Gianluca Anese | Marco Corazza | Michele Costola | Loriana Pelizzon

Impact of Public News Sentiment on Stock Market Index Return and Volatility

SAFE Working Paper No. 322

Leibniz Institute for Financial Research SAFE
Sustainable Architecture for Finance in Europe
info@safe-frankfurt.de | www.safe-frankfurt.de

Electronic copy available at: https://ssrn.com/abstract=3937901

Impact of public news sentiment on stock market index return and volatility*

Gianluca Anese†
Marco Corazza‡
Michele Costola§
Loriana Pelizzon¶

October 2021

Abstract

Recent advances in natural language processing have contributed to the development of market sentiment measures through text content analysis in news providers and social media. The effectiveness of these sentiment variables depends on the implemented techniques and the type of source on which they are based. In this paper, we investigate the impact of the release of public financial news on the S&P 500. Using automatic labeling techniques based on either stock index returns or dictionaries, we apply a classification problem based on long short-term memory neural networks to extract alternative proxies of investor sentiment. Our findings provide evidence that there exists an impact of those sentiments in the market on a 20-minute time frame. We find that dictionary-based sentiment provides meaningful results with respect to those based on stock index returns, which partly fails in the mapping process between news and financial returns.

Keywords: Public financial news, Stock market, NLP, Dictionary, LSTM neural networks, Investor sentiment, S&P 500

JEL Classification: G14, G17, C45, C63

*We thank Luca Coraggio of the University of Naples, Tomasz Gubiec of the University of Warsaw, Dian Kusumaningrum of Prasetiya Mulya University, Giovanni Zambruno of the University of Milano-Bicocca, and the other participants of the 2nd One-Day Workshop on Machine Learning for Finance, held at the Ca' Foscari University of Venice in 2020.

†Ca' Foscari University of Venice (Italy), Email: gianluca.anese@unive.it.

‡Ca' Foscari University of Venice (Italy), Email: corazza@unive.it.

§Ca' Foscari University of Venice (Italy) and Leibniz Institute for Financial Research SAFE, Email: michele.costola@unive.it.

¶Ca' Foscari University of Venice (Italy) and Leibniz Institute for Financial Research SAFE, Goethe University Frankfurt (Germany), Email: pelizzon@safe.uni-frankfurt.de. (Corresponding author.)

Electronic copy available at: https://ssrn.com/abstract=3937901

1 Introduction

Financial and economic news represents one of the main sources of public market knowledge that exerts an impact on stock prices. This type of information is available both at the macroeconomic (e.g., the periodic release of economic indicators) and microeconomic (e.g., quarterly release on firms' earnings) levels. For instance, Garcia (2013) studies the effect of the financial news from the *New York Times* during the twentieth century and shows its predictive ability on daily stock returns, particularly during recession periods. Recently, the availability of social media data from the Google search engine (Costola et al., 2020) and tweets (Iacopini and Santagiustina, 2021) has provided several insights about investor psychology and its impact on financial stock returns. Among others, Caporin and Poli (2017) and Xing et al. (2018) provide interesting surveys on firm-specific news and sentiment; Caporin and Poli (2017) show that augmented-data models including news variables provide superior forecasts.

Progress in natural language processing (NLP) has contributed to the development of market sentiment measures in economics and finance based on text sources like news providers and social media. For instance, Atkins et al. (2018) construct machine Learning (ML) models to represent information from news feeds and simple naïve Bayes (NB) classifiers to predict the market direction of movements. Empirical results from stocks and stock indices in the US market show that the average directional prediction accuracy for volatility on the arrival of new information is 56%, while that of the asset close price is no better than random. Souma et al. (2019) define news sentiment based on stock price returns averaged over one minute immediately after a news article has been released. They analyze the intraday Thomson Reuters News Archive and high-frequency DJIA 30 Index from 2003 to 2013 through a combination of deep learning (DL) methodologies and report good forecasting accuracy of this approach. Vicari and Gaspari (2020) investigate the possibility of trading on news sentiment through a long short-term memory (LSTM) neural network. They use this tool to forecast market sentiment using news headlines. The prediction is based on

Electronic copy available at: https://ssrn.com/abstract=3937901

the DJIA Index and is obtained by analyzing 25 daily news headlines available from 2008 to 2020. Testing is developed in real-world scenarios. The forecasting accuracy of this approach is around 58%. Wan et al. (2021) apply convolutional neural network with an LSTM neural network to extract news sentiment on 87 companies reported on Reuters for a period of seven years. They investigate the propagation of such sentiment in company networks and evaluate the associated market movements in terms of stock price and volatility. They also find significant abnormal market return and volatility in days with high sentiment levels. However, the effectiveness of these sentiment variables may vary by source.

In light of the above research, it is worth investigating the impact of publicly available financial news on the stock market immediately after its release. This is the aim of the present study. Using different automatic labeling techniques based on either stock index returns or dictionaries, we extract investor sentiments by means of a classification problem applied to financial news data by adopting *LSTM neural networks*.

The main contributions of the paper to the current literature are as follows. First, we analyze high-frequency market reactions to the release of financial news to the public to assess whether this information is to some degree informative, as privileged and private financial information generally is. Second, we make use of a novel technique to label financial news based on broad stock indices. Note that, by construction, the informativeness of these indices is more generic than that of narrow-based information, say, at the firm level. For instance, one of the pioneering studies in the field, Groß-Klußmann and Hautsch (2011), analyses high-frequency market reactions to stock-specific news flow and shows their ability to predict future prices. By making this choice, we aim to detect the overall reaction at the market level, which has not, to our knowledge, been previously investigated. Third, we couple the index-based labeling procedure for financial news with some alternative labeling techniques based on three *dictionaries* used widely in the financial literature to compare the outcomes coming from our stock index-based labeling approach (see Section 3) with those coming from standard dictionary-based ones. Finally, the obtained sentiment proxies are

Electronic copy available at: https://ssrn.com/abstract=3937901

used as exogenous variables on an EGARCH model fitted on S&P 500 Index intraday returns to check their out-of-sample explanatory power. Our findings show that these sentiments are significant market predictors in a 20-minute time window after the public release of news.

Operationally, financial news is obtained from Reuters, one of the main information providers in the market. Specifically, we performed web scraping on Reuters.com, which is freely available to the public. Each news item is labeled using stock index return- and dictionary-based approaches. The former classifies news based on S&P 500 Index and VIX Index log returns, while the latter makes use of dictionaries widely used in the financial literature: Loughran and Mc Donald's dictionary (Loughran and McDonald, 2011), Henry's dictionary (Henry, 2008) and the Harvard IV-4 General Inquirer dictionary (Harvard University, 1960). These are abbreviated LM, HE, and GI dictionaries, respectively. Then, LSTM neural networks are used to solve a three-class classification problem.

Our news-based sentiment shows the predictive power of using out-of-sample data, proving through the EGARCH model the existence of a relationship between news sentiment and stock returns and volatility. In particular, we find that dictionary-based sentiment provides meaningful results with respect to those based on stock index returns, which partly fails in the mapping process between news and financial returns. Indeed, the labeling technique that classifies news on the basis of stock index movements proved to be questionable as it can be affected by several forces that might negatively impact the accuracy of the classifier.

The remainder of this paper is organized as follows. Section 2 reviews the main supervised ML and dictionary-based techniques for text classification proposed in the literature. Section 3 presents the classification model and the main pre-processing techniques used to transform the raw texts into vectors representing the individual words of each news article. Section 4 describes the web-scraping approach used to collect data and the automatic labeling techniques to classify news articles. Section 5 discusses the findings of the LSTM learning process and reports the results of the classifications for the empirical analysis. Section 6 presents the sentiment variables obtained. Section 7 validates the informative content

Electronic copy available at: https://ssrn.com/abstract=3937901

of the sentiments obtained using an EGARCH model fitted on the log returns of the S&P 500 Index. The final section concludes.

2 Literature Review

In this section, we discuss the main text classification methods based on supervised ML techniques and dictionary-based approaches as presented in the literature.

Within the first research area Medhat et al. (2014) present the *NB classifier*, which computes the probability that a document belongs to a given class based on the distribution of the words in the document and assumes the independence of all its features. The authors also mention the *Bayesian network classifier*, which is rarely used because it assumes all features to be fully dependent and thus requires a complete joint probability distribution to be specified.

The *Maximum Entropy classifier*, as explained by Nigam et al. (1999), can be used when a joint probability distribution is unavailable. According to these authors, uniform distribution is preferred in these cases and updated using the constraints emerges from the training data. After selecting some relevant features from the text, they compute the expected frequency of the features words over the training data and set them as constraints on the conditional distribution.

Medhat et al. (2014) introduce *linear classifiers*, such as *support vector machines* (SVMs) and *artificial neural networks* (ANNs). The former separates data into classes, defining a separating hyperplane that maximizes the normal distance of any data points. ANNs, meanwhile, are universal function approximators that mimic the functioning of the human brain and are made up of units called (artificial) neurons. Basically, each neuron receives a vector of inputs $\overline{X_i} = \{x_{i,1}, .., x_{i,n}\}$ and combines them linearly using a vector of weights $A = \{a_1, .., a_n\}$ to produce $p_i = A \cdot \overline{X_i}$. In the case of binary text classification, the sign of p_i represents the label, positive or negative, to assign to the i-th document. The study we

Electronic copy available at: https://ssrn.com/abstract=3937901

present here uses a particular type of ANN, the aforementioned LSTM (see Section 3.2) for the main technical aspects, only recently been applied to this research field (e.g., Vicari and Gaspari, 2020; Wan et al., 2021).

Other supervised learning algorithms mentioned by Medhat et al. (2014) are *decision tree classifiers*, which follow a recursive approach to create sub-partitions of data based on the presence of one or more words, and *rule-based classifiers*, which divide the data defining sets based on *IF-THEN* rules: *IF* some rules are satisfied by a document, *THEN* that document can be assigned with the corresponding class label. Rules are defined during the training phase on the basis of criteria such as *support* and *confidence*. The former counts the number of times a specific rule, for instance the presence of a given word in the text, is satisfied in the whole training set, while the latter represents the conditional probability of observing a given label when the rule associated with that label is satisfied.

Yadav et al. (2019) use a supervised ML approach to classify real-time news headlines. The authors automatically label all headlines based on net buying pressure[1] patterns in the S&P NIFTY Index and then use NB classifiers and SVM for text classification, finding that the best alignment window for futures markets in India is five minutes.

Atkins et al. (2018) prove that financial news makes a better job in predicting stock market volatility than stock returns. To reduce data dimensionality, they use *Latent Dirichlet Allocation*, a generative technique that helps divide each document into a set of *topics* which are word sets and that are generated on the basis of the words they contain. Instead of having documents made up of a large number of words, or n-grams[2], each document is built up as a set of topics, thus reducing dimensionality. Specifically, they create a list of topics for each 60-minute time interval, by assigning a sparse feature vector that counts the number of times a topic appears in that interval. Then, they label these feature vectors based on the binary direction of volatility changes during the following time interval. Finally, they

[1] According to the authors, net buying pressure can be defined as "the difference between the number of buyer-initiated trades and the number of seller-initiated trades calibrated from the bid-ask quotes".

[2] An n-gram is a sequence of n consecutive words.

Electronic copy available at: https://ssrn.com/abstract=3937901

perform text classification using the NB classifier and justify this choice by the fact that it has high empirical performance even under the simplistic assumption that each feature is independent of the others. In our research, we refine this kind of investigation on the predictive capabilities of financial news using the sentiment proxies generated by our approach as exogenous variables in the mean and variance equation of an EGARCH model.

Souma et al. (2019) classify financial and economic news using an NLP ML approach. They train a recurrent neural network with LSTM units, which is particularly suitable for capturing long-term dependencies among words in a text. Before feeding the training algorithm, they perform word embedding using global vectors (GloVe) as a word representation method to convert words to vectors.[3] GloVe contains pre-trained word vectors based on Wikipedia documents. The authors assign labels to each news article based on stock returns over the next one-minute period and obtain a classification of positive and negative news.

Note that the prominent literature on financial news classification through ML techniques generally considers narrow-based indices (the DJIA 30 and the NIFTY Indexes are considered in Souma et al., 2019; Yadav et al., 2019, , respectively). Unlike this literature, here we take into consideration a broad stock index, namely, the S&P 500, to stress the exploratory capabilities of our approach. In our study, another important role is played by the techniques of labeling financial news based on dictionaries (see Section 1). In this research context, among the recent contributions in the literature, Li et al. (2014) first implement a generic stock price prediction framework, plug in six different models with different analytical approaches, and use GI and LM to construct a sentiment space in which textual news is projected. They conduct experiments on five years of historical Hong Kong Stock Exchange prices and news. Their main findings show that at the individual stock, sector, and index levels, model using sentiment analysis outperform the bag-of-words model in both validation and testing sets and that there is a minor difference between the models using the two different dictionaries.

[3]Embedding consists in assigning a vector to each word in the dictionary such that words with similar meaning are located close to each other in the vector space.

Electronic copy available at: https://ssrn.com/abstract=3937901

Wang et al. (2015) work on sentiment analysis retrieved from SeekingAlpha articles and StockTwits messages, two social media platforms, and analyze their correlation with S&P 500 Index movements, finding the former has better explanatory power, even though sentiment-based investment strategies have generally poor performances. They use a dictionary-based approach for SeekingAlpha analysis, relying on LM, and a supervised ML approach for StockTwits, using labels assigned directly by users. Their best model is that based on SVM.

Loughran and McDonald (2015) highlight the danger coming from possible misclassification when labeling financial words. With particular reference to the Diction platform, commonly used to assess the tone of business documents in the accounting and finance literature, they argue that it is inappropriate for gauging the tone of financial disclosures. Indeed, about 83% of the Diction optimistic words and 70% of the Diction pessimistic words appearing in a large sample are likely misclassified. They conclude that the LM appears better at capturing tone in business text than Diction.

Lastly, Mangee (2018) provides evidence that marketplace context matters for understanding stock price behavior. To this end, investor sentiment, extracted by reports from the Wall Street Journal and Bloomberg News outlets, is compared across two dictionaries: GI and LM. He finds a negative relationship between measures of investor pessimism and real stock returns and that this relationship is statistically significant only for the context-specific measures. These results suggest that contextualized investor sentiment is able to explain medium- to longer-term swings in aggregate stock prices.

In this paper, we continue this line of research on the use of dictionary-based labeling techniques. In particular, we deepen it by comparing the results from these techniques with those from our stock index-based labeling approach.

Electronic copy available at: https://ssrn.com/abstract=3937901

3 Methodology

In this section, we present the approach underlying our classification problem and the main pre-processing techniques used to transform the raw text into vectors representing the individual words of each article.

3.1 Pre-processing

Before being fed into the neural network, all financial news was pre-processed using the Deep Learning Toolbox in MATLAB. As for the learning phase, the first step was dividing data into three sets; namely, Training, Validation, and Testing sets. The Training set was used to find the optimal weights through backpropagation. Many algorithms, like stochastic gradient descent and Adam, divide the training set into subsets of observations named *mini-batches*. The neural network computes the total loss function at the conclusion of each mini-batch and updates the parameters, which can be used by the following mini-batch. The Validation set, which is itself divided into the same number of mini-batches, was used to test the model with the parameters optimized by the neural network at each mini-batch. The Testing set was used to test the trained neural network when the learning phase was completed.

In order to perform the learning task, we first divided the data set into two parts. The first one contains 75% of all articles and represents both Training and Validation sets, while the second part was used as the (out-of-sample) Testing set. Then, the first part of the data was divided into Training and Validation sets using the holdout cross-validation technique, applying the $^2/_3 : {}^1/_3$ ratio rule. After specifying the percentage of data to allocate to each of these subsets, the holdout cross-validation technique allows for the random selection of observations. Thus, the Training set represents 50% of all observations, while the Validation and Testing sets represent 25% each.

Pre-processing is an important phase and can heavily influence the accuracy of text classification. First, as is standard, documents were tokenized, and punctuation and stop

Electronic copy available at: https://ssrn.com/abstract=3937901

words were erased. Then, all words were converted to lower case, to avoid duplication. Another useful technique to reduce dimensionality is lemmatization, which consists of keeping only the roots of each word, as in stemming. The difference is that stemming uses general rules to cut off the final part of a word; for instance, it deletes suffixes like "ing" or "ed," while lemmatization is a more complex method that is able to turn each word into its dictionary form (Heidenreich, 2018). Words like "are" and "is" are transformed into "be," for instance. MATLAB provides pre-trained lemmatization tools based on widely available English dictionaries. To apply lemmatization, *part of speech* information is needed. This means words are categorized as nouns, verbs, adjectives, adverbs, and so on. In addition, words with less than two characters or more than fourteen characters were removed. Lastly, we removed "{R, r}euters" from the corpus, as it was unlikely to be informative.

Tokens went through encoding, which transformed them into numerical indexes, expressed by a sparse vector of length equal to the dictionary size and with all zero values but the one corresponding to the specific word in the vocabulary. In this way, tokens could be recognized by the classification algorithm.

Another important aspect to consider is document size. Each document is an article that contains an arbitrary number of words. However, best results could be achieved when the Training set contained documents of similar length, because infrequent long articles might have biased the learning phase; therefore, their length had to be reduced. Figure 1 represents the length of documents that belong to the Training set using a histogram. Most documents did not exceed 360 words, so this value can be used as an appropriate threshold. All articles that exceed this limit were truncated, while shorter articles were padded, meaning they were filled in with zero vectors.

3.2 LSTM neural network

In this section, we describe the six-layer ANN architecture (see Figure 2) that we used to divide news into three classes: positive, negative, and neutral. After encoding, the next

Electronic copy available at: https://ssrn.com/abstract=3937901

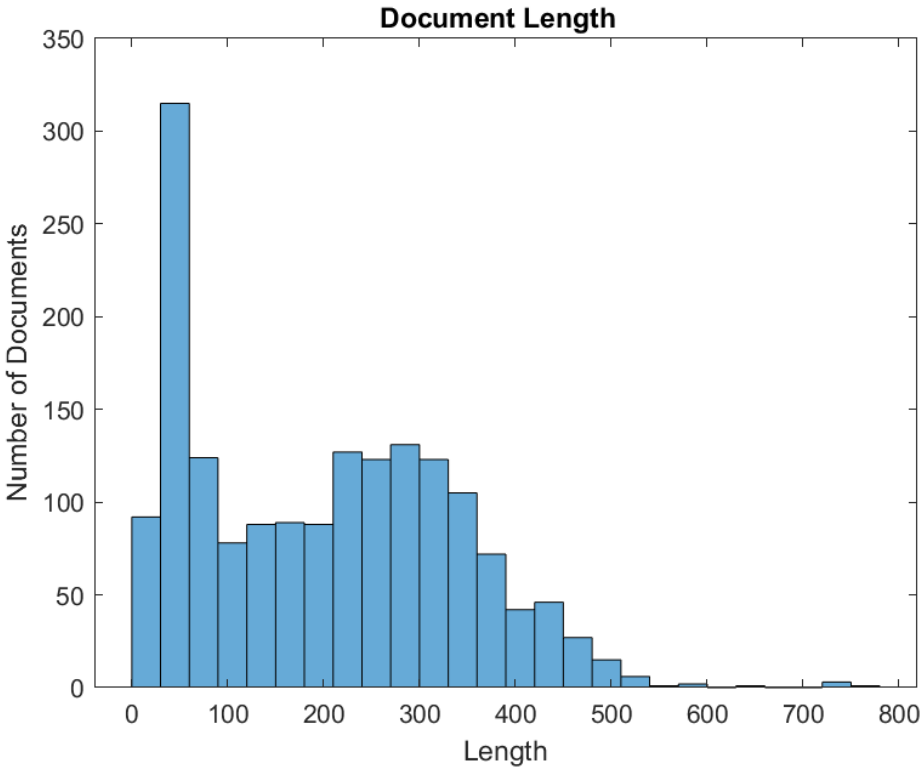


Figure 1: Length in words of news articles included in the Training set.

step consisted of word embedding, a very common practice before training a neural network. Embedding allows for dimensionality reduction and can be useful to nuances the semantic of words by assigning to each word a vector of real numbers. Among the different alternatives, we decided to include an embedding layer in the ANNs used to train the model. A different solution might be using pre-trained word embedding, but in this case, we likely would have had to rely on an overly generic training set, not specific to the financial context.

The first step of ANN learning is feeding the neural network with sequences of words in the form of encoding vectors with a length equal to the number of words in the dictionary, k. Therefore, the first layer was a one-dimensional *input layer* that took each word of the sequence and passed it on to an *embedding layer*, as shown in Figure 2. The Embedding layer mapped each word to a d-dimensional dense vector, with $d < k$, and adapted it to the corpus during the learning phase. The weighting matrix computed by the neural network for this layer is a $d \times k$ matrix.

Electronic copy available at: https://ssrn.com/abstract=3937901

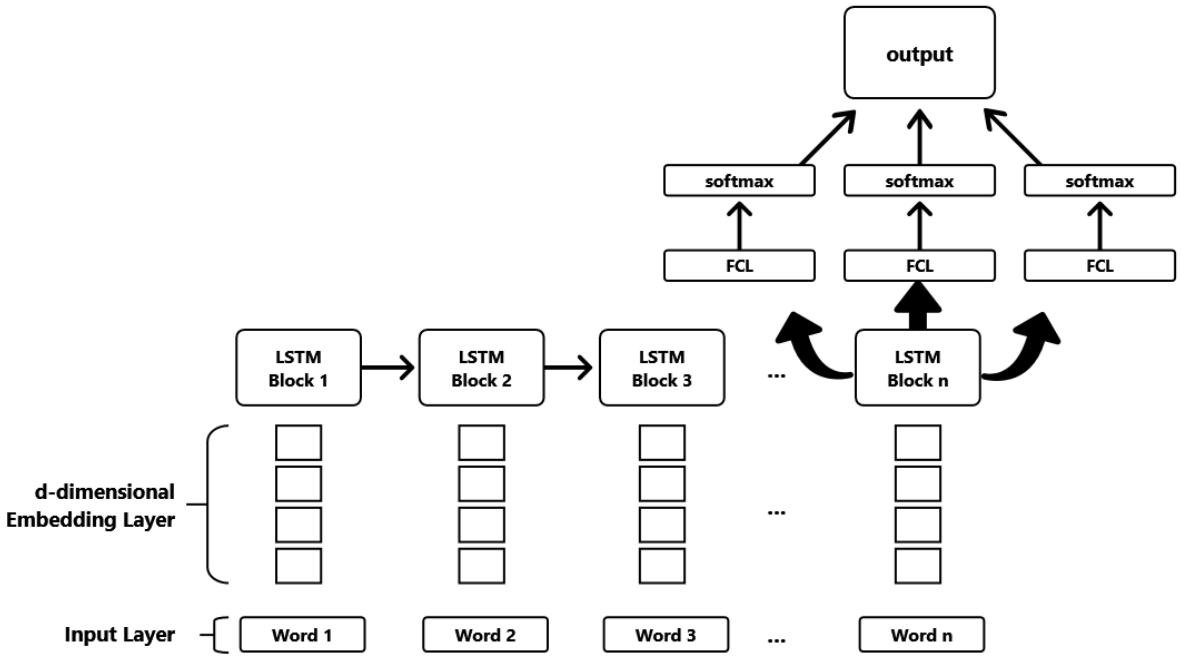


Figure 2: ANN's architecture with LSTM layer.

After the embedding layer, vectors were transferred to the *LSTM layer*, which learned long-term dependencies in the documents; these dependencies are comprised of as many LSTM blocks as there are words in the article under consideration. Each vector is associated with an LSTM block, which includes multiple LSTM units. The number of hidden units determines how much information is remembered between time steps. It should be fixed to an appropriate integer value, so as not to overfit the training data.

The input weights to compute for each LSTM block are defined by a $4u \times d$ matrix, where u is the number of units of each LSTM block, recalling that an LSTM unit has four layers (forget gate, input gate, candidate cell and output gate). In addition, each LSTM block has a $4u \times u$ recurrent weight matrix that must be computed by the neural network at each time step. Finally, the layer learns also a $4u$-dimensional vector of biases.

Electronic copy available at: https://ssrn.com/abstract=3937901

When the last final LSTM block is reached, outputs are passed on to a *fully connected layer* (FCL) with three activations, matching the number of classes (positive, negative, and neutral). The number of weights among these two layers is equal to the product of the three activations and the number of units of the LSTM block u. A weight matrix of size $c \times u$ is involved, where c is the number of classes. This fully connected layer receives as inputs the results of the activations of the LSTM layer and computes three weighted sums of these inputs, adjusting the weights together with all the others in the neural network.

When classification involves only two classes, only one activation is necessary for this layer and a hyperbolic tangent function (or a sigmoid) pushes its output between -1 and 1. The closer the value of this function is to -1, the more the document is associated with one class; the opposite holds true when the value approaches 1. In a three-class classification, three activations are needed, and their output is passed on to a *Softmax* function, which generalizes the sigmoid function for multi-class classification. Broadly speaking, the *Softmax* function computes the conditional probability of a class x_i, with $i = 1, 2, \ldots, k$, over all possible classes, as shown by Equation 1:

$$Softmax(x) = \frac{\exp(x_i)}{\sum_{j=1}^{k} \exp(x_j)}. \tag{1}$$

The $Softmax$ function takes values between 0 and 1, and the sum of the probabilities of all classes is 1, $\sum_{j=1}^{k} Softmax(x_i) = 1$.

The final layer is the *classification output layer* (Karpathy, 2015), which takes the probabilities provided by the $Softmax$ layer and computes a particular type of loss function, called *cross-entropy loss function*, for all documents. Loss function is used for classification tasks and is described by Equation 2.

$$Loss(y_i, \hat{y}_i) = -\sum_{i=1}^{n} \sum_{j=1}^{k} y_{ij} ln(\hat{y}_{ij}), \tag{2}$$

where y_{ij} represents the true class distribution of the i-th document and is an indicator

Electronic copy available at: https://ssrn.com/abstract=3937901

function that takes value of 1 when the processed i-th document belongs to the j-th class and zero otherwise. The output of the $Softmax$ layer is $\hat{y}_{ij}$ and represents the estimated probability of the i-th document belonging to the j-th class.

Equation 3 shows an example of the cross-entropy loss function computed in vector form for the individual i-th document, assuming the i-th document belongs to class number $j = 3$.

$$Loss(y_i, \hat{y}_i) = - \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} ln(\hat{y}_{i1}) & ln(\hat{y}_{i2}) & ln(\hat{y}_{i3}) \end{bmatrix} = -ln(\hat{y}_{i3}) \tag{3}$$

In this example, the sparse vector $y = [y_1, y_2, y_3] = [0, 0, 1]$ is multiplied by the logarithm of the transpose of the dense vector $\hat{y}$, $\hat{y}^T = [\hat{y}_1, \hat{y}_2, \hat{y}_3]^T$. As the three classes are mutually exclusive, only one element of the vector y_i has a non-zero value. Therefore, the result of the equation is simply represented by the last line of Equation 3.

The neural network selects the weights that minimize the loss function for each i-th document; this implies minimizing the negative log-likelihood of the true j-th class for training the i-th observation, $Loss(y_i, \hat{y}_i) = -ln(\hat{y}_{ij})$, which can be interpreted as performing maximum likelihood estimation to estimate the parameters (see Karpathy, 2015).

4 Data

In order to perform text classification and assess informative power, publicly available financial news articles related to S&P 500 Index were scraped from Reuters.com.

Web scraping was done using *Rvest* and *Rselenium*, two open source packages available in *R*. As Reuters web pages are dynamic, a dynamic approach was needed to retrieve the HTML code. Using Rselenium, we simulated browser activity to load web pages that contained older news articles and with Rvest we extracted the HTML code that included article headlines and links to those articles. Then, using those links, we extracted the remaining information.

Electronic copy available at: https://ssrn.com/abstract=3937901

Scraping techniques allowed us to download 20,728 news articles that appeared from March 19 to November 7, 2019, even though only 3276 was used for sentiment analysis, as explained in the next section. Article headlines and bodies were concatenated to obtain a unique text source for each article. As an additional step, we converted the time each piece of news was published to Eastern Time.

As news articles scraped from Reuters are not labeled, we introduced certain automatic labeling techniques, that are described below: a stock index returns-based approach and a dictionary-based approach.

4.1 Stock index returns approach

The stock index returns approach classifies news based on the subsequent returns as reflected in the S&P 500 Index.First, we have downloaded intraday S&P 500 Index prices from Bloomberg Terminal, choosing 10-minute and 20-minute time intervals. Then, we computed log-returns using the closing price of each time interval t, except for the first interval of each day, from 9:30 to 9:40, where log-returns were computed using the market opening price and the closing price of that time window.

All news articles were classified into three categories on the basis of the return recorded in the time interval when they were published. Articles published during time intervals that registered a "positive return", higher than the 55th percentile, were labeled *positive*. Those included in intervals that scored "negative", meaning returns below the 45th percentile, were classified as *negative*. For returns between the 45th and 55th percentile, articles were labeled as *neutral*. Furthermore, we applied a lagged labeling technique, meaning each article was labeled following index returns over the next time window. The choice of using three classes with these two percentile cutoffs derives from the attempt to capture only relevant price variations that are not due to common price fluctuations. In addition, these percentile levels allow positive and negative news to be almost relatively equally labeled in the data set, with

Electronic copy available at: https://ssrn.com/abstract=3937901

neutral news much less frequent.[4]

Clearly, a labeling approach based on fixed time intervals has the drawback that articles published at the beginning of each time window have more time to impact stock index returns, meaning they belong to that specific time interval and are labeled on the basis of price movements during that time interval, while articles published later in the same time window are labeled with price movements of that time window even if they only partly impact them.

In the 10-minute case, we analyzed is 3397 articles (3376 when considering lagged labeling), representing news published between 9:30 AM and 4:00 PM. In the 20-minute case, 3276 news articles were available for sentiment analysis (3198 when considering lagged labeling); they were published between 9:40 AM and 4:00 PM.

This automatic labeling produced the results shown in Table 1, where the number of negative items labeled on the basis of S&P 500 Index 10-minute (20-minute) returns is 1391 (1516), the number of neutral items is 322 (348), and the number of positive items is 1559 (1537). As shown in the table, similar results are obtained with the lagged labeled technique. With all classification methods, the number of negative news items is close to that of items. The number of neutral items is dramatically lower. Using 35th and 65th percentiles leads in many cases to a more uniform classification, but the overall results are unsatisfactory. The fact that the number of negative and positive labels are close to each other appears to be a good starting point to accurately train the network, as it ensures similar levels in learnability of the two classes.

4.2 Dictionary approach

In this section, we present our second labeling approach, which is based on dictionaries. LM (Loughran and McDonald, 2011) is one of several finance-specific dictionaries. They analyzed 10-K documents filed with the SEC between 1994 and 2008 and selected, among

[4]Before choosing these percentiles, we did some preliminary attempts with 35th and 65th, but the result was not satisfactory.

Electronic copy available at: https://ssrn.com/abstract=3937901

Classification Method	Negative	Neutral	Positive
S&P 500 10 min	1516	322	1559
S&P 500 10 min lag	1527	336	1513
S&P 500 20 min	1391	348	1537
S&P 500 20 min lag	1439	354	1405

Table 1: Automatic labeling of Reuters news.

words that occur in at least 5% of the documents, new terms to improve the GI.

The HE (Henry, 2008) uses words taken from the corpus of earnings press releases. To account for the context of each word, Henry assesses its directional meaning based on its relationships with close terms in the same sentence. Specifically, she started with an initial list of words and examined the three words preceding and following each word in the documents. She evaluated whether the word is positive and negative on the basis of its relationship with each of these close terms. Only words labeled either positive or negative in 80% of occurrences appear in HE.

As explained by Medhat et al. (2014), dictionary-based sentiment analysis consists of defining a list of words that are associated with a given sentiment state. This list of words is labeled on the basis of their meanings and expanded using synonyms and antonyms. A well-known publicly available list of opinion words is what we call the CI: *Psychological Harvard IV-4 Dictionary*, which is part of the General Inquirer (Harvard University, 1960) software for text analysis and classifying words as positive or negative. However, Loughran and McDonald (2011) note that the GI's poor performance with positive words is probably due to their frequent negation and that a negative list would be preferable.

We used all three dictionaries (GI, LM, and HE) to label articles based on the words they contain. These three dictionaries provide lists of positive and negative words that can be used to assess whether a document is positive or negative using a simple formula. Specifically, each article was labeled on the basis of how many positive and negative words it contains according to each of the three dictionaries.

Electronic copy available at: https://ssrn.com/abstract=3937901

Equation 4 shows the formula used to define each label. A "positive" class was attributed to the i-th article if the difference between the number of positive and negative words it contains is greater than zero. If this difference is negative, the assigned class is "negative." If this formula returns zero, then the class of the i-th article is "neutral."

$$class_i = p\_words_i - n\_words_i \tag{4}$$

where p_words_i and n_words_i indicate the number of positive and negative words, respectively.

As an example, we provide in Table 2 a sample of positive and negative words included in the three dictionaries.

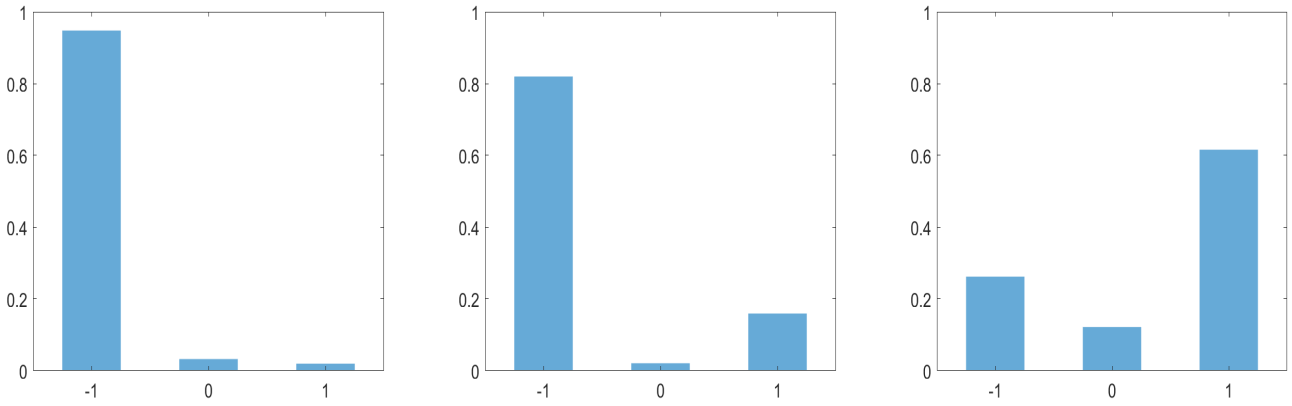
HE		GI		LM	
Positive	Negative	Positive	Negative	Positive	Negative
above	below	abide	abandon	able	abandon
accomplish	challenge	ability	abandonment	abundance	abandoned
accomplished	challenged	able	abate	abundant	abandoning
accomplishes	challenges	abound	abdicate	acclaimed	abandonment
accomplishing	challenging	absolve	abhor	accomplish	abandonments
accomplishment	decline	absorbent	abject	accomplished	abandons

Table 2: Sample of words contained in HE, LM, and GI. *Source: CRAN. Package 'SentimentAnalysis'.*

This classification produced similar results when comparing LM and GI, as presented in Figure 3. In both cases, the number of negative items is higher than the number of neutral and positive items. For the LM dictionary, the number of positive items is almost negligible. For the HE dictionary, the number of positive articles is greater than the number of negative articles. HE produced a more uniform classification, even though the number of positive items is more than twice the number of negative items. Given that the three dictionaries differ in terms of class distribution, a direct comparison among them provides interesting results in the sentiment analyses. Once all items were labeled according to each of the three dictionaries, we implemented and trained the LSTM neural network to predict the class of

Electronic copy available at: https://ssrn.com/abstract=3937901

each article.



(a) *LM.* **(b)** *GI.* **(c)** *HE.*

Figure 3: Histogram representation of article classes based on LM, GI, and HE.

5 Results

In this section, we present the results of the classification on the basis of the labeling technique used. The news articles are labeled and pre-processed and represent the inputs to the neural network for the training phase. Figure 4 shows the word cloud representation of the most frequent words in the text before and after pre-processing. Note that some of the words that appear in Figure 4a are excluded by Figure 4b. For example, words like *Reuters* or $\_x000D\_$ (which is an ASCII non-printable character) appear frequently in the text but are unlikely to have strong semantic relevance for an article.

5.1 Stock index returns approach

In the case of the stock index returns approach, considering all the possible combinations among 10- and 20-minute time intervals, non-lagged and lagged labeling and so on, 720 different settings were used to train the network. Preliminary attempts showed that no combination of parameters performs better than others. Therefore, all combinations are

Electronic copy available at: https://ssrn.com/abstract=3937901

(a) *Raw data.*

(b) *Pre-processed data*

Figure 4: Word cloud representation of raw training data and pre-processed training data.

Electronic copy available at: https://ssrn.com/abstract=3937901

retained, and those that showed the best results selected.

Recall that 10- and 20-minute are used, while news articles are classified on the basis of S&P 500 returns in the relevant time window or a lagged time window.

The *embedding dimension* indicates the length of the d-dimensional dense vector used by the embedding layer. In this respect, we select dimensions of 50, 100, and 150. For the *number of hidden units* used by the LSTM blocks, we select 50, 100, 150, and 200. The *gradient threshold* represents the value the gradient is not allowed to exceed. To avoid gradient diverging, which occurs when the Euclidean norm of the gradient becomes greater than this threshold, the algorithm clips it and returns it to the threshold. The values we checked as gradient thresholds are 1, 0.1, and 0.01.

The *initial learning rate* represents the step size used by the learning algorithm at each iteration to move toward the minimum of the loss function. First, we tried different values of the initial learning rate; we then decided to use only 0.0001, as we noticed that with larger values, the algorithm tends to run quickly and leads to unsatisfactory results.

The maximum number of *epochs* is set to 150. However, the *validation patience* represents an early stopping technique that halts the algorithm before approaching the maximum number of epochs and avoids network over-training. This technique establishes the number of times the loss function computed on the validation set can be larger than the previous smallest loss. It implies that the loss function on the validation set must decrease towards zero and if it increases too much compared to a prefixed threshold. Among the different combinations, we used 20, 30, 40, 50, and 60 as values for validation patience.

In addition, we set the following two parameters to the respective default valued provided by MATLAB. The *mini-batch size* is set to 128 and represents the number of training instances that are analyzed by the algorithm before updating the parameters. The *validation frequency* is set to 50 and indicates the number of instances used by the network to validate the training parameters on the validation set. This implies that the neural network assesses the accuracy of the model once every 50 instances.

Electronic copy available at: https://ssrn.com/abstract=3937901

Table 3 presents the results for the four models according to time window and the presence of the lagging method. The latter refers to models where classes are defined on the basis of current (not lagged) or subsequent (lagged) index returns. In order to extract the sentiment variables for each of the eight classes, we kept the 10 best results in terms of accuracy. In the table, we report only the best in class for each of the four models. In all cases, accuracy is very close to 50%, in line with many other similar studies.[5]

Best Accuracy	Time Window	Lag
0.50177	10 min	No
0.48815	10 min	Yes
0.47131	20 min	No
0.44556	20 min	Yes

Table 3: Results of LSTM neural network applied on articles labeled with the log returns of the S&P 500 Index. The percentiles used to define positive and negative classes are 45th and 55th, respectively.

5.2 Dictionary approach

To train the network with the dictionary-based approach, we tested different parameter settings. After a preliminary investigation, our network was built using 200 hidden units of LSTM blocks and 100 epochs. The other values are 100 for the embedding dimension, 0.1 for the gradient threshold, 0.001 for the initial learning rate, and 10 for the validation patience.

Table 4 shows the results of the training process for the three dictionary-based labeling techniques. The columns represent the accuracy of the model, the dictionary used for labeling, the embedding dimension, the number of units in each LSTM block, the gradient threshold, the number of epochs, the initial learning rate, and the validation patience used as early stopping technique.

The best result is obtained for LM, with an accuracy of 0.92462 on the testing set. This result is not so surprising when looking at Figure 3. Indeed, LM contains the highest relative

[5]Full results are available upon request to the authors.

Electronic copy available at: https://ssrn.com/abstract=3937901

Accuracy	Label
0.92462	LM dictionary
0.79034	GI dictionary
0.65135	HE dictionary

Table 4: Results of LSTM neural network applied to articles labeled with dictionary-based techniques.

number of negative words. On the one hand, positive words lead to poor performances. As noted in Loughran and McDonald (2011), the reason is that positive words are often followed or preceded by negations, which clearly change the meaning of the statement. On the other hand, almost all articles were classified as negative and, so the results may be biased. When it comes to GI and HE, accuracy decreases to 0.79034 and 0.65135, respectively, though both of them better than the random guess of 0.50.

6 Sentiment variables

In this section, we present the sentiment variables produced on the basis of the presented analysis. These variables are based on the classification of financial news belonging to the testing set. The out-of-sample test includes observations from October 1 to November 7, 2019. In the case of the stock index returns approach, we built eight types of sentiment variables, corresponding to the eight classification models presented in the previous sections. For the dictionary-based approach, there are six sentiment variables that depend solely on the dictionary and time window used.

6.1 Stock index returns approach

Table 5 presents the four sentiment variables built on the basis of the classification methods presented in Table 3. As already mentioned, for each labeling technique we selected the ten most accurate results. Therefore, each of these eight sentiment variables is a synthesis of a list of ten variables.

Electronic copy available at: https://ssrn.com/abstract=3937901

Sentiment variable	Time Window	Lag
S_{sp10}	10 min	no
$S_{sp10lag}$	10 min	yes
S_{sp20}	20 min	no
$S_{sp20lag}$	20 min	yes

Table 5: Sentiment variables based on LSTM neural network applied to articles labeled with S&P 500 Index log returns.

After all financial news items in the testing set were classified by applying the parameters obtained through the training process, the flow of items was divided into time intervals of 10 or 20 minutes based on their publication time; they were then aggregated to build the sentiment variables. Given that each article was assigned a value of 1 if positive, -1 if negative, and 0 if neutral, sentiment variables are defined as the sum of all news items n published during time interval t, as described by Equation 5:

$$S_t = \sum_i news_{t,i}, \tag{5}$$

where

$$news_{t,i} = \begin{cases} +1, & \text{if the news article is classified as positive.} \\ 0, & \text{if the news article is classified as neutral.} \\ -1, & \text{if the news article is classified as negative.} \end{cases}$$

The sentiment variables S_{sp10} and S_{sp20} are obtained by classifying news articles using the model that associates them with 10-minute and 20-minute log returns computed on the S&P 500 Index, respectively. Similarly, $S_{sp10lag}$ and $S_{sp20lag}$ are the sentiments obtained with subsequent 10-minute and 20-minute log returns, respectively, meaning with returns registered within the following 10-minute and 20-minute time interval.

As an example, Figure 5 shows the time series for one of the sentiment variables. Specifically, it refers to the most accurate sentiment variable built using the model that tracks the S&P 500 Index returns with 20-minute time intervals, S_{sp20}.

Electronic copy available at: https://ssrn.com/abstract=3937901

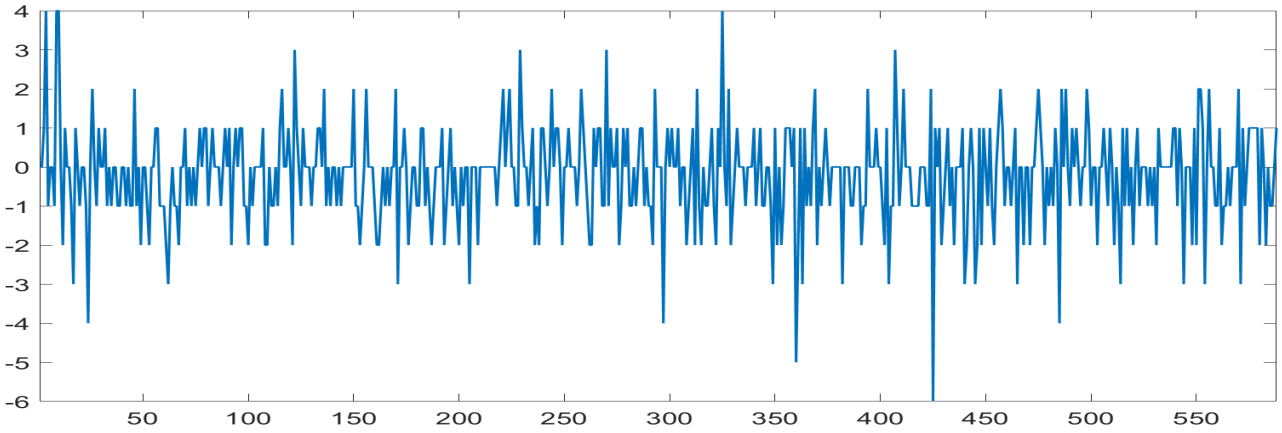


Figure 5: Time series of S_{sp20} which has the highest accuracy.

6.2 Dictionary approach

We built sentiment variables to test the ability of dictionary-based classifications to explain the S&P 500 Index performances. These sentiment variables are built on the basis of news published within 10-minute and 20-minute time intervals, analogously to the variables presented in the previous section, and are based entirely on data related to the Testing set. Table 6 presents six sentiment variables built aggregating all articles that were published within each interval. The aggregation method is the one presented in Equation 5.

The variables are S_{LM10}, which is based on LM and 10-minute time intervals, S_{GI10}, based on GI with 10-minute time intervals and S_{HE10}, which is based on HE with 10-minute time intervals. S_{LM20}, S_{GI20} and S_{HE20}, analogously, are based on 20-minute time intervals.

Sentiment Variable	Dictionary	Time Window
S_{LM10}	LM	10 min
S_{GI10}	GI	10 min
S_{HE10}	HE	10 min
S_{LM20}	LM	20 min
S_{GI20}	GI	20 min
S_{HE20}	HE	20 min

Table 6: Sentiment variables based on LSTM neural network applied to articles labeled with HE, GI, and LM.

Electronic copy available at: https://ssrn.com/abstract=3937901

Finally, Figure 6 shows the time series for the three sentiment variables based on 20-minute time intervals, S_{LM20}, S_{GI20} and S_{HE20}.

7 The EGARCH model and sentiment variables

In this section, we test whether the obtained out-of-sample news-based sentiment variables are informative in explaining the returns and volatility of the S&P 500 Index. We model the conditional volatility of stock index returns using an EGARCH model by including the sentiment variables into the mean and variance equations. The EGARCH model allows us to model conditional heteroscedasticity by introducing asymmetry between negative and positive shocks in volatility. Clearly, the aim is to analyze whether the obtained sentiment indicators represent meaningful predictors of market returns.

To account for skewness and fat tails in the log returns distribution, we adopt an EGARCH(1,1) specification using a skewed Student's t-distribution for innovations.

Equations 6 and 7 describe the EGARCH(1,1) model, where x_t is the index log-return, h_t is the volatility of the index log-return and S_t is the sentiment variable at time t:

$$x_t = \mu + \lambda_m S_t + \varepsilon_t, \quad \varepsilon_t \equiv h^{\frac{1}{2}} z_t, \quad z_t \sim \text{Skew Student}'\text{s } t \tag{6}$$

and

$$\log h_t = \omega + \alpha z_{t-1} + \beta \log h_{t-1} + \gamma(|z_{t-1}| - E(|z_{t-1}|)) + \lambda_v S_t. \tag{7}$$

The EGARCH parameters are defined as follows: μ (the constant in the mean equation), ω (the constant in the variance equation), α (the ARCH coefficient), β (the GARCH coefficient) and γ (the leverage coefficient). Finally, λ_m and λ_v are the coefficients for the sentiment variable S_t in the mean and variance equations, respectively.

Below, we present the results for the sentiment variables built using the stock index returns approach and the dictionary approach. First, we include only the results obtained

Electronic copy available at: https://ssrn.com/abstract=3937901

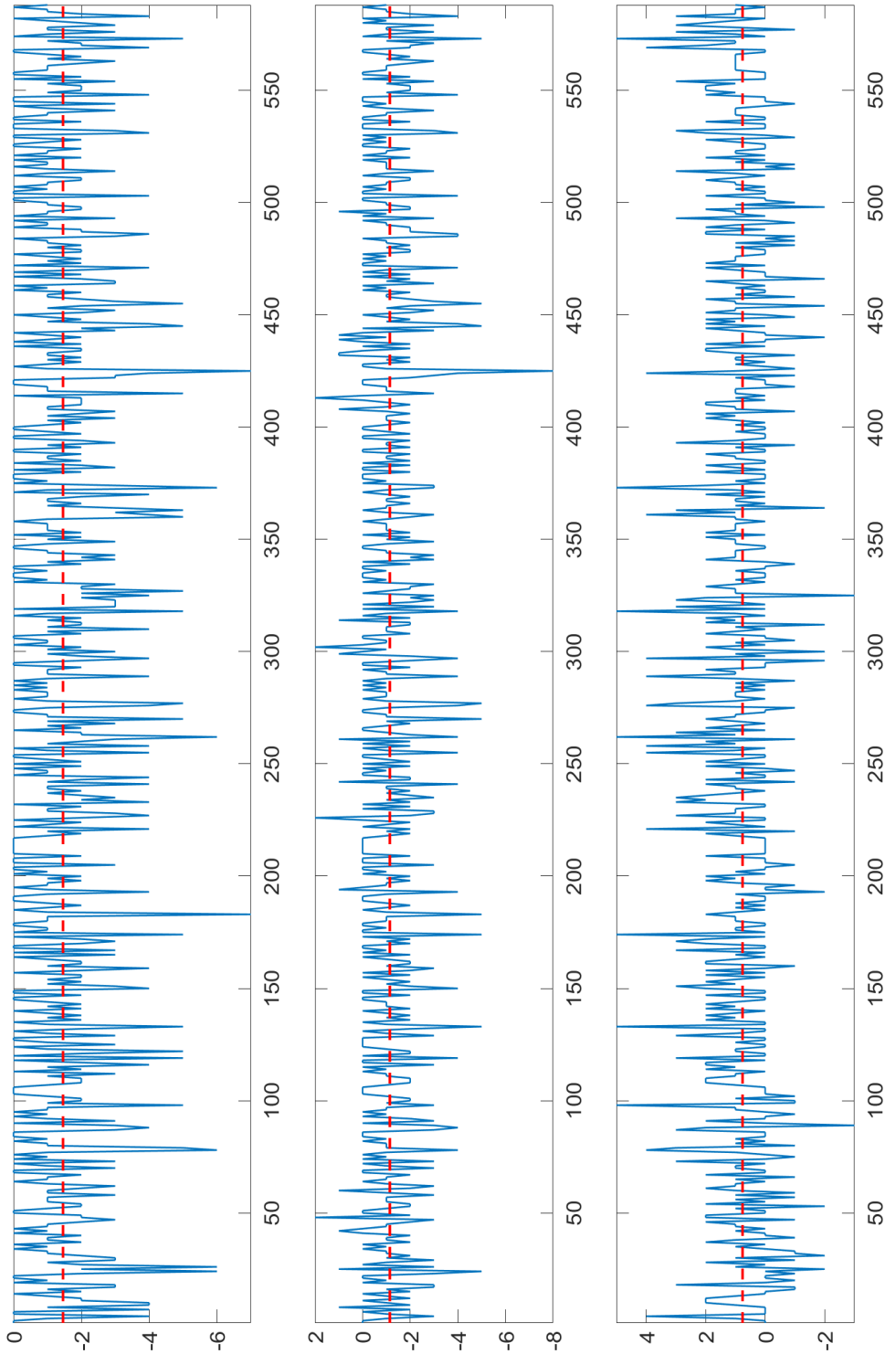


Figure 6: Time series of the sentiments using the three dictionaries, S_{LM20}, S_{GI20} and S_{HE20}. The red dashed line represents the average of the sentiment over the period.

Electronic copy available at: https://ssrn.com/abstract=3937901

using 20-minute time intervals, since the 10-minute time intervals did not produce any significant results.[6] This finding provides an interesting insight into the time propagation of public news and thus, the reaction of market participants. No impact of public news release was not detected in the first 10 minutes, which suggests that the information is discounted by the market on a larger time window. We stress here that this is an aggregated reaction since we are analyzing the response at the market level and not at the single-firm level.

Second, we show that the sentiment based on the stock index returns approach is particularly sensitive to the initial settings and thus can lead to different results in the EGARCH estimates. Conversely, the sentiment based on the dictionary approach provides quite similar results for all the three dictionaries. Figure 7 shows the out-of-sample time series of the S&P 500 Index intraday log returns from October 1 to November 7, 2019, using 20-minute time intervals (560 observations).

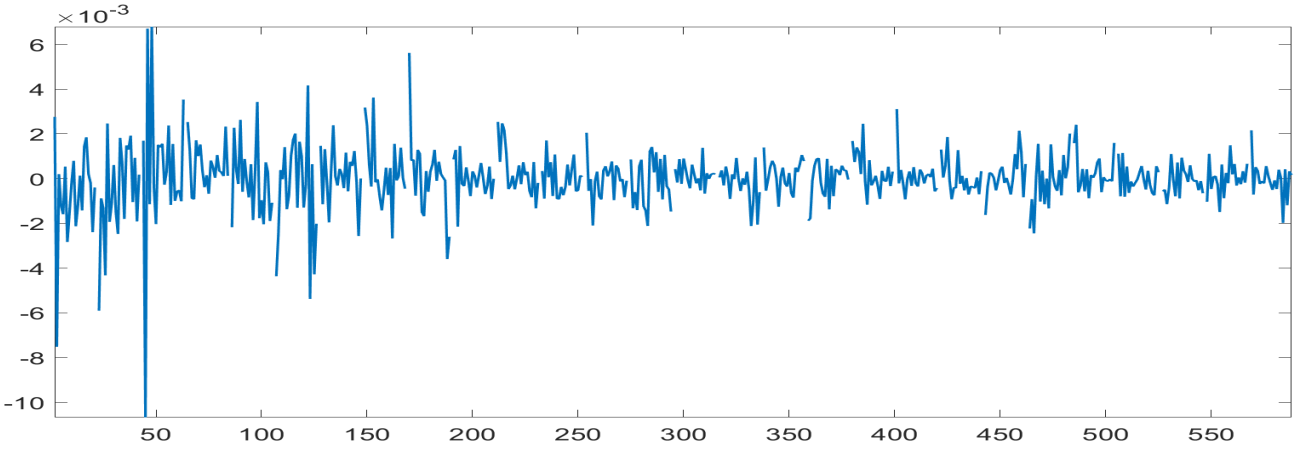


Figure 7: The plot of S&P 500 Index log returns with 20-minute time intervals from October 1 to November 7, 2019. *Source: Data from Bloomberg.*

7.1 Stock index returns approach

In this section, we present the results obtained by using as inputs in the mean and variance equations of the EGARCH(1,1) model fitted on the S&P 500 Index log returns, as presented

[6]Full results are available upon request to the authors.

Electronic copy available at: https://ssrn.com/abstract=3937901

in Table 5. We recall that each of these sentiment variables represents a list of ten variables, built using the ten most accurate classification results. We consider S_{sp20}, which is based on the classification that links news published within a given 20-minute time interval to index returns within the subsequent time interval, and $S_{sp20lag}$. Our findings show that the sentiment proxies built with the stock index returns approach have meaningful explanatory capabilities in four cases. Table 7 shows the estimates for the EGARCH(1,1) model using the ten versions of the sentiment variable S_{sp20} on the returns of the S&P 500 Index. Four of ten regressions have λ_m statistically different from zero (columns 2, 3, 4 and 9 in the table). In these cases, the impact on returns is relatively small and close to zero (i.e, lower than 10^{-4}) and is negative in two cases and positive in one. The coefficient in the volatility equation, λ_v, is negative and significant in three cases (columns 2, 3, and 9) and exhibits a larger magnitude with respect to the mean equation. This negative relationship implies that a higher value in the sentiment is associated with a lowering of volatility. Given that the sentiment builds on the returns, news items are mapped on positive and negative returns, so an increase in the sentiment underlines a positive news item that is related to positive returns. Table 8 includes the lagged versions of the sentiments $S_{sp20lag}$ as defined in Section 3. In the mean equation, λ_m is significant seven of ten cases (columns 1, 4, 5, 6, 7, 9, and 10). As for S_{sp20}, the magnitude is close to zero and is positive in three of seven cases.

Despite providing some interesting results, we conclude that the sentiment based on stock index returns does not represent a reliable approach for distinguishing between positive and negative news. In our view, this is due to the mapping process between news and financial returns. In the chosen methodology, the classification among positive, negative, and neutral news is performed through a direct match between the time release of each news items and the market price movements recorded in a subsequent time window. Clearly, price movements in the stock index are the result of several market forces and factors that could impact the accuracy of the classifier through spurious associations in the learning process.

Electronic copy available at: https://ssrn.com/abstract=3937901

Electronic copy available at: https://ssrn.com/abstract=3937901

	(1)		(2)		(3)		(4)		(5)		(6)		(7)		(8)		(9)		(10)	
μ	-0.00004		0.00000	***	-0.00003	***	0.00000		-0.00003		-0.00002		-0.00002		-0.00002		-0.00010	***	-0.00002	
	(0.00003)		(0.00000)		(0.00000)		(0.00003)		(0.00003)		(0.00004)		(0.00003)		(0.00003)		(0.00000)		(0.00003)	
λ_m	0.00003		-0.00002	***	-0.00000	***	0.00007	**	0.00004		-0.00000		0.00002		0.00002		0.00002	***	0.00006	
	(0.00003)		(0.00000)		(0.00000)		(0.00003)		(0.00003)		(0.00003)		(0.00003)		(0.00002)		(0.00000)		(0.00005)	
ω	-0.07631	***	-0.02435	***	-0.04894	***	-0.09188	***	-0.07626	***	-0.07924	***	-0.05132	***	-0.07034	***	-0.04163	***	-0.03857	***
	(0.00589)		(0.00000)		(0.00000)		(0.00488)		(0.00815)		(0.01175)		(0.00444)		(0.00349)		(0.00000)		(0.00279)	
α	-0.08292	***	-0.10543	***	-0.10124	***	-0.08049	***	-0.07884	***	-0.07616	***	-0.08171	***	-0.08012	***	-0.09992	***	-0.08111	***
	(0.02178)		(0.00001)		(0.00000)		(0.01641)		(0.01878)		(0.02443)		(0.02210)		(0.01145)		(0.00126)		(0.01858)	
β	0.99407	***	0.99729	***	0.99528	***	0.99287	***	0.99439	***	0.99448	***	0.99604	***	0.99470	***	0.99540	***	0.99721	***
	(0.00009)		(0.00005)		(0.00000)		(0.00006)		(0.00000)		(0.00009)		(0.00008)		(0.00010)		(0.00000)		(0.00018)	
γ	0.02824		-0.03273	***	-0.03235	***	0.03350	*	0.03846	***	0.03905		0.02653		0.03155		-0.01889	***	0.02583	
	(0.03472)		(0.00000)		(0.00012)		(0.01787)		(0.01088)		(0.03612)		(0.03510)		(0.03606)		(0.00056)		(0.03357)	
λ_v	-0.00874		-0.01592	***	-0.04364	***	0.02063		0.00115		0.00774		-0.01718		-0.00547		-0.03533	***	-0.02769	
	(0.01429)		(0.00000)		(0.00001)		(0.01693)		(0.01909)		(0.01728)		(0.01663)		(0.00962)		(0.00000)		(0.01802)	
Skew	0.95817	***	0.95619	***	0.95487	***	0.96444	***	0.94786	***	0.95495	***	0.95403	***	0.95424	***	0.90540	***	0.94840	***
	(0.05362)		(0.04449)		(0.00939)		(0.05329)		(0.05343)		(0.05438)		(0.05245)		(0.05207)		(0.04279)		(0.05219)	
Shape	3.81600	***	4.03436	***	4.35364	***	3.86145	***	3.79134	***	3.85639	***	3.86322	***	3.84175	***	3.89680	***	3.86577	***
	(0.62638)		(0.00093)		(0.00272)		(0.62194)		(0.61651)		(0.63172)		(0.64063)		(0.58708)		(0.04300)		(0.64898)	

Table 7: S_{sp20} as an exogenous variable in the mean and variance equations of the EGARCH(1,1) model with skewed Student's conditional t-distribution fitted on S&P 500 Index intraday log-returns with 20-minute time intervals.
Statistical significance at the 1% (***), 5% (**), 10% (*) levels.

	(1)		(2)		(3)		(4)		(5)		(6)		(7)		(8)		(9)		(10)	
μ	-0.00008	***	-0.00002		-0.00000		-0.00004	***	-0.00004	***	-0.00005	***	-0.00005	***	-0.00005	***	-0.00007	***	-0.00005	***
	(0.00000)		(0.00005)		(0.00004)		(0.00000)		(0.00000)		(0.00000)		(0.00000)		(0.00000)		(0.00000)		(0.00000)	
λ_m	-0.00001	***	0.00001		0.00002		-0.00000	***	0.00003	***	0.00000	***	-0.00000	***	0.00002		0.00003	***	0.00000	***
	(0.00000)		(0.00003)		(0.00002)		(0.00000)		(0.00000)		(0.00000)		(0.00000)		(0.00002)		(0.00000)		(0.00000)	
ω	-0.05884	***	-0.07526	***	-1.14235	***	-0.05164	***	-0.03120	***	-0.04635	***	-0.06704	***	-0.07157	***	-0.08434	***	-0.05189	***
	(0.00001)		(0.00427)		(0.31462)		(0.00000)		(0.00000)		(0.00002)		(0.00000)		(0.00000)		(0.00000)		(0.00000)	
α	-0.09741	***	-0.08161	***	-0.06853		-0.10978	***	-0.09938	***	-0.10154	***	-0.12273	***	-0.11465	***	-0.10967	***	-0.08163	***
	(0.00000)		(0.02799)		(0.05007)		(0.01041)		(0.00004)		(0.00052)		(0.00119)		(0.00005)		(0.00092)		(0.00031)	
β	0.99502	***	0.99427	***	0.91252	***	0.99588	***	0.99717	***	0.99766	***	0.99503	***	0.99454	***	0.99303	***	0.99579	***
	(0.00002)		(0.00014)		(0.02357)		(0.00004)		(0.00001)		(0.00004)		(0.00000)		(0.00000)		(0.00008)		(0.00008)	
γ	-0.03226	***	0.02252		0.35537	***	-0.02458	***	-0.02741	***	-0.02202	***	-0.02441	***	-0.02754	***	-0.03398	***	-0.02688	***
	(0.00005)		(0.05030)		(0.08959)		(0.00436)		(0.00018)		(0.00015)		(0.00050)		(0.00093)		(0.00045)		(0.00030)	
λ_v	-0.01741	***	-0.00771		-0.08553		-0.01361	***	-0.03596	***	0.02345	***	0.02903	***	0.01000	***	-0.03351	***	-0.03911	***
	(0.00001)		(0.01874)		(0.05399)		(0.00366)		(0.00005)		(0.00000)		(0.00063)		(0.00328)		(0.00009)		(0.00051)	
skew	0.88594	***	0.94944	***	0.98816	***	0.93553	***	0.92821	***	0.93301	***	0.92700	***	0.93750	***	0.90836	***	0.92973	***
	(0.01024)		(0.06107)		(0.05686)		(0.04569)		(0.04483)		(0.04679)		(0.04480)		(0.04524)		(0.04522)		(0.04451)	
shape	4.16569	***	3.78066	***	3.35728	***	3.94763	***	3.71794	***	3.99186	***	3.87340	***	3.82262	***	4.13009	***	3.79145	***
	(0.00105)		(0.64183)		(0.47949)		(0.03290)		(0.00294)		(0.06311)		(0.07607)		(0.01124)		(0.01093)		(0.00897)	

Table 8: $S_{sp20lag}$ as lagged exogenous variable in the mean and variance equations of the EGARCH(1,1) model with skewed Student's conditional t-distribution fitted on S&P 500 Index intraday log-returns with 20-minute time intervals.
Statistical significance at the 1% (***), 5% (**), 10% (*) levels.

Electronic copy available at: https://ssrn.com/abstract=3937901

7.2 Dictionary approach

Below, we present the analysis of the three sentiment variables based on positive and negative words in GI, LM, and HE. As for the sentiment variables built under the stock index returns approach, the sentiment variables presented in Table 6 are tested to prove their explanatory power. In this case, we also found that significant results in the 20-minute time intervals and not in the 10-minute time frames.

Table 9 shows the three estimated regressions where S_{LM20}, S_{GI20} and S_{HE20} are introduced as exogenous variables in the mean and variance equations of the EGARCH(1,1) fitted on the S&P 500 Index returns as included in Figure 7. Analogously, Table 10 includes the estimates for the sentiments lagged by one period $S(1)_{LM20}$, $S(1)_{GI20}$, and $S(1)_{HE20}$.

In both tables, all coefficients are always significantly different from zero at the 1% significance level. Table 9 shows that the sign for λ_m, is positive for all three dictionaries. As expected, this implies that sentiment is positively related to market returns. Conversely, λ_v is negative in all three dictionaries, indicating that higher sentiment values associated with a lowering of volatility.

When considering the lagged version of the sentiment, as described in Table 9, λ_m remains positive only for S_{LM20}, which is the sentiment variable built under the classification based on LM. For GI and HE, the sign becomes negative, indicating that a change in the relationship when the sentiment is lagged that could reflect a reversion toward the mean. Conversely, the estimated coefficient of the sentiment in the variance equation, λ_v, is negative, confirming the previous findings for S_{LM20}, S_{GI20}, and S_{HE20}. Overall, the results from the dictionary approach are clearly more stable since they provide similar results with all three dictionaries.

8 Conclusion

This paper contributes to the financial literature by providing some alternative proxies to estimate news sentiment, which is not directly observable and measurable. We provide evi-

Electronic copy available at: https://ssrn.com/abstract=3937901

	S_{LM20}		S_{GI20}		S_{HE20}	
μ	-0.00007	***	-0.00004	***	-0.00004	***
	(0.00000)		(0.00000)		(0.00000)	
λ_m	0.00000	***	0.00002	***	0.00002	***
	(0.00000)		(0.00000)		(0.00000)	
ω	-0.06525	***	-0.05202	***	-0.03938	***
	(0.00000)		(0.00000)		(0.00000)	
α	-0.09822	***	-0.11002	***	-0.10142	***
	(0.00213)		(0.00013)		(0.00003)	
β	0.99702	***	0.99688	***	0.99607	***
	(0.00003)		(0.00002)		(0.00008)	
γ	-0.01549	***	-0.02554	***	-0.03491	***
	(0.00012)		(0.00001)		(0.00000)	
λ_v	-0.03881	***	-0.02676	***	-0.04122	***
	(0.00000)		(0.00000)		(0.00001)	
Skew	0.91906	***	0.93127	***	0.95008	***
	(0.04482)		(0.06511)		(0.04773)	
Shape	4.16196	***	4.35039	***	4.01090	***
	(0.02944)		(0.02753)		(0.00147)	

Table 9: S_{LM20}, S_{GI20}, and S_{HE20} as exogenous variables in the mean and variance equations of EGARCH(1,1) model with skewed Student's conditional t-distribution fitted on S&P 500 Index intraday log returns with 20-minute time intervals.
Statistical significance at the 1% (***), 5% (**), 10% (*) levels.

Electronic copy available at: https://ssrn.com/abstract=3937901

	$S(1)_{LM20}$		$S(1)_{GI20}$		$S(1)_{HE20}$	
μ	-0.00004	***	-0.00008	***	-0.00004	***
	(0.00000)		(0.00000)		(0.00000)	
λ_m	0.00005	***	-0.00002	***	-0.00001	***
	(0.00000)		(0.00000)		(0.00000)	
ω	-0.06916	***	-0.06338	***	-0.07427	***
	(0.00000)		(0.00000)		(0.00000)	
α	-0.10050	***	-0.11082	***	-0.13714	***
	(0.00007)		(0.00019)		(0.00006)	
β	0.99721	***	0.99590	***	0.99379	***
	(0.00008)		(0.00000)		(0.00003)	
γ	-0.02730	***	-0.02281	***	-0.04008	***
	(0.00013)		(0.00014)		(0.00008)	
λ_v	-0.04682	***	-0.02376	***	-0.02212	***
	(0.00000)		(0.00005)		(0.00003)	
Skew	0.93598	***	0.92352	***	0.94147	***
	(0.07001)		(0.04582)		(0.01926)	
Shape	4.01275	***	3.96098	***	3.38238	***
	(0.00291)		(0.08599)		(0.00151)	

Table 10: $S(1)_{LM20}$, $S(1)_{GI20}$, and $S(1)_{HE20}$ as exogenous lagged variables in the mean and variance equations of the EGARCH(1,1) model with skewed Student's conditional t-distribution fitted on S&P 500 Index intraday log returns with 20-minute time intervals. *Statistical significance at the 1% (***), 5% (**), 10% (*) levels.*

Electronic copy available at: https://ssrn.com/abstract=3937901

dence that even publicly available news is informative and can explain short-lived movements in a broad index such as the S&P 500. Indeed, despite the relatively short time frame of our sample, our sentiment variables prove to be effective in explaining S&P 500 Index log returns and volatility, even when introduced as lagged variables. We also contribute to the literature by providing an alternative use of three widely used dictionaries (LM, HE, and GI) showing that the explanatory power of the sentiment is invariant to their use. Our findings show that the dictionary-based approach provides more reliable results with respect to the sentiment based on stock index returns approach. Finally, we show that the predictive power on the stock index is found in the 20-minute interval after a news article becomes publicly available.

Future research might focus on improving the accuracy of the classification models, using different ML tools and pre-processing techniques (such as the GloVe embeddings by Standford), improving the significance of all sentiment variables, and explaining the sign of the coefficients. To improve results, instead of considering fixed time intervals, articles might be labeled using index log returns computed over the 20-minute time interval that starts from the exact moment the article is published. Improvements might also come from the use of different financial news sources, which can be finance-specific, generic, or even news aggregators. With respect to dictionary-based automatic labeling, more complex equations than in Equation 4 can be used to improve classification accuracy. Even if the dictionaries used in this analysis are widely used in financial literature, other dictionaries can be used. Focusing on a less broad index or even an individual sector or stock might also be an interesting research path.

Electronic copy available at: https://ssrn.com/abstract=3937901

References

Atkins, A., Niranjan, M., Gerding, E., 2018. Financial news predicts stock market volatility better than close price. The Journal of Finance and Data Science 4, 120–137.

Caporin, M., Poli, F., 2017. Building news measures from textual data and an application to volatility forecasting. Econometrics 5, 1–46.

Costola, M., Iacopini, M., Santagiustina, C. R. M. A., 2020. Google search volumes and the financial markets during the COVID-19 outbreak. Finance Research Letters .

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

Groß-Klußmann, A., Hautsch, N., 2011. When machines read the news: Using automated text analytics to quantify high frequency news-implied market reactions. Journal of Empirical Finance 18, 321–340.

Harvard University, 1960. General Inquirer. http://www.wjh.harvard.edu/~inquirer/.

Heidenreich, H., 2018. Stemming? Lemmatization? What? https://towardsdatascience.com/stemming-lemmatization-what-ba782b7c0bd8/.

Henry, E., 2008. Are investors influenced by how earnings press releases are written? Journal of Business Communication 45, 363–407.

Iacopini, M., Santagiustina, C. R., 2021. Filtering the intensity of public concern from social media count data with jumps. Journal of the Royal Statistical Society: Series A (Statistics in Society) .

Jiang, G. J., Tian, Y. S., 2005. The model-free implied volatility and its information content. The Review of Financial Studies 18, 1305–1342.

Karpathy, A., 2015. CS231n Convolutional neural networks for visual recognition. Linear classification: Support Vector Machine, Softmax classifier. http://cs231n.github.io/linear-classify/#softmax.

Li, X., Xie, H., Chen, L., Wang, J., Deng, X., 2014. News impact on stock price return via sentiment analysis. Knowledge-Based Systems 69, 14–23.

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

Electronic copy available at: https://ssrn.com/abstract=3937901

Loughran, T., McDonald, B., 2015. The use of word lists in textual analysis. Journal of Behavioral Finance 16, 1–11.

Mangee, N., 2018. Stock returns and the tone of marketplace information: Does context matter? Journal of Behavioral Finance 19, 396–406.

Medhat, W., Hassan, A., Korashy, H., 2014. Sentiment analysis algorithms and applications: A survey. Ain Shams Engineering Journal 5, 1093–1113.

Nigam, K., Lafferty, J., McCallum, A., 1999. Using maximum entropy for text classification. In: *IJCAI-99 workshop on machine learning for information filtering*, Stockholom, Sweden, vol. 1, pp. 61–67.

Souma, W., Vodenska, I., Aoyama, H., 2019. Enhanced news sentiment analysis using deep learning methods. Journal of Computational Social Science 2, 33–46.

Vicari, M., Gaspari, M., 2020. Analysis of news sentiments using natural language processing and deep learning. Ai & Society pp. 1–7.

Wan, X., Yang, J., Marinov, S., Calliess, J.-P., Zohren, S., Dong, X., 2021. Sentiment correlation in financial news networks and associated market movements. Scientific reports 11, 1–12.

Wang, G., Wang, T., Wang, B., Sambasivan, D., Zhang, Z., Zheng, H., Zhao, B. Y., 2015. Crowds on Wall Street: Extracting value from social investing platforms. In: *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, ACM, pp. 17–30.

Xing, F. Z., Cambria, E., Welsch, R. E., 2018. Natural language based financial forecasting: A survey. Artificial Intelligence Review 50, 49–73.

Yadav, R., Kumar, A. V., Kumar, A., 2019. News-based supervised sentiment analysis for prediction of futures buying behaviour. IIMB Management Review 31, 157–166.

Electronic copy available at: https://ssrn.com/abstract=3937901

A Stock index returns approach using the VIX

In this appendix, we replicate the stock index returns approach using the VIX as the matching variable. The VIX is considered a superior predictor of historical volatility since it is based on option prices that reflect the future expectations of market participants (see, for instance, Jiang and Tian, 2005).

Due to shock asymmetry, volatility is usually higher when the S&P 500 Index returns are negative and might be lower when the S&P 500 Index returns are positive. Following the analysis in the main text, we do not impose any asymmetry in the weighting structure and therefore, process news for the VIX following the same method as for returns. As discussed in the paper, news articles were classified as *positive* in case of log returns higher than the 55th percentile, *negative* in case of log returns lower than the 45th percentile, and neutral otherwise. The results are presented in Table A.1. The period considered and number of articles analyzed were the same as for stock returns. Table A.2 shows the accuracy of the results for the four models according to the time windows and the presence of the lagging method. Finally, Table A.3 shows the four sentiment variables built on the basis of the classification methods presented in Table A.2.

Classification Method	Negative	Neutral	Positive
VIX 20 min	1468	271	1537
VIX 20 min lag	1448	276	1474
VIX 10 min	1516	202	1679
VIX 10 min lag	1524	206	1646

Table A.1: Automatic labeling of Reuters news.

Best Accuracy	Label	Time Window	Lag
0.4823	VIX index	20 min	No
0.48936	VIX index	20 min	Yes
0.48999	VIX index	10 min	No
0.51422	VIX index	10 min	Yes

Table A.2: Results of LSTM neural network applied to articles labeled with the log difference of the VIX Index.

The estimates for the EGARCH model are presented in Tables A.4 and A.5 for S_{vix20} and $S_{vix20lag}$, respectively. In both cases, we found similar evidence as for the sentiment built on the S&P 500. For instance, λ_m is significant and very close to zero in both S_{vix20} and $S_{vix20lag}$ in seven and nine of the ten cases, respectively. In addition, in this case, there is a discordant sign among the different versions of the sentiment, confirming that the approach

Electronic copy available at: https://ssrn.com/abstract=3937901

Sentiment Variable	Label	Time Window	Lag
S_{vx20}	VIX	20 min	no
$S_{vx20lag}$	VIX	20 min	yes
S_{vx10}	VIX	10 min	no
$S_{xv10lag}$	VIX	10 min	yes

Table A.3: Sentiment variables based on LSTM neural network applied on articles labeled with the log difference of the VIX Index.

based on stock index returns is highly sensitive to the initial settings. Analogously, λ_m is significant in six and nine of ten cases for S_{vix20} and $S_{vix20lag}$, respectively. As the tables shows, the coefficient exhibits different signs according to the different versions of sentiment. Also in the paper, we conclude that stock index returns do not represent a reliable approach since it fails in the mapping process between news and financial returns.

Electronic copy available at: https://ssrn.com/abstract=3937901

	(1)		(2)		(3)		(4)		(5)		(6)		(7)		(8)		(9)		(10)	
μ	0.00005	***	-0.00002		-0.00002		-0.00005	***	0.00003		0.00010	***	-0.00003		0.00004		-0.00008	***	-0.00003	
	(0.00001)		(0.00003)		(0.00004)		(0.00000)		(0.00004)		(0.00000)		(0.00003)		(0.00004)		(0.00000)		(0.00004)	
λ_m	-0.00007	***	0.00004		0.00000		-0.00011	*	-0.00006	**	-0.00000		-0.00006	**	-0.00004	*	-0.00004	***	-0.00001	
	(0.00001)		(0.00003)		(0.00003)		(0.00006)		(0.00003)		(0.00003)		(0.00002)		(0.00002)		(0.00000)		(0.00003)	
ω	-1.67612	***	-0.06083	***	-0.06543	***	-2.11980	***	-1.57159	***	-0.84343	***	-0.05771	***	-1.54431	***	-0.05078	***	-0.10591	***
	(0.59427)		(0.00314)		(0.00856)		(0.00207)		(0.53143)		(0.00052)		(0.00346)		(0.51385)		(0.00001)		(0.00762)	
α	-0.04219		-0.08343	***	-0.08007	***	-0.43339	***	-0.05000		0.76037	***	-0.07716	***	-0.02956		-0.09247	***	-0.08510	***
	(0.05456)		(0.01740)		(0.02181)		(0.00194)		(0.05326)		(0.00006)		(0.02075)		(0.05449)		(0.00002)		(0.02324)	
β	0.87276	***	0.99547	***	0.99512	***	0.82373	***	0.88250	***	0.92625	***	0.99574	***	0.88263	***	0.99536	***	0.99274	***
	(0.04505)		(0.00004)		(0.00008)		(0.00109)		(0.03966)		(0.00030)		(0.00008)		(0.03850)		(0.00008)		(0.00010)	
γ	0.46717	***	0.02248		0.02602		-0.29500	***	0.43321	***	0.05661	***	0.02628		0.44723	***	-0.03468	***	0.01547	
	(0.09940)		(0.01875)		(0.03749)		(0.00007)		(0.09921)		(0.00103)		(0.03224)		(0.09531)		(0.00002)		(0.03795)	
λ_v	-0.08553		0.00630		0.00095		0.08182	***	-0.07056		0.03942	***	0.00056		-0.11357	**	0.02588	***	-0.02024	
	(0.05699)		(0.01850)		(0.01244)		(0.01005)		(0.04884)		(0.00278)		(0.01424)		(0.05195)		(0.00000)		(0.01332)	
Skew	0.97068	***	0.95921	***	0.95316	***	0.91192	***	0.98683	***	1.04428	***	0.95773	***	0.99486	***	0.91853	***	0.94983	***
	(0.05173)		(0.05169)		(0.05176)		(0.03516)		(0.05793)		(0.03347)		(0.05217)		(0.05787)		(0.00586)		(0.05201)	
Shape	3.53390	***	3.92258	***	3.87590	***	2.16906	***	3.44743	***	2.11160	***	3.98012	***	3.58983	***	5.28774	***	3.99826	***
	(0.51945)		(0.62295)		(0.65053)		(0.00002)		(0.49902)		(0.00421)		(0.69790)		(0.53866)		(0.00096)		(0.69115)	

Table A.4: S_{vix20} as exogenous variable in the mean and variance equations of the EGARCH(1,1) model with skewed Student's conditional t-distribution fitted on S&P 500 Index intraday log returns with 20-minute time intervals.
Statistical significance at the 1% (***), 5% (**), 10% (*) levels.

Electronic copy available at: https://ssrn.com/abstract=3937901

	(1)		(2)		(3)		(4)		(5)		(6)		(7)		(8)		(9)		(10)	
μ	-0.00005	***	-0.00004	***	-0.00005	***	-0.00002		-0.00010	***	-0.00005	***	-0.00008	***	-0.00009	***	-0.00006	***	0.00000	
	(0.00000)		(0.00000)		(0.00000)		(0.00004)		(0.00000)		(0.00000)		(0.00000)		(0.00000)		(0.00000)		(0.00004)	
λ_m	-0.00005	***	0.00003	***	-0.00000	***	0.00000		0.00002	***	0.00001	***	0.00002	***	-0.00007	***	-0.00004	***	-0.00003	
	(0.00000)		(0.00000)		(0.00000)		(0.00003)		(0.00000)		(0.00000)		(0.00000)		(0.00000)		(0.00000)		(0.00002)	
ω	-0.06699	***	-0.07126	***	-0.06462	***	-0.07404	***	-0.04397	***	-0.07328	***	-0.08567	***	-0.07711	***	-0.05833	***	-1.07660	***
	(0.00000)		(0.00000)		(0.00000)		(0.00361)		(0.00000)		(0.00000)		(0.00004)		(0.00000)		(0.00000)		(0.30102)	
α	-0.10483	***	-0.10146	***	-0.10432	***	-0.08273	***	-0.11113	***	-0.11551	***	-0.10725	***	-0.12529	***	-0.09874	***	-0.05644	
	(0.00018)		(0.00001)		(0.00703)		(0.02495)		(0.00215)		(0.00114)		(0.00007)		(0.00069)		(0.00008)		(0.04746)	
β	0.99513	***	0.99416	***	0.99483	***	0.99449	***	0.99534	***	0.99361	***	0.99380	***	0.99448	***	0.99517	***	0.91845	***
	(0.00000)		(0.00016)		(0.00000)		(0.00009)		(0.00000)		(0.00019)		(0.00004)		(0.00008)		(0.00008)		(0.02244)	
γ	-0.02733	***	-0.03747	***	-0.02584	***	0.02129		-0.01695	***	-0.03403	***	-0.03892	***	-0.03014	***	-0.03116	***	0.33091	***
	(0.00017)		(0.00026)		(0.00004)		(0.04682)		(0.00069)		(0.00053)		(0.00013)		(0.00026)		(0.00008)		(0.09220)	
λ_v	-0.00880	***	-0.03856	***	0.00858	***	0.00403		-0.03213	***	-0.04035	***	0.02271	***	-0.02295	***	0.00894	***	0.10464	**
	(0.00063)		(0.00003)		(0.00050)		(0.01572)		(0.00002)		(0.00004)		(0.00005)		(0.00002)		(0.00000)		(0.04631)	
Skew	0.94924	***	0.94954	***	0.93079	***	0.94888	***	0.88842	***	0.92643	***	0.91354	***	0.89172	***	0.94920	***	1.00539	***
	(0.04620)		(0.04843)		(0.04484)		(0.05442)		(0.04221)		(0.03538)		(0.00595)		(0.01665)		(0.04619)		(0.05974)	
Shape	4.10436	***	4.06025	***	4.03054	***	3.80268	***	3.67073	***	3.57191	***	3.52227	***	3.79917	***	4.08413	***	3.46722	***
	(0.09502)		(0.00512)		(0.09364)		(0.64996)		(0.05956)		(0.00424)		(0.00251)		(0.00908)		(0.00149)		(0.50740)	

Table A.5: $S_{vix20lag}$ as lagged exogenous variable in the mean and variance equations of the EGARCH(1,1) model with skewed Student's conditional t-distribution fitted on S&P 500 Index intraday log returns with 20-minute time intervals.
Statistical significance at the 1% (***), 5% (**), 10% (*) levels.

Electronic copy available at: https://ssrn.com/abstract=3937901

LEIBNIZ INSTITUTE SAFE Sustainable Architecture for Finance in Europe

Recent Issues

No. 321	Ignazio Angeloni, Johannes Kasinger, Chantawit Tantasith	The Geography of Banks in the United States (1990-2020)
No. 320	Sebastian Steuer, Tobias H. Tröger	The Role of Disclosure in Green Finance
No. 319	Erik Theissen, Christian Westheide	Call of Duty: Designated Market Maker Participation in Call Auctions
No. 318	Kevin Bauer, Michael Kosfeld, Ferdinand von Siemens	Incentives, Self-Selection, and Coordination of Motivated Agents for the Production of Social Goods
No. 317	Volker Flögel, Christian Schlag, Claudia Zunft	Momentum-Managed Equity Factors
No. 316	Christian Mücke, Loriana Pelizzon, Vincenzo Pezone, Anjan Thako	The Carrot and the Stick: Bank Bailouts and the Disciplining Role of Board Appointments
No. 315	Kevin Bauer, Moritz von Zahn, Oliver Hinz	Expl(AI)ned: The Impact of Explainable Artificial Intelligence on Cognitive Processes
No. 314	Farshid Abdi, Mila Getmansky Sherman, Emily Kormanyos, Loriana Pelizzon, Zorka Simon	A Modern Take on Market Efficiency: The Impact of Trump's Tweets on Financial Markets
No. 313	Kevin Bauer, Andrej Gill	Mirror, Mirror on the Wall: Machine Predictions and Self-Fulfilling Prophecies
No. 312	Can Gao Ian Martin	Volatility, Valuation Ratios, and Bubbles: An Empirical Measure of Market Sentiment
No. 311	Wenhui Li, Christian Wilde	Separating the Effects of Beliefs and Attitudes on Pricing under Ambiguity
No. 310	Carmelo Latino, Loriana Pelizzon, Aleksandra Rzeźnik	The Power of ESG Ratings on Stock Markets
No. 309	Tabea Bucher-Koenen, Andreas Hackethal, Johannes Koenen, Christine Laudenbach	Gender Differences in Financial Advice
No. 308	Thomas Pauls	The Impact of Temporal Framing on the Marginal Propensity to Consume

Electronic copy available at: https://ssrn.com/abstract=3937901