Prediction of changes in the stock market using twitter and sentiment analysis

Iulian Vlad Serban, David Sierra González, and Xuyang Wu
University College London

***Abstract*— Twitter is an online social networking and microblogging service with over 200m monthly active users. Given this massive user base researchers have tried to mine the derived vast source of data for different purposes. In this work, we investigate the relationship between the market indicators for three companies (IBM, Intel and General Electric) and the volume of tweets mentioning their names or stock symbols. We consider additionally other factors, such as the predicted sentiment of the tweets, the number of followers/friends of the users and the presence of links on the tweets. With all this information a predictor is trained for each company to estimate the changes in the stock market price. An exhaustive feature selection procedure was performed, showing that the most correlated features with the stock market indicators were the number of tweets weighted by the number of friends. After selecting the four most correlated Twitter related features, and together with the stock market indicators at previous timesteps, six different approaches were studied as predictive models, namely, linear regression considering only the tweet counts, linear regression including sentiment features, non-linear regression considering higher-order interactions between the sentiment-based features and the stock market indicators, and the LASSO regularized versions of the three models. All models performed consistently better than two benchmark models (constant and random prediction) for the three stocks, according to the mean absolute error and mean squared error metrics. This confirms the existance of predictive power in the Twitter features. However, no significative difference was observed between the models using sentiment features and those considering only the tweet counts.**

I. Introduction

Twitter is a free microblogging service founded in 2006 by Jack Dorsey and Biz Stone. It enables users to send and read *tweets*, which are text messages limited to 140 characters. Registered users of Twitter are able to read and post tweets via the web, SMS or mobile applications. The user base of Twitter surpassed the 200 million active users in December 2012 [1].

With such an impressive user base researchers have become interested in mining Twitter data to extract patterns and trends. Understanding how and why people tweet seems like a reasonable first step. Twitter is currently being used for *daily chatter*, *conversations*, *sharing information/URLs*, and *reporting news*; and its users can be classified into the groups such as *information sources*, *friends*, and *information seekers* [2].

Already in 2004 there was an study correlating web buzz and stock market [3]. In this work, Antweiler and Frank analyse how Internet stock message boards are related to stock markets. They conclude with the thought that there is financially relevant information present. In 2006, blog sentiment was used to predict movie sales [4].

More recent research suggests that online social media (blogs, Twitter feeds, etc.) can predict changes in various economic and commercial indicators [5]. In particular, the mood of the tweets, when classified into the mood dimensions *Calm*, *Alert*, *Sure*, *Vital*, *Kind* and *Happy*, have been shown to be significantly correlated with the Dow Jones Industrial Average index (DJIA). In [6], Bollen et al. show that sentiment analysis of Twitter posts over a period of 5 months is correlated with fluctuations in macro-social and -economic indicators in the same time period. A similar approach is taken in [7], where the positive and negative mood of tweets on Twitter is analysed and compared with stock market indices such as Dow Jones, S&P 500, and NASDAQ over a period of 5 months. They found that the number of positive tweets is much higher than that of negative ones, more than double on average. However, the mood indicators (both positive and negative) proved to be always negatively correlated with DJIA, NASDAQ and S&P500.

In 2012, Mao et al. investigated the correlations between the number of tweets that mention S&P 500 stocks and the stock indicators [8]. They applied a simple linear regression model with the tweet counts as exogenous input (independent variable) to predict the stock market indicators. Testing the model on a short period of 17 days, they reported an accuracy of 68% in predicting the direction of change in the daily closing price at stock market level.

In this work, we will address the following research questions:

- How should we analyse and interpret the sentiment of thousands of emotional tweets?
- What is the intrinsic relationship between emotional tweets and stock market?
- What class of models can we expect to perform well on such stock price and trading volume prediction across several stocks?
- What metrics are suitable for evaluating a social media based model for stock market prediction?

The project workflow we have established is shown in figure 1. Initially, the dataset is parsed and processed. As the next step, we analyse the sentiment of each English tweet related to the selected companies. We use the sentiments and related information from the tweets to extract a set of features,

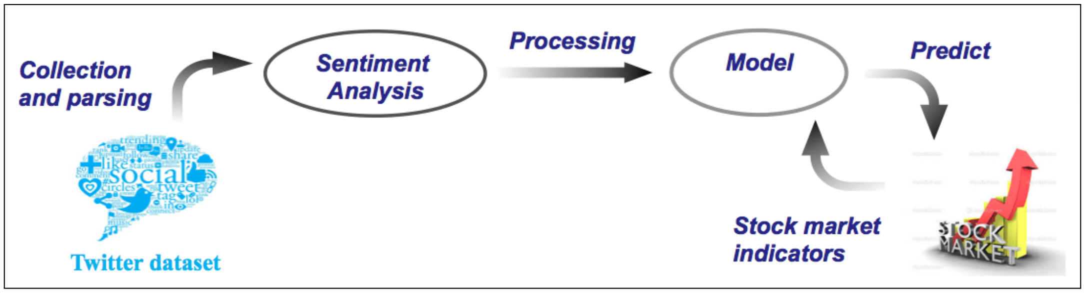

Fig. 1. Project workflow

which we may combine with stock market indicators to build models for predicting the stock prices and trading volume. Finally, we use the models to predict the price and volume changes over a fixed test period of time, which allows us to evaluate and compare their performance.

In our work we take a step-by-step incremental approach to extracting features and building models, such that we are able to isolate any additional predictive information from Twitter from the inherent stock market information. We will extract features which we have reason to believe are relevant for prediction, based on previous research, and construct models which have been proposed in the literature for similar problems. In particular, we will build models involving various levels of Twitter information while keeping the stock market features constant.

Software-wise, we have used Java with Eclipse to parse the data and extract any relevant features from the Twitter dataset. Matlab was then used to carry out the feature analysis, model implementation, and evaluation. All source code is distributed together with this report.

II. Materials and methods

A. Data description

From the Twitter dataset provided for this project we have considered the data for the following three companies: IBM, Intel (INTC), and General Electric (GE). For each of them we have processed all the tweets mentioning the name of the company between January 13th 2013 and March 3rd 2013, accounting for a total of 50 days of data.

The corresponding stock indicators for each of the three companies were obtained from Yahoo! Finance [9] as listed on the NASDAQ Stock Market. The market data is not available on weekends or festivities but we still have Twitter data for those days. Since the dataset was already too small (only 50 days) we used an autoregressive model to extrapolate the market stock data in the missing days. This synthetic data is used only for training. Testing and evaluation is performed only on true market information.

B. Feature extraction

The raw Twitter data contains a lot of information for each tweet, most of which is not relevant for the purpose of this work. Following the results from [10], where a study was conducted to weight the different factors that make information on Twitter credible, we limit the features we extract to certain properties of each tweet. In particular, the study found that tweets tend to be more credible when they cite external sources, i.e. when they provide an URL with the information they are propagating, and when they are retweeted many times. Tweets also tend to be more credible if they are sent by users with many friends and followers. This motivates us to weight tweets differently according to the number of friends and followers of the person, and according to whether or not they contain URLs. We choose not to use the number of retweets to limit our scope, and because it is not clear how this count is affected by the partial size of our data set (which is truncated w.r.t. both time and users).

With this in mind, we have extracted the following preliminary components for tweets in English language only:

- **Tweet creation time**. Used to know the date on which the tweet was posted.
- **Content of the tweet**. The actual content of each tweet, it contains the company name and/or company stock symbol.
- **Number of friends**. Number of friends of the user who posted the tweet.
- **Number of followers**. Number of followers of the user who posted the tweet.
- **Contains URL**. True if tweet contains an URL.

Depending on the content of the tweet it will be assigned to one of several categories. First, we analyse whether the tweet contains the stock name of the company ($IBM, $INTC or $GE) or the company name. Secondly, we consider if the tweet contains URLs in the text. Finally, we convert the tweet text to lower case letters, strip it of any URLs and perform sentiment analysis.

C. Sentiment analysis

For the sentiment analysis of the text of the tweets we have tried two different *off-the-shelf* platforms: Stanford's Deeply Moving [11] and LingPipe [12].

Stanford's Deeply Moving is a Deep Learning model based on a Recursive Neural Network that builds on top of grammatical structures. It builds up a representation of whole sentences based on the sentence structure and computes the sentiment based on how words compose the meaning of longer phrases. It was trained on the dataset Stanford Sentiment Treebank [11].

LingPipe is a toolkit for processing text using computational linguistics. LingPipe is used to do tasks like: find the names of people, organizations or locations in news, automatically classify Twitter search results into categories and suggest correct spellings of queries [12]. The method we use for sentiment classification is the *DynamicLMClassifier* and is described in detail by Pang et al [13].

In this project, we started off using Stanford's platform to predict the sentiment of tweets. However, we encountered two problems. Firstly, we obtained a disproportionate amount of negative tweets. Even if these sentiment labellings were correct by some definition, they would not work well for our application as they do not discriminate between tweets. When we then analysed the sentiment labels together with the actual text of the tweets, we found that the platform was indeed highly inaccurate. Secondly, the running time to predict the sentiment was extremely long, taking on average between 2 and 3 seconds for each tweet.

We then moved on to try LingPipe, using a classifier trained on random tweets written in English which can distinguish between positive, negative and neutral tweets with a reported accuracy of approximately 75% [14]. The data on which this classifier was trained contains 5513 hand-classified tweets from different topics, such as @apple, #google and #microsoft [15]. While testing Lingpipe we observed that the sentiment labellings were more uniformly distributed between neutral, positive and negative, while taking the neutral category for most of the tweets as we would expect. Comparing the predicted labels with the actual tweet texts, we again found that many tweets were misclassified. Nevertheless, we speculated that having the classifier trained on text from actual tweets would improve performance. For these reasons we choose to use LingPipe.

In conclusion, the disproportionate amount of negative sentiments returned by Stanford's Deeply Moving and its heavier computational demands are the main reason why we chose to use LingPipe for this project. Lingpipe's classifier was trained using random tweets written in English and returned more sensible results.

To compare both platforms we present the sentiment prediction counts over a period of 10 days (13/1 to 23/1). From each day 1000 random tweets were selected from which we only processed the English language tweets. Table I shows the total counts for Stanford's Deeply Moving and for the different companies. Likewise, Table II presents the results obtained with LingPipe for the same tweets.

Sentiment	IBM	Intel	GE
Positive	90	52	0
Neutral	243	185	27
Negative	2975	2202	172

TABLE I

STANFORD'S DEEPLY MOVING TOTAL RESULTS

The charts in figures 2 and 3 show the total counts of the different sentiment categories across the 10 days for Standford's platform and LingPipe, respectively.

Sentiment	IBM	Intel	GE
Positive	388	658	47
Neutral	2585	1329	133
Negative	335	452	19

TABLE II

LINGPIPE TOTAL RESULTS

D. *Sentiment features*

As argued earlier the number of friends and followers, as well as the existence of URLs, are important in weighting the credibility of the tweet. However, it is not clear how we should weight each of these factors. We hypothesize that the weight of friends and followers could be either linear or logarithic. Linear weighting would imply, for example, that twice as many friends will make a tweet twice as credible. Logarithmic weighting would imply, for example, that more friends correspond to more credible tweets, but that each additional friend only adds a decreasing marginal credibility. This would be the case if users having more than a certain number of friends, say a thousand friends, were all equally credible.

To keep the maximum amount of information we will consider the linear and logarithmic weightings, as well as no credibility weighting which would correspond to weighting each tweet with a constant. This yields $5 \times 2 \times 2 \times 3 = 60$ features in total, which we may partition according to:

- Weighting factor: constant, linearly in number of friends, log2(number of friends + 1), linearly in number of followers, log2(number of followers + 1)
- URL: contains URL, does not contain URL
- Tweet type: mentions stock symbol, mentions company name
- Sentiment: negative, neutral and positive

We used the logarithm in base 2, which would be equivalent to having a credibility difference of one between two people tweeting about the same stock where one has twice as many friends as the other.

In our implementation the tree has been folded out to one long vector from the top. To keep it simple we kept an additional copy of the constantly weighted tweets, which is why we have 72 features on some plots. That is, features $48 - 60$ are the same as features $1 - 12$.

E. *Feature selection*

To compare with the previous paper by Mao et al. [8] we performed a correlation analysis for each stock market features. Table III presents the correlation between the raw tweet counts and each of the stock market features. To analyse this visually, we can plot the stock features together with the tweet counts. This is done in figure 4 for Intel's volume traded and daily price change indicators. It seems that the volume traded correlates positively with the number of tweets, whereas the correlation in the case of the daily

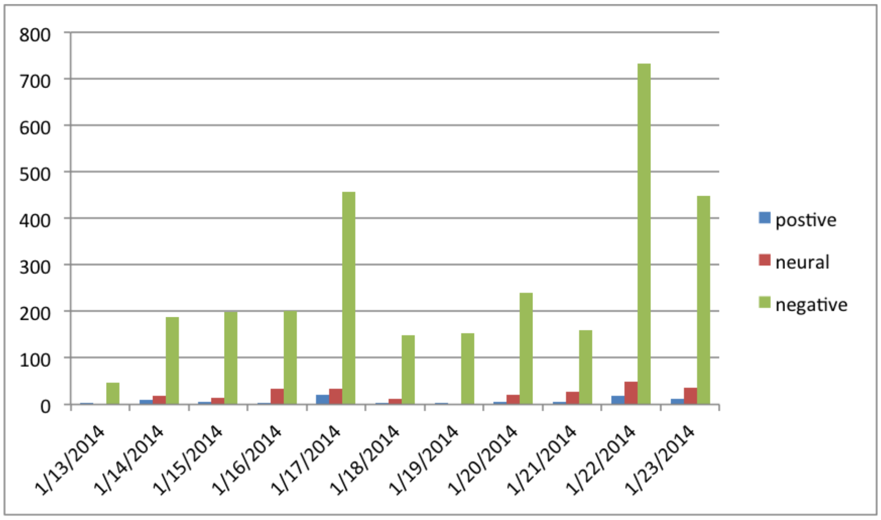

Fig. 2. Tweet sentiment counts for different days with Stanford's sentiment analysis platform

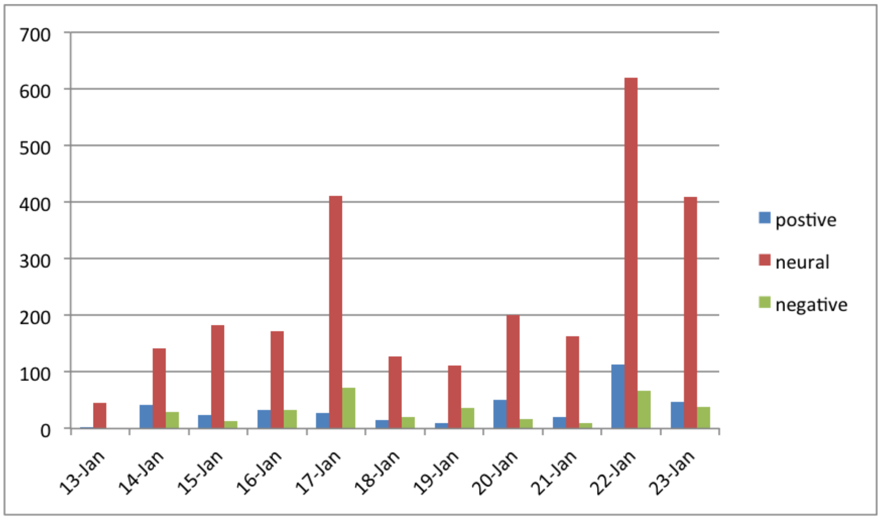

Fig. 3. Tweet sentiment counts for different days with LingPipe

price change is negative. This confirms the results obtained numerically.

We hypothesized that the three companies share a subset of relevant features. All three companies are publicly-traded stocks, technology-based, etc. Therefore they must share a set of relevant features. This is our background knowledge. To find a suitable subset of features, we performed a correlation analysis using the in-built Matlab function *corrcoeff* on all Twitter features w.r.t. all stock features (trading volume, closing price, price change, abs price change). This was done only on the training data. Figures 5 and 6 show an example of how the correlation test was performed for two of the market stock indicators of IBM, Volume traded and Price Change, respectively. Only a single coefficient appeared to be significant w.r.t. closing price, which we choose to discard as it may simply have been due to noise in the data. We then took the union of all the features which had a significant correlation coefficient w.r.t. trading volume, price change and abs price change. A t-test statistic was used to do this. See Matlab *corrcoeff* function for a description.

We then sorted all the significant Twitter features according to their coefficient for price change. If we are able to predict price change well, our trading system should work well.

Table IV shows the top 5 correlated features for each of the companies. For IBM and Intel the analysis was performed with a 95% confidence level. As only two features were significant at 95% confidence level for GE, we carry out the analysis at 80% confidence level.

It is a mix of URL and NO URL features. We assume that this is due to noise and that its importance does not depend on whether or not a tweet contains url. Number of Followers only appears to be an important feature for Intel, so we also discard this. For all companies, tweets related to the company name as well as to their stock name appear to be relevant. We therefore keep these features. All sentiment labels (negative, neutral and positive) appear to be important for all three stocks. A total of 4 final features were selected and are shown in table V. The final features selected, together with the stock

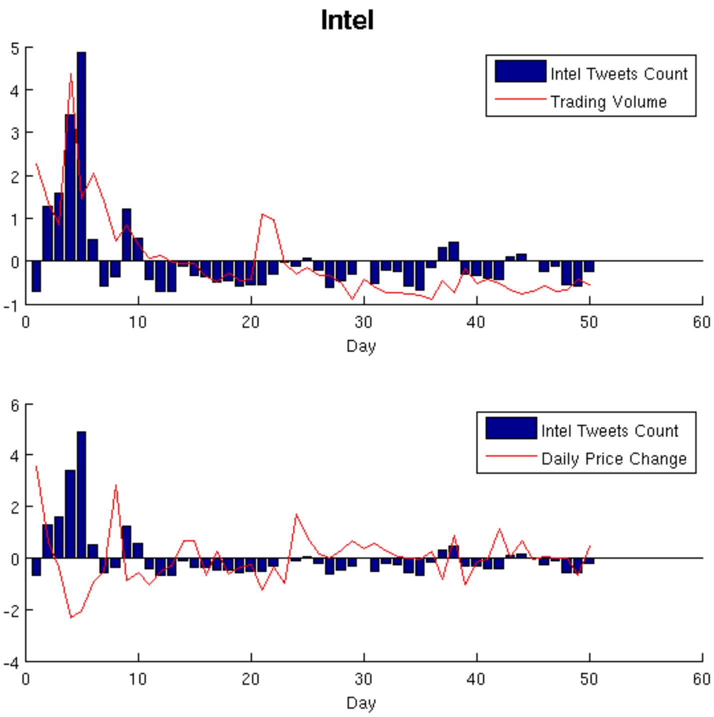

Fig. 4. Tweet counts and stock indicators for Intel

indicators used for training the models, are normalized to have null mean and unit standard deviation.

The model proposed by Mao et al. was linear. To analyse whether or not this is a suitable assumption for our data we plot the stock indicators vs the features, as shown in figure 7 for Intel. We observe across all stocks that most of the Twitter features exhibit some linear trend or no trend at all, i.e. the points are randomly scattered around some mean. An example of a clear linear trend is for the Trading volume of Intel vesus the third feature. An example of no trend is the closing price of Intel versus the fourth feature.

However, there is no evidence of systematic non-linear trends to the naked eye. This suggests that we should not use a non-linear model, at least the model should not be non-linear in the Twitter features.

F. Models

Six different models have been evaluated to build the predictor of the daily price change of the stock. The choice of using a regressor (regression model) rather than a classifier (classification model) was taken with the objective of maximizing the use of the training data. While a classifier trains only on the binary labels *up/down*, a regressor takes also into account how big the changes in the stock prices are.

The models predict the price stock change on a given day, while training on the 35 previous days. The stock changes are predicted for days rather than hours as we hypothesize that aggregated Twitter information cannot reflect future stock changes accurately, given that people tweet at different times of the day depending on their schedules. Furthermore, to the best of our knowledge, hourly prediction models based on Twitter information have not been attempted before in the literature. Establishing a daily prediction model will also avoid confounding predictive power obtained from Twitter sentiment information with a change in time-scale. For example, it might be that applying previous models from the literature on an hourly time-scale will improve the prediction significantly without the additional sentiment information. We might then have ended up wrongly concluding that the sentiment features improved performance, while the time-scale was the actual real reason. In addition to this, we also speculate that shorter prediction intervals (hourly or bi-daily) will contain a high degree of noise and lead to unreliable results.

The three main models we have analysed are: linear regression (baseline, only tweet counts), linear regression with sentiment (with sentiment features extracted above) and non-linear regression (with sentiment features extracted above). Due to the high number of variables we apply both least-squares linear regression and LASSO (least absolute shrinkage and selection operator) regularization. This yields a $3 \times 2 = 6$ models in total.

LASSO regularization penalizes the absolute weight of the parameters, which implies that it will set irrelevant parameters to zero. This is necessary because we have constructed models with up to many parameters, while our dataset only contains 50 samples (days) in total. In addition,

	IBM		Intel		GE	
	R	p-value	R	p-value	R	p-value
Volume	0.3588	0.0105	0.5798	0.0000	-0.1074	0.4580
Closing P	0.0217	0.8812	0.4546	0.0009	0.1169	0.4187
PC	-0.3400	0.0157	-0.4295	0.0019	-0.1161	0.4222
Abs. PC	0.4002	0.0040	0.3709	0.0080	0.0769	0.5954

TABLE III

CORRELATION ANALYSIS FOR THE RAW TWEET COUNTS WITH EACH OF THE MARKET FEATURES

	R	**Feature**
IBM	-6.80e-01	Friends x Tweets, URL, Stock Symbol, Neutral
	-6.70e-01	Friends x Tweets, URL, Stock Symbol, Negative
	+6.40e-01	Friends x Tweets, No URL, Company Name, Positive
	+6.40e-01	Friends x Tweets, No URL, Company Name, Neutral
	-6.10e-01	Friends x Tweets, No URL, Company Name, Negative
Intel	-4.90e-01	Followers x Tweets, No URL, Stock Symbol, Positive
	-4.90e-01	Followers x Tweets, No URL, Stock Symbol, Neutral
	-4.80e-01	Followers x Tweets, No URL, Stock Symbol, Negative
	-4.80e-01	1 x Tweets, URL, Company Name, Positive
	-4.70e-01	1 x Tweets, URL, Company Name, Neutral
GE	-3.80e-01	log2(Friends + 1) x Tweets, No URL, Stock Symbol, Positive
	-3.60e-01	log2(Friends + 1) x Tweets, No URL, Stock Symbol, Neutral
	-2.90e-01	log2(Friends + 1) x Tweets, No URL, Stock Symbol, Negative
	-2.80e-01	Friends x Tweets, URL, Company Name, Positive
	-2.80e-01	Friends x Tweets, URL, Company Name, Neutral

TABLE IV

TOP CORRELATED FEATURES FOR EACH COMPANY

Friends x Tweets, Company Name, Negative
Friends x Tweets, Company Name, Neutral
Friends x Tweets, Company Name, Positive
Friends x Tweets, Stock Symbol

TABLE V

FINAL SELECTED FEATURES FOR ALL 3 COMPANIES

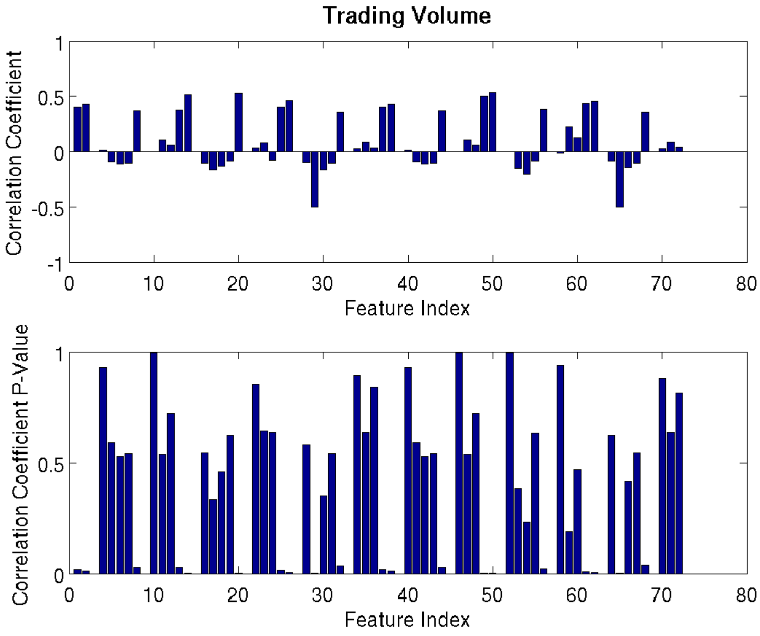

Fig. 5. Correlation analysis of the features considered with the Volume traded of IBM

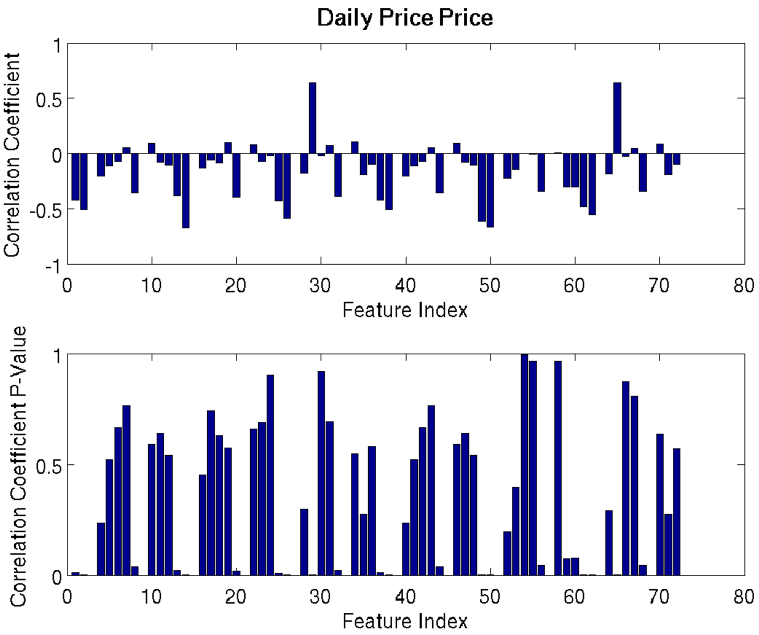

Fig. 6. Correlation analysis of the features considered with the Stock Price Change of IBM

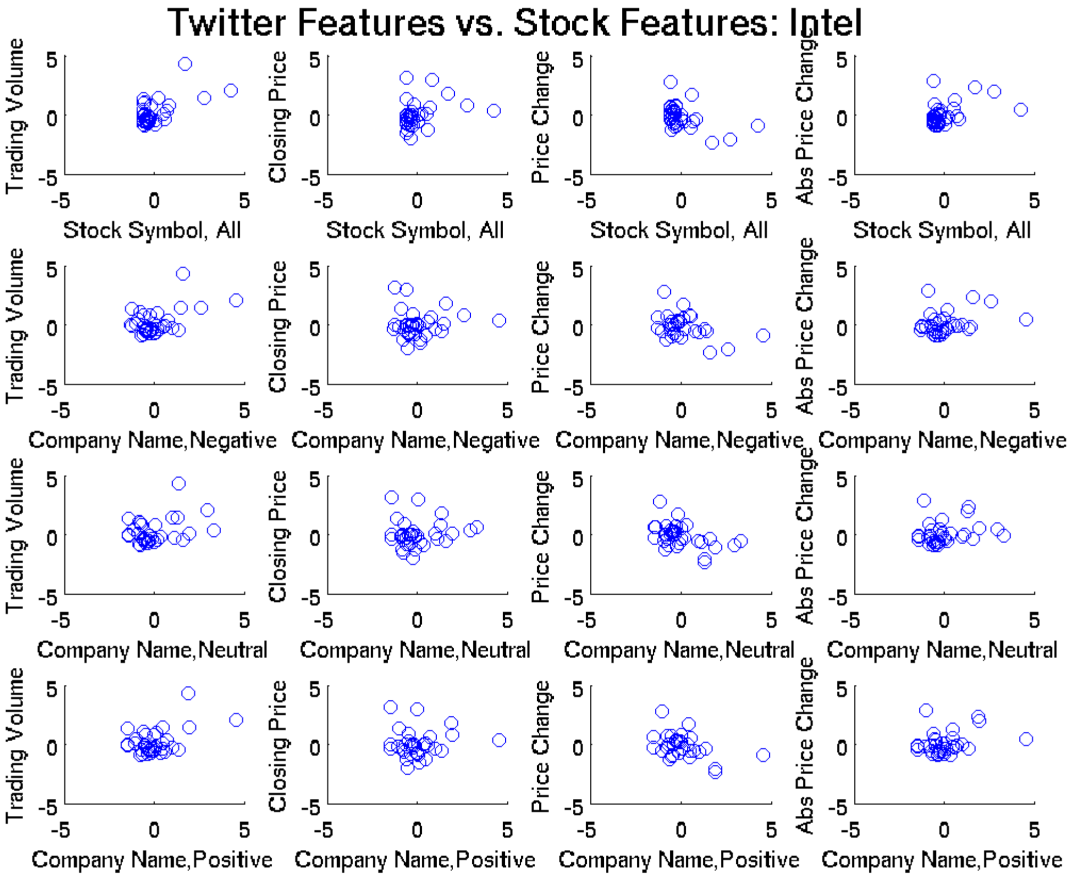

Fig. 7. Stock indicators vs Selected features for Intel

LASSO regression is more appropriate for our problem than the widely used Ridge regression because it favours a much smaller subset of variables. See [16] for a theoretical justification of this. A comparison example between LASSO and Ridge regression is given in [17].

We build a separate linear regression model for each of the stock features. For standard least-squares regularization we implemented our own Matlab method based on the lecture slides by Dr. Mark Herbster. For the LASSO regression implementation we first applied Matlab's built-in LASSO regression function, but this proved to be extremely slow. Instead, we made use of Glmnet, which is an optimized version of LASSO regression [18].

Linear regression (baseline): We establish our baseline model based on [8]:

$$\mathbf{Y}_t = \alpha + \sum_{i=1}^{m} \boldsymbol{\beta}_i \mathbf{Y}_{t-i} + \sum_{i=1}^{n} \boldsymbol{\gamma}_i X_{t-i} + \boldsymbol{\epsilon}_t,$$

where

$\mathbf{Y}_t$: Stock market indicators at day t,

X_t : Tweet count at day t.

Their model, which was based only on counting the number of tweets referring to the stock symbol, has yielded 68% accuracy at predicting the direction of change in price. We will use a cross-validation procedure on the previous 10 days to determine the optimal $m = 1, 2, 3$ and $n = 0, 1, 2, 3$ The LASSO regularization only changes the estimate of the parameters $\boldsymbol{\beta}_i$ and $\boldsymbol{\gamma}_i$.

Linear regression with sentiment features: Bollen et al. show in [5] that there are significant linear correlations between stock market indicators and aggregated emotional tweets, i.e. tweets which expresses some form of mood or emotion. we therefore append the sentiment features to the linear regression model:

$$\mathbf{Y}_t = \alpha + \sum_{i=1}^{m} \boldsymbol{\beta}_i \mathbf{Y}_{t-i} + \sum_{i=1}^{n} \boldsymbol{\gamma}_i \mathbf{Z}_{t-i} + \boldsymbol{\epsilon}_t$$

where

$\mathbf{Y}_t$: Stock market indicators at day t,

$\mathbf{Z}_t$: Tweet sentiment features as described in Table V.

We use the same cross-validation procedure as before to determine $m = 1, 2, 3$ and $n = 0, 1, 2, 3$. The LASSO regularization only changes the estimate of the parameters $\boldsymbol{\beta}_i$ and $\boldsymbol{\gamma}_i$.

Non-linear regression: Following the conclusion about feature selection from section II-E, we now take a step further and analyse how stock features at time t relate with stock features at time $t-1$ and Twitter features at time $t-1$. We carry out this analysis through several 3D graphs, where we plot a stock feature at time t together with a stock feature at time $t-1$ and Twitter feature at time $t-1$. We do this only for our training data.

We observe that there appears to be a non-linear manifold for several feature combinations. For example, figure 8 is a plot based on Price Change at time t, Trading volume at time $t-1$ and Friends x Tweets, Company Name, Neutral at time $t-1$, which has the shape of a smooth sigmoid function. Similar non-linear trends can be observed in 9. Though it is hard to characterize these, we suppose that a multiplicative interaction between the variables may be able to explain the trend. This suggests that we should extend our model to polynomial regression.

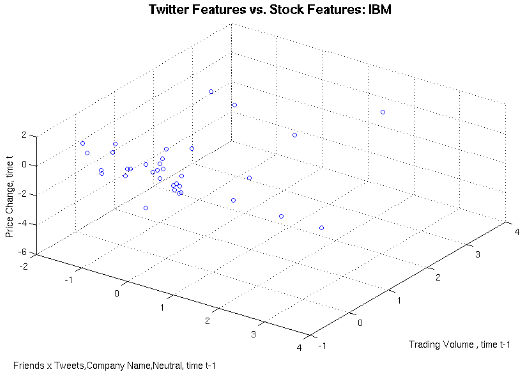

Fig. 8. Non-linearity analysis. IBM: price change at time t vs twitter features at time t-1 and vs volume traded at time t-1

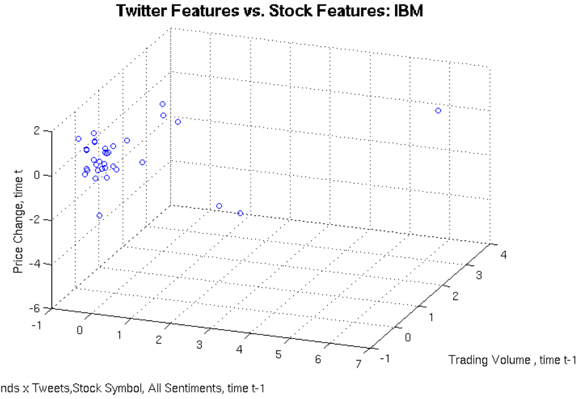

Fig. 9. Non-linearity analysis. IBM: price change at time t vs twitter features at time t-1 and vs volume traded at time t-1

We were unable to find similar structures in the plots for two Twitter features at time $t-1$ and a stock feature at time t. Most of the points on these plots appeared to follow a linear trend with noise. Furthermore, given our previous feature extractions (weighting tweets by followers, friends etc.) it does not make sense to consider interactions between these variables. To reduce the number of parameters we therefore set any interacting terms between two distinct stock features to zero. Furthermore, to compare with Mao et al.[8] and avoid adding additional stock-related information, which might confound any additional predictive performance from Twitter features with stock features, we also set all interacting terms between two stock features to zero. This is equivalent to transforming the data to a polynomial (in feature space) and then removing any second order terms consisting of only Twitter or only stock features.

This representation then gave us $4 + 4^2 = 20$ Twitter-related features for each day. This is a tremendous amount of features, which given our small data size, could easily throw off our linear regression. We therefore performed Principal Component Analysis (PCA) and Probabilistic Principal Component Analysis (PPCA) on the features to reduce them to only 4 features (the components explaining the highest amount of variation in the data). From visual inspection on the training data alone, PPCA features appeared to correlate more linearly with our stock market indicators than PCA. We therefore choose to use PPCA.

We note that our approach is a special case of Kernel LASSO Regression, where we apply a first order polynomial and fix certain terms to zero according to our prior knowledge. See lecture slides by Dr. Mark Herbster or Hastie et al. in [17]. The closely related Kernel Ridge Regression, based on a polynomial kernel, is known to work well on a range of financial time series [19].

Finally our non-linear regression model is:

$$\mathbf{Y}_t = \alpha + \sum_{i=1}^{m} \boldsymbol{\beta}_i \mathbf{Y}_{t-i} + \sum_{i=1}^{n} \boldsymbol{\gamma}_i \mathbf{Z}_{t-i} + \boldsymbol{\epsilon}_t \text{where}$$

$$\begin{aligned} \mathbf{Y}_t :& \quad \text{Stock market indicators at day } t, \\ \mathbf{Z}_t :& \quad \text{Four features extracted from PPCA.} \end{aligned}$$

We use the same cross-validation procedure as before to determine $m = 1, 2, 3$ and $n = 0, 1, 2, 3$. The LASSO regularization only changes the estimate of the parameters $\boldsymbol{\beta}_i$ and $\boldsymbol{\gamma}_i$.

G. Evaluation

Evaluating predictive performance on financial time-series depends to a large extend on the purpose of the prediction. We will assume that the purpose is to perform automatic trading, also known as algorithmic trading. That is, an online system which observes both the stock market indicators and the Twitter data stream, and uses this information to execute actions that maximizes its portfolio.

We will use the following evaluation metrics:

- **Mean squared-error**: An important metric is the mean squared-error (MSE). This is our primary evaluation method, since our models attempt to minimize the mean squared-error and as we will compare it to Mao et al. [8]. It is also good at evaluating difference for large deviations between predicted and actual values, which is often what a risk averse investor is looking for. It should also be appropriate for evaluating predictions of trading volume, as sudden spikes in these may correlate with certain economic events.
- **Mean absolute error**: The mean absolute error (MAE) is also important. Suppose the trading system has bought a stock, then a negative difference between the predictive price and actual price of the stock corresponds exactly to the monetary loss on that order (assuming the system will sell the stock the following day). The mean absolute error may also be applied to predicting the trading volume of the stock.
- **Positive vs. negative accuracy**: We will also consider the accuracy of predicting positive vs. negative values for each stock feature. Recall that our stock features have all been normalized to have mean zero and standard deviation one. If a stock feature was predicted to be positive, while its true (normalized) value was also positive it will be considered correctly classified and vice versa. This can be applied to all four stock features we consider, contrary to solely predicting the time-series movement direction (which can only be applied to price change and volume change). Due to normalization the number of positive and negative values are distributed quite evenly and we therefore do not need to apply the F1 score as discussed by Manning et al [20].

In order to assess the results obtained we have considered two additional benchmark models:

- **Constant Model**: A naive model which predicts each stock feature with the stock feature at the previous time. This would correspond to the optimal prediction of a time-series which is stationary over short periods.
- **Random Model**: A naive model, which predicts every stock feature with a random draw from a standard normal distribution N(0,1).

We note that there is much evidence against the assumptions posed by both the two models in the literature [21].

Based on $10,000$ simulations the Random Model produced the results shown in table VI.

III. Results

We set aside 30% (15 samples) of our total 50 sample data set for testing. The performance of each of our models, measured by the aforementioned metrics, are summarized in table VII. Only the results for the two most relevant stock features daily change price and volume traded are shown. Nevertheless, the results and discussion we present hold for the remaining two stock features close price and absolute change price.

Model	MAE	MSE	ACC
Random	1.130 ± 0.181	2.011 ± 0.127	0.494 ± 0.500

TABLE VI
Results for the random benchmark model

IV. Discussion

From the results in Table VII we observe that almost all models outperform the Random Model by a large margin. Aside from this, the results appear to be mixed. In particular, the Constant Model outperforms all the models on accuracy for IBM and GE. We may speculate that these two companies are, in fact, harder to predict. However, when we consider

Model	Company	Trading Volume			Price Change		
		MAE	MSE	Acc	MAE	MSE	Acc
Constant	IBM	**0.323 ± 0.220**	0.361 ± 0.192	**0.878 ± 0.32**	0.902 ± 0.403	1.709 ± 0.539	**0.673 ± 0.469**
	Intel	0.403 ± 0.149	0.587 ± 0.083	0.939 ± 0.239	0.847 ± 0.347	1.440 ± 0.340	0.592 ± 0.492
	GE	0.546 ± 0.181	0.790 ± 0.127	**0.878 ± 0.327**	0.986 ± 0.524	1.592 ± 0.877	**0.592 ± 0.492**
Linear regression (#tweets)	IBM	0.470 ± 0.220	0.266 ± 0.192	0.600 ± 0.126	0.492 ± 0.403	0.393 ± 0.539	**0.733 ± 0.114**
	Intel	**0.239 ± 0.149**	0.078 ± 0.083	**1.000 ± 0.000**	0.412 ± 0.347	0.282 ± 0.340	0.533 ± 0.129
	GE	0.284 ± 0.181	**0.111 ± 0.127**	0.867 ± 0.088	0.488 ± 0.524	0.494 ± 0.877	0.333 ± 0.122
Linear regression (sentiment)	IBM	0.447 ± 0.245	0.256 ± 0.206	0.733 ± 0.114	0.500 ± 0.396	0.396 ± 0.533	0.600 ± 0.126
	Intel	0.245 ± 0.149	0.081 ± 0.085	**1.000 ± 0.000**	0.412 ± 0.355	0.287 ± 0.355	0.533 ± 0.129
	GE	**0.277 ± 0.208**	0.117 ± 0.178	0.733 ± 0.114	0.465 ± 0.528	**0.476 ± 0.929**	0.467 ± 0.129
Non-Linear regression	IBM	0.465 ± 0.227	0.264 ± 0.215	0.533 ± 0.129	0.492 ± 0.406	0.397 ± 0.546	0.667 ± 0.122
	Intel	0.243 ± 0.139	**0.077 ± 0.074**	**1.000 ± 0.000**	**0.400 ± 0.357**	**0.279 ± 0.338**	0.533 ± 0.129
	GE	0.312 ± 0.177	0.126 ± 0.146	0.867 ± 0.088	0.500 ± 0.537	0.519 ± 0.980	0.333 ± 0.122
Linear regression (#tweets) /w LASSO	IBM	0.359 ± 0.216	0.172 ± 0.154	0.467 ± 0.129	0.510 ± 0.386	0.399 ± 0.519	0.467 ± 0.129
	Intel	0.363 ± 0.209	0.172 ± 0.181	**1.000 ± 0.000**	0.439 ± 0.350	0.307 ± 0.356	0.533 ± 0.129
	GE	0.318 ± 0.202	0.139 ± 0.157	0.600 ± 0.126	**0.459 ± 0.533**	0.476 ± 0.979	0.467 ± 0.129
Linear regression (sentiment) /w LASSO	IBM	0.345 ± 0.206	**0.158 ± 0.145**	0.600 ± 0.126	**0.490 ± 0.384**	**0.378 ± 0.503**	0.333 ± 0.122
	Intel	0.344 ± 0.193	0.153 ± 0.141	0.933 ± 0.064	0.447 ± 0.360	0.321 ± 0.365	0.467 ± 0.129
	GE	0.345 ± 0.228	0.167 ± 0.175	0.467 ± 0.129	0.513 ± 0.570	0.567 ± 1.024	0.333 ± 0.122
Non-Linear regression /w LASSO	IBM	0.354 ± 0.212	0.167 ± 0.150	0.600 ± 0.126	0.534 ± 0.383	0.422 ± 0.496	0.467 ± 0.129
	Intel	0.344 ± 0.203	0.157 ± 0.179	0.933 ± 0.064	0.435 ± 0.373	0.319 ± 0.390	0.600 ± 0.126
	GE	0.325 ± 0.205	0.145 ± 0.147	0.533 ± 0.129	0.497 ± 0.513	0.492 ± 0.973	0.267 ± 0.114

TABLE VII

Mean squared-error, mean absolute error and mean accuracy together with their standard deviations for the 6 implemented models and the Constant Model. The best models for each evaluation metric are marked in bold font.

the MAE and MSE we note that the majority of models consistently outperform the Constant Model. This suggests that either the previous stock features or the previous Twitter features do play a significant role. Taken together with our earlier correlation analysis, which showed that tweet counts were significantly correlated with stock features, we conclude that some Twitter features do contain predictive information when combined with stock features.

For all companies and all stock market features, we also observed that $n > 0$ was chosen for at least one day based on cross-validation. This also supports our conclusion that Twitter features do contain predictive information.

When we compare our models across regularization methods, we observe that the best performing models are based on simple least-squares with no regularization. Despite the large number of features, we do not need any form of regularization to improve our models. This could be due to the very low number of samples, which might throw off the cross-validation procedure used to select the regularization parameters.

Now, if we discarded the LASSO regularization models, the Non-Linear regression model would be the best performing model for IBM and Intel and the second best performing model for GE. However, when compared w.r.t. MSE its results are very close to the Linear Regression (#Tweets) to a point where we cannot favour any model more than the other. This can also be observed from figure 10 and figure 11, where it is hard to spot any visible differences between the two models. Since this is the best model out of all our proposed models, our results strongly indicate that there is no significant amount of predictive information in our additional sentiment features.

V. Further Work

In our work we have only considered three companies over a period of 50 days. This is clearly extremely limited. We should consider more companies over a longer period to obtain more reliable results. Indeed by considering more companies, we may also be able to uncover whether or not sentiment plays a larger predictive role for certain stocks than others.

It would be very interesting to train a sentiment classifier based on actual stock-related tweets. Since the LingPipe classifier was only trained on random English tweets, this may give us a significantly better result. It would also be interesting to consider other features, such as subjective (sentimental) versus objective (no sentiment) tweets.

Following the results from [10], [22], an interesting expansion of this project would be to *handpick* the users whose tweets are used to build the prediction, as users who often tweet about the stock market are in general more reliable. This is somewhat related to the PageRank idea, that certain nodes in a network (users in our case) are more reliable than others [20].

VI. Conclusion

In this project, we investigated the relationship between stock market indicators for the three companies and tweets mentioning their name on Twitter. Grounded in previous research, we extracted a set of sentiment-based features which took into account the number of followers, friends and the presence of URLs. We applied the two sentiment analysis platforms Stanford's Deeply Moving and LingPipe, and found that LingPipe performed best [11] [12].

We performed a correlation analysis, which indicated that the number of tweets related to a certain stock is significantly

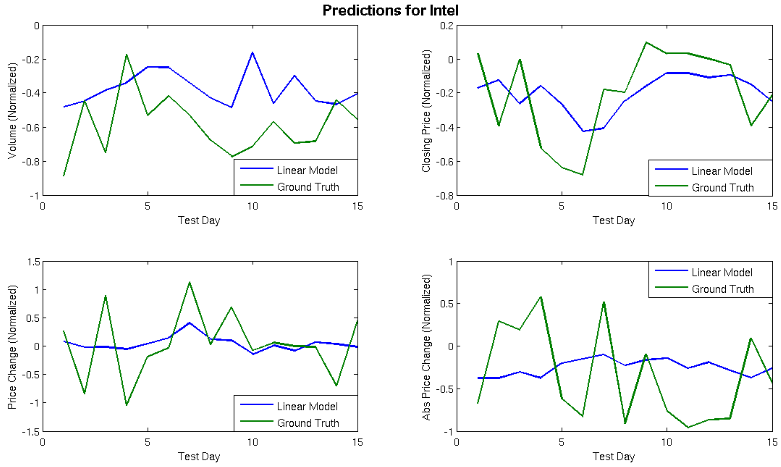

Fig. 10. The predicted stock market indicators w.r.t. the least-squares linear regression model based only on tweet counts and the actual values for Intel.

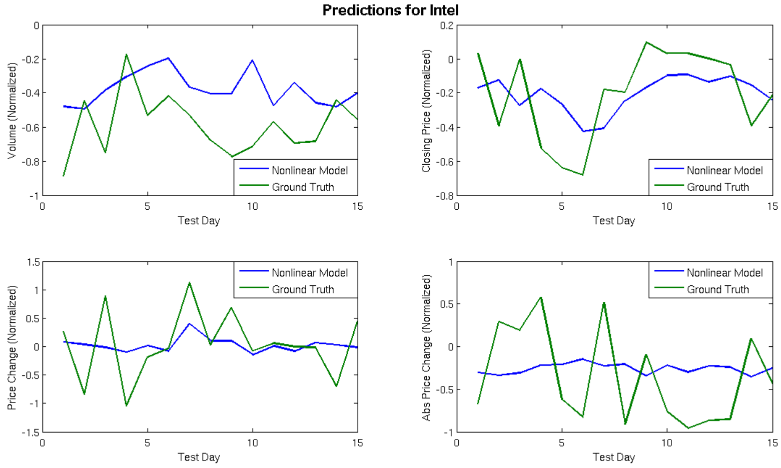

Fig. 11. The predicted stock market indicators w.r.t. the least-squares non-linear regression model and the actual values for Intel.

correlated with several stock market indicators. This confirms previous research [8]. We also performed an extensive feature analysis, which showed that the number of tweets weighted by the number of friends were the most important features correlated with stock market indicators. We used this analysis to select the four features which were most likely to contain predictive power w.r.t. future stock market indicators.

Based on our stock market indicators and Twitter-based features, we then trained three models for predicting stock market indicators. We trained two linear regression models, which took as input respectively the number of tweets and sentiment-based tweet features. We trained a third non-linear regression model, which took into account higher-order interactions between the sentiment-based features and the stock market indicators. To fit the models we applied standard least-squared error and LASSO (least absolute shrinkage and selection operator) regularization. We evaluated all models with respect to mean squared-error, mean absolute error and accuracy. Finally we found that all models performed consistently better than a naive or random prediction, which indicates that Twitter related features in fact do contain predictive information. However there was no significant difference between models taking sentiment-based features as input and models taking only the number of tweets as input. This strongly indicates that sentiment-based features do not add any predictive information in addition to the simple tweet count. This disagrees with some previous research [6] [7].

References

[1] Techcrunch news. http://techcrunch.com/2012/12/18/twitter-passes-200m-monthly-active-users-a-42-increase-over-9-months/. Accessed: 2014-04-12.

[2] Akshay Java, Xiaodan Song, Tim Finin, and Belle Tseng. Why we twitter: Understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis*, WebKDD/SNA-KDD '07, pages 56–65, New York, NY, USA, 2007. ACM.

[3] Werner Antweiler and Murray Z. Frank. Is all that talk just noise? the information content of internet stock message boards. *The Journal of Finance*, 59(3):1259–1294, 2004.

[4] Gilad Mishne and Natalie Glance. Predicting movie sales from blogger sentiment. In *AAAI Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW)*, pages 155–158, 2006.

[5] Johan Bollen, Huina Mao, and Xiao-Jun Zeng. Twitter mood predicts the stock market. *CoRR*, abs/1010.3003, 2010.

[6] Johan Bollen, Alberto Pepe, and Huina Mao. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. *CoRR*, abs/0911.1583, 2009.

[7] Xue Zhang, Hauke Fuehres, and Peter A. Gloor. Predicting stock market indicators through twitter ”i hope it is not as bad as i fear”. *Procedia - Social and Behavioral Sciences*, 26:55–62, January 2011.

[8] Yuexin Mao, Wei Wei, Bing Wang, and Benyuan Liu. Correlating S&P 500 stocks with twitter data. In *Proceedings of the First ACM International Workshop on Hot Topics on Interdisciplinary Social Networks Research*, HotSocial ’12, pages 69–72, New York, NY, USA, 2012. ACM.

[9] Yahoo! finance. http://finance.yahoo.com/. Accessed: 2014-04-12.

[10] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. Information credibility on twitter. In *Proceedings of the 20th International Conference on World Wide Web*, WWW ’11, pages 675–684, New York, NY, USA, 2011. ACM.

[11] Deeply moving: Deep learning for sentiment analysis. http://nlp.stanford.edu/sentiment/. Accessed: 2014-04-12.

[12] Alias-i. 2008. lingpipe 4.1.0. http://alias-i.com/lingpipe/. Accessed: 2014-04-12.

[13] Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity. In *Proceedings of ACL*, pages 271–278, 2004.

[14] How to: Sentiment analysis of tweets using java. http://cavajohn.blogspot.co.uk/2013/05/how-to-sentiment-analysis-of-tweets.html. Accessed: 2014-04-12.

[15] Sanders analytics: Twitter sentiment corpus. http://www.sananalytics.com/lab/twitter-sentiment/. Accessed: 2014-04-14.

[16] Massimiliano Pontil. Ucl supervised learning (module: Gi01/m055) slides. 2013-2014.

[17] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: Data mining, inference and prediction*. Springer, 2001.

[18] Glmnet in matlab. lasso and elastic-net regularized generalized linear models. http://www.stanford.edu/~hastie/glmnet_matlab/. Accessed: 2014-04-16.

[19] Peter Exterkate, Patrick JF Groenen, Christiaan Heij, and Dick van Dijk. Nonlinear forecasting with many predictors using kernel ridge regression. Technical report, Tinbergen Institute Discussion Paper, 2011.

[20] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[21] Andrew W Lo and Archie Craig MacKinlay. Stock market prices do not follow random walks: Evidence from a simple specification test. *Review of financial studies*, 1(1):41–66, 1988.

[22] Roy Bar-Haim, Elad Dinur, Ronen Feldman, Moshe Fresko, and Guy Goldstein. Identifying and following expert investors in stock microblogs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP ’11, pages 1310–1319, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.