CashTagNN: Exploiting the Use of CashTags to Predict Stock Market Prices Using Convolutional Networks

Neeraj Rajesh, Lisa Gandy
Central Michigan University
Mount Pleasant, MI
+1-989-774-3696
rajes1n, gandy1l@cmich.edu

ABSTRACT

In this paper, the authors present a system, CashTagNN, which uses the sentiment and subjectivity scores of tweets that include cashtags to model stock market movement, and in particular, predict opening stock market prices. Currently, the system focuses on two companies: Apple and Johnson \& Johnson. The system uses two machine learning methods for prediction, a feed-forward neural network, and a deep convolutional neural network. The authors use stock market prices in March and October 2016 as training data, with stock market prices in November 2016 as test data. The time series used for training and testing consists of stock prices recorded at one minute, five minutes, and one-hour intervals. Results show that the Feed Forward Network Model, in this case, outperforms the Deep Convolutional Network model.

CCS Concepts

• **Computing Methodologies** → **Neural Networks**
• **Information Systems** → **Web and Social media network**

Keywords

Convolutional Networks; Feed Forward Neural Networks; Stock Market Prediction; Social Media

1. INTRODUCTION

A significant amount of data available via Twitter has allowed researchers unprecedented access to real-time conversations about a wide array of topics. The prevalence of Twitter data has also allowed data scientists to predict a broad range of behaviors and occurrences such as flu trends [1], postpartum behavior [2],voting intentions [3] using tweets. Not surprisingly researchers have also begun to use Twitter data in regards to studying sales and the consumer industry. Chamlertwat et al. [4] have investigated how to predict product demands using Twitter, and Jaring et al. [5] among many others [6] [7] [8], have investigated how to market products on Twitter effectively. Moving to the stock market,researchers such as Bollen et al. [9], Souza [10] and Zheludev [11] have shown that sentiment of tweets predicts the movement of the stock market with high accuracy.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ICACS'20, January 6–8, 2020, Rabat, Morocco
© 2020 Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-7732-4/20/01…$15.00
DOI: https://doi.org/10.1145/3423390.3423392

This work is exciting as it suggests that researchers can use tweets as a variable in regards to stock market trading and investment. Though a significant amount of work has focused on stock market price prediction, the work has not utilized the presence of cashtags to concentrate on specific stocks. Cashtags are stock market symbols that used by stock market savvy tweeters in their tweets. When preceded by a dollars sign (for example $JNJ in regards to Johnson and Johnson) become clickable. By clicking on a cashtag, the user can also view other current tweets with the same cashtag. Cashtags are a valuable resource for researchers and traders as they allow both to focus on financial news or discussion regarding specific companies. This particular information is also useful because at times a companies stock market movement might not mirror that of the stock exchange as a whole. The authors demonstrate that by using tweets with cashtags along with measures of text sentiment, we can accurately model the stock movement of two highly traded stocks: Apple and Johnson & Johnson. The paper then discusses CashTagNN, a system that collects tweets with the appropriate cashtags, finds the polarity and subjectivity scores of these tweets using TextBlob and then uses both a feed forward neural network and a deep neural network to predict opening prices of the two stocks. Also, this work varies the two machine learning models with decreasing time slices of stock market prices. Finally, the authors discuss findings and conclude with future work.

2. PRIOR WORK

As noted in the introduction, Twitter data has been used to study a broad range of topics such as flu trends, postpartum behavior, voting intentions, among many others. In regards to consumer industry Chamlertwat et al. [4] have investigated how to predict product demands using Twitter, and Jaring et al. [5], among others [6] [7] [8], have investigated how to market products on Twitter effectively.

In regards to stock market prediction via tweets, Bollen et al. [9] have shown that sentiment analysis can be used to predict stock market movement as a whole. They did not focus on tweets that specifically referenced the stock market, but rather they gathered a wide variety of tweets (approximately 9 million) which contained the phrases ?I feel?, ?I am feeling?, ?I a?m feeling?,?I don?t feel?, ?I?m?, ?I am?, and ?makes me?. They then tested whether the mood of the public measured via these tweets would accurately predict stock market trading volumes on the Dow Jones Industrial Index. Their research concluded that 86.7% of cases there was a positive correlation between public sentiment and the upward/downward movement of the Dow Jones.

Souza et al. [10] find that the overall Twitter sentiment in regards to five retail companies (Abercrombie and Fitch, Nike, Home Depot, Mattel, and Gamestop) has a statistically significant relation between stock returns and volatility. Zheludev et al. [11] also contrast the information content of both the sentiment of

tweets and volume of tweets regarding their influence on stock prices. They find that the sentiment of tweets contains significantly more lead-time information about the costs than the number of tweets by itself, they look at a vast array of financial instruments and find that only twelve are statistically significant. Xu [12] uses prior prices of AAPL stocks from September 2007 to August 2012 along with hand coded news stories (positive or negative) from yahoo finance to predict stock prices. This prediction is completed using ARIMA time series analysis. Xu also uses Google Trend to measure the impact of each news article concerning AAPL.

This work serves as a complement to Ranco et al. [13] who examine tweets with cashtags representing companies on the Dow Jones Index. However, this work differs from Ranco in the following manner: first, Ranco et al. build an hourly times series whereas this paper uses time slices of one minute, five minutes and an hour. Secondly, Ranco et al. hand annotate sentiment whereas this study uses automated sentiment analysis throughout. Thirdly, Ranco's use of sentiment is divided into negative, positive or neutral categories whereas this paper uses a continuous sentiment designation. Finally, Ranco et al. use Pearson correlation and Granger causality tests for prediction whereas this work makes use of both feed forward and deep convolutional neural networks.

In regards to deep convolutional networks and financial markets, Chao et al. [14] use a deep belief network (DBN) to predict the British Pound/US dollar and Indian rupee/US dollar exchange rates. They compare the DBN method to a feedforward neural network. Their experiments indicate that deep belief networks achieve better performance than feedforward neural networks.

In regards to Deep Convolutional Networks and stock market prices, Ding et al. [15] model three events concerning Google as features (via a technique called event embeddings). They then use a deep neural network to refine the input event features further, and then use a pooling layer to extract the best global features per event. Finally, they use a feed forward neural network to extract to associate the features with stock trends with good results. A comparison of results listed in this paper with results from previous studies is given in Table 6.

This work is a refinement and continuation of our previous paper [16]. This paper uses the same data as our previous work; however, a different machine learning and data analytics approach is employed. In regards to the feed forward neural network, 1, 5- and 10-minute time slices are used in training, whereas previous work uses daily time slices. Using smaller time slices gives us more data and a finer grained view of tweets and their sentiment per day. This paper also includes the use of a convolutional neural network to evaluate the data which 1, 5, and 10-minute time slices.

3. CASHTAGNN SYSTEM

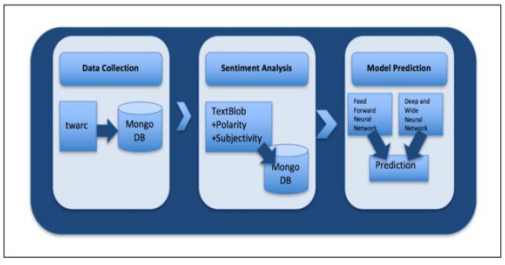


Figure 1. CashTagNN System Figure.

The CashTagNN System works in three stages: data collection and storage, sentiment assignment, and price prediction. In this section, the authors discuss each step in detail. For further reference, please view the system figure (Figure 1).

3.1 Machine Learning Model

3.1.1 Data Collection

In regards to data collection; the authors give the number of tweets collected per month (March, October, November 2016) in Table 1 In regards to data storage, the authors used the NoSQL database MongoDB [17]. The authors chose MongoDB as they preferred to keep data in the JSON data format, and MongoDB works well with JSON. Before storing data in MongoDB, the authors used twarc [18] to clean and filter out any information that was not needed. The database is indexed based on the date and time a tweet originates to increase lookup speed.

3.1.2 Attributes Used for Prediction

The next step in the system is to find the sentiment of the tweets collected. TextBlob [19] was used to this end. TextBlob is a natural language processing package built on top of the NLTK (Natural Language Toolkit) framework.

With regards to sentiment, the system uses two measures: subjectivity and polarity. Subjectivity is the measure of how subjective or conversely, objective, a text is. Polarity is the measure of how positive or negative the sentiment in the tweet is. In regards to the TextBlob scoring system, a text's subjectivity ranges from [0, 1] with zero being completely objective and one being entirely subjective. The Polarity ranges from [-1, +1], with -1 being very negative and +1 being very positive. The system labels a tweet objective if its subjectivity value is less than 0.5, and subjective if the value is greater than 0.5. The system considers a tweet to have positive sentiment if its polarity score is a positive number and to have negative sentiment if its polarity score is a negative number. The data used to train TextBlob's sentiment and subjectivity analysis are 2888 words scored for polarity, subjectivity, intensity, and reliability (this work uses only the polarity and subjectivity scores). These words included in TextBlob's training corpus are mostly adjectives with no nouns present.

In regards to the stock market, an example of a negative tweet (with a sentiment score of -0.025) would be: @grayund The iPhone was a decade long product. $AAPL cannot sustain 500 billion+ market cap over the next decade with iphone sales [20]. TextBlob also scores the tweet as slightly objective, with a score of 0.2. An example of a positive tweet, with a sentiment score of 0.29, would be 3,500% Gains On Stock Picks In 5 Months And An Amazing New Pick Is Coming Soon! Get It Early: https://t.co/Ikbnn3W31x $DO $ALL $AAPL [21]. TextBlob scored this tweet as slightly subjective with a score of 0.55. In Table 1 the authors give the numbers of positive/negative, subjective/objective tweets scored by the system. In regards to the data, there are ten attributes used as machine learning features (see the list below). The system uses nine of the attributes as predicting features, and the remaining attribute is the predicting class (closing stock price).

(1) Number of Positive Tweets
(2) Number of Negative Tweets
(3) Number of Objective Tweets
(4) Number of Subjective Tweets
(5) Average Polarity
(6) Average Subjectivity

(7) Total Subjectivity
(8) Total Polarity
(9) Opening Stock Price
(10) Closing Stock Price

Table 1: Number of Positive, Negative, Objective and Subjective Tweets

Company/Month	Total	#Pos	#Neg	#Obj	#Subj
AAPL / Mar 2016	122,047	112,892	9,155	78,589	43,458
AAPL / Oct 2016	129,320	116,787	12,533	87,317	42,003
AAPL / Nov 2016	95,051	86,244	8,851	62,362	32,733
JNJ / Mar 2016	203,224	197,000	6,224	180,039	23,185
JNJ / Oct 2016	141,579	135,130,	6,449	118,400	23,179
JNJ / Nov 2016	164,869	157,875	6,994	134,440	30,428

Table 2: Example of Features and Classes Used in CashTagNN

Time	# Pos	# Neg	# Obj	# Subj	Avg Pol	Avg Subj	Total Subj	Total Pol	Open Price	Close Price
9:31	5	0	5	0	0.02	0.12	0.10	0.62	96.87	97.08
9:32	1	0	1	0	0.00	0.15	0.15	0.00	97.90	97.32
9:33	2	0	1	1	0.20	0.45	0.90	0.40	97.09	97.31
9:34	12	1	11	2	0.04	0.18	2.40	0.50	97.13	97.12
9:35	1	0	1	0	0.24	0.37	0.24	0.37	97.12	97.15

These metrics apply to tweets and stock prices in each time slice. As an example, consider Table 2 above. The table includes data collected between 9:31 and 9:35 am on 2/29/16. In the table, the attributes # Pos, # Neg, # Obj and # Subj all apply to tweets created at each listed time. At the point, the system calculates the 6he total and average subjectivity and polarity scores. The system collects opening and closing prices from FinAM.ru. As described above, all attributes but the closing price are predicting features, and the system then predicts the closing price.

3.2 Prediction

The last task for the system is to predict the closing prices per company (Apple and Johnson and Johnson) based on the sentiment/subjectivity of tweets with the company's cashtags. Per this end, the system uses both a Feed Forward Neural Network and a Deep Convolutional Neural Network which are both described and compared below.

3.2.1 Feed Forward Neural Network

The first neural network used is a feed-forward neural net, created using the ffnet library [22]. The neural network uses nine attributes (nine inputs) and has a hidden layer of size 180 x 90 x 270 x 81 x 27. For ffnet neural networks, input units use the identity activation function, and all other units use the sigmoid activation function. FFNET automatically normalizes data both input and output, with a linear mapping to the range (0.15, 0.85). Also, the ffnet implementation considers input and output separately (the linear map is unique for each input and output). The function minimized during training is a sum of squared errors of each output for each training pattern.

In regards to JNJ data, Table 5 displays the results for median MSE scores. As can be observed, testing on October 1 minute, 5 minutes and 1-hour data results in a median MSE below 0.15. For testing on November 5 minute data, all October time slices had a median MSE below 0.2. Finally, for testing on November 1 hour data, all October time slices had a median MSE below 1.677.

3.2.2 Feed Forward Neural Network

To achieve the deep convolutional network learning model the authors use TensorFlow [23] The model created is a combination of a wide and deep model, and it works by summing the output logs of the TensorFlow graph. The data is parsed and bucketed into time slices as mentioned earlier and each of the inputs for the learning is passed along with the target and trained. The dimensions of hidden layers in the deep neural network are the same as those given in the feed forward neural network. The model uses the Adadelta optimizer.

4. RESULTS

4.1 Feed Forward Neural Network

For both AAPL and JNJ datasets training was implemented on March 2016 1 minute, 5 minute and 10 minutes data, as well as October 2016 1 minute, 5 minute and 10-minute data, and then tested on November 2016 1 minute, 5 minute and 1 hour time slices using ten fold cross validation. Table 3 below gives the median, mean squared error (MSE) for AAPL data. For 1-minute training data, all MSE scores are below 1.2, and for 5-minute training data, all MSE scores are below 2. In regards to 1-hour training data, only March 1 minute, 5 minute and 1-hour MSE have an MSE below 3.

In regards to JNJ data, Table 4 displays the results for median MSE scores. As can be observed, testing on October 1 minute, 5 minutes and 1-hour data results in a median MSE below 0.15. For testing on November 5 minute data, all October time slices had a median MSE below 0.2. Finally, for testing on November 1 hour data, all October time slices had a median MSE below 1.677.

Table 3: AAPL Results median MSE: a) FFNet training for 1 minute, 5 minute or 10 minute time slices in March or October and then testing each time slice on November 1 minute data, November 5 minute data, and November 1 hour data

	October 1 min	October 5 min	October 1 hour	March 1 min	March 5 min	March 1 hour
Nov 1 min	0.98	0.99	1.11	0.13	0.15	0.37
Nov 5 min	1.77	1.78	1.88	0.25	0.30	0.51
Nov 1 hour	14.86	14.69	14.64	2.91	2.78	2.86

Table 4: JNJ Results median MSE: a) FFNet training for 1 minute, 5 minute or 10 minute time slices in March or October and then testing each time slice on November 1 minute data, November 5 minute data, and November 1 hour data

	October 1 min	October 5 min	October 1 hour	March 1 min	March 5 min	March 1 hour
Nov 1 min	0.05	0.06	1.14	30.81	30.66	30.83
Nov 5 min	0.10	0.11	0.19	30.44	30.30	30.45
Nov 1 hour	1.68	1.64	1.20	34.21	34.07	34.21

Table 5: Median MSE JNJ and AAPL results using TensorFlow on November test data with a one to one time slice correspondence

	October 1 min	October 5 min	October 1 hour	March 1 min	March 5 min	March 1 hour
AAPL	5.96	2.95	55.21	6.79	6.89	5.96
JNJ	3.07	10.87	84.02	1.88	5.55	30.67

4.2 Deep Convolutional Network

To begin, the authors trained the TensorFlow model on time slices of 1 minute, 5 minutes and one hour for March 2016 and October 2016 and then tested the data on equivalent time slices for November 2016 using ten fold cross validation for both AAPL and JNJ data points.

The authors did not use the one to many models featured in the Feed Forward Network as the setup time per training and testing was time prohibitive. Table 5 provides the median MSE per data point. Only October 5 minute data had a median below three (2.95). In regards to JNJ data, only two data points (October 1 minute and March 1 minute) had a data point close to or less than 3.

5. DISCUSSION AND FUTURE WORK

The FFNet machine learning model was significantly more accurate than the TensorFlow model, as the TensorFlow model only had an MSE of close to or below 3 (2.95) for three data points, whereas a majority of FFNet results had an MSE below three. In regards to FFNet results, we see a sharp increase in the MSE between one and five minute data (less than 0.153 and less than 0.188 respectively) to a median MSE of 1.677 for one hour time slices. We speculate that the model does not capture in fine enough granularity the movement of the stock market and the sentiment of tweets at the one-hour time slice. How- ever a median MSE of 1.677 is still small, and therefore could be a relatively accurate predictor of stock market prices.

In regards to FFNet data, we see that JNJ prices appear to be seasonal as October test data can accurately predict November prices but is not accurate when predicting prices in March. Conversely, in regards to AAPL data, October data more accurately predicts March data than November data, but not as strong as that of JNJ. In the future, the authors plan to tune further the Deep Convolutional network as well as test a one to many model such as the one we used in FFNet. The authors, however, feel that this test will not reveal different results as this work tested the primary data points. The authors do plan to, however, test a combination of Deep Convolutional Network and a feed forward network to see if the combination might yield better results than that of the two models alone.

Table 6: Comparison of CashTagNN with other systems

Paper	Twitter Data	Stock Data	Method	Results
Ranco et al.	DJIA tweets June 2013 - Sept. 2014	DJIA June 2013 - Sept. 2014	Cumul Abnormal Returns	$\alpha \leq .01$
Bollen et al.	10% tweets Feb. - Dec. 2008	DJIA Feb. - Dec. 2008	Self Organizing FNN	87.6% accuracy
Zheludev et al.	10% tweets Dec 2012 - March 2013	S&P 500 Index Futures Dec 2012 - March 2013	Mutual Info	$\alpha \leq .01$
Souza et al.	Twitter SemEval 2013	ANF, NKE, HD, MAT, GME Nov 2013 - Sept 2014	Granger	$p \leq 0.05$
Rajesh et al.	tweets AAPL, JNJ Mar, Oct, Nov 2016	AAPL JNJ Mar, Oct, Nov 2016	FFNN	MSE≤2
Rajesh et al.	tweets AAPL, JNJ Mar, Oct, Nov 2016	AAPL JNJ Mar, Oct, Nov 2016	Deep Conv Network	MSE≤3

6. REFERENCES

[1] Alex Lamb, Michael J Paul, and Mark Dredze. 2013. Separating Fact from Fear: Tracking Flu Infections on Twitter.. In HLT-NAACL. 789–795.

[2] Munmun De Choudhury, Scott Counts, and Eric Horvitz. 2013. Predicting post-partum changes in emotion and behavior via social media. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 3267–3276.

[3] DanielGayo-Avello.2013.A meta-analysis of state-of-the-art electoral prediction from Twitter data. Social Science Computer Review (2013), 0894439313493979.

[4] Wilas Chamlertwat, Pattarasinee Bhattarakosol, Tippakorn Rungkasiri, and Choochart Haruechaiyasak. 2012. Discovering Consumer Insight from Twitter via Sentiment Analysis. J. UCS 18, 8 (2012), 973–992.

[5] Päivi Jaring, Asta Bäck, Marko Komssi, and Juha Käki. 2015. Using Twitter in the acceleration of marketing new products and services. Journal of Innovation Management 3, 3 (2015), 35–56.

[6] Xi Y Leung, Billy Bai, and Kurt A Stahura. 2015. The marketing effectiveness of social media in the hotel industry a comparison of facebook and twitter. Journal of Hospitality & Tourism Research 39, 2 (2015), 147–169.

[7] Craig Dempster, David S Williams, and John Lee. 2015. The Rise of the Platform Marketer: Performance Marketing with Google, Facebook, and Twitter, Plus the Latest High-growth Digital Advertising Platforms. John Wiley & Sons.

[8] Sharad Goel, Jake M Hofman,Sébastien Lahaie,David M Pennock,and DuncanJ Watts. 2010. Predicting consumer behavior with Web search. Proceedings of the National academy of sciences 107, 41 (2010), 17486–17490.

[9] Johan Bollen, Huina Mao, and Xiaojun Zeng. 2011. Twitter mood predicts the stock market. Journal of Computational Science 2, 1 (2011), 1–8.
[10] Thársis Tuani Pinto Souza,Olga Kolchyna,Philip C Treleaven,and Tomaso Aste. Twitter sentiment analysis applied to finance: A case study in the retail industry. arXiv preprint arXiv:1507.00784 (2015).
[11] Ilya Zheludev, Robert Smith, and Tomaso Aste. 2014. When can social media lead financial markets? Scientific reports 4 (2014), 4213.
[12] Selene Yue Xu. 2014. Stock Price Forecasting Using Information from Yahoo Finance and Google Trend. URL https://www. econ. berkeley.edu/sites/default/files/Selene% 20Yue% 20Xu. pdf (2014).
[13] Gabriele Ranco, Darko Aleksovski, Guido Caldarelli,Miha Grčar,and Igor Mozetič. 2015. The effects of Twitter sentiment on stock price returns. PloS one 10, 9 (2015), e0138441.
[14] Jing Chao, Furao Shen, and Jinxi Zhao. 2011. Forecasting exchange rate with deep belief networks. In Neural Networks (IJCNN), The 2011 International Joint Conference on. IEEE, 1259–1266.
[15] Jing Chao, Furao Shen, and Jinxi Zhao. 2011. Forecasting exchange rate with deep belief networks. In Neural Networks (IJCNN), The 2011 International Joint Conference on. IEEE, 1259–1266.
[16] Neeraj Rajesh and Lisa Gandy. 2016. CashTagNN: Using sentiment of tweets with CashTags to predict stock market prices. In Intelligent Systems: Theories and Applications (SITA), 2016 11th International Conference on. IEEE, 1–4.
[17] Kyle Banker. 2011. MongoDB in action. Manning Publications Co.
[18] Ed Summers. 2016. GitHub twarc. (2016). https://github.com/edsu/twarc.
[19] Steven Loria. 2014. TextBlob: simplified text processing. Secondary TextBlob: Simplified Text Processing (2014).
[20] MLGY Family. 2016. @grayund The iphone was a decade long product. $AAPL cannot sustain 500 billion+ market cap over the next decade with iphone sales. Twitter (16 2 2016).
[21] BinaryOptionsGuard. 2015. RT https://twitter.com/Stockfiendz/status/675714628144508929? 3,500% Gains On Stock Picks In 5 Months And An Amazing New Pick Is Coming Soon! Get It Early. Twitter (12 12 2015).
[22] M Wojciechowski. 2011. Ffnet: Feed-forward neural network for python, http://ffnet.sourceforge.net/ (2011).
[23] Martín Abadi and et. al. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems http://tensorflow.org/ Software available from tensor-flow.org.