Identifying Hyped Stocks on Reddit Using Natural Language Processing

Malene Hansen Sebastian Harvej

May 2021

Abstract

Identifying hyped stocks on Reddit requires understanding and processing of huge data sets with text. If it can be done successfully, it can be compared to the stock market and ultimately predict stock prices. To detect hyped stocks it is necessary to collect mentions and determine their context. This paper attempts to accomplish that by applying Natural Language Processing.

Contents

1 Introduction

In recent times a new trend has appeared on the stock market, where the price is no longer necessarily connected to earnings or potential of a company. The phenomenon is known as Meme stocks: Stocks who's value primarily comes from hype on social media [9]. This was well illustrated by the GameStop stock that recently skyrocketed from $ 20 to $ 347 in less then a month [11] for no apparent reason. Other than people on Reddit talking it up. [1]

If people writing about a stocks on Reddit can affect stock prices it rises the question of whether it is measurable. This paper aims to figure out whether it is possible to identify hyped stocks on Reddit and to explore how it can be done, using natural language processing. To determine whether a stock is being hyped, it is crucial to identify the parameters to search for. The first step is therefore to define "hype".

1.1 Defining hype

Before attempting to identify a hyped stock it is necessary to reflect on the definition of "hype".

According to the Cambridge Dictionary the meaning of hype is:

> "A situation in which something is advertised and discussed ...
> a lot in order to attract everyone's interest ..." [2]

To identify a hyped stock it therefore seems reasonable to observe the quantity of mentions. And also to pay attention to the emotional tone connected, to distinguish advertising contributions. Which can be achieved with Natural Language Processing.

1.2 Natural Language Processing

Identifying and analysing stock mentions, written in comments on Reddit, necessitates interpreting of human language. In practice that can be done using natural language processing (NLP).

NLP is an umbrella term concerning the practice of making computers understand natural human language. There are a lot of language concepts that are making it difficult for computers to understand human language. Things like irony, tone and slang can alter the meaning of a text.

2 Identifying Stock Mentions

Stocks are mentioned in different formats: Typically by the organization name e.g. GameStop or GameStop Corp. or by their ticker symbol, e.g. GME or $GME.

It becomes complicated as we discover that there exist ticker symbols that can be confused with commonly used words like for example ONE, GO or AI.

To capture stock mentions by searching through posts and comments, looking for a list of words, is therefore inadequate. The words must be understood in their context. Which is the core use case of Named Entity Recognition.

2.1 Named Entity Recognition (NER)

Named Entity Recognition (NER) is a form of Natural Language Processing (NPL). NER is the task of detecting and classifying "real-world" objects, like for example people, places or organizations, from a body of text. [10]

NER relies on trigger words: Words that help understand the meaning of an entity. For example would "had lunch at" in the sentence "he had lunch at The New Yorker" indicate that "The New Yorker" is a restaurant, and not a citizen of New York. [**ner2**]

2.2 NER using python and spaCy

spaCy is an open-source library for Python that offers various pre-trained pipeline packages for Named Entity Recognition.

Here is a demonstration of how to use it:

Installation

```
pip install spacy

python -m spacy download en_core_web_sm
```

`en_core_web_sm` is an english model trained on blogs, newz and comments provided by spaCy. [3]

```
# Import libraries
import spacy

# Load model
nlp = spacy.load('en_core_web_sm')
```

`nlp()` loads the trained NER pipeline that we have downloaded.

Extracting entities

```
text = 'Apple is looking at buying U.K. startup for $1 billion'

doc = nlp(text)

# Print Label and Entity Name
for ent in doc.ents:
    print(ent.label_,': ', ent.text)
```

`nlp(text)` processes our text body and returns a spaCy Document Object. The Document Object holds the entities that spaCy has identified within the text and can be accessed by the property `.text` and their label by the property `.label_`. This is the output from our sample text:

```
ORG :  Apple
GPE :  U.K.
MONEY :  $1 billion
```

Visualizing entities

```
from spacy import displacy
displacy.render(doc, style='ent')
```

Discovering labels

The definition of the labels can be programmatically with `spacy.explain()` :

```
ORG = spacy.explain('ORG')
ORG
```

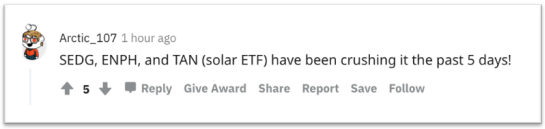
```
'Companies, agencies, institutions, etc.'
```

2.3 NER on subreddit dataset

On a test data set, consisting of 145.000 posts and comments collected from the subreddit `https://www.reddit.com/r/investing` [4] we can loop through each comment to identify and extract entities with the ORG label:

```
# Creating a list of ORG labeled entities
org_list = []
for text in data_set:
    doc = nlp(text)
    for entity in doc.ents:
        if entity.label_ == 'ORG':
        org_list.append(entity.text)
```

Taking out samples revealed that ticker names are some times not identified; like in this comment, where the ticker name SEDG is ignored:



SEDG, ENPH ORG , and TAN ORG (solar ETF) have been crushing it the past 5 days DATE !

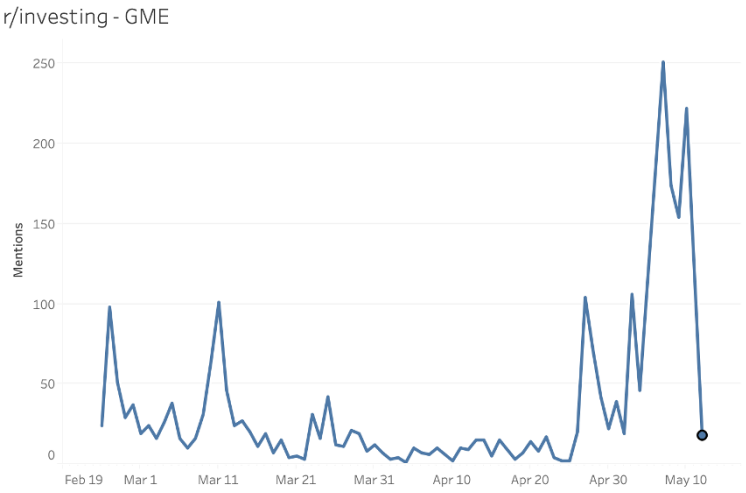
To obtain an overview over the identified entities, we count the discovered entities respectively and print out the top 20:

```
# Print 20 most mentioned ORGs
from collections import Counter
org_freq = Counter(org_list)
org_freq.most_common(20)
```

```
[('gme', 7540),
 ('amazon', 6785),
 ('apple', 6540),
 ('tesla', 4960),
 ('ford', 2965),
 ('intel', 2810),
 ('microsoft', 2720),
 ('amd', 2500),
 ('sony', 2500),
 ('vanguard', 2456),
 ('spy', 2430),
 ('aapl', 2365),
 ('btc', 2335),
 ('pe', 2215),
 ('pltr', 2140),
 ('ebay', 2140),
 ('s&p', 1976),
 ('gm', 1925),
 ('spotify', 1860),
 ('ai', 1760)]
```

Entities like "vanguard" and "sp" do not represent single stocks and therefore needs to be ruled out. And entities like "Apple" and "aapl" refer to the same stock. But overall there is a solid basis for further analyse on classified submissions.

When extracting dates and counting the comments, a line chart can be created. Like the line chart below, representing mentions of the GameStop Stock on the subreddit "investing". It looks like something that could be compared to a stock chart:



3 Sentiment analysis

The quantity of mentions in it self does not necessarily mean that a stock is hyped. The mentions must also be of a positive sentiment. The sentiment of a text can be determined with the use of sentiment analysis. Sentiment analysis is the practice of predicting a sentiment using machine leaning. In this case the most basic form is used with only two possible outcomes; positive or negative. But sentiment analysis can also be used to detect neutral sentiment or even things like mood (sad, happy or angry). It all depends on the available data sets, and what makes sense in the context of the system that is being developed.

3.1 Data gathering

To train any machine leaning model data is needed. It can be data that one have collected or data found on the internet. In this example review data from IMDb, Amazon and Yelp is used. The data we use can be found here [5]. The data set consists of 2748 review texts with a sentiment value 0 for negative and 1 for positive. The ratio of positive and negative reviews are split evenly 50/50. In the table below you can see an example from each of the 3 data sets.

Review	Sentiment
I advise EVERYONE DO NOT BE FOOLED!	0
The Songs Were The Best And The Muppets Were So Hilarious.	1
Not tasty and the texture was just nasty.	0

3.2 Data preparation

To use the data it has to be prepared for training. The code displayed below is the code used here to clean the data. First the sentence is converted into a spaCy doc, which is a sequence of tokens where the tokens are the individual words. Then the collection is iterated over and the token is then converted into its base form, unless it is a pronoun which do not have any base form. A check is then made to know if the word is a punctuation or stop word. Stop words are a collection of some of the most commonly used words and they do not give much value.

```
def text_data_cleaning(sentence):
    doc = nlp(sentence)

    tokens = []
    for token in doc:
        if token.lemma_ != "-PRON-":
            temp = token.lemma_.lower().strip()
        else:
            temp = token.lower()
        tokens.append(temp)
    cleaned_tokens = []
    for token in tokens:
        if token not in stopwords and token not in punct:
            cleaned_tokens.append(token)
    return cleaned_tokens
```

3.3 Training the model

The data is then split into training and test data. 80 % are used for training and the remaining 20 % for testing. The model is then being fitted with the data, so it is able to predict the sentiment outcome.

3.4 Testing of the model

Testing of a model like this is done by comparing the predicted values of the test data set, with the correct result in the data set. The comparison can then be visualized in a classification report. A classification report visualizes how many percent of the calculated values are right.

	precision	recall	f1-score	support
0	0.82	0.83	0.82	262
1	0.84	0.83	0.84	288
accuracy			0.83	550
macro avg	0.83	0.83	0.83	550
weighted avg	0.83	0.83	0.83	550

Figure 1: classification report results.

As the figure shows, the results of the test was okay: Humans are agreeing on sentiment 80-85 % of the time [8], so this is a good baseline to aim for. But it can also become necessary to look at the purpose of the task and its critically to determine if the result is good enough in specific cases. In this case, it is considered that since the focus is not on the individual messages but rather on a large amounts of data the results are accepted.

3.5 Testing with reddit data

The model can then be tested with real data from reddit. Testing on real data can give completely different results than with test data. This result is very important because it will give an idea about what result should be expected when it is used for its intended purpose.

This can be done by performing an analysis manually on parts of the Reddit data and then comparing them with the models' result on the same data. In this case a small sample is being used; 5 positive and 5 negative. This is a relatively small sample and it is probably not sufficient to show the entire picture but it clearly shows that there is an issue analysing the reddit lingo.

When comparing the result of the models result with the ones predetermined sentiment score, the result is not very good. The model only predicts right in 60 % of the cases, which is only a small improvement compared to if it was just guessing randomly.

There can be multiple reasons for such a big error ratio. It could be because the test samples are too small or that the model needs more training to be able to predict it correctly. But in this case the main issue is most likely that the training data and the Reddit data are not closely enough related to each other.

4 Conclusion

By defining hype, as something that is "advertised and discussed a lot"[2], it is possible to identify hyped stocks by measuring sentiment and quantity of their mentions. Natural Language Processing can be applied to break down and interpret the text retrieved from Reddit: Mentions can be successfully collected using Named Entity Recognition and the sentiment be analysed using Sentiment Analysis.

For a more accurate result on discovering stock mentions, spaCy's preprocessing token handling can be modified. [6] Adding regex expressions, looking for 2-4 letter words initiated by a dollar sign and 2-4 letters words written in block capital, would probably capture the tickers that are currently missed. For the sentiment analysis to assist the mention count, in determining hype, it must be trained to take the specialized slang in to account. Words as "Rocket" and "Diamond hands" are not obviously standing out, but are known on the subreddit "wallstreetbets" to be very promoting in favor of a stock. [7] Emojis are heavily used as well, and might be benificial to take into account.

For further investigation taking the sentiment of comments, not mentioning stocks but replying to such, into account, would definately contribute to additional nuance.

References

[1] URL: https://www.cnbc.com/2021/01/25/gamestop-shares-skyrocket-but-experts-say-it-cant-last.html. (accessed: 27.05.2021).

[2] URL: https://dictionary.cambridge.org/dictionary/english/hype.

[3] URL: https://spacy.io/models. (accessed: 27.05.2021).

[4] URL: https://github.com/PBASOFT/Data-Science-Project/tree/main/data/r_investing_sample.

[5] URL: https://github.com/laxmimerit/NLP-Tutorial-8---Sentiment-Classification-using-SpaCy-for-IMDB-and-Amazon-Review-Dataset/tree/master/datasets.

[6] URL: https://spacy.io/usage/rule-based-matching. (accessed: 25.05.2021).

[7] URL: https://www.cnet.com/personal-finance/investing/gamestop-stock-surge-lingo-heres-what-reddits-wallstreetbets-vocabulary-means/. (accessed: 26.05.2021).

[8] Paul Barba. *Sentiment Accuracy: Explaining the Baseline and How to Test It*. URL: https://www.lexalytics.com/lexablog/sentiment-accuracy-baseline-testing. (accessed: 06.05.2021).

[9] Erin Gobler. *What Is a Meme Stock?* URL: https://www.thebalance.com/what-is-a-meme-stock-5118074. (accessed: 05.05.2021).

[10] Christopher Marshall. *What is named entity recognition (NER) and how can I use it?* URL: https://medium.com/mysuperai/what-is-named-entity-recognition-ner-and-how-can-i-use-it-2b68cf6f545d. (accessed: 04.05.2021).

[11] *yahoo finance*. URL: https://finance.yahoo.com/quote/GME/chart/. (accessed: 05.05.2021).