12-2019

Stock Price Using Domain Specific Lexicons

Zane Turner

Stock Price Prediction Using Domain Specific Lexicons

An Undergraduate Honors College Thesis

in the

Department of Computer Science

College of Engineering

University of Arkansas

Fayetteville, AR

By

Zane Turner

zmt001@uark.edu

Abstract

Sentiment analysis is a broad and expanding field that aims to extract and classifying opinions from textual data. Lexicon-based approaches are based on using a sentiment lexicon, a list of words each mapped to a sentiment score, to rate the sentiment of a text chunk. Our work focuses on predicting stock price change using a sentiment lexicon built from financial conference call logs. We introduce a method to generate a sentiment lexicon based upon an existing probabilistic approach. By using a domain-specific lexicon, we outperform traditional techniques and demonstrate that domain-specific sentiment lexicons provide higher accuracy than generic sentiment lexicons when predicting stock price change.

1 Introduction

For years, conference calls have been analyzed by investors to help evaluate the price of stock. However, due to their size and nature, it has been a challenge to extract and analyze their content making them an uncommon choice to use for stock market prediction via machine learning techniques.

The expanded availability of financial data and news allows researchers to further study their content in order to determine what causes stock price changes (Batra and Daudpota, 2018). Our work tackles this problem by performing sentiment analysis on such financial data. Sentiment analysis allows us to map each word present in these conference call logs to a corresponding positive or negative value that reflects a word's effect on stock price. Sentiment analysis approaches can be divided into two broad categories: machine learning approaches, and lexicon-based approaches (Muhammad, 2019). Machine learning approaches usually consist of constructing and training a Classifier using labeled data such as stock prices. The performance of the Classifier is then evaluated by classifying unlabeled data and by measuring the accuracy. Lexicon approaches, on the other hand, train a Classifier on a set of data, and typically uses a set of words with predetermined assigned positive or negative weights to predict an outcome such as if a stock price rises or decreases (Basak et al., 2019). In the case that a lexicon holds a word with a certain orientation that is not representative of what the words orientation should actually be, the performances of the sentiment analysis task will be negatively impacted.

There are many approaches to word polarity annotations that determine how words' sentiment scores are represented and therefore computed. A commonly used one is discrete polarity annotation that labels words with a discrete value among positive, negative, or neutral. Such a polarity is used in lexicons such as the MPQA subjectivity lexicon (Deng and Wiebe, 2015). Another common approach is a continuous polarity annotation that assigns words a decimal value within a range (+1 to -1) that reflects the positivity, negativity, or neutrality of the word, such as ChatterBox (Abbasi et al., 2014). Another less frequent approach is assigning a value to a word from a set of predetermined emotions such as joy, anger, sadness, disgust, surprise, fear, etc. Additionally, because words can often be partly positive and partly negative, a 3-tuple of positive numbers that sum up to one is used. The values that the 3-tuple reflects is positive, negative, and neutrality, an example could be (.3,.5,.2), which would be a word that is mostly neutral but slightly more positive than negative. SentiWordNet 3.0, which is a popular lexicon, applies the 3-tuple polarity annotation.

A key advantage of using a lexicon is that once the lexicon is built, it can be applied to other areas, especially areas where there is not enough information to use machine learning approaches. This is different than supervised learning sentiment analysis techniques such as naïve Bayes, which tends to perform poorly when applied to a problem that the Classifier was not trained on.

In our work, we focus on building domain-specific lexicons without needing any prior knowledge, as opposed to a general sentiment lexicon such as the widely used SentiWordNet. Domain-specific lexicons helps to better evaluate the sentiment of a word in regard to a context which improves the accuracy of the Classifier. Indeed, words can have opposite sentiment orientation, i.e., positive and negative, depending on the context. For instance, "fire" is a negative word in a corpus consisting of apparel-related products whereas it is a positive word when used in a video game context. We demonstrate that computational domain-specific lexicon techniques can be applied to the financial world so as to create a financial lexicon, and that using such a lexicon can help predict the direction of stock prices changes. Financial lexicons have been created before but have not historically assigned each individual word a non-binary sentiment. All previous financial lexicons have been binary, where our proposed lexicon has each word assigned a discrete polarity of -1 to 1 (Loughran and Mcdonald, 2011; Henry, 2008).

The rest of the paper is organized as follows: We first review over existing works, then we describe how we create the domain-specific dictionaries, next we present our results, and finally we summarize our work and describe what future work can be done and what improvements can be made.

2 Related Work

The recent increase of textual data availability has made interest in sentiment analysis grown tremendously. Usually sentiment analysis is broadly categorized into two different areas: opinion mining and opinion summarizing. The former is usually concerned with predicting whether the text represents a positive or negative value according to what we are trying to predict, while the latter is usually concerned with summarizing what has been written (Derakhshan and Beigy, 2019). Similarly, there are three general approaches to machine learning problems semi-supervised, supervised, and unsupervised, which also applies to sentiment analysis. Usually supervised sentiment analysis tasks train a Classifier such as naïve Bayes or SVM that is then used to predict future results of a particular dataset. These approaches usually result in high accuracy for that specific problem. The other method, unsupervised learning, is usually done by utilizing a lexicon. Lexicons are just a list of words with corresponding sentiment strength and orientation.

Sentiment analysis can be performed on various granularity. In Dunder and Pavlovski (2019), the authors perform sentiment analysis on the sentence level, which aims at classifying a single sentence as opposed to classifying the sentiment of an entire document. On the other hand, sentiment analysis can also be performed on the document level such as in our work. Additionally, sentiment analysis can be performed on the aspect level where words can be weighted according to specific aspects or feature (Yiran and Srivastava, 2019).

Because lexicon-based approaches can be used in sentiment analysis on every level (sentence, document, and feature) it is particularly important to accurately gauge the connotation of each word. Lexicons have generally been created using three main methods: manual creation, using an existing lexicon, and using a corpus of documents. Also, combinations of these different methods are used to create lexicons. Muhammad (Muhammad et al., 2014) showed that adapting a general lexicon to a particular domain could improve the overall accuracy of the Classifier.

Lexicons can be applied to wide variety of tasks, but one of special interest is stock price prediction. Stock price prediction has been a widely explored application of machine learning using a variety of techniques including Genetic Algorithms and Support Vector Machines to predict the future stock price (Patel et al., 2015). Another prediction that is commonly done is the direction of the stock price, rather than the stock price itself. This problem can also be seen as a classification problem, which also has been done using many different methods such as neural networks (Hu et al., 2018).

Data for stock price prediction can vary significantly, sometimes the data used to predict the stock price is simply the historical stock price, other times it a combination of different features that can seem more abstract like weather (Hirshleifer and Shumway), or a form of news may be used to predict the future stock price.

Lexicons and machine learning techniques such as LSTM are commonly used for predicting stocks based on the news, often using various forms of news such as social media (Makrehchi, Laio 2013; Akita etal., 2016). Similarly, researchers may apply sentiment analysis techniques on actual news articles instead of social media for predicting the direction of stock price movements (Matsubara et al., 2018). Finance sentiment lexicons have been created historically, but have only scored words in a binary way, that is the words are scored either positive or negative (Loughran and Mcdonald, 2011; Henry, 2008). In Henry's work a financial lexicon was created by counting the frequency of various positive and negative words obtained by a piece of software known as Diction 5.0. The most frequently occurring positive words were placed in the positive lexicon, and the most frequent negative words were placed in the lexicon, acting as a subset of the original Diction 5.0 lexicons.

Loughran and McDonald created their financial sentiment lexicon by handpicking words that occur in 5% or more of 10-ks and placing those words into corresponding lexicons.

In this work we primarily focus on building a domain-specific lexicon to predict the future direction of a stock price. Our approach varies from other domain-specific financial lexicons our lexicon is computationally built with sentiment weights attached to each word, instead of the traditionally used "positive" or "negative" label. Additionally, we introduce a new scheme for weighting words sentiment values (Kim et al., 2014; Prollochs et al., 2015)

3 Estimating Words Scores

The method used for calculating the sentiment scores for each unigram is intuitive yet powerful. We first measure the positive weight of a given word *w* by computing the quotient of its total number of occurrences across all documents and the total number of words appearing in positive documents. Likewise, the negative weight of a word is the quotient of its total number of occurrences across all documents and the total number of words appearing in negative documents. We then compute the probability of a word to be positive by dividing its positive weight by the sum of its positive and negative weights. Similarly, the probability of a word to be negative is the result of its negative weight divided by the sum of its positive and negative weights. Finally, the sentiment score of a word is the difference between its probability of being positive and the probability of being negative, which is inspired from the probabilistic approach of the work of Labille et al. (2017).

$$Score\ (w) = prob_{pos}(w) - prob_{neg}(w) \tag{1}$$

Where:

$$prob_{pos}(w) = \frac{weight_{pos}(w)}{weight_{pos}(w)+weight_{neg}(w)} \tag{2}$$

$$prob_{neg}(w) = \frac{weight_{neg}(w)}{weight_{pos}(w)+weight_{neg}(w)} \tag{3}$$

And:

$$weight_{pos}(w) = \frac{n_w}{\sum_{doc\epsilon pos} \quad n_{w,doc}} \tag{4}$$

$$weight_{neg}(w) = \frac{n_w}{\sum_{doc\epsilon neg} \quad n_{w,doc}} \tag{5}$$

n_w is the number of times a given word appears, $doc \in pos$ are the documents that appear in the positive class and $doc \in neg$ are the documents belonging to the negative class. $n_{w,doc}$ is the total number of words that appear in a given document. Additionally $\sum_{doc\epsilon neg} \quad n_{w,doc}$ is the total number of words appearing in the negative class and $\sum_{doc\epsilon pos} \quad n_{w,doc}$ is the total number of words appearing in the positive class.

Furthermore, the positive weight and negative weight of a word are multiplied by a positive stock price weight and negative stock price weight respectively. The stock price weight of a word is a coefficient that measures the average stock price change of the word. That is, words that appear in conference calls that greater affect the stock price receive a heavier weight. Therefore, $weight_{pos}(w)$ and $weight_{neg}(w)$ respectively become:

$$weight_{pos}(w) = \frac{n_w}{\sum_{doc\epsilon pos} \quad n_{w,doc}} * \lambda_{w\epsilon pos} \tag{6}$$

$$weight_{neg}(w) = \frac{n_w}{\sum_{doc\epsilon neg} \quad n_{w,doc}} * \lambda_{w\epsilon neg} \tag{7}$$

where λ is the average stock price change across all the documents a given word occurs after controlling for the number of times the word occurs in the documents, calculated as follows:

$$\lambda_{w\epsilon pos} = \frac{\sum_{doc\epsilon pos} \quad \lambda * n_w}{n_{doc\epsilon pos}} \tag{8}$$

$$\lambda_{w\epsilon neg} = \frac{\sum_{doc\epsilon neg} \quad \lambda * n_w}{n_{doc\epsilon neg}} \tag{9}$$

Where $doc \in pos$ are the documents that belong to the positive class and $doc \in neg$ are the documents belonging to the negative class. λ is the weight of the class for that document and n_w is the number of times that word appears in that document. $n_{doc\in neg}$ is the number of negative documents and $n_{doc\in neg}$ is the number of positive documents. Additionally, all words in the final lexicon that appear at a frequency less than one in ten thousand words are ignored because their limited occurrences do not provide enough information the word to have an accurate sentiment.

Table 1: Dataset Statistics

	Minimum Stock Change Percent Threshold					
	1%	3%	5%	7%	10%	20%
#Positive doc	34,380	18,617	11,050	6,945	3,683	502
#Negative doc	34,763	19,127	11,771	7,901	4,739	1,400
Total	69,143	37,744	22,821	14,846	8,422	1,902
Training						
#Positive doc	27,540	14,926	8,812	5,583	2,947	309
#Negative doc	27,774	15,269	9,444	6,293	3,790	832
Testing						
Total	55,314	30,195	18,256	11,876	6,737	1,141
#Positive doc	6,840	3,691	2,238	1,362	736	193
#Negative doc	6,989	3,858	2,327	1,608	949	568
Total	13,829	7,549	4,565	2,970	1,685	761

Experiment

To evaluate the impact of using a discrete domain-specific lexicon built on financial data against previously created lexicons, we ran a controlled experiment wherein we compare our domain-specific lexicons to the widely used generic lexicon SentiWordNet 3.0 and two popular domain-specific financial sentiment lexicons referred to for the rest of this paper as Henry's lexicon and Loughran's lexicon, while our dictionary is referred to as the domain-specific lexicon. A sentiment analysis task was performed and was evaluated through state-of-the-art evaluation metrics such as Recall, Precision, F-1 Score, and Accuracy (Baccianella et al., 2010).

3.1 Dataset

In order to build a domain-specific lexicon, we first obtained access to a large amount of earnings conference calls from `seekingalpha.com`. While the website is public, permission was required to extract data through web-scraping. Once permission to scrape their website was granted, we were able to extract 120,431 conference calls which are used as our main dataset. The conference calls span from the year 2008 to 2018, and consist of calls that Seeking Alpha tracks, which are limited to the 4,500 companies which have the most subscribers for Seeking Alphas real-time alerts product. This list of companies has varied overtime, so a total of 8,689 firms have been covered since 2008. The dataset is split into three subsets depending on the adjusted stock price change. Adjusted stock price change is the stock price adjusted according to the direction of the market that day, so if the market moved up 0.5% in a given day, the stock of a given day would have to drop 1.5% or rise 2.5% to be included in the sample.

Conference calls were deemed negative when an adjusted stock price drops by the end of the trading day and deemed positive when the adjusted stock price rose by the end of the trading day. Calls that had an adjusted stock price change of less than 1% were considered neutral and thus eliminated from the dataset, yielding a total of 69,143 conference calls. We created a second, smaller subset of 37,744 conference calls that had an adjusted stock price change greater than 3%, a third subset containing 22,821 conferences calls with an adjusted stock price change greater than 5%, a fourth subset containing 11,876 conference calls with an adjusted stock price greater than 7%, a fifth subset containing 8,422 conference calls with an

Table 2: Results

	Minimum Stock Price Change Threshold					
	1%	3%	5%	7%	10%	20%	Average
Accuracy							
Domain-specific	0.555	0.586	0.616	0.637	0.686	0.804	0.647
Henry	0.511	0.504	0.513	0.49	0.481	0.345	0.474
Loughran	0.53	0.551	0.558	0.572	0.606	0.726	0.591
SentiWordNet	0.504	0.499	0.508	0.484	0.462	0.343	0.467
Recall							
Domain-specific	0.513	0.512	0.557	0.568	0.604	0.49	0.541
Henry	0.898	0.896	0.9	0.905	0.906	0.843	0.891
Loughran	0.393	0.404	0.397	0.402	0.414	0.435	0.408
SentiWordNet	0.92	0.919	0.931	0.922	0.924	0.948	0.927
Precision							
Domain-specific	0.558	0.594	0.612	0.626	0.653	0.696	0.623
Henry	0.503	0.496	0.502	0.471	0.453	0.256	0.447
Loughran	0.534	0.556	0.571	0.545	0.567	0.454	0.538
SentiWordNet	0.499	0.494	0.499	0.468	0.444	0.271	0.446
F1-Score							
Domain-specific	0.267	0.275	0.292	0.298	0.298	0.288	0.286
Henry	0.323	0.319	0.322	0.31	0.31	0.197	0.297
Loughran	0.226	0.234	0.234	0.231	0.231	0.222	0.23
SentiWordNet	0.324	0.321	0.325	0.31	0.31	0.211	0.3

adjusted stock price greater than 10%, finally a sixth subset containing 1902 conference calls with an adjusted stock price greater than 20%. Conference calls were preprocessed so that stop words and punctuation marks were removed and stemmed only when building the domain-specific dictionaries and calculating the results of the corresponding domain-specific lexicon. The conference calls were not stemmed when considering the other lexicons because their lexicons only consisted of complete words and were not meant to be used on stemmed data. The remaining words contained in the conference calls were assigned a sentiment score using the formulae described in section 3.

Furthermore, each of our datasets were randomly split into two partitions, where we used 80% for training and 20% for testing.

3.2 Experimental Results

We evaluate our domain-specific lexicons by comparing them to our baselines SentiWordNet 3.0, Henry's lexicon, and Loughran's lexicon, then report our results in table 2. Table 2 reports the F1-Score, precision, recall, and accuracy achieved by all lexicons. Our results show that domain-specific lexicons achieve an accuracy average accuracy of 64.7% while SentiWordNet, Henry's lexicon, and Loughran's lexicon have respective average accuracy of 47.4%, 59.1%, and 46.7%. Additionally, our domain-specific lexicons were more accurate in all cases, not just with averages.

Table 3: Sentiment Scores

	Lexicon								
	1%	3%	5%	7%	10%	20%	SWN	Henry	Loughran
congrat	0.386	0.417	0.509	0.547	0.609	0.558	0.125	0	0
congratul	0.326	0.385	0.443	0.511	0.568	0.757	0.125	0	0
nice	0.275	0.328	0.373	0.399	0.447	0.41	0.875	0	0
job	0.194	0.259	0.297	0.323	0.396	0.484	0	0	0
sustain	0.19	0.236	0.244	0.292	0.317	0.42	0	0	0
upside	0.19	0.179	0.224	0.233	0.269	0.222	0	0	0
solid	0.182	0.221	0.253	0.187	0.297	0	0.875	1	0
excel	0.179	0.219	0.26	0.259	0.257	0.317	0	0	0
strength	0.164	0.194	0.247	0.236	0.299	0.414	0.375	1	1
impress	0.164	0.182	0.248	0.262	0.299	0.236	0	0	1
delay	-0.231	-0.257	-0.296	-0.354	-0.429	-0.506	0.125	0	-1
reconcil	-0.202	-0.218	-0.246	-0.275	-0.295	-0.514	0.125	0	0
lose	-0.19	-0.237	-0.292	-0.283	-0.25	-0.513	-0.5	0	-1
weaker	-0.189	-0.28	-0.314	-0.421	-0.369	-0.581	-0.375	0	-1
weak	-0.181	-0.192	-0.236	-0.3	-0.329	-0.595	-0.0375	-1	-1
issu	-0.18	-0.197	-0.228	-0.251	-0.266	-0.323	0.125	0	-1
lost	-0.177	-0.204	-0.227	-0.239	-0.308	-0.064	-0.25	0	-1
slowdown	-0.165	-0.205	-0.217	-0.289	-0.263	-0.551	0	0	-1
loss	-0.145	-0.185	-0.203	-0.207	-0.244	-0.149	-0.5	0	-1
deceler	-0.145	-0.215	-0.21	-0.31	-0.211	-0.284	0	0	0

We also computed and compared the recall, precision, and F1-score achieved by both lexicons. Results are reported in Table 2. Our domain-specific lexicon achieved an average recall of 54.1%, while Henry's lexicon had 89.1%, Loughran's lexicon had 40.8%, and SentiWordNet had 92.7%. Although SentiWordNet and Henry's lexicon achieved a higher recall, it does not necessarily mean that it performed better. Indeed, such high recall values were achieved because the two lexicons predicted the positive class in most cases while our domain-specific lexicon predicted both classes relatively evenly.

Our domain-specific lexicon achieved an average precision 62.3% and Henry's lexicon, Loughran's lexicon, and SentiWordNet achieved corresponding average accuracies of 44.7%, 53.8, and 44.6%. This demonstrates that our domain-specific lexicon was better at predicting the correct class. Additionally, as seen in the table 2, the domain-specific lexicon's precision was higher than the other three lexicons in all experiment's cases.

Finally, we evaluated the performances of both lexicons through the F1-score. Our domain-specific lexicon achieved an average F1-score of 28.6%, while Henry's lexicon, Loughran's lexicon, and SentiWordNet achieved average F1-scores of 29.7%, 23.0% and 30.0%. The higher values of F1-Score achieved by SentiWordNet and Henry's lexicon are due to the extremely high and biased recall values.

The lexicons varied in size, with our domain-specific lexicons containing 462, 470, 476, 511, 519, and 542 positive words for the corresponding

1%, 3%, 5%, 7%, 10%, and 20% adjusted stock price change data datasets. They also contained 446, 478, 512, 501, 523, and 597 negative words for same previously corresponding datasets. Henry's lexicon contained 105 positive words and 85 negative words, and Loughran's lexicon contained 2355 negative words and 254 positive words.

We ran a paired two-tailed student t-test on the accuracy of each pair of lexicon's results (domain-specific vs. Henry's lexicon, domain-specific vs. Loughran's lexicon, and domain-specific vs. SentiWordNet) to test for statistical significance. This process was done with all six domain-specific lexicons. Results show that our results are statistically significant in all cases, meaning that our domain-specific lexicon was indeed more accurate than the other three tested lexicons.

This supports our intuition that domain-specific lexicons with each word assigned a non-binary sentiment better catch the sentiment of the words in a given context as opposed to a generic lexicon or binary domain-specific lexicons, and therefore are more accurate for performing classification tasks. Our results show that domain-specific lexicons can be accurately used to predict the direction of future stock price.

4 Discussion

In order to gain further insight on how domain-specific lexicon outperform other lexicons when used to predict stock price direction, we examine the content of each of the domain-specific lexicons. We searched the top ten positive and top ten negative words within the domain-specific lexicons and compared their score with that of the other lexicons. Our findings are summed up in Table 3.

When looking at conference calls that affect the stock price greater than 1%, our ten most positive words were *congrat*, *congratul*, *nice*, *job* and *sustain*, *upside*, *solid*, *excel*, *strength*, and *impress* while the ten most negative words were *delay*, *reconcile*, *weaker*, *weak, issu, lost, slowdown, loss,* and *deceler.* We then looked at the sentiments for each the top words in our 1% lexicon in our other domain-specific lexicons. We find that there are no cases in our examples where we have a positive sentiment for a word in one of our domain-specific lexicons that then occurs as a negative value in another domain-specific lexicon, or the vise-versa.

We first notice from Table 3 that words from our lexicons carry a similar sentiment across all six datasets within the finance domain, while the sentiment of some words differs greatly when compared to a the SentiWordNet, Henry, and Loughran lexicon. For instance, the word *congratulations* has a sentiment of 0.125 in a generic lexicon while it has a sentiment that ranges from 0.326 to 0.757 in the finance domain, meaning that congratulations is approximately three to six times more positive in that particular domain. Likewise, *lose* seems to be a negative word in the finance domain with a score ranging from -0.19 to -0.513 while it is generally more negative in the SentiWordNet lexicon with a score of -0.500 and it is completely negative in Loughran's lexicon, but *lose* does not exist within Henry's lexicon.

We also notice that some words actually have opposite sentiment when used in a particular domain, for instance, the word *delay* is considered negative in our financial lexicon with a score ranging from -0.231 to -0.506, while it is considered positive in the SentiWordNet lexicon with a score of 0.125.

Similarly, *reconcil* has a value of -.202 to -.514 in the domain-specific dictionaries, while reconcile carries a positive value of .125 in SentiWordNet.

Finally, we notice that there are words such as *nice, job, sustain, upside, excel, impress, slowdown, and deceler* that carry either a positive sentiment or negative sentiment in our lexicons while they carry no sentiment in the SentiWordNet and are therefore deemed neutral. Additionally, words *congrat, congratul, nice, job, sustain, solid, excel, impress, delay, reconcile, lose, weaker, issu, lost, slowdown, loss,* and *deceler* and their pre-stemmed counterparts do not exist in Henry's lexicon. Also, *congrat, congratul, nice, job, sustain, upside, solid, excel, reconcile,* and *deceler* and their pre-stemmed counterparts do not exist within Loughran's lexicon.

This highlights our intuition that, within the financial realm, certain words carry their own unique meanings that do not often translate well to other domains like SentiWordNet. While words in the Henry's and Loughran's financial lexicons often match the sign of our domain-specific lexicon, they are not able to capture the differences in effect of each individual word because they only are assigned a negative or positive label.

Furthermore, we notice that the sentiment of words within our lexicons across all six datasets gets stronger as the adjusted stock price change gets bigger. This means that a word carries a stronger sentiment in more extreme situations. This is actually intended and is due to the stock price weight λ introduced in our formula, which intends to weight words more heavily when the adjusted stock price is higher.

5 Conclusion

In this paper we introduced a method for generating a domain-specific lexicon using probabilistic theoretic weights. This work is different than the traditional approaches in that we create the domain-specific lexicons without a-priori knowledge. This means we do not have to adapt our lexicon from a generic lexicon. This solution also helps overcome certain performance issues that can arise when using a transferred supervised Classifier (Tan, et al., 2009).

We measure the effectiveness of our domain-specific lexicon by comparing the performances of our domain-specific lexicon against that of the widely used generic lexicon SentiWordNet 3.0, Henry's financial lexicon, and Loughran's financial lexicon. Experimental results show that our domain-specific lexicon is 4.4% to 45.9% more accurate than Henry's lexicon depending on the stock change threshold used, 2.5% to 8% more accurate than Loughran's lexicon, and 4.4% to 46.1% more accurate than SentiWordNet. Our domain-specific lexicon better predicts the stock price direction and also gets more accurate than the three other lexicons as the weighted price change increases (1% vs 20% adjusted change in price following the call.)

Our results indicate that domain-specific non-categorical lexicons are more accurate than generic lexicons when performing sentiment analysis tasks applied to financial data and also more accurate than the two binary domain-specific dictionaries. Additionally, our results show that domain-specific scores better reflect word sentiment than generic sentiment scores do.

Future work could include predicting the actual stock price change instead of just the direction. In addition, we could experiment with using deep learning and word embedding for sentiment lexicon creation.

REFERENCES

Abbasi, A., Hassan, A., & Dhar, M. (2014, May). Benchmarking Twitter Sentiment Analysis Tools. In *LREC* (Vol. 14, pp. 26-31).

Akita, R., Yoshihara, A., Matsubara, T., & Uehara, K. (2016, June). Deep learning for stock prediction using numerical and textual information. In *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)* (pp. 1-6). IEEE.

Baccianella, S., Esuli, A., & Sebastiani, F. (2010, May). Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In *LREC* (Vol. 10, No. 2010, pp. 2200-2204).

Basak, G. K., Das, P. K., Marjit, S., Mukherjee, D., & Yang, L. (2019). British Stock Market, BREXIT and Media Sentiments-A Big Data Analysis (No. 7760). CESifo Group Munich.

Batra, R., & Daudpota, S. M. (2018, March). Integrating StockTwits with sentiment analysis for better prediction of stock price movement. In *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)* (pp. 1-5). IEEE.

Deng, L., & Wiebe, J. (2015). Mpqa 3.0: An entity/event-level sentiment corpus. In *Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics: human language technologies* (pp. 1323-1328).

Derakhshan, A., & Beigy, H. (2019). Sentiment analysis on stock social media for stock price movement prediction. *Engineering Applications of Artificial Intelligence*, *85*, 569-578.

Dunder & Pavlovski, M. (2019, *International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (pp. 577-582). IEEE.

Henry, E. (2008). Are investors influenced by how earnings press releases are written?. *The Journal of Business Communication (1973)*, *45*(4), 363-407.

Hu, H., Tang, L., Zhang, S., & Wang, H. (2018). Predicting the direction of stock markets using optimized neural networks with Google Trends. *Neurocomputing*, *285*, 188-195.

Hirshleifer, D., & Shumway, T. (2003). Good day sunshine: Stock returns and the weather. *The Journal of Finance*, *58*(3), 1009-1032.

Kim, Y., Jeong, S. R., & Ghani, I. (2014). Text opinion mining to analyze news for stock market prediction. *Int. J. Advance. Soft Comput. Appl*, *6*(1), 2074-8523.

Labille, K., Gauch, S., & Alfarhood, S. (2017, August). Creating domain-specific sentiment lexicons via text mining. In *Proc. Workshop Issues Sentiment Discovery Opinion Mining (WISDOM)*.

Loughran, T., & McDonald, B. (2011). When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. *The Journal of Finance*, *66*(1), 35-65.

Madhoushi, Z., Hamdan, A. R., & Zainudin, S. (2015, July). Sentiment analysis techniques in recent works. In *2015 Science and Information Conference (SAI)* (pp. 288-291). IEEE.

Matsubara, T., Akita, R., & Uehara, K. (2018). Stock Price Prediction by Deep Neural Generative Model of News Articles. *IEICE TRANSACTIONS on Information and Systems*, *101*(4), 901-908.

Muhammad, A., Wiratunga, N., & Lothian, R. (2014, November). A hybrid sentiment lexicon for social media mining. In *2014 IEEE 26th International Conference on Tools with Artificial Intelligence* (pp. 461-468). IEEE.

Muhammad, S. H. (2019, January) An overview of Sentiment Analysis Approaches.

Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications*, *42*(1), 259-268.

Pröllochs, N., Feuerriegel, S., & Neumann, D. (2015, May). Generating Domain-Specific Dictionaries using Bayesian Learning. In *ECIS*.

Tan, S., Cheng, X., Wang, Y., & Xu, H. (2009, April). Adapting naive bayes to domain adaptation for sentiment analysis. In *European Conference on Information Retrieval* (pp. 337-349). Springer, Berlin, Heidelberg.

Yiran, Y., & Srivastava, S. (2019, June). Aspect-based Sentiment Analysis on mobile phone reviews with LDA. In Proceedings of the 2019 4th International Conference on Machine Learning Technologies (pp. 101-105). ACM.