Who drives the market?
Sentiment analysis of financial news posted on Reddit and the Financial Times

Bachelor’s Thesis by Michael Lubitz

January 23, 2018
Overview

- Motivation
- Data Sources
  - Reddit
  - The Financial Times
  - Data Retrieval
- Experimental Set-up
- Sentiment Analysis
  - Using a dictionary
  - Using machine learning
- Evaluation
Motivation / Goals

- Researchers found out that the sentiment of financial news articles have a certain power to predict falling or rising stock indices\(^1\) as they are the main source for investors\(^2\).

- The sentiment of social media posts (especially on twitter) has also been proven to have a even higher accuracy of predictions\(^3\).

- Reddit combines both worlds.

---

\(^1\) R. P. Schumaker and H. Chen, »Textual analysis of stock market prediction using breaking financial news«
\(^2\) X. Li, H. Xie, L. Chen, J. Wang, and X. Deng, »News impact on stock price return via sentiment analysis«
\(^3\) J. Bollen and H. Mao, »Twitter mood as a stock market predictor«
Task

- Creation of the Reddit dataset
  - Posts and news articles
- Creation of the Financial Times dataset
- Find suitable models for our predictions
- The task of this thesis was to compare the accuracy of predictions based on Reddit to the results of a classic news paper analysis
»The front page of the internet«

- Founded 2005
- Social News Aggregator
  - Each registered user can share links with the community (submission)
  - Other users can comment and vote on the relevance and importance of a submission
Subreddits 1/2

/reddit
/r/economics
/r/politics
/r/science
/r/movies
Subreddits 2/2

Economics subreddit
https://reddit.com/r/economics
Many different news from many different sources
- Most articles of trustworthy newspapers
- The quality of a source is essential for good predictions
- A news text has to be business related
  - We have chosen the subreddit /r/economics
  - The moderators of this subreddit ensure that there are sources related to this topic only
Why Reddit? 2/2

- Reddit Community decides what news appear on Reddit
- Decides position and visibility of submissions
- Moderators
- Votes and Comments
- Submissions
- Removal of non-relevant news for this topic

- Most Reddit users contribute to a single subreddit only
- We assume they are familiar with the respective topic

---

1 C. Buntain and J. Golbeck, »Identifying social roles in reddit using network structure«
Stock Index: S&P 500

- One of the most important stock indices worldwide
- Grounded on 500 companies listed on American stock exchanges
- According to Reddit 54% of its visitors come from the United States

Source: finance.yahoo.com
Framework

The Financial Times

Reddit
- Submissions
- Votes
- Comments

Extract news texts → Preprocessing → Sentiment → Weighting → Evaluation → S&P 500
Data Retrieval

- **Reddit:**
  - 
  - Extraction of all submissions
  - Follow links and extract news texts
  - Link 1
  - Link 2
  - ...
  - Link n

- **Financial Times:**
  - ft.com news archive
  - Extraction of news texts
  - Database

- Python crawler
- Time period: January 2008 to July 2017
Preprocessing 1/3

1. Remove special characters from a news text (from now on called document $d$)

2. Remove common stop words (e.g. and, the, for)

3. Convert text to a list of words (bag of words) / a feature vector

- Problem: The natural language has different word forms and linked words which is not respected by list of words and feature vectors
Noun phrases:

- We used OpinionFinder to extract nouns from the documents.
- Idea: Only use nouns to reduce noise (especially machine learning).

Stemming:

- Not used in combination with machine learning.
- Stemming reduces words to their stem (get rid of different word forms).
- We used the Porter 2 stemming algorithm (Python package stemming) on documents and dictionary.
Doc2Vec:

- Used in combination of machine learning
- Problem with bag of words approach: Words are usually related or linked to other words in a text
- Doc2Vec tries to find such related words and stores them as vectors
  - Example: Paris ←→ France; Paris ←/→ desert
- Python package gensim
Labeling with dictionaries

- Each word in the dictionary is labeled with either **positive** or **negative**, or in a numeric representation +1 or -1

- For each document \( d \) we count the number of positive words \( p \) and negative words \( n \)

- Then, we calculate a sentiment score \( s_d \) as follows:

\[
s_d = \frac{p - n}{p + n} \quad ; \quad -1 \leq s_d \leq 1
\]

- If \( s_d < 0 \) we consider document \( d \) as negative and otherwise
Labeling with machine learning

- Naive Bayes and Random Forests work with probabilistic models to classify documents either as positive or negative.
- Support Vector Machines (SVM) is a large margin classifier which uses a decision boundary to separate the documents into positive and negative.

- Training set:
  - Collection of already labeled financial news texts
  - The sentiment score will be either -1 or +1
Weighting

- Weighting the sentiment value $s_d$ of a document with the
  1. Number of votes $v$
  2. Number of comments $c$
  3. Number of votes and comments
- Multiply the sentiment value with $\max\{\log(v+1), v_{\min}\}$
or / and $\max\{\log(c+1), c_{\min}\}$
where $v_{\min}$ and $c_{\min}$ are minimum weights (found through optimization)
Scoring

- Not in combination with machine learning
- Use of **BM25** scoring to put emphasis on the relevance of words
- We calculate the **BM25** score for each word in a document \( d \) that also appears in the dictionary
- Determine the sentiment of \( d \) (positive or negative) as before and replace the sentiment score with the **BM25** score (positive or negative)
Prediction

- Group all documents by its publishing date
- Sum up all sentiment scores of a day and normalize it by the number of documents on that day

Prediction:

\[ \text{Sum of sentiment scores} \]

\[ \begin{align*}
\text{Predict rising/steady} & : \geq 0 \\
\text{Predict falling} & : < 0
\end{align*} \]
### Results 1/2

- **Baseline (guessing based on majority class):** 54.20%

<table>
<thead>
<tr>
<th>Scoring</th>
<th>Financial Times</th>
<th>Unweighted</th>
<th>Votes weighted</th>
<th>Comments weighted</th>
<th>Votes and comments weighted</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bag of words</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>None</td>
<td>54.19</td>
<td>54.59</td>
<td>54.68</td>
<td>54.80</td>
<td>54.76</td>
</tr>
<tr>
<td>BM25</td>
<td>54.83</td>
<td>54.76</td>
<td>54.30</td>
<td>54.26</td>
<td>53.84</td>
</tr>
<tr>
<td><strong>Stems</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>None</td>
<td>54.20</td>
<td>54.39</td>
<td>54.47</td>
<td>54.43</td>
<td>54.55</td>
</tr>
<tr>
<td>BM25</td>
<td>54.20</td>
<td>54.30</td>
<td>54.30</td>
<td>54.30</td>
<td>53.84</td>
</tr>
<tr>
<td><strong>Nouns</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>None</td>
<td>54.24</td>
<td>54.45</td>
<td>55.45</td>
<td>55.56</td>
<td>55.56</td>
</tr>
<tr>
<td>BM25</td>
<td>54.02</td>
<td>54.83</td>
<td>54.41</td>
<td>54.52</td>
<td>54.11</td>
</tr>
</tbody>
</table>

---

*Sentiment Analysis based on a dictionary*
Results 2/2

- Baseline (guessing based on majority class): **54.20%**

<table>
<thead>
<tr>
<th>Feature representation</th>
<th>Financial Times</th>
<th>Unweighted</th>
<th>Votes weighted</th>
<th>Comments weighted</th>
<th>Votes and comments weighted</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Naive Bayes</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bag of words</td>
<td>54.20</td>
<td>54.85</td>
<td>54.43</td>
<td>54.43</td>
<td>54.72</td>
</tr>
<tr>
<td>Nouns</td>
<td>54.24</td>
<td>55.24</td>
<td><strong>55.45</strong></td>
<td><strong>56.39</strong></td>
<td><strong>56.49</strong></td>
</tr>
<tr>
<td>Doc2Vec</td>
<td>45.80</td>
<td>46.27</td>
<td>46.48</td>
<td>46.23</td>
<td>46.31</td>
</tr>
<tr>
<td><strong>Random Forest</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bag of words</td>
<td>54.20</td>
<td>54.51</td>
<td>54.42</td>
<td>54.43</td>
<td>54.43</td>
</tr>
<tr>
<td>Nouns</td>
<td>54.24</td>
<td><strong>55.45</strong></td>
<td><strong>55.45</strong></td>
<td>55.45</td>
<td>55.45</td>
</tr>
<tr>
<td>Doc2Vec</td>
<td>47.89</td>
<td>50.50</td>
<td>51.13</td>
<td>51.93</td>
<td>51.63</td>
</tr>
<tr>
<td><strong>Support Vector Machines</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bag of words</td>
<td>54.20</td>
<td>54.85</td>
<td>55.02</td>
<td>54.89</td>
<td>54.97</td>
</tr>
<tr>
<td>Nouns</td>
<td>54.24</td>
<td>55.42</td>
<td><strong>55.45</strong></td>
<td>55.45</td>
<td>55.56</td>
</tr>
<tr>
<td>Doc2Vec</td>
<td>54.20</td>
<td>53.86</td>
<td>53.77</td>
<td>54.07</td>
<td>53.52</td>
</tr>
</tbody>
</table>

Reddit

Sentiment Analysis based on a machine learning
Conclusion

‣ Almost all of our results outperform baseline and Financial Times

‣ Therefore we state that Reddit has a certain power to predict stock index changes

‣ Generally the use of votes and comments for weighting purposes is reasonable