
```


thank you
شكراً جزيلاً
ngiyabonga
рахмат
danke
謝謝
teşekkür ederim
Баярлалаа
mersi
kia ora
barka
welalin
tack
spas
tapadh leat
mahalo
спасибо
faafetai lava
vinaka
blagodaram
dank je
misaotra
matondo
paldies
grazzi
спасибі
kiitos
dankie
nanni
nandri
dhanyavad
gracias
хвала
asante
manana
obrigada
tenki
bedankt
bayarlalaa
gràçie
hvala
mauruuru
köszönöm
dankon
ačiū
akun
enkosi
dziękuję
mochchakkeram
djiere dieuf
tau
murakoze
дякую
mamnun
chnorakaloutioun
gratias ago
gràcies
sulpáy
taiku
go raibh maith agat
chokrane
sobodi
dėkuji
obrigado
sagolun
sukriya
kop khun krap
arigatô
takk
dakujem
trugarez
mèsi
didi madloba
ありがとう
najis tuke
tanemirt
grazie
terima kasih
rahmet
shukriya
merce
мерси
kam sah hamnida
rahmat
diolch
dhanyavadagalu
তোমাকে ধন্যবাদ
감사합니다
xièxie
ευχαριστώ
merci