Digital Finance (2020) 2:1–13
https://doi.org/10.1007/s42521-019-00014-x

ORIGINAL ARTICLE

Sentiment analysis and machine learning in finance: a comparison of methods and models on one million messages

Thomas Renault[1]

Received: 20 March 2019 / Accepted: 11 September 2019 / Published online: 18 September 2019

Abstract
We use a large dataset of one million messages sent on the microblogging platform StockTwits to evaluate the performance of a wide range of preprocessing methods and machine learning algorithms for sentiment analysis in finance. We find that adding bigrams and emojis significantly improve sentiment classification performance. However, more complex and time-consuming machine learning methods, such as random forests or neural networks, do not improve the accuracy of the classification. We also provide empirical evidence that the preprocessing method and the size of the dataset have a strong impact on the correlation between investor sentiment and stock returns. While investor sentiment and stock returns are highly correlated, we do not find that investor sentiment derived from messages sent on social media helps in predicting large capitalization stocks return at a daily frequency.

Keywords Social media · StockTwits · Sentiment analysis · Machine learning · Asset pricing

JEL classification G10 · G12 · G14

1 Introduction

Can the stock market be predicted by extracting and analyzing large collections of publicly available textual documents? Recently, the exponential increase in the number of textual data available (big data) and the development of new techniques to analyze textual content (machine learning, natural language processing) have widely changed the ability for researchers to answer this question. Since the seminal papers of Antweiler and Frank (2004) and Tetlock (2007), finance researchers and market

✉ Thomas Renault
thomas.renault@univ-paris1.fr

[1] CES Sorbonne, Université Paris 1 Panthéon-Sorbonne, CES & LabEx RéFi, Maison des Sciences Économiques, 106-112, boulevard de l'Hôpital, 75013 Paris, France

participants have indeed extensively used computer approach to content analysis to explore the relation between textual sentiment—defined as the degree of positivity or negativity in qualitative content—and stocks prices. In the literature, a vast number of textual content have been analyzed, including media articles (Tetlock et al. 2008; Garcia 2013; Ahmad et al. 2016), corporate disclosures (Li 2010; Loughran and McDonald 2011; Price et al. 2012) and user-generated content on the Internet (Sprenger et al. 2014; Renault 2017; Bartov et al. 2017). While all projects are unique given the source of data used and the effect of textual sentiment on equity valuation (permanent impact, temporary impact or no impact), at the early stage of those projects, all researchers were faced with the following question: how to convert textual content (qualitative) into a textual sentiment indicator (quantitative).

Quantifying unstructured and noisy textual content is complex and involves numerous methodological issues related to the preprocessing of the data and the optimization of the algorithm used to quantify textual content. The number of text preprocessing that can be implemented is numerous (lowercase, stemming, lemmatization, part-of-speech tagging, stopwords removal, punctuation removal, etc.) and it is not easy to identify which transformation increases (decreases) the accuracy of the classification. The same is true for the choice of the algorithm: the large number of algorithms (Naive Bayes, SVM, logistic regression, random forest, multilayer perceptron, etc.) and the even greater number of hyperparameters for each algorithm lead to an immense number of combinations.[1] Furthermore, the answers relative to those methodological issues strongly depend on the type of data used (informal or formal content, short or long text), on the size of the dataset (few hundreds or millions of documents), on the availability of pre-classified messages (supervised or unsupervised learning), and on the type of documents (domain-specific or generic documents). While there is no one-fits-all solution, we nonetheless believe that some guidance and tips can help researchers to avoid common mistakes.

In this paper, we use a dataset of one million messages sent on the microblogging platform StockTwits to evaluate the performance of a wide range of machine learning algorithms and preprocessing methods for sentiment analysis in finance. Data from StockTwits are increasingly used in the literature (Oliveira et al. 2016; Renault 2017; Mahmoudi et al. 2018; Chen et al. 2019) and are relevant for sentiment analysis as users can express their sentiment—bearish (negative) or bullish (positive)—when they publish a message on the platform. This unique feature allows researchers to construct large datasets of classified messages without any manual classification. This is especially interesting as the size of the dataset is of the utmost importance for machine learning supervised classification, as we will demonstrate in the paper. We choose to adopt the practical point of view of a researcher in finance facing an arbitrage between the benefit of constructing the best possible sentiment indicator from a given collection of documents, and the cost associated with the creation of this indicator (time, complexity, lack of transparency and computing power costs). In practice, there are several hundreds of text processing, hundreds of machine learning

[1] We do not explore the impact of word embedding in this article (GloVe, Word2Vec). We let this for future research.

algorithms and millions of fine-tuning parameter combinations. However, as we will demonstrate, adopting a limited number of rules of thumbs is often enough to obtain a robust and reliable textual sentiment indicator.

To do so, we construct two datasets: one balanced dataset containing 500,000 positive messages and 500,000 negative messages, and one unbalanced dataset containing 800,000 positive messages and 200,000 negative messages. First, and considering a simple multinomial Naive Bayes model (tenfold cross-validation), we analyze the impact of the size of the dataset on the accuracy of the classification (the percentage of documents correctly classified out-of-sample). We find that the accuracy increases strongly with the size of the dataset: the performance of the classifier increases by nearly 10 percentage points when the size of the dataset increases from 1000 messages to 10,000 messages, and by 4.3 percentage points when the size of the dataset increases from 10,000 messages to 100,000 messages. The marginal accuracy improvement is decreasing and the accuracy reaches a plateau around 500,000 messages.[2]

Then, we consider the impact of considering continuous sequence of words (ngrams) instead of unigrams. We find that considering bigrams strongly improves the accuracy of the classification. For example, for a dataset of 250,000 messages, adding bigrams improves the accuracy of the classification by 2.2 percentage points compared to a processing with unigrams only. Trigrams and four-gram have no significant impact on the accuracy. Those results are consistent with the recent work of Mahmoudi et al. (2018) who find that bigrams significantly boost the performance of the classification. We also analyze the impact of a wide range of preprocessing techniques: stopwords removal, punctuation removal, text stemming, part-of-speech tagging and the inclusion of emojis. We find that the inclusion of emojis significantly increases the accuracy of the classification (+ 0.3 percentage points). The inclusion of punctuation, such as question mark and exclamation mark, also increases the accuracy (+ 0.3 percentage points).

Afterward, we compare a wide range of machine learning methods used in the literature (maximum entropy, support vector machine, random forest and multilayer perceptron) to our benchmark Naive Bayes model. We perform a grid search for hyperparameter optimization and we find that the best performance is achieved by a Maximum Entropy classifier, closely followed by a Support Vector Machine classifier. More complex and time-consuming algorithms do not improve the accuracy of classification. This result suggests that machine learning algorithms may be more decisive in complex context or linguistic structure, but that for short texts published on social media, the text preprocessing method plays a bigger role.

Then, we construct a sentiment indicator for five stocks (Apple, Microsoft, Facebook, Amazon, and Google) and we explore the relation between investor sentiment and stock returns during the year 2018. We compare our results when investor sentiment is computed without preprocessing and considering unigrams (benchmark method) and when sentiment is computed by including emojis, punctuation, and

[2] The accuracy only increases by 0.31 point of percentage when the size of the dataset increase from 500,000 messages to 1 million messages.

bigrams (optimal method). We find that the correlation between sentiment and stock returns is significantly higher when sentiment is derived using the optimal method (+55% on average). However, we do not find any strong evidence that investor sentiment helps to predict large capitalization stock returns at the daily level, consistent with findings from Sprenger et al. (2014) and with the efficient market hypothesis.

The paper is structured as follows. Section 2 describes the data. Section 3 presents the accuracy of classification for a wide range of text processing methods and machine learning algorithms. Section 4 explores the relation between investor sentiment and stock returns. Section 5 concludes.

2 Data

StockTwits is a microblogging platform where users can share ideas and opinions about the stock market. Since its creation in 2008, more than 150 million messages have been sent on the platform by a total of more than 100,000 users. For researchers in finance and computing science, StockTwits is a very valuable source of data as (1) messages sent on StockTwits are all related to financial markets and (2) users on StockTwits can use a toggle button to self-classify their messages as bullish (positive) or bearish (negative) before posting the message on the platform. This feature allows the construction of large datasets of labeled messages dedicated to financial markets.

We use the StockTwits Application Programming Interface to extract messages sent on StockTwits. We end up with a database of more than 35 million messages: 1,779,957 negative messages (5%), 7,887,332 positive messages (22%) and 26,238,809 non-classified messages (73%). The 4-to-1 ratio between positive and negative messages has already been documented in the literature: online investors tend to be bullish about the stock market on social media (see Antweiler and Frank (2004) or Avery et al. (2015)). This is true even for a year like 2018 during which the stock market has experienced a sharp drop from September to December (the S&P500 decreases by more than 20% during the period Sept–Dec 2018, and by 12.5% during the whole year 2018).

We construct two datasets: one balanced dataset containing 500,000 positive messages and 500,000 negative messages, and one unbalanced dataset containing 800,000 positive messages and 200,000 negative messages. We use tenfold cross-validation to measure the performance of the classifiers. We consider two indicators to quantify the quality of the prediction: the classification accuracy score and the Matthews correlation coefficient as in Mahmoudi et al. (2018). Considering the number of true positives (TP), the number of true negatives (TN), the number of false positives (FP) and the number of false negatives (FN), the classification accuracy score (AC) and the Matthews correlation coefficient (MCC) can be defined as:

$$\mathrm{AC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{1}$$

$$\mathrm{MCC} = \frac{\mathrm{TP} * \mathrm{TN} - \mathrm{FP} * \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}} \tag{2}$$

We use the NLTK[3] package for the text preprocessing and scikit-learn[4] to implement the machine learning classification. As in Oliveira et al. (2016) and Renault (2017), we replace all cashtags ($AAPL, $MSFT, $FB...) by a common word "cashtag", all urls (http / https) by a common word (linktag) and all users (@howardlindzon, @OpenOutcrier...) by a common word usertag. We impose a minimum of three words to include a message in the dataset. We tokenize each text using NLTK.

3 Results

3.1 Size of the dataset

What is the optimal dataset size for training a machine learning algorithm? While having more data tends to improve the accuracy of the classification, getting more data also has a cost (time to collect the data and time to run the algorithm on a larger dataset). In this subsection, we consider various dataset size to analyze the marginal improvement of having a larger dataset. We start from a very small dataset of 500 messages up to a very large dataset of 1 million messages. We consider a Naive Bayes classifier with default parameters (alpha=1.0, fit_prior=True, class_prior=None).[5] We remove punctuation and stopwords as in the default parameters of the *CountVectorizer()* function in scikit-learn. Table 1 presents the results for both the balanced and unbalanced dataset.

As expected, the accuracy of the classification strongly increases with the size of the dataset. For the balanced dataset (Panel A), the accuracy increases from 59.6% for a dataset of 500 messages up to 73.08% for a dataset of one million messages. The marginal improvement is decreasing: doubling the size of the dataset from 500,000 messages to 1 million messages only increases the accuracy by 0.3 percentage points, while doubling the size of the dataset from 25,000 to 50,000 increases the accuracy by 1.34 percentage points. This result suggests that, even if having more messages is always better, a decent accuracy can be attained with a dataset of 100,000 to 250,000 messages.[6] Similar results have been found on the unbalanced dataset (Panel B)—from 78% accuracy for a dataset of 1000 messages up to 82.612% for a dataset of one million messages.

This result questions the accuracy of the classification method used in numerous papers on the literature. For example, Antweiler and Frank (2004) use a training dataset of 1000 messages ; Das and Chen (2007) a dataset of 300–500 messages, Sprenger et al. (2014) a dataset of 2500 messages. While constructing large training dataset from unlabeled messages from Twitter is expensive or time-consuming[7],

[3] Natural Language Toolkit -https://www.nltk.org/.

[4] https://scikit-learn.org/stable/.

[5] https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html.

[6] We also find some that the accuracy reaches a plateau around 250,000–500,000 messages when a Logistic Regression algorithm is used.

[7] Ranco et al. (2015) state that "to achieve the performance of human experts, a large enough set of tweets has to be manually annotated, in our case, over 100,000".

Table 1 Size of the dataset and classification accuracy

Dataset size	AC	MCC
Panel A: balanced dataset		
500	59.6	0.197
1000	57.7	0.16
2500	62.44	0.249
5000	65.22	0.304
10,000	66.97	0.339
25,000	69.396	0.388
50000	70.738	0.415
100,000	71.27	0.425
250,000	72.407	0.448
500,000	72.777	0.456
1,000,000	73.084	0.462
Panel A: unbalanced dataset		
500	78.6	0.092
1000	78.0	0.134
2500	79.56	0.141
5000	79.04	0.194
10,000	79.31	0.206
25,000	79.908	0.254
50,000	80.708	0.291
100,000	81.441	0.327
250,000	82.094	0.354
500,000	82.372	0.368
1,000,000	82.612	0.378

This table presents the accuracy (AC) and the Matthews correlation coefficient (MCC) for various dataset size and for both balanced and unbalanced datasets. Results are based on a Naive Bayes classifier (tenfold cross validation), unigram text features and no text preprocessing

using pre-labeled data from StockTwits allows researchers to work with a very large dataset without the need to manually annotate messages. Extracting a few hundred thousands of messages using the StockTwits API is free and relatively easy. We believe that researchers working on unlabeled data from Twitter could take advantage of the pre-labeled data from StockTwits to train their classifiers.

3.2 Number of ngrams

Is considering a contiguous sequence of words (ngrams) instead of single words (unigrams) always improve the classification accuracy? We consider a dataset of 250,000 messages, following results from the previous subsection, and we analyze the precision of the classification when features are composed of unigrams (single words), unigrams and bigrams (sequences of two words), trigrams, and four-gram.

Table 2 Number of ngrams and classification accuracy

Ngrams	AC	MCC
Unigrams	72.407	0.448
Unigrams + Bigrams	74.618	0.493
Unigrams + Bigrams + Trigrams	74.668	0.494
Unigrams + Bigrams + Trigrams + 4-gram	74.522	0.491

This table presents the accuracy (AC) and the Matthews Correlation Coefficient (MCC) for different number of grams. Results are based on a Naive Bayes classifier (tenfold cross validation) and no text preprocessing. The dataset is composed of 250,000 messages (125,000 positive and 125,000 negative)

We present the results for the balanced dataset as results are similar on the unbalanced dataset. Table 2 presents the results.

As documented in Mahmoudi et al. (2018), considering bigrams improves significantly the accuracy of the classification. Classification accuracy (MCC) is equal to 72.407% (0.448) when features are composed only of unigrams and to 74.617% (0.493) when both unigrams and bigrams are considered as text features. Adding bigrams increases the precision of the classification as bigrams tend to capture more context around each word and allows a better understanding of the sequence of words preceded by "not" or other negating words. However, including trigrams and four-gram have no significant effect on the precision of the classification, consistent with Wang and Manning (2012).

3.3 Text preprocessing

Which preprocessing method should be implemented before running a classifier? In this subsection, we consider various preprocessing methods and we discuss the value of each technique: stopwords removal, emojis inclusion, punctuation inclusion, stemming and Part-Of-Speech (POS) tagging. We use the stopwords corpus from NLTK for stopwords removal, the Porter Stemmer for stemming, and the Penn Treebank part-of-speech tagset for POS tagging. We consider the following punctuation and signs to be included: "! ? % + – = : ;) (]" and we use unicode names to add emojis. Table 3 presents the results. The benchmark model is the same model as in Table 2 (unigram and bigram text features without text preprocessing based on a balanced dataset of 250,000 messages).

We find that including emojis and punctuation improves the precision of the classification, respectively, by 0.38 and 0.30 percentage points. This result is consistent with Mahmoudi et al. (2018) for emojis and with Renault (2017) for punctuation. The best preprocessing method on our sample is achieved by including both emojis and punctuation (accuracy of 75.32%). Emojis are widely used on social media and special attention should be paid to the inclusion of those features. Standard "bag-of-words" approaches often focus on words (a string of characters from A to Z) and, by removing special characters and punctuation, do not exploit the specificity of messages sent on social media.

Table 3 Text preprocessing methods and classification accuracy

Preprocessing	AC	MCC
Benchmark	74.618	0.493
Punctuation	74.92	0.499
Stem	74.312	0.486
Emoticons	74.996	0.5
StopWords	73.025	0.461
PosTagin	74.095	0.482
Emojis + punctuation	75.322	0.507

This table presents the accuracy (AC) and the Matthews Correlation Coefficient (MCC) for various preprocessing techniques. Results are based on a Naive Bayes classifier (tenfold cross validation), unigram and bigram text features. The dataset is composed of 250,000 messages (125,000 positive and 125,000 negative)

We also find that removing stopwords using the NLTK stopwords corpus significantly decreases the accuracy of the classification. We believe that this result is due to the fact that the stopwords corpus from NLTK includes words that could be very useful for sentiment analysis in finance such as “up”, “down”, “below” or “above”. Thus, researchers should not use the standard NLTK list and should consider a more restrictive list of stopwords for sentiment analysis (“a”, “an”, “the”...). This result is consistent with Saif et al. (2014) who show that Naive Bayes classifiers are more sensitive to stopword removal and that using pre-existing lists of stopwords negatively impacts the performance of sentiment classification for short-messages posted on social media. POS tagging and stemming also decrease the accuracy of the classification. When the dataset is large, stemming does not increase the accuracy, as also documented in Renault (2017). For example, the words “short”, “shorts”, “shorted”, “shorter”, “shorters” and “shorties” are used by online investors to express very distinct feelings. Stemming those words to a common root (“short”) might decrease the accuracy.

3.4 Machine learning methods

Are more complex machine learning methods always better than more simple techniques? In this subsection, we consider five machine learning algorithms: a Naive Bayes algorithm (NB), a Maximum entropy classifier (MaxEnt), a linear Support Vector Classifier (SVC), a Random Forest classifier (RF) and a MultiLayer Perceptron classifier (MLP). We perform a grid search for hyperparameter optimization using the scikit-learn package. Table 4 presents the results. The benchmark model is the same model as in Table 3 (unigram and bigram text features, including emojis and punctuation, based on a balanced dataset of 250,000 messages and considering a Naive Bayes classifier) except that we use a threefold cross-validation and we impose a minimum word frequency of 0.0001% to remove very infrequent words and reduce calculation time (GridSearchCV is very time consuming). For each

Table 4 Machine learning methods and classification accuracy

Algorithm	AC	Time
Multinomial Naive Bayes	73.568	2 s
Maximum entropy	74.451	8 minutes
Support vector machine	74.292	7 min
Random forest	71.665	170 min
Multilayer perceptron	73.829	32 min

This table presents the accuracy (AC) for various machine learning methods. Results are based on threefold cross-validation, with unigram and bigram as text features, including emojis and punctuation. The dataset is composed of 250,000 messages (125,000 positive and 125,000 negative). We impose a minimum word frequency of 0.0001% (or 25 occurrences) to remove very infrequent words and reduce computation time for GridSearchCV

algorithm, we also include the time needed to run the classification (for the optimal combination of hyperparameters).[8]

We find that more complex algorithms (Random Forest and Multilayer Perceptron) do not improve the precision of the classification compared to more simple methods such as Maximum Entropy or Support Vector Machine. Given the cost associated with the optimization of the hyperparameters (time, complexity, lack of transparency and computing power costs), we believe that a simple classifier such as Naive Bayes, support vector machine, or maximum entropy), as in Antweiler and Frank (2004) and Sprenger et al. (2014) will often do the trick for social media sentiment analysis.

4 Investor sentiment and stock returns

In this section, we explore the relation between investor sentiment and stock returns at a daily frequency. To do so, we extract all messages sent on Stocktwits during the year 2018 for the five biggest U.S. listed stocks: Apple ($AAPL), Amazon ($AMZN), Facebook ($FB), Google ($GOOG) and Microsoft ($MSFT). For each stock, we construct two daily investor sentiment indicators by averaging the sentiment of all the messages containing the cashtag of the company ($ sign followed by the ticker of the company) sent on Stocktwits between 4 p.m. and day $t-1$ and 4 p.m. on day t. We define $S_{i,t}$ the sentiment about stock i on day t. We compute two distinct sentiment indicators for each stock: one based on unigram without any text preprocessing (benchmark method) and the other one including bigrams, emojis, and punctuation (optimal method). The accuracy of the classification is equal to 72.41% for the benchmark method and to 75.32% for the optimal method. We extract daily price data from Thomson Reuters and we denote $R_{i,t}$ the daily return for

[8] The time will differ depending on the computing power and the optimization of the script.

Table 5 Correlation matrix—investor sentiment and stock reruns

Stock	Return/benchmark investor sentiment	Return/optimal investor sentiment
AAPL	0.3079	0.468
AMZN	0.3197	0.3821
FB	0.2556	0.419
GOOG	0.1952	0.3611
MSFT	0.0663	0.1108

This table presents the correlation between stock returns and investor sentiment at a daily frequency. Benchmark investor sentiment is computed by considering unigram and without text preprocessing. Optimal investor sentiment is computed by considering both unigrams and bigrams and by including emojis and punctuation

Table 6 Forecasting investor sentiment

Stock	β_1	β_2	R^2
AAPL	0.4836***	− 0.6895	20.54
AMZN	0.364***	− 0.1796	12.17
FB	0.5162***	− 0.5112	23.86
GOOG	0.2167***	0.6511	7.3
MSFT	0.5105***	− 0.259	25.48

This table presents the results of the equation $S_{i,t} = \alpha + \beta_1 * S_{i,t-1} + \beta_2 * R_{i,t-1} + \epsilon_t$ for each stock during the year 2018 (251 observations). Standard errors are computed using White's heteroskedasticity robust standard errors. The superscripts ***, **, and * indicate statistical significance at the 1%, 5%, and 10% levels, respectively

stock i on day t. Table 5 shows the correlation between investor sentiment and stock returns for both indicators (benchmark and optimal).

We find that the correlation between investor sentiment and stock returns is much higher when bigrams, emojis, and punctuation are used to derive investor sentiment from individual messages sent on social media. For example, for Apple (AAPL), the correlation is equal to 0.3079 when we use the benchmark method for sentiment analysis, compared to 0.468 when we use the optimal method. Similar results have been found for other stocks. This result shows that an increase in classification accuracy by less than 3 percentage points (i.e., the difference in accuracy between the benchmark model in Table 1 and the optimal model in Table 3) can have a significant impact on the precision of the investor sentiment indicators and on the correlation between sentiment and stock returns.

Last, for each stock, we consider the following equations to analyze (1) if previous day sentiment and return ($(t-1)$ forecast sentiment in t, (2) if previous day sentiment and return $(t-1)$ forecast return in t. Tables 6 and 7 present the value of the β_1 and β_2 coefficients for the two following equations:

Table 7 Forecasting stock returns

Stock	β_1	β_2	Adj R^2
AAPL	− 0.0111	0.0703	0.6
AMZN	− 0.0157	− 0.0165	0.51
FB	0.0106	− 0.074	0.59
GOOG	0.0085	− 0.0217	0.26
MSFT	0.0018	− 0.182**	3.22

Note: This table presents the results of the equation $R_{i,t} = \alpha + \beta_1 * S_{i,t-1} + \beta_2 * R_{i,t-1} + \epsilon_t$ for each stock during the year 2018 (251 observations). Standard errors are computed using White's heteroskedasticity robust standard errors. The superscripts ***, **, and * indicate statistical significance at the 1%, 5%, and 10% levels, respectively

$$S_{i,t} = \alpha + \beta_1 * S_{i,t-1} + \beta_2 * R_{i,t-1} + \epsilon_t \tag{3}$$

$$R_{i,t} = \alpha + \beta_1 * S_{i,t-1} + \beta_2 * R_{i,t-1} + \epsilon_t \tag{4}$$

We find that sentiment on day t is strongly related to sentiment on day $t-1$ (sentiment persistence) consistent with previous results on the literature. However, we do not find any evidence that investor sentiment helps in predicting stock returns, consistent with the efficient market hypothesis and with previous results from Sprenger et al. (2014) or Antweiler and Frank (2004) (i.e., the β_1 coefficients on Table 7 are not significant in all regressions). This result should not discourage researchers to explore more in depth the relation between online investor sentiment and stock prices. Investor sentiment might not have any predicting power at a daily frequency for large capitalization stocks, but opposite results might be found at the intraday level (Renault 2017), around specific events (Ranco et al. 2015), or for small-capitalization stocks (Leung and Ton 2015). We encourage future research in this area by emphasizing the importance of the size of the dataset and of the preprocessing method to derive investor sentiment indicators from short texts published on social media.

5 Conclusion

All projects exploring the relation between investor sentiment and stock returns are unique given the source of data used and the effect of textual sentiment on equity valuation (permanent impact, temporary impact or no impact). While there is no one-fits-all solution, we provide in this paper some guidelines to help researchers in finance in deriving quantitative sentiment indicators from textual content published on social media.

We find that the size of the dataset is of utmost importance—and in fact more important than the preprocessing method or the choice of the machine learning algorithm. In that regard, using pre-labeled data from StockTwits allows researchers to work with a very large dataset without the need to manually annotate messages

and can be useful for researchers working on unlabeled data from Twitter. While having more messages always improves the accuracy of the classification, we find that the marginal improvement is decreasing and that a dataset of approximately 100,000–250,000 labeled messages provides reliable indicators.

Then, we provide evidence on the importance of the text preprocessing method, and more precisely of the positive effect of considering emojis (emoticons) and punctuation. Messages on social media are very different from articles published on traditional media. In that regard, preprocessing methods widely used on longer texts (such as stemming, stopwords removal or POS tagging) are not necessarily well suited for sentiment analysis of short financial text published on social media. The impact might also depend on the size of the dataset: while stemming is interesting for a small dataset and can be useful to reduce the number of features, it does not always lead to an increase in accuracy for short texts.

We also demonstrate that more complex algorithms do not increase the classification accuracy. The results presented in this paper suggest that simple algorithms (Naive Bayes, Maximum Entropy) might be sufficient to derive sentiment indicators from textual messages published on the Internet.

Last, and exploring the relation between investor sentiment and stock returns, we do not find any empirical evidence that sentiment helps in forecasting large capitalization stock returns at a daily frequency. However, our findings suggest that the preprocessing method used to derive investor sentiment indicators from textual content has an important impact on the correlation between investor sentiment and stock returns. In that regard, we believe that previous results from the literature in which sentiment is derived using machine learning methods on a dataset of fewer than 10,000 messages, such as Sprenger et al. (2014), should be reassessed in the light of the results of this paper.

References

Ahmad, K., Han, J., Hutson, E., Kearney, C., & Liu, S. (2016). Media-expressed negative tone and firm-level stock returns. *Journal of Corporate Finance*, *37*, 152–172.

Antweiler, W., & Frank, M. Z. (2004). Is all that talk just noise? The information content of internet stock message boards. *The Journal of Finance*, *59*(3), 1259–1294.

Avery, C. N., Chevalier, J. A., & Zeckhauser, R. J. (2015). The "CAPS" prediction system and stock market returns. *Review of Finance*, *20*(4), 1363–1381.

Bartov, E., Faurel, L., & Mohanram, P. S. (2017). Can Twitter help predict firm-level earnings and stock returns? *The Accounting Review*, *93*(3), 25–57.

Chen, C. Y.-H., Despres, R., Guo, L., & Renault, T. (2019). 'What makes cryptocurrencies special? investor sentiment and return predictability during the bubble', *Working Paper* .

Das, S. R., & Chen, M. Y. (2007). Yahoo! for Amazon: Sentiment extraction from small talk on the web. *Management Science*, *53*(9), 1375–1388.

Garcia, D. (2013). Sentiment during recessions. *The Journal of Finance*, *68*(3), 1267–1300.

Leung, H., & Ton, T. (2015). The impact of internet stock message boards on cross-sectional returns of small-capitalization stocks. *Journal of Banking & Finance*, *55*, 37–55.

Li, F. (2010). The information content of forward-looking statements in corporate filings—A naïve bayesian machine learning approach. *Journal of Accounting Research*, *48*(5), 1049–1102.

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.

Mahmoudi, N., Docherty, P., & Moscato, P. (2018). Deep neural networks understand investors better. *Decision Support Systems*.

Oliveira, N., Cortez, P., & Areal, N. (2016). Stock market sentiment lexicon acquisition using microblogging data and statistical measures. *Decision Support Systems*, *85*, 62–73.

Price, S. M., Doran, J. S., Peterson, D. R., & Bliss, B. A. (2012). Earnings conference calls and stock returns: The incremental informativeness of textual tone. *Journal of Banking & Finance*, *36*(4), 992–1011.

Ranco, G., Aleksovski, D., Caldarelli, G., Grčar, M., & Mozetič, I. (2015). The effects of Twitter sentiment on stock price returns. *PloS One*, *10*(9), e0138441.

Renault, T. (2017). Intraday online investor sentiment and return patterns in the US stock market. *Journal of Banking & Finance*, *84*, 25–40.

Saif, H., Fernández, M., He, Y., & Alani, H. (2014). 'On stopwords, filtering and data sparsity for sentiment analysis of twitter'.

Sprenger, T. O., Sandner, P. G., Tumasjan, A., & Welpe, I. M. (2014). News or noise? Using Twitter to identify and understand company-specific news flow. *Journal of Business Finance & Accounting*, *41*(7–8), 791–830.

Tetlock, P. C. (2007). Giving content to investor sentiment: The role of media in the stock market. *The Journal of Finance*, *62*(3), 1139–1168.

Tetlock, P. C., Saar-Tsechansky, M., & Macskassy, S. (2008). More than words: Quantifying language to measure firms' fundamentals. *The Journal of Finance*, *63*(3), 1437–1467.

Wang, S., & Manning, C. D. (2012). Baselines and bigrams: Simple, good sentiment and topic classification, in 'Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2'. *Association for Computational Linguistics*, 90–94.