Lexicon-Based Sentiment Analysis for Stock Movement Prediction

Zane Turner, Kevin Labille, Susan Gauch

Abstract—Sentiment analysis is a broad and expanding field that aims to extract and classify opinions from textual data. Lexicon-based approaches are based on the use of a sentiment lexicon, i.e., 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.

Keywords—Computational finance, sentiment analysis, sentiment lexicon, stock movement prediction.

I. 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 [1]. 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. 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 positive or negative weights to predict an outcome such as if a stock price rises or decreases [2]. 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 [3]. 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 [4]. 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 in which it is used consequently improving 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 value. Indeed, all previous financial lexicons use a binary sentiment strength, i.e., 0 or 1, whereas our proposed lexicon uses a discrete polarity annotation ranging from -1 to 1 [5], [6].

The rest of the paper is organized as follows: we first review existing works, then we describe how we create the domain-specific lexicons, next we present our results, and finally, we summarize our work and describe what future

Zane Turner, Kevin Labille, and Susan Gauch are with the Department of Computer Science and Computer Engineering at the University of Arkansas, Fayetteville, AR 72701 (e-mail: zmt001@email.uark.edu, kclabill@uark.edu, sgauch@uark.edu).

work can be done and what improvements can be made.

II. RELATED WORK

The recent increase of textual data availability has made interest towards 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 [7]. 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 a specific problem they were trained on. The other method, unsupervised learning, is usually done through the use of a lexicon. Lexicons are just a list of words mapped to a value that represent the sentiment strength and orientation.   Sentiment analysis can be performed on various granularity. Dunder and Pavlovski perform sentiment analysis on the sentence level, which aims at classifying a single sentence as opposed to classifying the sentiment of an entire document [8]. 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 [9].

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 showed that adapting a general lexicon to a particular domain could improve the overall accuracy of the classifier [10].

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 [11]. 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 tackled using many various methods such as neural networks [12].

The type of data used for stock price prediction can vary significantly, while most of the time the data is finance related such as the historical stock price, other times it is a combination of different features not related to the financial aspect such as the weather or news [13].

Lexicons and deep learning techniques such as LSTM neural networks are commonly used for predicting stocks based using news items, often extracted from social media [14], [15]. Similarly, researchers may apply sentiment analysis techniques on actual news articles instead of social media for predicting the direction of stock price movements [16]. Financial sentiment lexicons have been created in the past, but they typically assign a binary value to express the sentiment, that is, a value of 1 is mapped to a word that is deemed positive and a value of 0 if assigned if the word is deemed negative [5], [6]. In Henry's work a financial lexicon was manually created. Loughran and McDonald created their financial sentiment lexicon by manually choosing 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 in that we computationally evaluate the sentiment weights of the words instead as opposed to assigning sentiment manually, and we use a continuous polarity annotation as opposed to using a binary "positive" or "negative" label. Additionally, we introduce a new scheme for weighting words sentiment values [17], [18].

III. ESTIMATING WORD 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 [19].

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

where:

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

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

and:

$$weight_{pos}(w) = \frac{n_w}{\sum_{doc \in pos} \ n_{w,doc}} \qquad (4)$$

$$weight_{neg}(w) = \frac{n_w}{\sum_{doc \in neg} \ n_{w,doc}} \qquad (5)$$

In (4) and (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 $\sum_{doc \in neg} n_{w,doc}$ is the total number of words appearing in the negative class and $\sum_{doc \in pos} n_{w,doc}$ 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:

$$weight_{pos}(w) = \frac{n_w}{\sum_{doc \in pos} n_{w,doc}} * \lambda_{w \in pos} \qquad (6)$$

$$weight_{neg}(w) = \frac{n_w}{\sum_{doc \in neg} n_{w,doc}} * \lambda_{w \in neg} \qquad (7)$$

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:

$$\lambda_{w \in pos} = \frac{\sum_{doc \in pos} \lambda * n_w}{n_{doc \in pos}} \qquad (8)$$

$$\lambda_{w \in neg} = \frac{\sum_{doc \in neg} \lambda * n_w}{n_{doc \in neg}} \qquad (9)$$

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 about the word to evaluate an accurate sentiment.

TABLE I
DATASET STATISTICS

Training + Testing	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
Total	55,314	30,195	18,256	11,876	6,737	1,141
Testing						
#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

IV. 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 lexicon is referred to as the domain-specific lexicon [5], [6]. 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 [20]. Since SentiWordNet assigns 3 polarity scores to each word (positive score, neutral score, and negative score), we compute a new sentiment score based on the positive and negative score for each word. Specifically, for each word in the SentiWordNet lexicon we subtract its negative score to its positive score. If the positive score exceeds the negative score, the result will be a positive sentiment score and the word will be deemed positive. Likewise, if the negative score is greater than the positive

score, the result of the subtraction will be negative, and the word deemed negative.

A. 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 techniques. Once the 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 Alpha 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 several 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.

TABLE II
RESULTS

Accuracy	Minimum Stock Price Change Threshold						
	1%	3%	5%	7%	10%	20%	Average
Domain-specific	**0.555**	**0.587**	**0.617**	**0.638**	**0.687**	**0.804**	**0.648**
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	0.529	0.548	0.546	0.562	0.616	0.49	0.549
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	**0.557**	**0.588**	**0.616**	**0.628**	**0.65**	**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	0.271	0.284	0.29	0.297	**0.316**	**0.288**	0.291
Henry	0.323	0.319	0.322	**0.31**	0.302	0.197	0.296
Loughran	0.226	0.234	0.234	0.231	0.239	0.222	0.231
SentiWordNet	**0.324**	**0.321**	**0.325**	**0.31**	0.3	0.211	**0.299**

Conference calls were deemed negative when an adjusted stock price dropped 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, therefore 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 adjusted stock price greater than 10%, and finally a sixth subset containing 1,902 conference calls with an adjusted stock price greater than 20%. Conference calls were preprocessed so that stopwords and punctuation marks were removed. Additionally, stemming was used when creating our domain-specific lexicons whereas the baseline lexicons that we used were not stemmed as per choice of their respective authors. When building our financial sentiment lexicons, unigrams extracted from the conference calls were assigned a sentiment score calculated using the formulae described in Section III. Furthermore, each of our datasets were randomly split into two subsets, where 80% is used for training to build our lexicons, and the remaining 20% is used for testing purposes.

B. Experimental Results

We evaluate the performances of our domain-specific lexicons by comparing them to our baselines SentiWordNet 3.0, Henry's lexicon, and Loughran's lexicon, and report our results in Table II. A simple sentiment analysis approach was employed for evaluating the lexicons. Specifically, for each word within a call log we look up its sentiment score from the lexicon and sum them up. If the resulting score is greater than 0, the call is thought to be positive, conversely, if the resulting sentiment score is less than 0, the call is thought to be negative. Table II reports the F1-Score, precision, recall, and accuracy achieved by all lexicons across all testing datasets. Our results show that on average, our domain-specific lexicons outperform all baselines, with an average accuracy of 64.7% against 47.4%, 59.1%, and 46.7% for SentiWordNet, Henry's lexicon, and Loughran's, respectively. Similarly, our approach outperforms all baselines on average in terms of precision with 62.3% for our lexicon against 44.7%, 53.8%, and 44.6% for Loughran's, Henry's and SentiWordNet's lexicons, respectively. In terms of recall, our domain-specific lexicon achieved an average of 54.1%, while Henry's lexicon achieved 89.1%, Loughran's lexicon achieved 40.8%, and SentiWordNet achieved 92.7%. Although SentiWordNet and Henry's lexicon achieved a higher recall, it does not necessarily reflect a better performance. Indeed, the dataset is biased towards the positive class (c.f. Table I), therefore, by predicting a higher amount of the positive class than the negative class, a high recall can be achieved while achieving a relatively low accuracy. As opposed to Henry and SentiWordNet, our lexicon predicted both classes more evenly which results in a lower recall but higher accuracy.

When looking at the accuracies and precision achieved on the individual datasets (namely, the 1%, 3%, 5%, 7%, 10%, and 20%), results show that our lexicon outperforms all baselines on all six datasets. SentiWordNet achieve a higher recall on all datasets, however, as explained earlier this is due to the imbalanced dataset. We further notice that our lexicon becomes more accurate as the stock price change threshold increases whilst both Henry and SentiWordNet lexicons

actually become less accurate. Although Loughran's lexicon also increases in accuracy as the datasets stock price threshold increases, it does not increase as much as our lexicon does (25.4% vs 19.6% accuracy increase). Our intuition is that the increase of accuracy achieved by our domain-specific lexicons is due to the way we compute the words sentiment scores. Indeed, our sentiment scores take into consideration the stock price weight (through the λ factor) when computed, and are continuous values as opposed to binary values (0 or 1). They therefore should be able to adapt to the data by better capturing the sentiment of the words. Specifically, the higher the stock price weight, the higher the sentiment scores should be for a same word.

TABLE III
SIZE OF THE VARIOUS LEXICONS

	1%	3%	5%	7%	10%	20%	SWN	Henry	Loughran
#Positive words	586	586	598	564	583	590	13,127	105	254
#Negative words	595	595	582	611	585	583	14,724	85	2,355
Total	1,181	1,181	1,180	1,175	1,168	1,173	27,851	190	2,609

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 intuitions that (1) domain-specific lexicons better reflect the sentiment of the words in a given context as opposed to generic lexicons, and (2) that, the use of continuous sentiment values more accurately catch the sentiment of the words as opposed to using binary values, and therefore are more appropriate for performing classification tasks. Our results demonstrate that sentiment analysis with domain-specific lexicons can be used to predict the direction of future stock price.

V. DISCUSSION

In order to gain further insight on how our domain-specific lexicon outperforms other lexicons when used to predict stock price direction, we first looked at the size and content of the various lexicons and report our findings in Table III.

Table III shows the size of the various lexicons. As we can see, our domain specific lexicons contain almost a balanced amount of positive words and negative words in most cases, with a size ranging from 1,168 to 1,181 words overall. Conversely, Loughran's lexicon is very unbalanced with almost 10 times more negative words than positive words. Henry's lexicon is almost balanced; however, its size is rather small with only 190 words as opposed to the SentiWordNet lexicon which contains over 25,000 words in total. We believe that both the balance and size of the lexicon is important. While Henry's lexicon is balanced, its size negatively impacts its performances since it will not allow the lexicon to cover all of the vocabulary found and will thus fail to know the sentiment of many words. Although Loughran's lexicon has a size of 2,609 and therefore covers most of the vocabulary, it is very unbalanced, which could indicate that the polarity of some negative words is wrong.

We then examine the content of each of the domain-specific lexicons. We searched the top 10 positive and top 10 negative words within the domain-specific lexicons and compared their sentiment scores with that of the other lexicons. Our findings are summed up in Table IV. Since we used stemming during our preprocessing phase, some of our top positive and negative words appear as the base form, e.g., *congratul, congrat, reconcil*. These words are not found in other lexicons and will therefore appear with a value of N/A in Table IV. Similarly, when one of our top words is not found in other lexicon, it will appear with a value of N/A rather than 0, which would mean that the word is actually found but has a sentiment value of 0.

We first notice from Table IV 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 SentiWordNet, Henry, and Loughran lexicons. For instance, the word *nice* has a sentiment of 0.875 in a generic lexicon while it has a sentiment that ranges from 0.275 to 0.447 in the finance domain, meaning that *nice* is approximately two to three times less positive in that particular domain. Likewise, *loss* has a score ranging from -0.145 to -0.244 in the finance domain while it is more negative in the SentiWordNet lexicon with a score of -0.500 and it is completely negative in Loughran'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.

Finally, we notice that there are words such as *job, upside, excel, impress,* and *slowdown,* 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* do not exist in Henry's lexicon. Likewise, *congrat, congratul, nice, job, sustain, upside, solid, excel, reconcile,* and *deceler* do not exist within Loughran's lexicon.

We further 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.

TABLE IV
SENTIMENT SCORES OF TOP TEN POSITIVE AND TOP TEN NEGATIVE WORDS

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

These observations support our intuition that the size of the lexicon matters as it should cover as much vocabulary as possible in order to achieve better results. It also 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 are only assigned a negative or positive label.

VI. CONCLUSION

In this paper, we introduced a method for generating a domain-specific lexicon using a probabilistic approach combined with financial-based knowledge, i.e., the stock price weight. This work is different than traditional approaches in that the domain-specific lexicons are created automatically and the sentiment scores are computed by taking into account the prior stock price change. Furthermore, 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 [21].

We measure the effectiveness of our domain-specific lexicon by comparing the performances of our domain-specific lexicons against that of widely used generic lexicons such as 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 lexicons better predict 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 binary-weight domain-specific lexicons. 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

[1] R. Batra, S. M. Daudpota, "Integrating StockTwits with sentiment analysis for better prediction of stock price movement," in *2018 Int. Conf. on Computing, Mathematics and Engineering Technologies,* pp. 1-5.

[2] G. K. Basak, P. K. Das, S. Marjit, D. Mukherjee, and L. Yang, "British Stock Market, BREXIT and Media Sentiments-A Big Data Analysis," unpublished.

[3] L. Deng, J. Wiebe, "Mpqa 3.0: An entity/event-level sentiment corpus," in *Proc. conf. of the North American chapter of the association for computational linguistics: human language technologies*, 2015, Minnesota, pp. 1323-1328.

[4] A. Abbasi, A. Hassan, and M. Dhar, "Benchmarking Twitter Sentiment Analysis Tools," *LREC,* vol. 14, pp. 26-31, May 2014.

[5] T. Loughran, B. McDonald, "When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks". *The J. of Finance*, vol. 66, no.1, pp. 35-65, Feb. 2011.

[6] E. Henry, "Are investors influenced by how earnings press releases are written?," The *J. of Business Communication*, vol. 45, no. 4, pp. 363-407, Oct. 2008.

[7]   A. Derakhshan, H. Beigy, "Sentiment analysis on stock social media for stock price movement prediction," *Engineering Applications of Artificial Intelligence*, vol. 85, pp. 569-578, Oct. 2019.

[8]   I. Dunder, M. Pavlovski, "Computational concordance analysis of fictional literary work," MIPRO, In *2018 41st Int. Conv. on Information and Communication Technology, Electronics and Microelectronics,* pp. 644-648.

[9]   Y. Yiran, S. Srivastava, "Aspect-based Sentiment Analysis on mobile phone reviews with LDA," in *Proc. 4th Int. Conf. on Machine Learning Technologies*, Austria, 2019, pp. 101-105.

[10] A. Muhammad, N. Wiratunga, and R. Lothian, "A hybrid sentiment lexicon for social media mining," in *2014 IEEE 26th Int. Conf. on Tools with Artificial Intelligence*, pp. 461-468.

[11] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques," *Expert Syst. with Applications*, vol. 42, no. 1, pp. 259-268, Jan. 2015.

[12] H. Hu, L. Tang, S. Zhang, and H. Wang, "Predicting the direction of stock markets using optimized neural networks with Google Trends," *Neurocomputing*, vol. 285, pp. 188-195, Apr. 2015.

[13] D. Hirshleifer, T. Shumway, "Good day sunshine: stock returns and the weather," *The J. of Finance*, vol. 58, no. 3, pp. 1009-1032, Jun. 2013.

[14] M. Makrehchi, S. Shah, and W. Liao, "Stock prediction using event-based sentiment analysis," in *Proc. IEEE/WIC/ACM Int. Joint Conf. on Web Intelligence and Intelligent Agent Technologies*, Georgia, 2013, vol. 1, pp. 337-342.

[15] R. Akita, A. Yoshihara, T. Matsubara, and K. Uehara. "Deep learning for stock prediction using numerical and textual information," In *2016 IEEE/ACIS 15th Int. Conf. on Computer and Information Science*, pp. 1-6.

[16] T. Matsubara, R. Akita, and K. Uehara, "Stock Price Prediction by Deep Neural Generative Model of News Articles," *IEICE Transactions on Information and Syst.*, vol. 101, no. 4, pp. 901-908, Apr. 2018.

[17] Y. Kim, S. R. Jeong, and I. Ghandi, "Text opinion mining to analyze news for stock market prediction," *int. J. Advance. Soft Comput. Appl*, vol. 6, no. 1, pp. 2074-2087, Mar. 2014.

[18] N. Pröllochs, S. Feuerriegel, and D. Neumann, "Generating Domain-Specific Dictionaries using Bayesian Learning," in *2015 conf. ECIS,* Paper 144.

[19] K. Labille, S. Gauch, and S. Alfarhood, "Creating domain-specific sentiment lexicons via text mining" in *WISDOM Proc. Workshop Issues Sentiment Discovery Opinion Mining*, Halifax, 2017.

[20] S. Baccianella, A. Esuli, and F. Sebastiani, "Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining," in *LREC*, Vol. 10, No. 2010, pp. 2200-2204, May 2010.

[21] S. Tan, X. Cheng, Y. Wang, and H. Xu, "Adapting naive bayes to domain adaptation for sentiment analysis," in 2009 *European Conf. on Information Retrieval*, pp. 337-349.