**Examining Correlation between Twitter Sentiment & Stock Price Fluctuation of Auto-Manufacturers**

**Georgetown Data Science Certificate**

**Capstone Project**

**Fall 2015**

Maximo Almodovar

Jonathan Creem

Andrew Miller

Caroline Morris

1. **Abstract**

This paper examines the following fundamental question: **Is Twitter sentiment correlated with fluctuation in stock price?** The presented analysis examines Twitter sentiment of three auto-manufacturers (General Motors, Volkswagen, and Toyota) and attempts to look for correlation between twitter sentiment and fluctuation in company stock price over a designated time period. We find that the daily Twitter sentiment has a positive correlation with the closing stock price of the company. In particular, we find that a 1% increase in the average Twitter sentiment of a company contributes to approximately 1.2% increase in its stock price, and that this relationship is statistically significant. Additionally, our paper describes both the challenges in developing an accurate twitter sentiment scoring model and with the overall project topic.

1. **Background**

Since its inception in 2006, Twitter has become an established social media outlet for corporations and other organizations. Because of its large scale and diverse user base, the ability to leverage the sentiment of Tweets provides for a unique avenue into the conscious of potential consumers and companies’ investors.

In this paper, we analyze if Twitter sentiment can be leveraged in order to forecast stock pricing. There are many papers proposing this approach (Si, et al., 2013, pp. 24-29) and many other papers and online resources where Twitter sentiment is used to derive conclusions and analysis on many different areas: politics (Bakliwal, et al., 2013), sports (Waldron, 2015), health (Ji, 2013), etc.

First, this paper explains the problem we want to solve and what goals and hypothesis we have set. The next section goes over the structure of the data that we collected from Twitter. Then, we described the structure of our tool following a standard Data Science pipeline: ingestion, wrangling, machine learning, analysis and visualization. Finally, we summarize our findings and discuss some future improvements and lessons learned from this process.

1. **Twitter Sentiment & Stock Prices**

As history has shown, predicting the stock market is an extremely challenging operation. The rise of big data, social media, and technology provides a new, important, and unique element for market forecasting. However, these products are new, and many are either skeptical or lack the knowledge to accurately incorporate them into forecasting techniques. Specifically, since Twitter accounts for, in part, investor sentiment, it could be a potentially useful forecasting tool. Our analysis attempts to solve the underlying question: Can Twitter sentiment analysis be used to predict stock price fluctuation?

Hypothesis 1: By looking at the Twitter sentiment of potential or current consumers of Volkswagen, Toyota, and General Motors, we believe we can predict the changes in the stock value of these companies.

This hypothesis creates the two first limits of our analysis. First, we decided to focus on the US securities of these 3 companies (i.e. we are not trying to predict changes in the stock value of Volkswagen in European markets or of Toyota in Japanese markets). Subsequently, we focus our Twitter sentiment analysis on English tweets published in the whole Twitter universe. We will go over the Twitter ingestion process in future sections.

Hypothesis 2: The more tweets that are ingested and calibrated, the more accurate the sentiment analysis will become, and the more accurate our machine learning and regression models will also become.

This hypothesis derives from the Twitter sentiment analysis process itself. As explained in (Luce, 2012), the foundation of a Twitter sentiment analysis tool is the definition of a tweet corpus defining whether a text is positive or negative. Although we start with an initial corpus, we decided to use an approach by which this corpus will be updated every date with additional tweets classified as positive or negative. This improvement in the foundational corpus should create improvements in the accuracy of our machine learning and regression models.

Once we have the problem well defined and our set of 2 hypotheses, we need to come up with a clear set of goals for our tool. In particular, we decided to achieve the following goals:

* Develop an accurate model for Twitter sentiment analysis.
* Perform a regression analysis under the assumption of “stock price changes are derived by Twitter sentiment”.
* Use visualization tools to show the correlation between Twitter sentiment and stock price fluctuation, or vice versa.
* Create additional visualizations to show other unexpected features of our data: top words, changes derived by news about the companies, etc.

1. **Description of Data Set Used**

In terms of tweets, we have used the Twitter API (Twitter, Inc, 2015) to ingest tweets mentioning General Motors, Volkswagen, and Toyota over a period from early November to mid-December. The tweets were stored in MongoDB as JSON documents. The existence of an API eases the process of data ingestion and cleaning since the data that the developer can access is well structured and organized. However, there are some drawbacks and limitations that will be explained in future sections.

On the financial side, the stock prices of the US securities for General Motors, Volkswagen, and Toyota were obtained from Yahoo Finance (Yahoo! Inc., 2015) in the form of CSVs. A complete list of our data pipeline tools is listed in **Appendix A.**

1. **Methodology Used – Models**
2. Twitter Sentiment Model

There is a broad literature on different methods for modelling Twitter Sentiment. In our case, we have followed the techniques and guides described in (Luce, 2012) and (Czerny, 2015).

Our first step was to select a corpus of text annotated with sentiment values on which to build our sentiment analysis model; the syntax, relationships between words, and vocabulary in the corpus teaches the algorithm how to process natural language, and then how to classify the sentiment of a text. We insisted on using a tweet-based corpus on the assumption that, although the language is not domain specific (the tweets are not generally about automobiles or stock prices), it would best prepare the model for sentiment analysis of the language patterns found in our ingested tweets. The corpus used (Think Nook, 2012), created from prior sentiment classification work (University of Michigan, 2011 and Sanders, 2011), is comprised by a million and a half tweets marked positive or negative.

The corpus serves two purposes in our sentiment analysis model. First, following (Czerny, 2015), we use the Google Word2Vec model (Google, Inc., 2013) on the full corpus to compute vector representations of words; the result is a ‘model’ word vector file built on the vocabulary of the corpus. Second, for sentiment analysis of the ingested tweets, we create a **Smaller Corpus** (a term I will use to describe this truncated version of the full corpus later in this paper) from a subset of the full corpus (roughly 100,000 tweets), ‘calibrated’ with additional ingested tweets that are assigned sentiment value by the team (see Section 6.2 for details). The choice of a leaner version of the full corpus for the actual sentiment analysis is due to consideration of both the full corpus’s large file size and a desire to give our calibrated tweets more importance in the smaller corpus.

1. Regression Model

A standard log-log regression model was used to analyze the effect twitter sentiment has on the closing price of the automobile companies. The model is defined below.

LogClose*jt*= LogSent*jt* + LogAgg\_RT*jt*+ *ejt*

Where *jt* is defined as the auto-company *j* on date *t* and *e* is an error term assumed to be i.i.d.

The dependent variable being studied will be the natural log of the closing stock price of each company on a particular trading day. The independent or predictive variables we will apply are as followed:

LogSent = Defined as the natural log the average daily sentiment

LogAgg\_RT = Defined as the natural log of the total number retweets for each company, on each day.

Taking the log of both sides of the equation is needed because stock prices that are expressed in dollars cannot compared directly to the units we have for Twitter sentiment. To account for this a log-log model expresses the relationship of the dependent and independent variables in percentage terms.

As a form of evaluation, we will compare the performance of our model to that of the standard Random Walk Model

1. **Methodology Used –Data Pipeline Steps:**
2. Ingestion

Data ingestion is *"the process of obtaining, importing, and processing data for later use or storage in a database "* (Gibilisco, 2013)*.* According to this definition we define the following processes as the basic pipeline of our own Twitter sentiment analysis tool:

* Obtaining and processing tweets via Twitter API
* Obtaining and processing stock prices via a public Financial Data provider
* Storing the data in a MongoDB database

Thanks to the Twitter API, any Twitter user can create custom applications that read the Twitter universe which mainly includes:

* **Users:** Entities that create the information via tweeting, retweeting, following, etc. They tend to represent people, companies or whatever entity interested in participating in the Twitter worldwide conversation.
* **Tweets:** Basic units of textual information created by Users.

The Twitter API is well documented but for practical purposes is much more efficient to use an existing Twitter API library for Python. There are many libraries available for coding in many different languages. We have chosen to use Tweepy (Roesslein, 2015), which is broadly used and many examples can be found in the Internet.

With the Tweepy library we can easily obtain tweets from a user timeline. While in terms of coding this approach is very simple, there is a major disadvantage in terms of accuracy of the results. If we try to measure the Twitter sentiment on a particular company by using only its own tweets, we would be measuring the self-sentiment of the company. Although this approach may be interesting in terms of understanding the self-esteem of the company, we need external data to determine correlations with stock prices. Since a company stock price is mainly derived from external factors, it is important to include some external sentiment in the equation. This led us to another option.

The final option we choose is based on "streaming" data from the Twitter universe by looking for particular terms in the Tweets that may refer to the companies of study. In (Bonzanini, 2015), we find a good example of how to get real time tweets from the Twitter universe. However, this approach captures tweets from present and the future (depending on how long you execute it in your machine), but not from the past. Historical data in twitter cannot be gathered unless you hire a company to do so, which is very expensive for the purposes of our study.

The use of a “Streamer” has been a major drawback in our development. Until the required data set for future steps in the pipeline was not clear enough and the ingestion code properly coded, we couldn’t capture tweets. This implies that we don’t have enough historical data to prove if our model works.

After we plug-in our “Streamer”, the tweets read are processed and filtered since we do not need all the fields of a Tweet entity for our analysis. Every tweet is saved in a MongoDB database structured in a way that each date is represented with a collection.

The ingestion of financial data is made via downloading the CSV files for the dates that we need from the specific security website in Yahoo! Finance. This data is only consumed by our regression analysis model that tries to find correlation between Twitter sentiment and stock price changes.

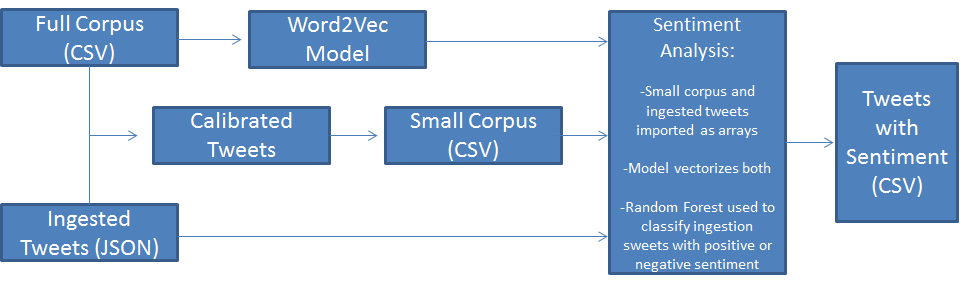
1. Munging & Wrangling

Although data obtained through Twitter API is well structured and very clean, we need to implement some wrangling and munging options that are key for our machine learning algorithm.

To prepare ingested tweets for sentiment analysis, we load a day’s worth of ingested tweets (in the form of a JSON file) into a pandas database. Following the techniques described in (Bonzanini, 2015), we use a ‘wrangler’ comprising NLTK language processing and simple text cleaning to prepare each tweet as a tokenized list of words, removing punctuation and filler language from the text.

1. Machine Learning/Modeling

The sentiment analysis module has three steps to assign sentiment to tweets from a given day: 1) training the Word2Vec model on the corpus, 2) creating a calibrated Smaller Corpus as training data for a classifying the ingested tweets, 3) calculating average vectors for both ingested tweets and tweets from the most recently updated small corpus, and 4) using a Random Forest Classifier to assign sentiment value to the ingested tweets.



First we train the model on the full corpus to compute vector representations of words; the result is a ‘model’ word vector file built on the vocabulary of the corpus. We do this by importing the million and a half tweets from the corpus into a panda array and then using the same ‘wrangler’ method detailed above to clean and tokenize the text, and then running Word2Vec over all tweets to create the model. We used Word2Vec’s default parameters for vectorization, except where we followed the example of (Kaggle, 2015) and (Bonzanini, 2015) to adjust minimum counts for words to be considered, as well as the number of featured vectors for extraction, for the relatively limited size and vocabulary of tweets from consumers.

As mentioned in section 5.1, we create a Small Corpus from a subset of the full corpus (roughly 100,000 tweets), plus a couple hundred tweets each from the day being analyzed and every day analyzed prior; the added tweets are annotated with a sentiment value manually by the team, with the intention of ‘calibrating’ the small corpus with human discretion over the sentiments being expressed towards a given car company (i.e., assessing the sentiment value of Toyota’s popularity with ISIS, which we consider negative, is something the model likely needs to be calibrated to understand). A user manually annotates tweets by using our calibration module to i) randomly look for unclassified tweets (i.e. tweets not classified by a user) or ii) assigning a sentiment to a subset of tweets including a word specified by the user.

With both a Word2Vec model and a calibrated Small Corpus, we used the Word2Vec model to obtain vector scores for each tweet in both the ingested tweets and the Small Corpus. To do this, we followed an example (Kaggle, 2015) for obtaining the average vector value per tweet; each tweet is evaluated as an array of all words for which the model has a vectorized value, and then the average of all word vectors present is assigned to the tweet as its vector score.

Vectorization gives us two sets of data; the ingested tweets with each tweet assigned a vector score, and the smaller corpus consisting of vector-scored tweets paired with sentiment values of positive or negatve. We then use a Random Forest Classifier, with the vectorized tweets and their assigned sentiments from the smaller corpus as training data, to assign positive or negative sentiments to each ingested tweet. The resulting array of ingested tweets with assigned sentiment value is exported to a CSV file.

**Note:** for the purposes of our project, we added the calibrated tweets from all our days into the million and a half tweet original corpus to train the model once, and used that file of vectorized words as the model for training each pair of calibrated small corpora and the day’s ingested tweets.

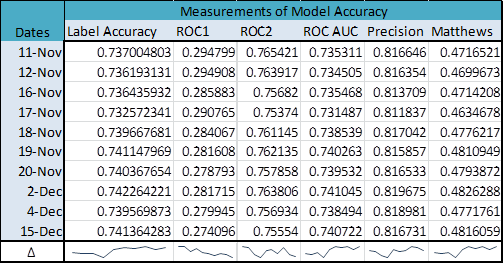
*Sentiment Analysis Model Accuracy*

As described in Section 3, we have the hypothesis that as we read more data our model’s accuracy will improve, which is why we have the ‘calibration’ component in our pipeline; with each new day of tweets ingested, we manually assign sentiment value to a number of tweets and incorporate them into the corpus and modeling, ostensibly improving the model’s reading comprehension of tweets about our car companies.

By updating our Small Corpus each day with the user-labeled tweets, we gradually calibrated our Sentiment Analysis module to classify ingested tweets more accurately. Below are the measures of accuracy we used to evaluate the model after each calibration, and a chart showing the results.

* Label Accuracy - accuracy for label classication
* ROC1 - increasing false positive rates
* ROC2 - increasing true positive rates
* ROC AUC - area under curve from prediction scores for classication
* Average precision score - average precision from prediction score
* Matthews correlation coefficient - quality of binary classifications, taking into account true and false positives.

Table 1: Calibration effect on Sentiment Model



*Note: Due to technical issues, we were not able to calibrate for November 13th.*

Formatting our data to be compatible with running the log-log model was done, mostly using SAS. The code for that data manipulation can be provided if needed. Once formatted correctly, the data was loaded into Python using Pandas DataFrame, and the StatsModel module was out primary tool to run the linear regression and analyze its performance.

1. Data Visualization

Tableau has been used to create the data visualizations. The results of our Data Science Pipeline are exported into CSV files that can be used in Tableau in order to create charts. Other visualization tools used are shown in **Appendix B.**

1. **Visualizing Twitter Sentiment**

Figure 1 shows the total number of tweets per day by company. By looking at this graph we discovered that General Motors does not have too many tweets. One possible reason is that we should have looked for terms like GMC or GM instead of General Motors.

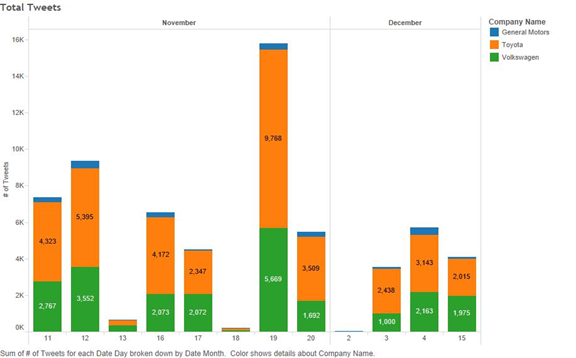


Figure 1: # of tweets per day by company

Another interesting fact is that the highest amount of tweets for each company occurred on November 11, November 19, and December 4. We performed a Google search to see what was happening on high tweet days. Results are shown in Figure 2**.** It appears there were significant events happening on days that had the highest number of tweets. We did not run regressions on this because it was not the major objective of the project. However, including some information about news and events could be a further refinement into making our model more accurate.

Our main objective is to visualize Twitter sentiment on these dates. Figure 3 shows the evolution of Twitter sentiment during the analyzed period. The percentage indicates the percentage of tweets that were classified as positive for a particular company and a particular date. We can clearly see that Volkswagen is the company with the most negative Twitter sentiment which makes sense considering the recent emissions scandal. General Motors seems to have a “neutral” sentiment, but this is probably derived by a low amount of tweets. Finally, Toyota seems to have the most positive sentiment.

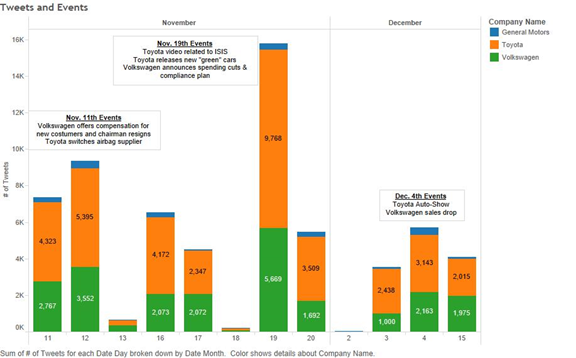


Figure 2: Looking for significant events

We append more figures showing the Twitter sentiment of each company during time and including some trend analysis

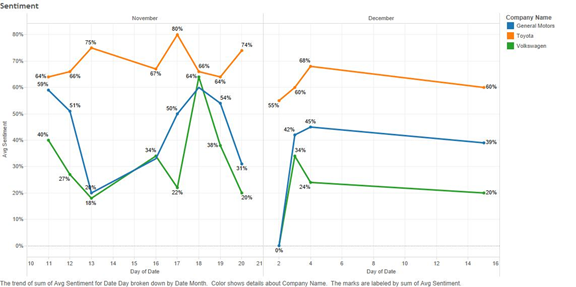


Figure 3: Twitter sentiment by company

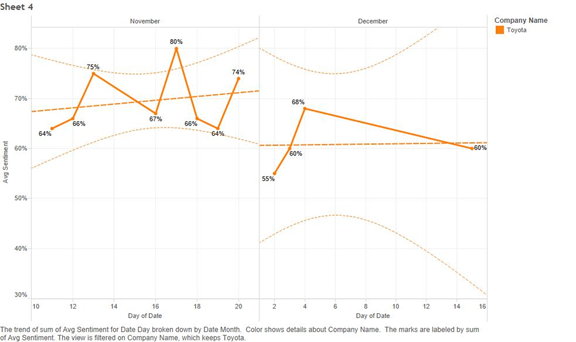


Figure 4: Toyota sentiment trends

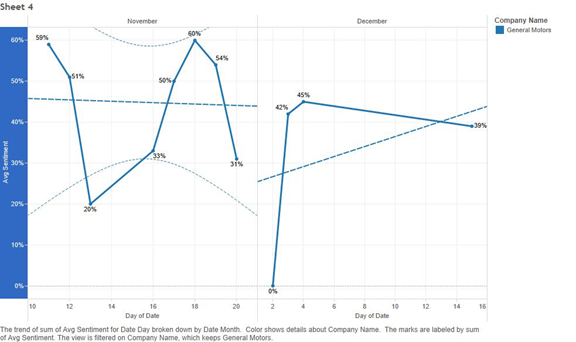


Figure 5: General Motors sentiment trend

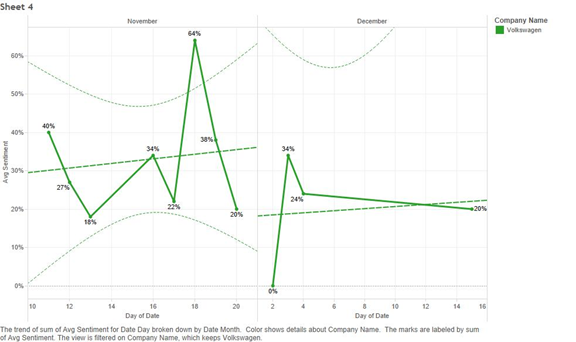


Figure 6: Volkswagen sentiment trends

1. **Results – Regressions**

A full summary of our regression can be found in Table 2. As expected, our regression indicates that closing stock price is positively influenced by level of average daily sentiment and is significant at the 99% confidence level. This shows that the public sentiment of a company can influence its stock price.

To further our analysis the daily aggregate number of Retweets was implemented in the model and its relationship to the closing stock price was examined. Our model shows that, all else equal, a 1% increase in the aggregate number of retweets cause a .086% increase in stock price. From an intuitive standpoint, this would suggest that if news of an auto company is impactful enough the increased pace to which the news spread has an impact on the market price.

The performance of all predictive models attempting to forecast stock prices, however, should be evaluated against the performance of a standard Random Walk. We have concluded that even though our sentiment variable have some predictive power with an R2 = 0.6435, F Value = 26.17, and Mean Squared Error = 0.18688, it failed to outperform the Random Walk. The R2 of the Random Walk returned a value of 0.9993, its F Value returned a value of 48921.9 and a Mean Squared Error of 0.00032. This evaluation reinforces the idea that the best indicator of stock price fluctuation is price of the previous period.

The final indication that our model fails to outperform the Random Walk model is the large and significant explanatory power of the intercept variable. Because of its impact of the dependent variable, we can conclude that the close price is most likely being affected by variable not accounted for in our model. The reverse is seen when looking at the Random Walk.

1. **Challenges & Conclusion**

This project presented several challenges in the data ingestion and data wrangling phases. For one, the amount of historical Twitter data that can be obtained for free is extremely limited and our approach makes our model extremely reliable on the user (i.e. the user has to connect the “Streamer” to Twitter in order to read data). Thus, our model for Twitter Sentiment and our Regression Model are limited in accuracy. However, as per the results shown Table 1, the more Twitter data that is ingested and calibrated, the more accurate our model becomes. Our model accuracy could be further improved through the incorporation of Google trends to help further understand the correlation between major news events, twitter sentiment, and stock price.

1. **GitHub Location**

Our repository, including our Python code, JSON files, and CSVs can be found at: <https://github.com/georgetown-analytics/auto-sentiment>

Table 2: Log-log Regression of Twitter Sentiment on Stock Close Price and Random Walk Model

|  |  |  |
| --- | --- | --- |
| **LogClose** | **Sentiment Model** | **Random Walk** |
| **Variables** | **Coeff.** | **Coeff.** |
| **Intercept** | **4.51467** | **0.00802** |
|  | **(17.38)\*\*\*** | **(0.45)** |
| **LogL1Close** |  | **0.99825** |
|  |  | **(221.18)\*\*\*** |
| **LogSent** | **1.23512** |  |
|  | **(6.97)\*\*\*** |  |
| **LogAgg\_RT** | **0.08585** |  |
|  | **(2.09)\*\*** |  |
|  |  |  |
| **Number of Obs** | **32** | **36** |
| **MSE** | **0.18688** | **0.0003242** |
| **F Value** | **26.17** | **48921.9** |
| **Pr > F** | **<.0001** | **<.0001** |
| **R2** | **0.6435** | **0.9993** |
| **Adj R2** | **0.6189** | **0.9993** |

***LogL1Close = Natural Log of the first daily lag of close price***

***t values are listed under the coefficiants in parentheses***

***\* indicates significance at the 90% confidence level***

***\*\* indicates significance at the 95% conficdence level***

***\*\*\* indicates significance at the 99% confidence level***

**References**

Bakliwal, A., Foster, J., van dr Puil, J., O'Brien, R., Tounsi, L., & Hughes, M. (2013). Sentiment Analysis of Political Tweets: Towards an Accurate Classifier. *Association for Computational Linguistics* , 49-58.

Bonzanini, M. (2015). *Marco Bonzanini Tech Diary*. Retrieved from http://marcobonzanini.com/2015/03/02/mining-twitter-data-with-python-part-1/

Czerny, M. (2015). *Modern Methods for Sentiment Analysis*. Retrieved from District Data Labs: http://districtdatalabs.silvrback.com/modern-methods-for-sentiment-analysis

Gibilisco, S. (2013). *WhatIs.com*. Retrieved from http://whatis.techtarget.com/definition/data-ingestion

Google, Inc. (2013). *Word2Vec*. Retrieved from https://code.google.com/p/word2vec/

Ji, X. (2013). Monitoring Public Health Concerns Using Twitter Sentiment Classifications. *2013 IEEE International Conference on Healthcare Informatics (ICHI)*, 335 - 344.

Luce, L. (2012). *Twitter sentiment analysis using Python and NLTK*. Retrieved from Laurent Luce's Blog: http://www.laurentluce.com/posts/twitter-sentiment-analysis-using-python-and-nltk/

Roesslein, J. (2015). *Tweepy*. Retrieved from https://github.com/tweepy/tweepy

Sanders, N. (2011). *Sanders Analytics*. Retrieved from Twitter Sentiment Corpus: http://www.sananalytics.com/lab/twitter-sentiment/

Si, J., Mukherjee, A., Liu, B., Li, Q., Li, H., & Deng, X. (2013). Exploiting Topic based Twitter Sentiment for Stock Prediction. *Association for Computational Linguistics*, 6.

*Think Nook*. (2012). Retrieved from Twitter Sentiment Analysis Training Corpus (Dataset): http://thinknook.com/twitter-sentiment-analysis-training-corpus-dataset-2012-09-22/

Twitter, Inc. (2015). *Twitter API Documentation*. Retrieved from Twitter: https://dev.twitter.com/overview/documentation

University of Michigan. (2011). *Kaggle*. Retrieved from UMICH SI650 - Sentiment Classification: https://inclass.kaggle.com/c/si650winter11

Waldron, M. (2015). *SuperBowl XLIX in Tweets: Sentiment Analysis of 4 Million Tweets*. Retrieved from Text Analysis Blog | Aylien: http://blog.aylien.com/post/110191391148/superbowl-xlix-in-tweets-sentiment-analysis-of-4

Yahoo! Inc. (2015). *Yahoo Finance*. Retrieved from http://finance.yahoo.com/