Influencer Tweet NLP for Equity Trading Signals

Roman Paolucci

James Madison University

Abstract

Artificial intelligence posits a unique opportunity to analyze large amounts of text data for the development of equity trading models. This paper analyzes the forecasting power and trading signal generated by tweets containing the hashtag TSLA (#TSLA) from *influencers* with a high follower count (>10,000 followers) against those with a low follower count (≤10,000). Using a combination of Word2Vec and a scoring system laid out herein compared to a generic VADER model, it is generally observed that influencers generate better trading signals.

I. Prior Literature

In a financial setting, it isn't uncommon to use AI and NLP to develop short term forecasting models and trading signals. Rao and Srivastava (2012) show a high correlation (.88) between market returns and the sentiment of short-term Twitter discussions with a forecasting R^2 of over 95%[1]. Smailović, Grčar, Lavrač, and Žnidaršič (2013) use support vector machines to classify tweets as having a positive, negative or neutral sentiment that were capable of indicating a stock price movement to occur in the following days. These results support the notion of a behavioral response to public mood and opinion, and they don't stop there. Bollen, Mao, and Zeng (2011) use a number of dimensions to classify mood (including calm, alert, sure, vital, kind, and happy). They find by including certain dimensions of public mood, 86.7% of daily up and down changes in the DJIA can be predicted.

There is significant empirical evidence suggesting public mood and opinion carry weight in short term future returns. Behrendt, and Schmidt (2018) take it a step further and use high-frequency Twitter data to attempt to forecast intraday movements for individual equities. The interesting thing to note about their research is the lack of improvement and significance of high-frequency tweet sentiment at the intraday tick level. Nevertheless, these results still validate the previously described behavioral response to public mood, as it can be argued that the majority does not use or have access to high-frequency tweet streaming capabilities. Further, it would be fair to perpetuate the idea that there is a range of time tweet sentiment can be productive to forecasting power and trading signal. These results in aggregate lead us to hypothesize that Twitter influencer opinion and mood may carry more weight in explaining a behavioral response.

[1] It is worth noting here that a large coefficient of determination and a valid trading signal are two different animals. This is further discussed in the chapters on regression and trading signal analysis.

II. Data

This article analyzes tweets categorized by accounts with high follower counts (>10,000, designated as influencers in this article) and low follower counts (≤10,000), containing the hashtag TSLA (#TSLA*)*. Retweets were filtered out--only original tweets were considered. Twitter's Developer API restricts tweet queries to the previous week. Consequently, 2,000 tweets were pulled at an arbitrary time every business day for 8 weeks. The result after categorization was two matrices containing a total of 78,000 tweet objects as an observation indexed by time. The tweet object contains relevant information including favorites, retweets, if the account is verified, follower count, and of course, tweet text. This article will focus specifically on the tweet text field. After the analysis laid out herein, each time index (day) receives an average *VADER* and *Lexicon Sensitivity* score which is matched to the return[2] of TSLA for the *following* day[3]. The result is a set of forecasting regression that determines the amount of variability in the following day's TSLA returns explained by tweets from influencers and accounts with low follower counts. These regressions are discussed in chapters IV and V.

III. Natural Language Processing

a.) Cleaning and Preprocessing

The tweet text itself is in poor condition for mathematical computations due to the unstructured nature of tweet text. To combat this, standard natural language processing techniques were used to clean and preprocess the *documents* (where a *document* is a tweet's text). This process included removing punctuation, case sensitivity, and stop words from each

[2] Future return is computed by the following day's opening price to closing price for actionable trading
[3] Friday scores are assigned to the following Monday's return

document. Cleaning and preprocessing allow for more effective numerical representations of words in each document and act as a sort of dimensionality reduction.

b.) The Unlabeled Problem

Though it may be obvious, there is no dataset comprised of millions of tweets containing the hashtag TSLA that is labeled on a scale of [-1, 1] for *financial sentiment*. This is a common issue in natural language processing, as the problem requires unsupervised learning solutions rather than supervised ones. Fortunately, there exists a wide variety of libraries and pretrained unsupervised models for *sentiment* analysis, but they come with some caveats. The caveats of these pretrained models are seen in the difference between what we define as *sentiment* and *financial sentiment*.

c.) Valence Aware Dictionary and sEntiment Reasoner (VADER) Model

VADER is a pretrained sentiment analysis model open-sourced under the MIT License. This model was developed specifically to detect and classify sentiment from social media text. Take for example the document "*Tesla is the worst stock ever".* After successfully downloading and importing the VADER model we can compute a polarity score with the following output…

'neg': 0.451, 'neu': 0.549, 'pos': 0.0, 'compound': -0.6249

The model can identify sentiment of this arbitrary document as overall negative (reflected in the negative compound score). The model is also capable of identifying a change of tone. Consider the document "*Tesla is the worst, but I love it anyway!".* This results in the following score…

'neg': 0.153, 'neu': 0.481, 'pos': 0.366, 'compound': 0.6749

The model acknowledges the change in sentiment consequent of "*but I love it anyway!",* impressive! Though this model is very effective at classifying positive, negative, and neutral

tones in *general,* the specific lexicon pertaining to the financial industry makes this model less than optimal. Consider the document "*I am buying Tesla stock"* with the subsequent score…

'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0

The tone detected by VADER is neutral, why? In general, the VADER model is trained for general sentiment analysis. This means financial lexicon lacks weight in either direction, words like *buy* and *long* or *sell* and *short* do not contribute to the sentiment score as we would like them to. Further training of this model with industry specific lexicon using a labeled dataset can aid in this issue. However, as mentioned previously, there does not exist a large dataset of tweets containing the hashtag TSLA labeled for this *financial sentiment*. If there were this would instead be a supervised learning problem, and a variety of other models and techniques would be at our disposal. Instead, this article explores and develops a new model for assigning *financial sentiment* scores to documents using what we are calling a *Lexicon Sensitivity Model.*

d.) Word Embeddings

The datasets are now prepared to be converted to word embeddings. One way to do this is to one-hot encode each document for use with linear algebra operations. The issue with one-hot encoding is the lack of a projection of one word into another's dimensional space. That is, each token (word) is orthogonal to one another in an N-dimensional space. Consider the tweet (disregard preprocessing and cleaning) "*Tesla is a buy, invest*.", one-hot encoding would result in the following vectors:

Tesla = [1, 0, 0, 0, 0]

is = [0, 1, 0, 0, 0]

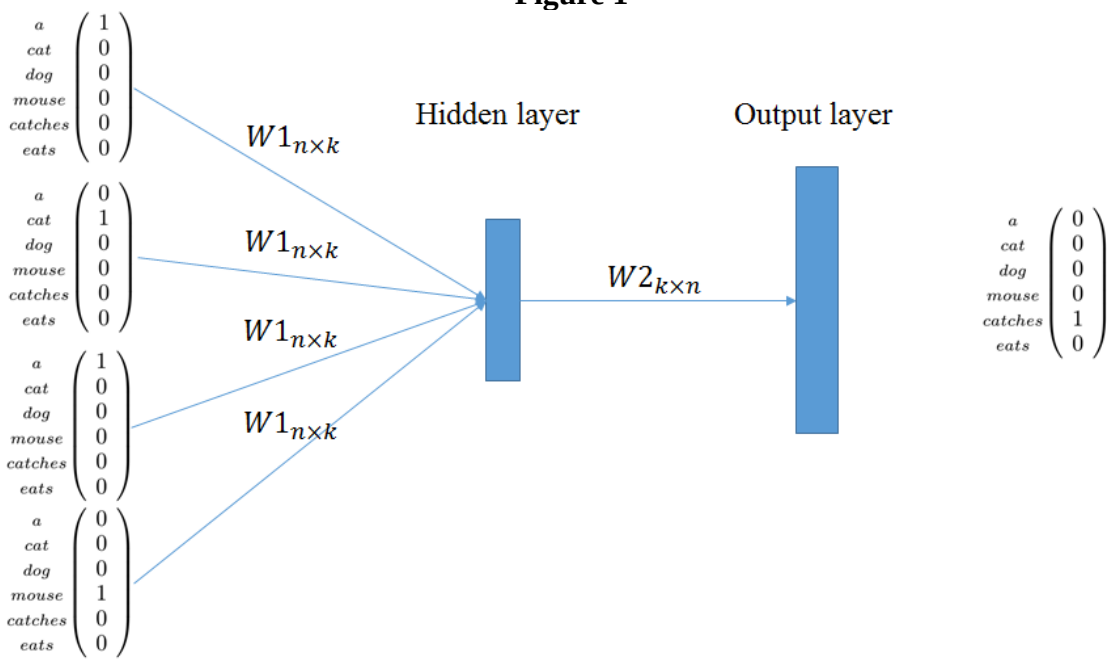
a = [0, 0, 1, 0, 0]

buy = [0, 0, 0, 1, 0]

invest = [0, 0, 0, 0, 1]

Consequent linear algebra operations will have the words "*Tesla"* and "*is"* as similar as "*buy"* and "*invest,* which obviously isn't the case. To combat this, and develop better word embeddings, we use a common trick by building and training a neural network for the task of predicting subsequent words. Instead of using the model as intended, we strip the weights from the hidden layer to use as the word embeddings. The notion is quite simple: given a matrix of documents, each document is used to train a neural network with a single hidden layer to predict the following word given the preceding one (from an input of one-hot encoded documents). Fortunately, *Word2Vec* with a CBOW approach in Python makes this process incredibly easy. Figure 1 depicts this idea with an arbitrary document and a *window size* (the number of preceding words in the document to be used as an input for the following target word) of 4.

Figure 1[4]



Through standard backpropagation, the neural network's loss function is optimized to minimize the error of predictions. The tweets from those with low follower counts were used to

[4] From Ferrone and Zanzotto (2019) and their work on discrete symbol representation in neural networks.

train the neural network. Using Word2Vec with a window size of 3 and a minimum word frequency of 100, the weight matrix was extracted from the hidden layer and used as word embeddings. The condition of minimum frequency ensures that every word in the document appears at least 100 times within the training set. We will now find that we can compute the similarity between words by taking the cosine of the angle between two word vectors: the *cosine similarity*. Originally, the cosine similarity of two different words after one-hot encoding is zero, as the vectors are orthogonal to one another. However, after stripping the weight matrix from the neural network to use as word embeddings, we find that we can quantify the similarity of two words in the N-dimensional space.

In this case, we find that the cosine similarity of words that are related to investing have a high degree of similarity (e.g. the words *buy* and *sell* in either dataset have a similarity score close to 1). This is incredibly informative and provides evidence Word2Vec was productive. However, the goal of this research is to conduct financial sentiment analysis on tweets to determine a positive, neutral or negative tone for use in a forecasting regression and trading signal. To accomplish this, we further unsupervised classification of text with industry specific lexicon by developing a model dependent on the word embeddings from Word2Vec.

e.) Lexicon Sensitivity Model

The *Lexicon Sensitivity Model* was developed to better classify tweets containing jargon related to a specific industry. The notion is quite simple. Following the methodology used to develop word embeddings, each word in a document can be expressed as a vector. This vector can be compared to words that have *financial connotations* by their cosine similarity value. Words like *buy* or *long* would be considered generally positive, while words like *sell* or *short* would be considered generally negative. Take for example the preprocessed document

containing the words [*buy, shares, tesla*]. In the sentence, the cosine similarity is cross computed for each word within two sets of prespecified words considered generally positive and negative in financial context.

Figure 2

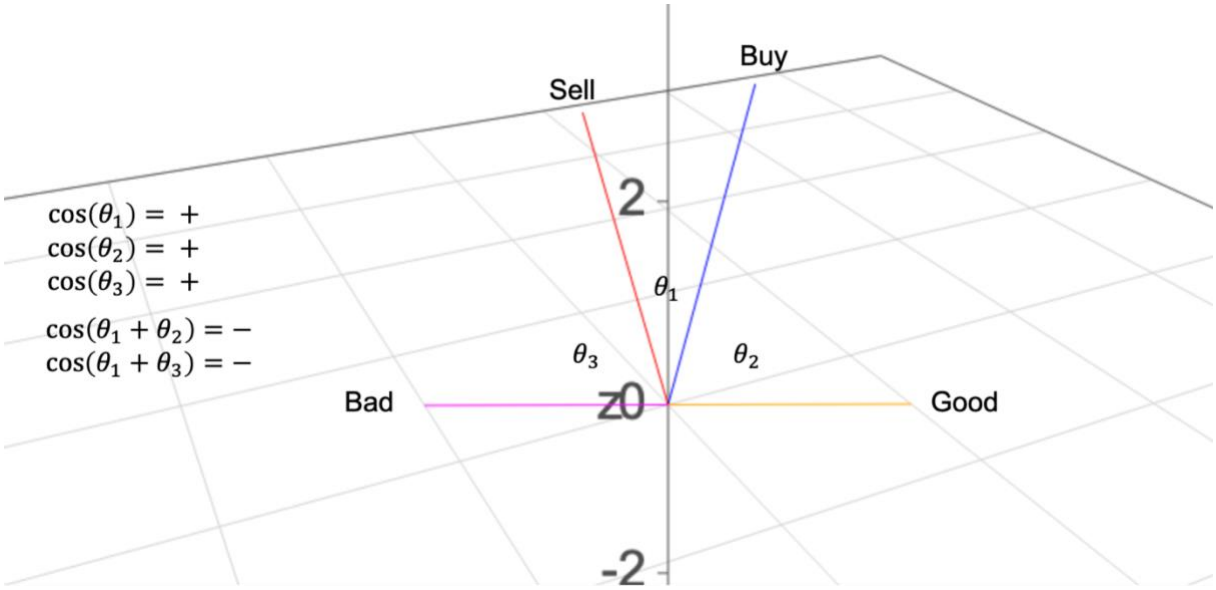


Figure 2 displays this notion in a three-dimensional space to aid in visualization. The word embeddings for the words *buy, sell, good,* and *bad* are drawn in this space. Those particularly keen to what is going on will acknowledge the representation of four words in a three-dimensional space, which does not accurately reflect the Word2Vec process of extracting word embeddings from documents. Word2Vec would ensure each word had a representation in four-dimensions. Nevertheless, for the sake of this aid, consider the word bad as being derived from the word good. To understand the mechanics of the *Lexicon Sensitivity Model,* recall the cleaned and preprocessed document [*buy, shares, tesla*]. Consider the cosine of the angle between vectors *buy* and *good*. The result, since the angle is less than 90 degrees, will be positive. This value, since the word *good* is a part of our set of words with positive financial

intent, is then added to the sentences scoring sum. This same process is repeated for the set of words with negative financial intent. Further, consider the cosine of the angle between the words *buy* and *bad*. Notice how the angle is greater than 90 degrees. The resulting score will be negative; however, this value is *subtracted* from the overall sentence score as the word *bad* is a part of our set of words with negative financial intent; meaning the value added to the score will be positive (further representing the intent associated with this document in financial context). This process continues until the similarity score is computed for each word in the sentence against each word in the sets representing positive or negative financial intent. The result is a score that represents the similarity of each word in the sentence to *prespecified financial lexicon*. It is worth noting that the prespecified financial lexicon is ensured to exist in the word embeddings. It's also worth noting that arbitrary words like *shares* also have a place in the N-dimensional space that is the word embeddings. But using the three-dimensional example given above, its location in the space relative to words like *buy* and *sell* (and other financial lexicon for that matter) contribute to the sentence's overall *financial sentiment* score.

This model derives more accurate financial sentiment scores.

"Doubling down on Tesla to the moon!" → Positive Financial Intent

Lexicon Sensitivity Score: Positive, (1, 0.0063)

VADER Polarity Score: Neutral, {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}

"Selling all shares of Tesla" → Negative Financial Intent

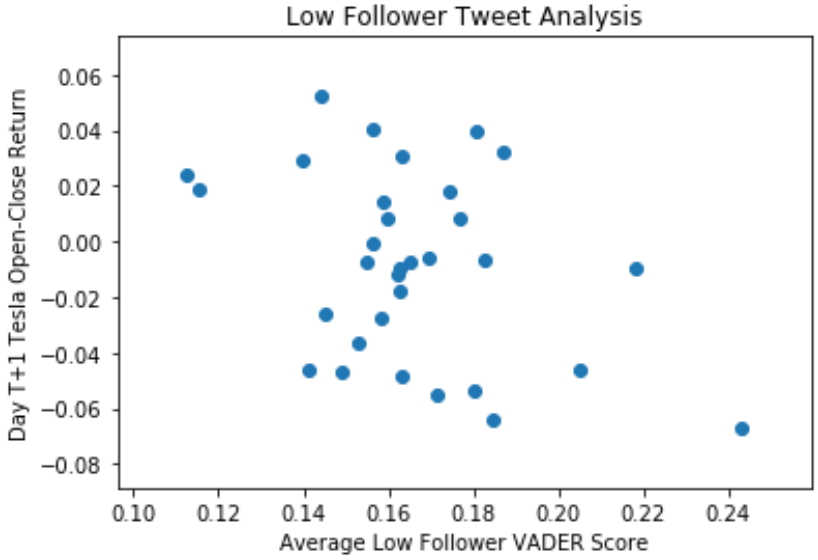
Lexicon Sensitivity Score: Negative, (0, -0.2098)

VADER Polarity Score: Positive, {'neg': 0.0, 'neu': 0.645, 'pos': 0.355, 'compound': 0.296}

IV. VADER Regressions

The following regressions take the average VADER sentiment score for each tweet on day T and look at their explanatory power against the following day's (T+1) return of TSLA from open to close. Figure 3 displays the results for the low follower tweets scored with the VADER model.

Figure 3



Interestingly, the result is a weak negative linear relationship between average daily low follower tweet VADER sentiment score and the following day's open to close return for TSLA. The regression considers the heteroscedasticity of TSLA returns, and the OLS regression results can be seen in Figure 4. It is critical to note that though there is some statistically significant relationship identified, it is entirely possible these results were generated coincidentally. The lack of a behavioral explanation for the negative linear correlation furthers this notion; it will help to identify the potential of this scoring system as a trading signal to determine whether the results are meaningful. This is further discussed in chapter VI.

Figure 4

OLS Regression Results

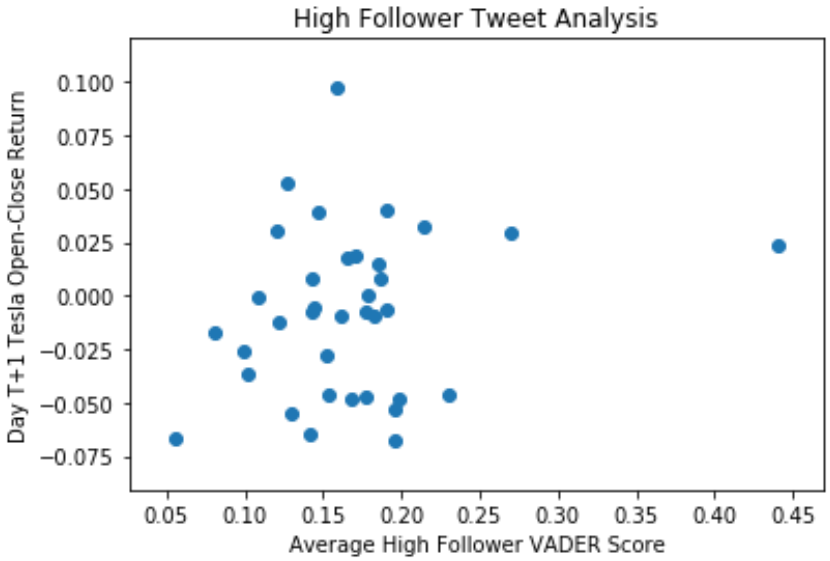
Dep. Variable:	y	R-squared:	0.134
Model:	OLS	Adj. R-squared:	0.105
Method:	Least Squares	F-statistic:	7.485
Date:	Sun, 14 Mar 2021	Prob (F-statistic):	0.0103
Time:	21:58:25	Log-Likelihood:	66.021
No. Observations:	32	AIC:	-128.0
Df Residuals:	30	BIC:	-125.1

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.0703	0.029	2.421	0.015	0.013	0.127
x1	-0.4774	0.175	-2.736	0.006	-0.819	-0.135

Omnibus:	2.773	Durbin-Watson:	1.923
Prob(Omnibus):	0.250	Jarque-Bera (JB):	1.361
Skew:	0.061	Prob(JB):	0.506
Kurtosis:	1.997	Cond. No.	40.6

Figure 5 and Figure 6 depict the same regression process for the high follower dataset.

Figure 5



These results are interesting when compared to the low follower dataset. Using the VADER model with the high follower dataset, a regression shows a positive linear relationship between average daily high follower tweets and the following day's open to close return on TSLA. This carries more weight behaviorally as accounts with large audiences that tweet generally positive things on day T about TSLA may explain some variability in returns for the

following day. This is more compelling (behaviorally) than a negative relationship exhibited in the low follower returns – but again, this is further considered in chapter VI.

Figure 6

OLS Regression Results

Dep. Variable:	y	R-squared:	0.041
Model:	OLS	Adj. R-squared:	0.012
Method:	Least Squares	F-statistic:	3.980
Date:	Sun, 14 Mar 2021	Prob (F-statistic):	0.0541
Time:	21:58:15	Log-Likelihood:	67.697
No. Observations:	36	AIC:	-131.4
Df Residuals:	34	BIC:	-128.2

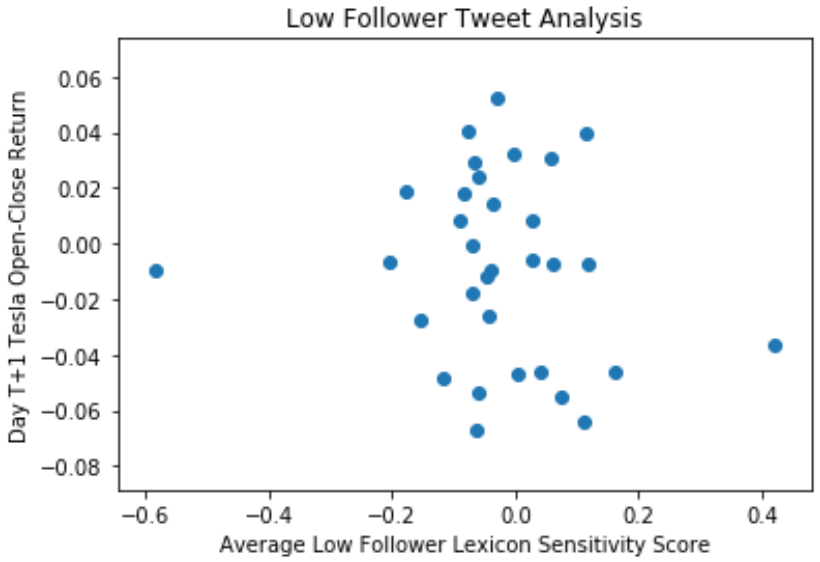
	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.0286	0.013	-2.259	0.024	-0.053	-0.004
x1	0.1220	0.061	1.995	0.046	0.002	0.242

Omnibus:	2.302	Durbin-Watson:	2.398
Prob(Omnibus):	0.316	Jarque-Bera (JB):	1.432
Skew:	0.475	Prob(JB):	0.489
Kurtosis:	3.230	Cond. No.	16.5

V. Lexicon Sensitivity Model Regression

The same regression process was used with the Lexicon Sensitivity model to derive the average daily sentiment scores. Figure 7 and Figure 8 show this regression process with the low follower dataset.

Figure 7



Clearly, the model considering the relationship tweets have to industry specific jargon in sentiment scoring suppressed both the explainable power and statistical significance generated by the VADER model.

Figure 8

OLS Regression Results

Dep. Variable:	y	R-squared:	0.024
Model:	OLS	Adj. R-squared:	-0.008
Method:	Least Squares	F-statistic:	1.621
Date:	Sun, 14 Mar 2021	Prob (F-statistic):	0.213
Time:	21:52:41	Log-Likelihood:	64.120
No. Observations:	32	AIC:	-124.2
Df Residuals:	30	BIC:	-121.3

	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.0096	0.006	-1.623	0.105	-0.021	0.002
x1	-0.0339	0.027	-1.273	0.203	-0.086	0.018

Omnibus:	2.658	Durbin-Watson:	1.957
Prob(Omnibus):	0.265	Jarque-Bera (JB):	1.320
Skew:	0.004	Prob(JB):	0.517
Kurtosis:	2.005	Cond. No.	6.57

Figure 9 and Figure 10 show this same regression process with the high follower dataset.

Figure 9

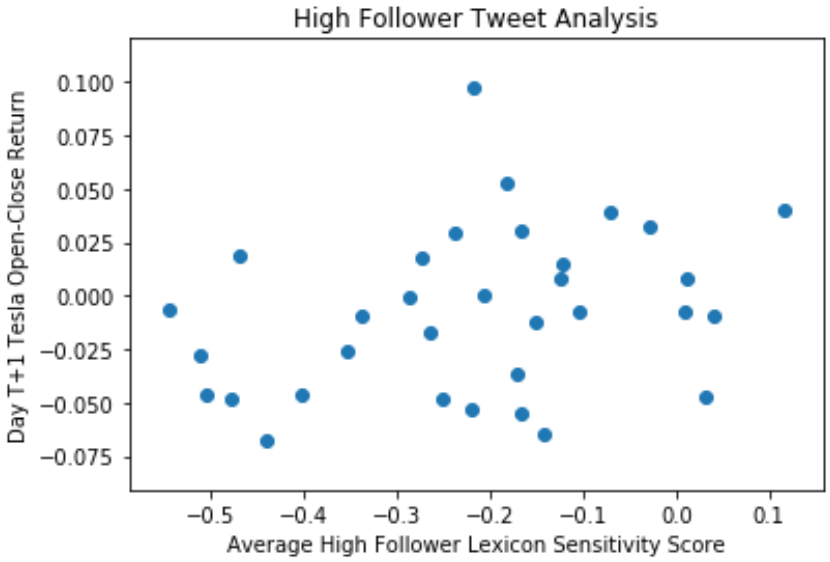


Figure 10

OLS Regression Results

Dep. Variable:	y	R-squared:	0.096
Model:	OLS	Adj. R-squared:	0.067
Method:	Least Squares	F-statistic:	5.674
Date:	Sun, 14 Mar 2021	Prob (F-statistic):	0.0235
Time:	21:52:47	Log-Likelihood:	63.116
No. Observations:	33	AIC:	-122.2
Df Residuals:	31	BIC:	-119.2

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.0073	0.009	0.788	0.431	-0.011	0.025
x1	0.0674	0.028	2.382	0.017	0.012	0.123

Omnibus:	2.819	Durbin-Watson:	2.383
Prob(Omnibus):	0.244	Jarque-Bera (JB):	1.645
Skew:	0.499	Prob(JB):	0.439
Kurtosis:	3.447	Cond. No.	6.07

The average daily lexicon sensitivity score from tweets by accounts with high followers show a weak positive linear relationship that explains more variability in the following day's open to close return of TSLA than the VADER model. However, in terms of R^2, the high follower dataset at its best ($R^2$of 9.6%) is still less than the low follower dataset at its best ($R^2$of 13.4%). The next chapter makes an argument using trading signal analysis that high follower tweets generate better trading signals using the Lexicon Sensitivity model than any other combination of model and dataset laid out herein.

VI. Trading Signal Analysis

This chapter determines if the scores derived by the tweets analyzed by the VADER and Lexicon Sensitivity model (and the two datasets of tweets) have potential as a trading signal to be deployed in an algorithmic trading system. The R^2 of a signal against the future return of its traded security is one thing, but if the signal broken down into quartiles does not have a monotonic increase it is entirely possible the regression's results are being generated coincidentally. Figure 11 depicts what would be considered a promising trading signal.

Figure 11

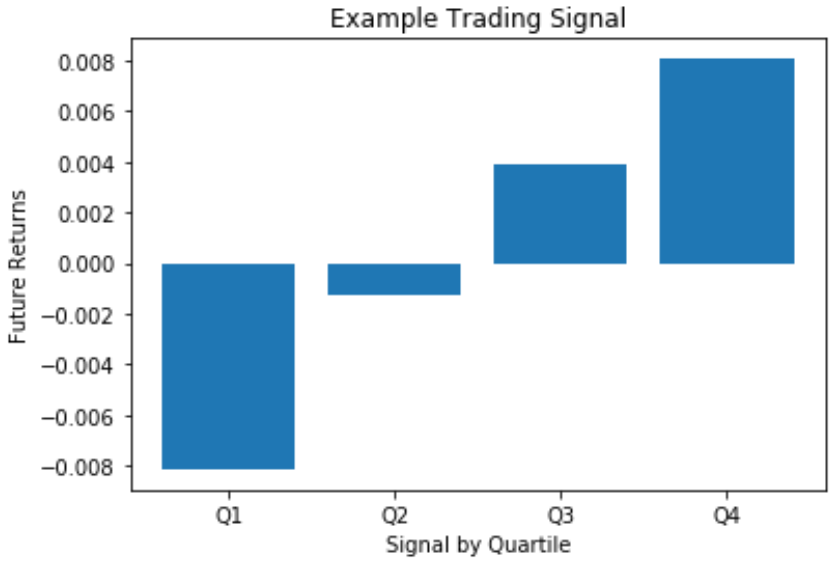


Figure 11 depicts a trading signal broken down into its quartiles. Each bar is the average return generated by the signals in that quartile. Notice how the first quartile contains most of the negative returns, and the fourth quartile contains most of the positive returns, while the intermediate quartiles preserve the previously stated desired monotonic increase. Figure 12 shows the trading signal generated by the low follower dataset with the VADER scoring system.

Figure 12

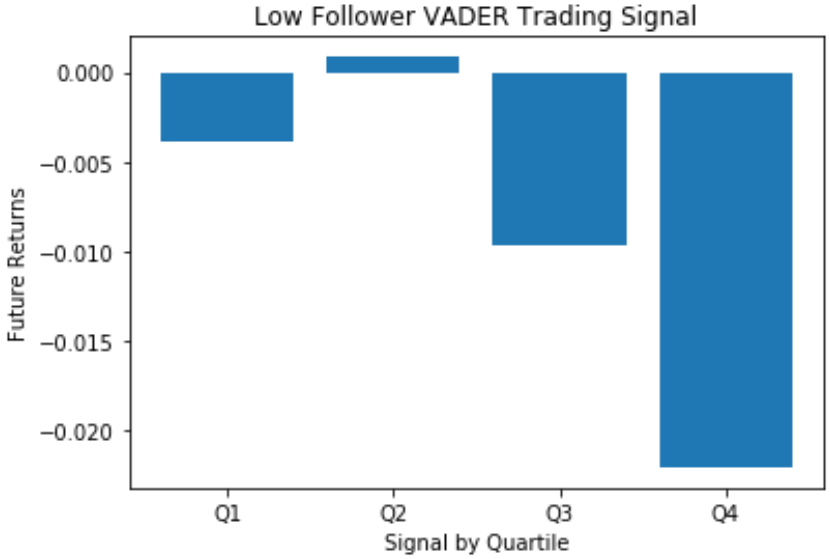


Figure 12 shows a generally stochastic signal generated by the low follower tweets and VADER scores. This is evidence that the low follower regression with the VADER model was generating the results randomly. As the signal value (average daily VADER sentiment score) increases the average returns for that bucket should also be increasing. Similar results are seen in both the high follower tweets with VADER scores (Figure 13) and low follower tweets with lexicon sensitivity scores (Figure 14).

Figure 13

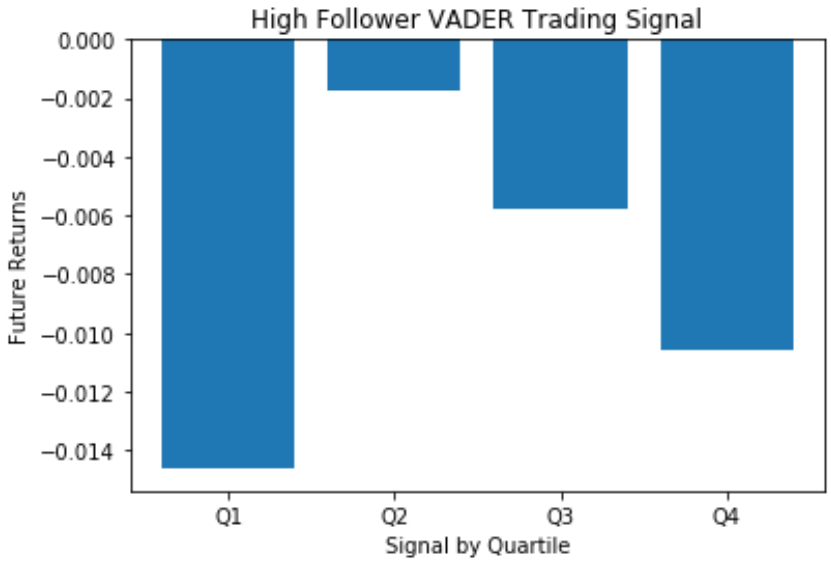
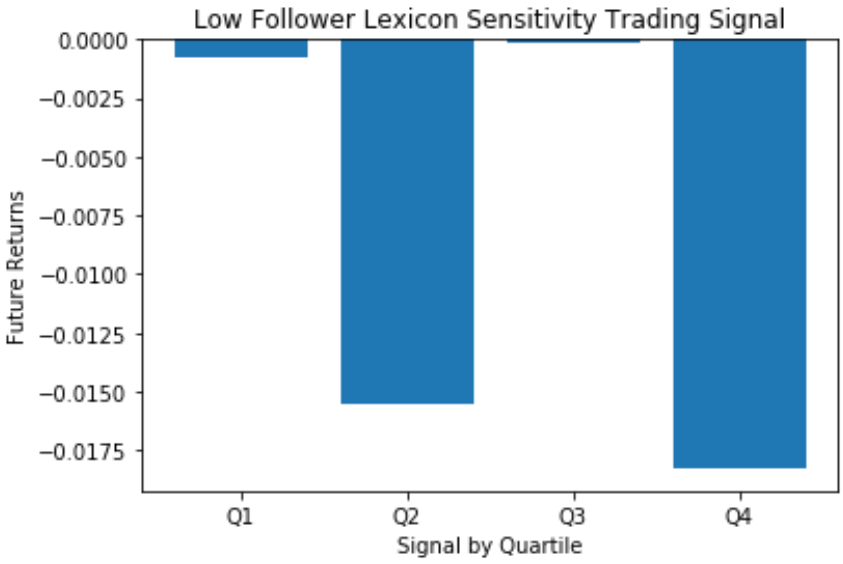
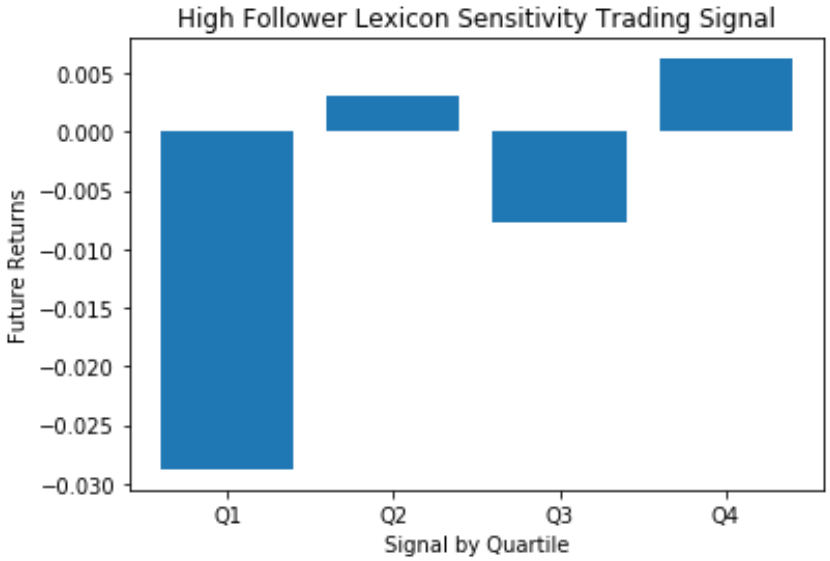


Figure 14



There is, however, one exception to these poor trading signals. Figure 15 shows the combination of the high follower dataset with lexicon sensitivity scores.

Figure 15



Though the relationship is still far from what we would like to see, it is much more promising than the seemingly random trading signals generated by the other combinations of models and data.

VII. Conclusion

In this article we provide evidence that, using a Lexicon Sensitivity model, tweets from accounts with a high follower count generate better trading signals than any other combination of model and data. There are several ways to improve on the findings of this article, including latent semantic analysis of tweets for the generation of buy and sell word sets in a Lexicon Sensitivity model and lemmatization for use in VADER and other models. More complex analysis in these domains may develop better models which in turn produce higher quality trading signals and forecasting regressions.

References

3D Vector Plotter. https://academo.org/demos/3d-vector-plotter/

Behrendt, S., & Schmidt, A. (2018, September 29). The twitter myth revisited: Intraday investor sentiment, twitter activity and individual-level stock return volatility. https://www.sciencedirect.com/science/article/pii/S0378426618302115?casa_token=BDRx-fj291sAAAAA%3AlzjnEnXJkBzKrXIB4-IKv2VU8gip2c-unNfBHfDvT_F3OGWERvLKc4WQv5S64PtCi2a_c0_CmB4

Beri, A. (2020, May 27). Sentimental Analysis using VADER. https://towardsdatascience.com/sentimental-analysis-using-vader-a3415fef7664

Bollen, J., Mao, H., & Zeng, X. (2011, February 02). Twitter mood predicts the stock market. https://www.sciencedirect.com/science/article/abs/pii/S187775031100007X

Burchell, J. (2017, April 8). Using VADER to Handle Sentiment Analysis with Social Media Text. https://t-redactyl.io/blog/2017/04/using-vader-to-handle-sentiment-analysis-with-social-media-text.html

Cjhutto. VADER Github Repository. https://github.com/cjhutto/vaderSentiment

Ferrone, Lorenzo & Zanzotto, Fabio Massimo. (2019). Symbolic, Distributed, and Distributional Representations for Natural Language Processing in the Era of Deep Learning: A Survey. 10.3389/frobt.2019.00153.

Karani, D. (2020, September 02). Introduction to Word Embedding and Word2Vec. https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa

Kirlić, Ajla and Orhan, Zeynep and Hasovic, Aldin and Kevser-Gokgol, Merve, Stock Market Prediction Using Twitter Sentiment Analysis (January 2, 2018). Invention Journal of Research Technology in Engineering & Management (IJRTEM), Volume 2 Issue 1 ǁ January. 2018 ǁ PP 01-04, Available at SSRN: https://ssrn.com/abstract=3266569

Li, Z. (2019, June 01). A beginner's guide to word embedding With Gensim word2vec model. https://towardsdatascience.com/a-beginners-guide-to-word-embedding-with-gensim-word2vec-model-5970fa56cc92

Rao, T., & Srivastava, S. (2012). Analyzing Stock Market Movements Using Twitter Sentiment Analysis. http://eprints.lincoln.ac.uk/id/eprint/11274/1/ASONAM%202012.pdf

Reuters. (2020, July 20). These retail investors bet big on TESLA early on and are now reaping the riches. https://financialpost.com/investing/these-retail-investors-bet-big-on-tesla-early-on-and-are-now-reaping-the-riches

Smailović J., Grčar M., Lavrač N., Žnidaršič M. (2013) Predictive Sentiment Analysis of Tweets: A Stock Market Application. In: Holzinger A., Pasi G. (eds) Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data. HCI-KDD 2013. Lecture Notes in Computer Science, vol 7947. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-39146-0_8

Wojcik, R. (2020, March 02). Unsupervised sentiment analysis. https://towardsdatascience.com/unsupervised-sentiment-analysis-a38bf1906483

Yalcin, O. G. (2021, February 02). Sentiment analysis in 10 minutes With Rule-Based Vader and NLTK. https://towardsdatascience.com/sentiment-analysis-in-10-minutes-with-rule-based-vader-and-nltk-72067970fb71