Produced with a Trial Version of PDF Annotator - www.PDFAnnotator.com

The Performance Evaluation of Textual Analysis Tools in Financial Markets

Regular Paper

Tianyou Hu
University of Auckland
Address
Owen G Glenn Building, 12 Grafton Road, Auckland, New Zealand
t.hu@auckland.ac.nz

Arvind Tripathi
University of Auckland
Address
Owen G Glenn Building, 12 Grafton Road, Auckland, New Zealand
a.tripathi@auckland.ac.nz

Abstract

The rapid development of textual analysis techniques has paved the way for automatic sentiment analysis. Many studies have been focusing on using sentiments disclosed from social media to predict the financial market performance. However, the underlying value of the messages collected from social media is still not fully understood due to lack of comprehensive apprehending of the performance for different textual analysis methods. In this research, we shed light on this problem by comparing the different classification accuracy of dictionaries and machine-learning techniques in the financial context.

Introduction

Social media platforms and online discussion forums have become popular places to share and learn about products, services, and even financial markets. Therefore, one would expect that the sentiments expressed in social media and investor discussion forums may contain value-relevant information, which will be incorporated by the financial market. Indeed, recent

Electronic copy available at: http://ssrn.com/abstract=2661064

Produced with a Trial Version of PDF Annotator - www.PDFAnnotator.com

studies (e.g., Tetlock, 2007; Loughran and Mcdonald, 2011; Hu and Tripathi, 2015) have shown that the percentage of negative words used in an article captures the tone of varied financial reports and even influence the stock market. Antweiler and Frank (2004) and Li (2008) have also shown the impact of qualitative information on equity valuation. The words selected by news article authors, social media content contributors and company reports have been proved to have explanatory or even predictive power for stock returns, earnings and even managing deceptive activities. There is much research towards how to extract the sentiments from the financial articles and user-generated content on social media using machine-learning algorithms, however, we are yet to have a good understanding of performance accuracy of textual analysis tools.

Textual analysis, especially sentiment analysis, is a domain dependent problem. An expression that has a clear sentiment in one domain may be ambiguous in other domains. This issue is particularly strong in the financial context analysis, as there are specialized concepts and limited use of effective words. It has been shown that dictionaries developed for other disciplines misclassify common words in the financial context (Loughran and Mcdonald, 2011). For instance, liability is a neutral word in financial context. The L&M dictionary developed by Loughran and Mcdonald (2011) has been widely used (Chen et al., 2014) in financial context analysis since its first appearance in academia.

Besides the L&M dictionary, researchers have used other machine learning algorithms for classification. For example, Leung and Ton (2015) use Naïve Bayes (NB) algorithm to classify message sentiments and find that sentiments positively relate to the returns of small-cap stocks. Malo et al. (2013) compare the performance of the support vector machine (SVM) with varied underlying pattern analysis algorithms and find that performance is significantly improved

Electronic copy available at: http://ssrn.com/abstract=2661064

when different algorithms are combined. Tirunillai and Tellis (2012) implement both NB and SVM to show that SVM outperform NB in five markets and underperform in one market.

Although previous studies have tested the sentiments from different social media posts or articles, we are still unaware of the strengths and limitations of various approaches for sentiment analysis of user-generated content (UGC) in financial markets. In this research, we contribute to the literature by comparing the performance efficiency of a widely used dictionary in financial context (Loughran and Mcdonald 2011) and two classifiers, Naïve Bayes Classifier, and Support Vector Machine Classifier.

The rest of the paper is organized as follows. The second section presents the previous studies; the third section describes the dataset, the fourth section explains the method and the last two sections present the results and discussion.

Literature Review

The textual analysis in financial markets is an emerging area, therefore, the corresponding taxonomies are still not clear. Researchers have used various dictionaries and machine learning algorithms to extract user sentiments from posts and articles in different contexts with varied degrees of success. To address this issues, recently, Loughran and McDonald (2015) compared four most widely used dictionaries, which are Henry (2008), Harvard's General Inquirer (GI), DICTION, and the L&M (Loughran and Mcdonald 2011) and showed that each word list has its expertise in different contexts, but the L&M dictionary is better than the rest three dictionaries in financial context for the following two reasons. First, it is more comprehensive than the rest, without missing commonly appearing negative or positive

Electronic copy available at: http://ssrn.com/abstract=2661064

words. Second, the L&M dictionary was created for the financial context analysis while GI and DICTION are not specifically designed to analyze financial communications.

Loughran and Mcdonald (2011) created six different word lists, out of which we will only use the "negative" and "positive" word lists, which have been most widely used in literature (Gurun and Butler 2012; Garcia 2013; and Chen et al. 2014). Our main focus is the fraction of negative words rather than the positive words as positive words are often negated to convey negative feelings (e.g., not good). A recent work (Chen et al. 2014) using the L&M dictionary has shown that the fraction of negative words in articles and comments from SA (SeekingAlpha.com) is negatively related to the subsequent abnormal return with three months holding period. Besides social media, researchers have also tested other traditional media like newspapers. Dougal et al. (2012) study the Wall Street Journal's (WSJ) columns and find that more pessimistic column tone are linked with more negative market returns in the following day based on the earlier work of Tetlock (2007). The research has also been extended to study documents that are related to corporate fundamentals. Huang et al. (2014) find that sentiments of an earnings press release misinforms market participants. Although these papers have used the L&M dictionary in many varied financial contexts, we have little understanding if this dictionary performs better or worse compared to machine learning techniques, such as Naïve Bayes and support vector machines (SVM).

Besides the wide usage of predefined dictionaries, machine-learning algorithms are also becoming popular for analyzing financial articles or comments. In the pool of algorithms, two are most extensively used to analyze the financial context, which are Naïve Bayes, and Support Vector Machine. Machine learning algorithms are trained on the training set. We could then apply the "knowledged" algorithms to the remaining sample dataset or out-of-sample dataset.

When all sentences are classified, by comparing the classified sentiments and self-disclosed sentiments, one could compute the accuracy of the algorithms. A common challenge in a couple of studies is the unavailability of self-disclosed sentiments, which makes it difficult to assess the accuracy of the classification scheme. For example, using a dataset from Yahoo! Finance message board, Kim and Kim (2014) found no evidence that investor sentiments will influence the stock return in the following days. Since, only 25.9% of the total messages have self-disclosed sentiments, it is hard to test the accuracy of the out-of-sample dataset. In comparison, our research could compute the exact accuracy, as all of the posts in our dataset have self-disclosed sentiments by the investors.

The Naïve Bayes algorithm, one of the oldest, is called "naive" because it assumes the words are independent of each other, even though it is quite unlikely. It has been used to classify messages (Leung and Ton 2015) as bullish, neutral or bearish, however, these studies didn't compare the performance of Naïve Bayes with other classification approaches. Antweiler and Frank (2004) is one of the first studies to implement NB classifier and demonstrate that positive message board posts are followed by negative returns on the next day. Although the authors have reported significant misclassification using NB algorithm, they don't use other classifiers for comparison or propose other alternative high-accuracy algorithms.

SVM is another extensively used classifier in the financial context. Malo et al. (2013) show that substantial performance could be improved by combining different underlying pattern analysis algorithms. Instead of comparing different classifiers, this research focuses on comparing the underlying algorithms and the combination effect of algorithms.

A few studies have implemented both NB and SVM classifiers. Tirunillai and Tellis (2012) compare the accuracy and sensitivity of the two algorithms. The overall results show that

SVM performs better than NB in the products review context. Zhang et al. (2012) conduct a comprehensive research on eight widely applied text classifiers to stock message board data. They find that NB performs better than SVM in the out-of-sample test.

There are two streams of studies in the literature. One stream is about comparing the performance of predefined dictionaries, which are used to analyze sentiments in a financial context. The other stream has focused on different machine learning algorithms to classify the sentiments into several groups. However, Chen et al. (2014) didn't report the accuracy of their classification approach by using the L&M word list. Tirunillai and Tellis (2012) has reported the classification accuracy using NB and SVM, but it was not in a financial context. Leung and Ton (2015) didn't report the accuracy for the out-of-sample data and was only using NB. To sum up, none of the studies has compared the performance of dictionaries and the machine learning algorithms in analyzing financial context in one research. Our research contributes to the literature by comparing the performance of one dictionary and two classifiers that are widely used in the financial context.

Data

The data for this study was obtained from Hot Copper (HC) message board (Hotcopper.com.au). HC allows users to make recommendations (posts) or comments on existing posts about stocks listed in ASX. For this study, we focus on all the fifty stocks, which are listed in the S&P/ASX 50 index. The sample contains all the messages for these stocks from January 2014 to March 2015.

We downloaded and saved the messages in a database with date and time stamp, user name, length of the post, content and sentiment. HC requires users to disclose a sentiment and

position along with their posts. Users can choose sentiment from "Sell", "Hold", "Buy" or "None" and position from "Not Held" and "Held". Users have to make a choice before the posts are eligible to be seen by other HC users. There is no such requirement on many other message boards, such as Yahoo! Finance. Some of the recent studies (Kim and Kim, 2014) show that only 26% messages from Yahoo Finance have sentiments disclosed by the users, compared to 100% for our dataset.

Methodology

This study compares the performance of a dictionary and two classifiers in a financial context. Specifically, we compare the performance of the L&M dictionary, Naïve Bayes classifier and Support Vector Machine based classification.

When using the L&M dictionary, we calculate the percentage of negative words for each post and get the median of its overall distribution. A post is bullish if its fraction of negative words is below this median; a post is bearish if its fraction of negative words is above the median (Chen et al. 2014).

For NB classifier, we define a message $\{W_k\}(k=1,...K)$ as a sequence of words, in which each word $\{W_k\}$ is indexed by k. The words from W_k could be categorized in a message of class C (e.g., bullish) or in counter class $\tilde{C}$ (e.g., bearish). The number of occurrence of words from W_k in class C or counter class $\tilde{C}$ are denoted as m_k and $\tilde{m}_k$.The total number of words that exist in class C or counter class $\tilde{C}$ are described as n and $\tilde{n}$. We observe the conditional probabilities of words contained in messages from the training set as $P(W_k \mid C) = m_k / n$ and

$P(W_k \mid \tilde{C}) = \tilde{m}_k / \tilde{n}$. Then we could calculate the posterior probability $P(C \mid W_k)$ from the prior probability $P(C)$ with: $P(C \mid W_k) = \frac{P(W_k \mid C)P(C)}{P(W_k)} = \frac{P(W_k \mid C)P(C)}{P(W_k \mid C)P(C) + P(W_k \mid \tilde{C})P(\tilde{C})}$ (1)

where $P(C)$ is the initial possibility that a message belongs to class C and $P(\tilde{C}) = 1 - P(C)$, while $P(C \mid W_k)$ is the probability that a message belongs to class C given that word W_k is observed. *PS* stands for a post. Using the naïve independence assumption the possibility for a sentence in class *C* is: $P(C \mid PS) = P(C \mid W_1, ..., W_n) = \frac{P(C)\prod_{i=1}^{n} P(W_i \mid C)}{P(W_1, ..., W_n)}$ (2)

Since $P(W_1, ..., W_n)$ is constant given the input, we use the following classification rule:

$$P(C \mid W_1, ..., W_n) \propto P(C)\prod_{i=1}^{n} P(W_i \mid C) \quad (3)$$

Finally, the classification with the highest posterior probability is chosen.

$$\hat{C} = \arg\max_{C} P(\mathrm{C})\prod_{i=1}^{n} P(W_i \mid C) \quad (4)$$

The main difference between NB classifiers lies in the assumptions they make towards the distribution of $P(W_i \mid C)$. For example, Multinomial NB utilizes the NB algorithm on multinomially distributed data. $P(W_i \mid C)$ is the probability that feature *i* appearing in a message that belongs to class *C*. The parameters $P(W_i \mid C)$ may be zero. In this case, the NB-algorithm uses Lidstone smoothing or Laplace smoothing.

Bernoulli Naïve Bayes focus on the data that is multivariate Bernoulli distributed. Thus, samples to be processed by this classifiers need to be represented as binary-valued feature

vectors. If the samples are other kind of data, the Bernoulli NB will binarize the input. And $P(W_i \mid C)$ is calculated based on: $P(W_i \mid C) = P(i \mid C)W_i + (1 - P(i \mid C))(1 - W_i)$ (6)

This method gives a penalty for the non-occurrence of a feature i and while the multinomial NB only ignore a non-occurrence feature.

Support Vector Machines (SVMs) are a set of supervised learning methods widely used for classification. A separation boundary will be produced by an SVM in a feature set. We consider *n* training observations, x_i, each of which is a p-dimensional vector of features. Each training set has an associated class label, which could be self-disclosed or manually classified. In our case, the labels y_i are "Buy" or "Sell". Thus, (x_i, y_i) represents pairs of features and labels. Then a hyperplane is constructed that could separate the training dataset "perfectly" according to the labels. To alleviate the problem of over-fitting, new parameters are introduced into the model: slack values ε_i, and budget *C*. To get the maximal margin hyperplane (MMH), we need to find the largest margin, which is the aggregate smallest perpendicular distance to a training observation from the hyperplane (as shown by the first expression in expression set (7)).

$$y_i(\beta_0 + \sum_{j=1}^{p} \beta_j X_{ij}) \geq M, \forall i = 1, ..., n \quad \sum_{j=1}^{p} \beta_j^2 = 1 \quad \varepsilon_i \geq 0, \sum_{i=1}^{n} \varepsilon_i \leq C \text{ (7)}$$

M is the margin, ε_i states the location for the ith observation relative to the margin and hyperplane, C controls how much the individual ε_i can be modified to violated the margin. In essence, C governs the bias-variance trade-off for the SVM. Besides, there are multiple kernels that SVM could implements, including linear, polynomial, rbf, sigmoid or precomputed.

Results and Contribution

The dataset is pre-processed before classification. Our raw data contains a total of 36557 posts. We deleted all the posts with self-disclosed sentiments "None", removed all the posts with zero "Length_of_Post" and got 19985 posts, which will be our main dataset (dataset M). Then we removed all the posts with "Hold" sentiments, and the remaining dataset contains 12920 messages (dataset S).

As all the posts have self-disclosed sentiments ("Buy", "Hold", or "Sell") in our dataset M, therefore, we don't need to classify the training set for NB and SVM manually. Following the literature (Leung and Ton 2015), we randomly draw 1000 posts and make an N-fold classification run. In N-fold cross-validation, the original 1000 posts are partitioned in N equal size subsamples. In the N subsamples, a single subsample is retained as the testing dataset and the rest N-1 subsamples are considered as the training dataset. Then we repeat the cross-validation process until each of the N subsamples has been used as the testing data for one time. We report the average of all these runs. However, if we only focus on the posts with "Buy" and "Sell" sentiments (dataset S), the accuracy is substantially increased (see Figure 1).

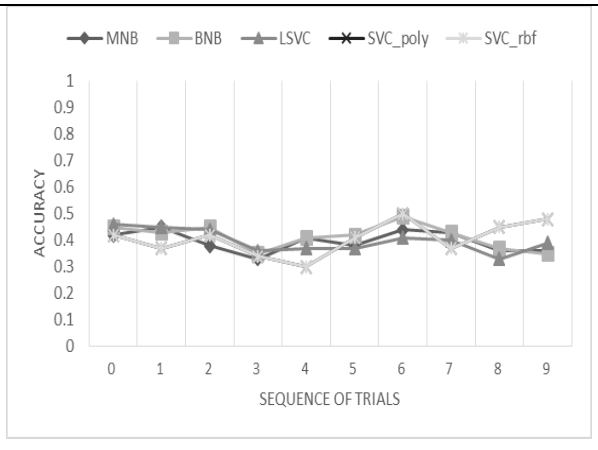

Figure 1a. 10 Fold Validation on Dataset with

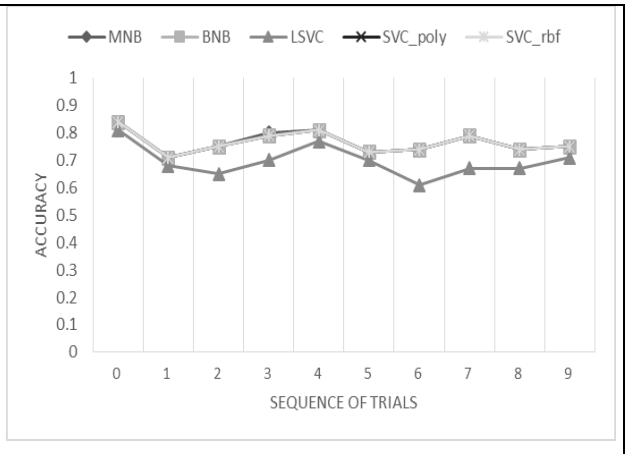

Figure 1b. 10 Fold Validation on Dataset

"Hold" Sentiment	without "Hold" Sentiment

In Figure 1, we report results for Multinomial Naïve Bayes (MNB), Bernoulli Naïve Bayes (BNB), Linear Support Vector Classifier (LSVC), SVM with radial basis function kernel (SVC_rbf) (Chang et al. 2010), and SVM with the polynomial kernel (SVC_poly) (Shashua 2009). The x-axis is the sequence number of trials, and the y-axis is the accuracy. The datasets for Figure 1a shows the classification accuracy of 1000 posts randomly chosen from the dataset M (with "Hold" sentiments). Figure 1b shows the classification accuracy of 1000 posts randomly chosen from the dataset S (without "Hold" sentiments).

It is clear that after removing the posts with "Hold" sentiments from the dataset M, we get a much better accuracy for all the classifiers. The reason could be that, although authors of these posts with "Hold" sentiment didn't indicate clear sentiment, the sentences that they normally use contain words that will show the tone tendency, which will make it hard to classify this kind of posts. To explore a bit further, we run a ten-fold validation on all the posts with only "Buy" and "Sell" sentiments. Dataset S is used for Figure 2. The results for SVM with rbf and poly kernel are quite similar, so the lines are almost identical. It is clear that the overall accuracy for the SVM with rbf or poly kernel is better than NB or other SVM classifiers.

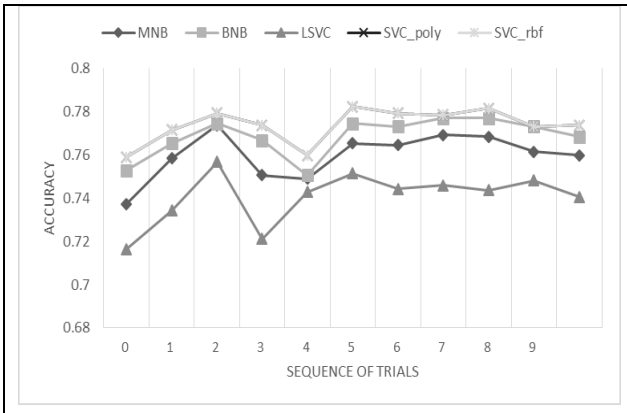

Figure 2. 10-Fold Validation for Dataset S

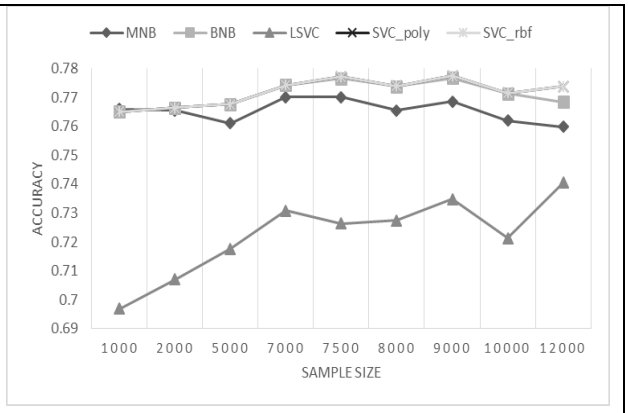

Figure 3. 10-Fold Validation for Different Size of Dataset

Figure 3 shows the result for 10-fold cross-validation on the dataset S with different sample size. In-sample test means you test the accuracy on the training dataset. Out-of-sample test means you test the accuracy on a dataset other than the training dataset. The x-axis tells the number of messages taken into consideration for the 10-fold cross-validation. The y-axis shows the performance accuracy. It is clear that the accuracy does not always increase with the accumulation of the number of messages. The accuracy for SVMs reaches peaks at 7500 and 9000 messages, which means that 750 and 900 messages are used as the training set. Antweiler and Frank (2004) and Leung and Ton (2015) hand-coded 1000 messages as the training set, but they didn't report the accuracy for the out of sample classification as their dataset didn't have self-disclosed sentiments (e.g. buy or sell). Tirunillai and Tellis (2012) collected 347,628 reviews and reported 78% average precision. Their result is comparable to ours (77.4%), but they didn't disclose their choice of training set, and they were not focusing on the financial context. Kim and Kim (2014) used 4000 messages as the training set and reported only 62.7% out-of-sample accuracy. In this case, the researchers need to choose the size of the training set wisely. Too big training set is not a good choice.

Figure 4 shows the accuracy for machine learning classifiers with different number of characters in each message taken into consideration. We could see that for most of the algorithms, the accuracy reaches the top with 1000 characters in each messages considered. This means that we don't need to consider all the characters in all of the messages to get the best accuracy. This is consistent with the result that 93.1% of the messages consume less than 1000 characters.

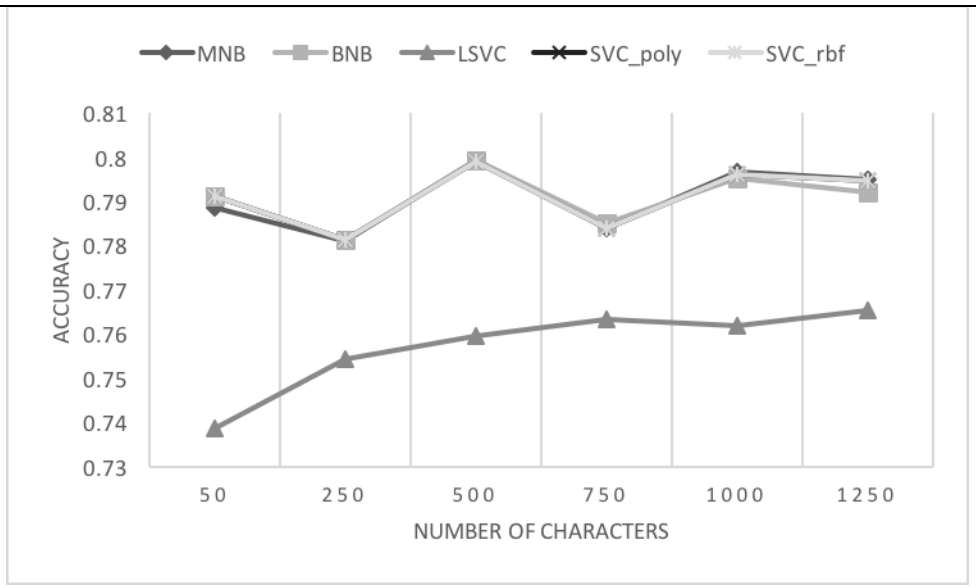

Figure 4. Accuracy for Machine Learning Classifiers with Different Number of Characters Taken into Consideration

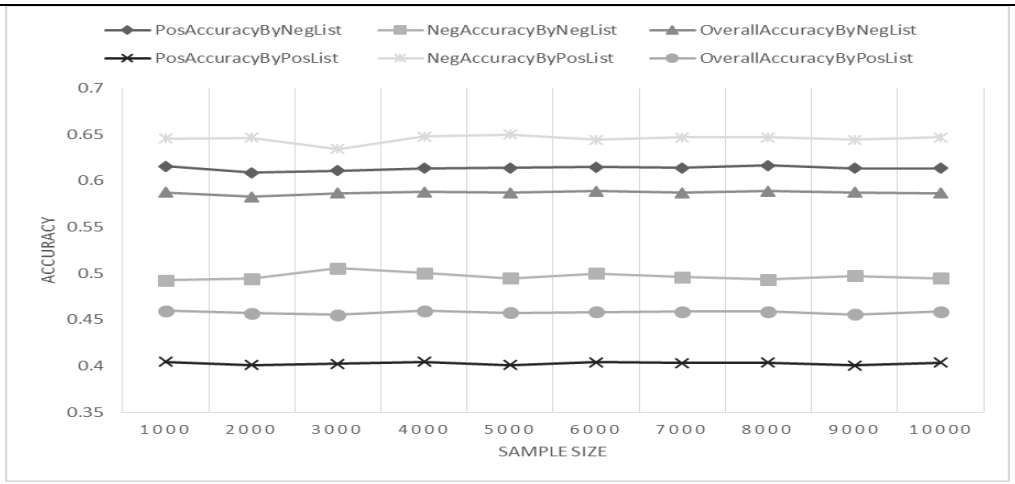

Figure 5. Accuracy for L&M Dictionary with Different Sample Size

After demonstrating the results for the machine learning classifiers, we test the accuracy of the classification by using the L&M dictionary. As we have discussed before, we classify a post as bullish if the percentage of negative words is below the median of the overall

distribution; a post is bearish if the percentage of negative words is above median (Chen et al. 2014). Also, we try to use the median of the percentage of positive words as another discriminative value. If the percentage of positive words is above the median of the overall distribution, the post is bullish. If not, the post is considered bearish. Figure 4 shows the result for L&M dictionary using dataset S.

Table 1. Comparison of Accuracy

Classifier	L&M	MNB	BNB	LSVC	SVC_poly	SVC_rbf
Accuracy	0.587	0.759799	0.768474	0.740511	0.773819	0.773819

In Figure 5, PosAccuracyByNegList means the classification accuracy for positive messages using the negative word list, and the rest coordinates on the x-axis are named accordingly. The over accuracy is 59% by using the negative word list. (Chen et al. 2014) has also implemented this tool, but they didn't report the accuracy as their dataset (equity research articles) didn't have the self-disclosed sentiment. Their research should have similar poor classification accuracy as this is not as sophisticated as the machine-learning classifiers. The overall accuracy using negative word list is better than the positive word list as the positive words could be negated to express the negative sentiment. It is a bit surprising that the classification accuracy for the negative words by the positive word list is much higher than the rest classification in L&M word list. The reason could be that the messages without any positive words have a higher tendency to convey negative sentiment.

As shown by Table 1, we find that the overall accuracy of machine learning classifiers is much better than the L&M dictionary on the individual message level, even though the L&M is

designed for the financial context. Among the tested NB and SVM classifiers, SVM with rbf or poly kernel performs best in the financial context.

Conclusion

This study examines the different textual analysis powers of one dictionary and two machine-learning classifiers in the financial context. Most of the dataset used by the previous research does not have self-disclosed sentiment for all the messages in their testing set. Thus, it's impossible for these research to report the classification accuracy. Even some research reported their classification accuracy; their out-of-sample accuracy was very poor or they failed to describe the number of messages in their training set. Due to these short comings, it was impossible for the previous studies to compare the classification accuracy of the L&M dictionary and these two machine-learning classifiers.

In this research, we fill the gap by showing that SVM with rbf or poly kernel performs best in the financial context analysis. In addition, researchers should try to train the machine learning algorithms on a reasonable size dataset to get better classification accuracy. This finding could help information systems and finance researchers to get better results when trying to analyze the financial context.

Reference

Antweiler, W., and Frank, M. Z. 2004. "Is all that talk just noise? The information content of Internet stock message boards," *Journal of Finance* (59:3), pp. 1259–1294 (doi: 10.1111/j.1540-6261.2004.00662.x).

Chang, Y.-W., Hsieh, C.-J., Chang, K.-W., Ringgaard, M., and Lin, C.-J. 2010. "Training and Testing Low-degree Polynomial Data Mappings via Linear SVM," *The Journal of Machine Learning Research* (11), JMLR.org, pp. 1471–1490 (available at http://dl.acm.org/citation.cfm?id=1756006.1859899).

Chen, H., De, P., Hu, Y. Y. (Jeffrey), and Hwang, B.-H. B.-H. 2014. "Wisdom of Crowds: The Value of Stock Opinions Transmitted Through Social Media," *Rev. Financ. Stud.* (27), p. hhu001– (doi: 10.1093/rfs/hhu001).

Dougal, C., Engelberg, J., Garcia, D., and Parsons, C. A. 2012. "Journalists and the Stock Market," *Review of Financial Studies* (25:3), pp. 639–679 (doi: 10.1093/rfs/hhr133).

Garcia, D. 2013. "Sentiment during Recessions," *The Journal of Finance* (68:3), pp. 1267–1300 (doi: 10.1111/jofi.12027).

Gurun, U. G., and Butler, A. W. 2012. "Don't Believe the Hype: Local Media Slant, Local Advertising, and Firm Value," *The Journal of Finance* (67:2), pp. 561–598 (doi: 10.1111/j.1540-6261.2012.01725.x).

Henry, E. 2008. "Are Investors Influenced By How Earnings Press Releases Are Written?," *Journal of Business Communication* (45:4), pp. 363–407 (doi: 10.1177/0021943608319388).

Hu, Tianyou, and Tripathi, Arvind 2015. "The Effect of Social Media on Financial Market Liquidity," *Forthcoming at the Proceedings of Thirty Sixth International Conference on Information Systems (ICIS),Fort Worth, Texas, USA*.

Kim, S.-H., and Kim, D. 2014. "Investor sentiment from internet message postings and the predictability of stock returns," *Journal of Economic Behavior & Organization* (107), Elsevier B.V., pp. 708–729 (doi: 10.1016/j.jebo.2014.04.015).

Leung, H., and Ton, T. 2015. "The impact of internet stock message boards on cross-sectional returns of small-capitalization stocks," *Journal of Banking & Finance* (55:July 1998), Elsevier B.V., pp. 37–55 (doi: 10.1016/j.jbankfin.2015.01.009).

Loughran, T., and Mcdonald, B. 2011. "When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10‐Ks," *The Journal of Finance* (66:1), Blackwell Publishing Inc, pp. 35–65 (doi: 10.1111/j.1540-6261.2010.01625.x).

Loughran, T., and McDonald, B. 2015. "Textual Analysis in Finance and Accounting: A Survey," *SSRN Electronic Journal* (doi: 10.2139/ssrn.2504147).

Malo, P., Sinha, A., Takala, P., Ahlgren, O., and Lappalainen, I. 2013. "Learning the roles of directional expressions and domain concepts in financial news analysis," *Proceedings - IEEE 13th International Conference on Data Mining Workshops, ICDMW 2013*, pp. 945–954 (doi: 10.1109/ICDMW.2013.36).

Shashua, A. 2009. "Introduction to Machine Learning: Class Notes 67577," Learning, , p. 109 (available at http://arxiv.org/abs/0904.3664).

Tetlock, P. C. 2007. "Giving content to investor sentiment: The role of media in the stock market," *Journal of Finance* (62:3), pp. 1139–1168 (doi: 10.1111/j.1540-6261.2007.01232.x).

Tirunillai, S., and Tellis, G. J. 2012. "Does Chatter Really Matter? Dynamics of User-Generated Content and Stock Performance," *Marketing Science* (31:November 2014), pp. 198–215 (doi: 10.1287/mksc.1110.0682).

Zhang, Y., Swanson, P. E., and Prombutr, W. 2012. "MEASURING EFFECTS ON STOCK RETURNS OF SENTIMENT INDEXES CREATED FROM STOCK MESSAGE BOARDS," *Journal of Financial Research* (35:1), pp. 79–114 (doi: 10.1111/j.1475-6803.2011.01310.x).