


Article

# Financial Context News Sentiment Analysis for the Lithuanian Language

Rokas Štrimaitis <sup>1,\*</sup>, Pavel Stefanovič <sup>1</sup>, Simona Ramanauskaitė <sup>2</sup>  and Asta Slotkienė <sup>1</sup>

<sup>1</sup> Department of Information Systems, Vilnius Gediminas Technical University, LT-10223 Vilnius, Lithuania; pavel.stefanovic@vilniustech.lt (P.S.); asta.slotkiene@vilniustech.lt (A.S.)

<sup>2</sup> Department of Information Technology, Vilnius Gediminas Technical University, LT-10223 Vilnius, Lithuania; simona.ramanauskaite@vilniustech.lt

\* Correspondence: rokas.strimaitis@vilniustech.lt

**Abstract:** Financial area analysis is not limited to enterprise performance analysis. It is worth analyzing as wide an area as possible to obtain the full impression of a specific enterprise. News website content is a datum source that expresses the public's opinion on enterprise operations, status, etc. Therefore, it is worth analyzing the news portal article text. Sentiment analysis in English texts and financial area texts exist, and are accurate, the complexity of Lithuanian language is mostly concentrated on sentiment analysis of comment texts, and does not provide high accuracy. Therefore in this paper, the supervised machine learning model was implemented to assign sentiment analysis on financial context news, gathered from Lithuanian language websites. The analysis was made using three commonly used classification algorithms in the field of sentiment analysis. The hyperparameters optimization using the grid search was performed to discover the best parameters of each classifier. All experimental investigations were made using the newly collected datasets from four Lithuanian news websites. The results of the applied machine learning algorithms show that the highest accuracy is obtained using a non-balanced dataset, via the multinomial Naive Bayes algorithm (71.1%). The other algorithm accuracies were slightly lower: a long short-term memory (71%), and a support vector machine (70.4%).

**Keywords:** sentiment analysis; financial news analysis; supervised machine learning; Lithuanian language



**Citation:** Štrimaitis, R.; Stefanovič, P.; Ramanauskaitė, S.; Slotkienė, A. Financial Context News Sentiment Analysis for the Lithuanian Language. *Appl. Sci.* **2021**, *11*, 4443. <https://doi.org/10.3390/app11104443>

Academic Editor: Julian Szymanski

Received: 19 March 2021

Accepted: 12 May 2021

Published: 13 May 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In the business field, companies are always attempting to discover the best solutions on how to optimize the business process by implementing new technologies. There are various reasons to do so, such as increasing company profits, discovering new reliable partners for expansions, etc. As a rule, companies always have stored data about the clients and partners, but this information is more official and provided by the users themselves. Therefore, one solution is to find more unstructured information about clients in social networks, blogs, news websites, or other public sources, and then perform the sentiment analysis. Current research mostly focuses on user or client comments, but we analyze news website texts, where the structures of the texts are different from comments. Sentiment analysis is the field of data analysis, where the main aim is to assign the sentiment to unknown texts, usually positive, negative, or neutral, or in other words, emotion extraction [1]. It can help to identify the viewpoint about something solely from written text; it is the first step of context analysis. Sentiment analysis can be used in various fields, such as analysis of customer reviews [2], YouTube user comments [3], social networks [4], websites news [5], etc., but the main concept of sentiment analysis is the same—to retrieve the short texts and assign the sentiment.

The most commonly used ways to perform sentiment analyses are via lexicon-based methods and classification algorithms. The goal of the lexicon-based method [6] is to prepare the positive and negative list of words, and later, use tagging algorithms to assign

the values (positive, negative) to the words in the texts. In such a way, according to the tagged word frequency, the score is calculated and the sentiment is assigned. The main problem of this method is to correctly prepare the lists and to find the best score, which allows separating datasets into the sentiment groups, especially when the neutral sentiment is used. To use the classification algorithms [7], the main aim is to correctly prepare and preprocess the dataset, and preferably use a large number of dataset items to train a model. The various classifiers and their parameters can be used to obtain the best results, so deep analysis must be conducted.

The difficulty of sentiment analysis is that, even for a human, it is sometimes hard to decide the difference among positive, negative, and neutral texts. Usually, in the same text, there can be two sentiments, so sentiment analysis accuracy is often not so high. In this paper, both methods were used to assign the sentiment analysis of financial context Lithuanian news. The newly collected dataset from four main Lithuanian news websites were collected and experimentally analyzed [8]. There is no database where this kind of information would be freely accessed. Most sentiment datasets are oriented on comment sentiment estimation; therefore, it is unknown how accurate sentiment can be estimated on financial news topics published on websites. Therefore the paper's main contribution is datasets for sentiment analysis for financial news, as well as research on how effective existing classification methods are for sentiment estimation with this dataset.

To perform the lexicon-based approach we used the positive and negative bigrams prepared with financial company experts. The supervised machine learning model was implemented to discover the best classifier for our collected dataset. The multinomial Naive Bayes, support vector machine, and a long short-term memory algorithm were analyzed and compared. To evaluate the results, the model accuracy was calculated. Moreover, hyperparameters optimization was used to discover the best parameters for each classifier. In Section 2, the related works are analyzed. The implemented supervised machine model for sentiment analysis are described and presented in Section 3. Moreover, in this section, the newly collected dataset is described. Finally, the experimental investigation and results comparison are presented in Section 4.

## 2. Related Works

These days, the most common solution for a smart system is the application of artificial intelligence (AI) techniques. AI helps to solve various tasks, such as classification, prediction, clustering, etc. One sub-area of artificial intelligence is machine learning (ML). The main aim of ML in practice is to analyze data, learn from obtained results, and then make a decision or prediction about something in the analyzed area. The more the ML model learns by feeding it data, the better the results will be. Nowadays, machine learning models are used in different areas [9–11], but in all of them, the main model concept is generally the same. There are various research studies on using machine learning for sentiment analysis. For example, the Twitter sentiment analysis was performed using a sentiment analysis model based on Naive Bayes and Support Vector Machine [12]. In other research, the experiments were made using five machine learning algorithms to run sentiment analyses of movie reviews [13]. Sentiment analysis solutions are used to predict the stock market price based on Twitter message sentiment as well [14]. They all had the same task—to find the best model for a specific dataset, but results can be obtained in various ways. Machine learning models, according to the learning style, can be grouped as supervised, unsupervised, and hybrid learning, where the main factor of choosing one of them is the analyzed dataset (classified data, unclassified data).

Sentiment analysis is a classification task that attempts to predict whether a positive or negative emotion is expressed in a text. Generally, it is a binary classification task—the text is assigned to one of two classes (positive or negative), but often, as in this paper, the third class is also relevant—a neutral opinion. In that case, it is considered the multiclass classification task. Sentiment analysis can be applied in different areas, such as user opinions on received products or movies recommendations, according to the

user comments, etc. However, financial or business area texts have their specifics, which have to be taken into account. General-purpose models are not effective enough, as some specialized term and event text style is used in a financial context. Therefore, close to general-purpose models for language understanding [15], pre-trained language models for the financial area arise [16].

In 2018, Sahar Sohangi published a paper [17] in which he tried to extract sentiment from user messages on the financial social platform StockTwits. In other words, an attempt was made to predict whether the stock price would increase or not (binary classification) based on sentiments of user messages using traditional machine learning (doc2vec) and deep learning (LSTM and CNN) methods. The results showed that doc2vec is not effective in extracting sentiment from the large financial news dataset and an accuracy of 0.67623 was achieved. Long short-term memory (LSTM) was constructed using the average pooling layer and logistic regression layer in the last step. This solution also managed to achieve an accuracy of 0.6923. The best accuracy was achieved with a convolutional neural network (CNN)—0.9093 in 10,000 steps. Filters with sizes of 3, 4, and 5, max-pooling layer, dropout regularization, and softmax as the last layer were used to construct this model. The result of CNN was quite high, but it should be noted that only two classes were used in this classificatory.

Comparative analysis of machine learning methods to extract sentiments from user messages on the StockTwits platform was performed by Thomas Renault [18]. The research was conducted with different sizes of balanced and unbalanced data and the results showed that the size of the dataset had a greater effect on accuracy than the choice of method. It was also shown that accuracy does not improve significantly when text features are composed of more than two words (bigrams). The highest accuracy was obtained with the maximum entropy classifier—0.74451. More complex methods do not necessarily guarantee higher accuracy, so research should not rule out simpler methods.

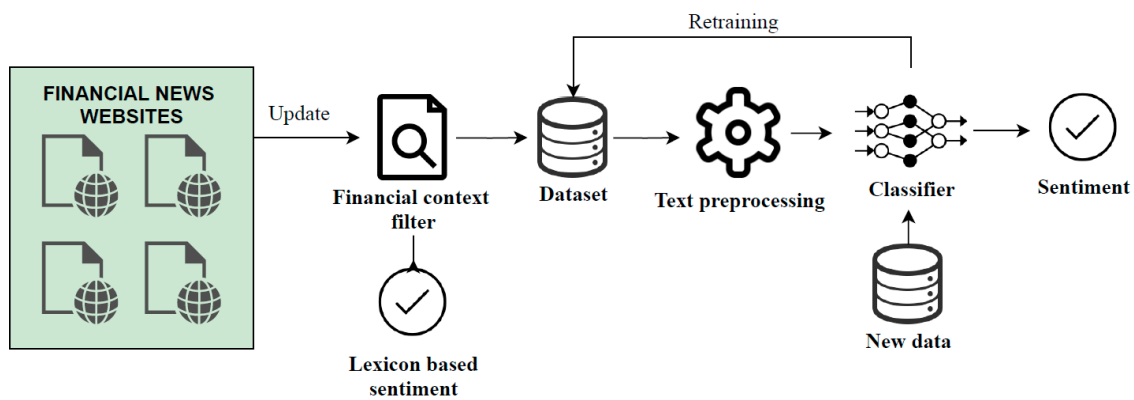
Several authors applied natural language processing (NLP) techniques to extract sentiment from financial data. Anita Yadav published a paper [19] in which financial sentences were tagged with different parts of speech, and it was examined which pairs (of parts) of speech revealed sentiment the most. The best accuracy of 0.79 was obtained using a combination of noun and verb pairs. Xingchen Wan, in his paper [20], extracted a value of sentiment for each company mentioned in financial news articles on the Reuters website. A state-of-the-art attentive neural network model was used to identify sentiments and they were used to discover a correlation between sentiment and associated market movements.

While financial area texts are more specific than general texts, non-English language analysis has its specifics as well. The spelling of the Lithuanian language is complicated, because of word form variety as well as sentence structure. Therefore, Lithuanian language text analysis is still evolving [21,22]. Models for sentiment analyses of financial context texts do not exist at the moment; however, research on different text classifications for Lithuanian language exists. Jurgita Kapočiūtė-Dzikienė, in her paper, attempted to classify the Lithuanian comments from the news portal as positive, negative, or neutral [23]. Traditional machine learning methods—support vector machines (SVM) and Multinomial Naïve Bayes (MNB)—and deep learning methods—LSTM and CNN—were used for sentiment analysis. The best results were achieved with a full unbalanced dataset including emoticons: MNB achieved an accuracy of 0.735 and SVM—0.723 (both with restored diacritics). CNN was able to achieve an accuracy of 0.705 with an unbalanced full dataset including emoticons, but eliminated/excluding diacritics. The worst results were achieved with LSTM, using a balanced dataset without preprocessing—0.612. The results of the Lithuanian language text sentiment analysis are promising; however, it is unclear how efficient these methods would be for financial context texts.

### 3. Supervised Machine Learning Model

In this paper, the supervised machine learning model was used as a classification task to assign the sentiments for the financial news dataset collected from Lithuanian news

websites. The scheme that represents the concept of our suggested model is presented in Figure 1.



**Figure 1.** The suggested supervised machine learning model.

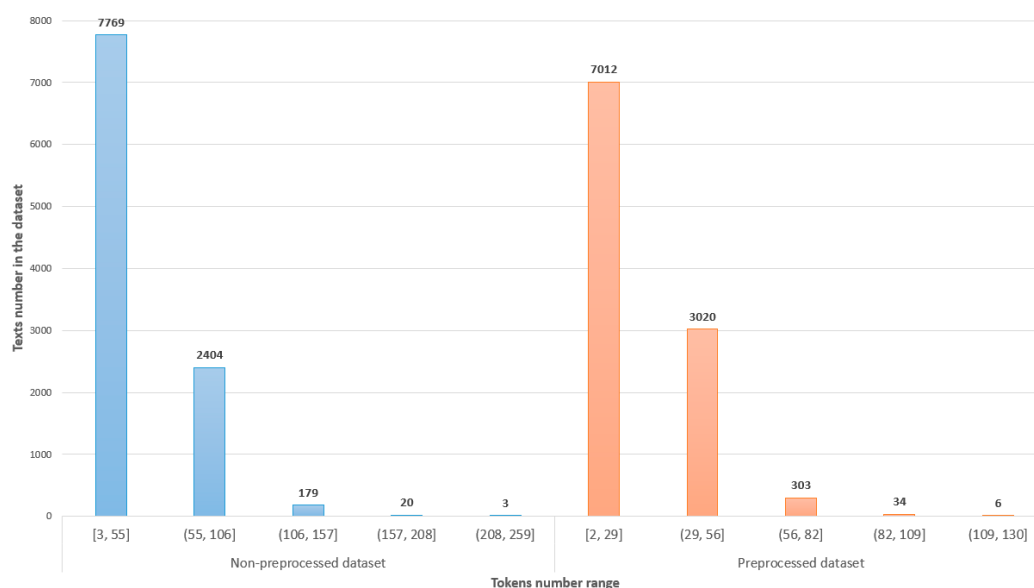
This model can be considered as a two parts model: lexicon-based sentiment and machine learning sentiment assignment. First, WEB scraping was used to collect the financial news from four main Lithuanian news websites and store it in a database as a text. To avoid the problems of some retrieved news not being strongly related to the financial area, the dataset was preprocessed by the suggested financial context filter. This filter can also be used as a sentiment assigned model to assign the sentiment to each dataset item: positive, negative, and neutral. In the last three steps, the data are preprocessed and trained using three commonly used classifiers in the sentiment analysis field: Naive Bayes, support vector machine, and long short-term memory algorithms.

To evaluate the model quality and to find the best classifier for our model, the accuracy was measured and compared. Moreover, the hyperparameter optimization using grid search was implemented to find the best parameters for each classifier with the highest accuracy. After the specified period, the dataset was updated, so the classifier had to be retrained. When we obtained the trained classifier, the new data could be given to the model as new input, and sentiments were assigned by the classifier.

### 3.1. Dataset

In this paper, performing experimental research, we used data collected from the four most popular Lithuanian language news websites. The data were collected from September 2020 to March 2021, using the “business news” category. The collected records were highly related to the financial area, where the texts were about financial operations, business projects, investments, political decisions that influenced companies, business expansion, law enforcement, etc. The data in the dataset were labeled by a company; the main field was accounting and business management software development with employees having more than 30 years of experience. Five experts from the financial department assigned sentiment (positive, negative, neutral) to the dataset manually. A discussion was executed between these employees to drive clear opinions on labeling guidelines. Moreover, portions of already labeled data were occasionally relabeled, to test labeling quality (different persons labeled the same records; differences in the labeling results were analyzed). The developed dataset was published and is available in Kaggle [8].

The analyzed dataset consisted of 10,375 texts, where 5780 texts were assigned to the class positive (POS), neutral (NEU)—1997 texts, and 2598 texts were negative (NEG). The total number of words and punctuation in the text, so-called tokens, were calculated. The distribution of the non-preprocessed and preprocessed dataset tokens are presented as a histogram in Figure 2.



**Figure 2.** The distribution of non-preprocessed and preprocessed datasets.

As we can see, the majority of dataset texts received less than 100 tokens and just 202 texts received more. To avoid problems of choosing too many preprocess filters (such as a frequent number of tokens), some text would become empty, so the main filters were selected: numbers were removed, all letters were changed to lowercase, punctuation erased, lower than 3-letter tokens were removed, Lithuanian language snowball stemmer was used [24], and Lithuanian language common words stop list (451 words) were included. After the dataset was preprocessed, all dataset items mostly received less than 82 tokens and the highest token numbers was equal to 130. The experimental investigation was performed using both non-preprocessed and preprocessed datasets.

The collected dataset was not balanced; the dominant class was positive, so it could be the reason why the classifier could learn improperly. For such a reason, we balanced the dataset in three ways and used it in an experimental investigation (Table 1). There was no other way to balance the dataset not using the artificial methods, so, in this paper, we chose to balance it by removing some parts of the dataset. Artificial methods to balance datasets were not suitable for such kind of research, because the text would have lost the natural text structure. According to the number of each class text, the dataset was POS balanced, NEU balanced, and NEG balanced. In the POS balanced way, the original dataset was used, because the dataset items from other class numbers were lower than positive, so all of them were included. In the case of NEU balance, the neutral class texts were taken, but the number of positive and negative class texts were shuffled and reduced randomly to have the same amount of text as the neutral class. In the NEG balanced way, all negative and neutral class texts were included, the positive class texts were reduced to the same number as the negative class.

**Table 1.** The dataset balancing ways.

Balancing	Records of Positive Class	Records of Neutral Class	Records of Negative Class	Total Number of Texts
POS balance	5780	1997	2598	10,375
NEU balance	1997	1997	1997	5991
NEG balance	2598	1997	2598	7193

### 3.2. Financial Context Filter

One of our suggested supervised machine learning model steps is a financial context filter based on a lexicon of positive (597) and negative (510) bigrams. The lists of bigrams were generated by five experts individually and then compared to eliminate conflicting bigrams.

As above-mentioned, this filter can also be used as a sentiment assignment model of financial news data. The base of sentiment analysis is text data analysis, so various techniques can be used to analyze such kinds of data [25]. The general suggested financial context filter scheme is given in Figure 3. At the first step, the dataset was preprocessed according to the selected filters: numbers were removed, all letters were changed to lowercase, punctuation erased, lower than 3-letter words were removed, and the Lithuanian language snowball stemmer was used. In this step, the common Lithuanian word stop list was not included as a filter, to avoid the wrong interpretation of bigrams in the sentiment.

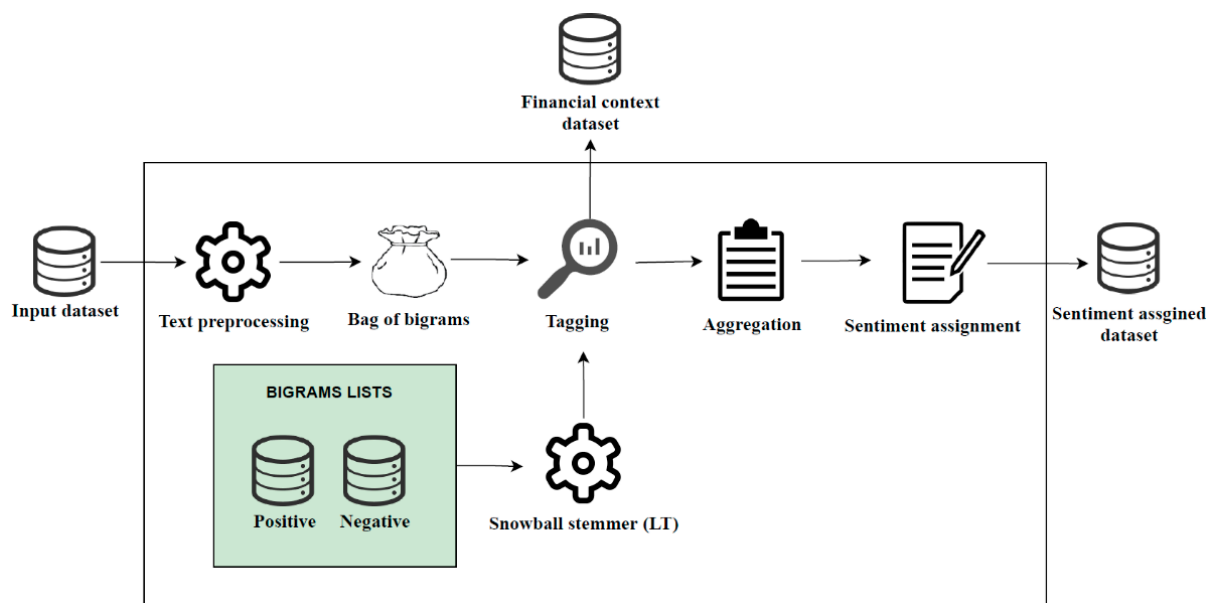


Figure 3. Financial context filter.

After the dataset was preprocessed, the bag of bigrams was created. According to the prepared positive and negative bigrams lists [8], the bag of bigrams was tagged with two tags: positive or negative. Before tagging was made, the Lithuanian snowball stemmer was used for the bigrams list, to make it a similar meaning to a bag of bigrams of the original dataset. The tagging was made with reverse bigrams too, in case bigrams were written in a different order. The bigrams list was prepared, cooperated with a financial company, and correctness confirmed and evaluated by five experts.

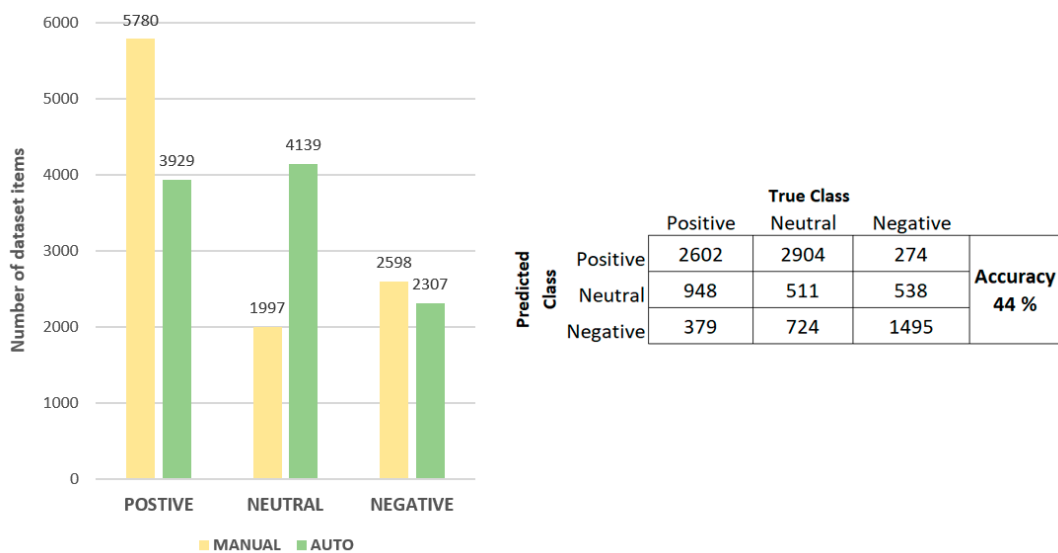
After the tagging process, the filtered dataset was given as an output, the experts could then assign the sentiment manually. At the aggregation step, the positive and negative number of tagged bigrams was calculated, which was later used to make a sentiment assignment. In the literature, we can find many different scores of sentiment assignments [26], but mostly in all of them, the total number of words or bigrams of analyzed data (O), the number of tagged positive (P) and negative (N) bigrams were used. The most common sentiment scores are presented in Table 2.

**Table 2.** Sentiment scores.

Description 1	Sentiment Score	
Absolute proportional difference. The score falls in the interval [0, 1].	$\frac{P-N}{P+N+O}$	(1)
Relative proportional difference. The score falls in the interval [-1, 1].	$\frac{P-N}{P+N}$	(2)
Logit scales. The score falls in the interval $(-\infty, +\infty)$ .	$\log(P + 0.5) - \log(N + 0.5)$	(3)

The primary research showed that there is no significant difference in regards to which sentiment score would be used with our dataset, so we chose the relative proportional difference score (2). In the last stage of this model, the sentiment was assigned according to the obtained score in such a way: positive—if the score was from the interval (0, 1], negative—if the measure was from the interval [-1, 0), neutral—if the measure was equal to 0.

In Figure 4, the distribution of the dataset is presented, where the sentiment is assigned by experts (manual) and using our suggested financial context filter (automatic).



**Figure 4.** The distribution of dataset (automatic and manual sentiment assignment) and confusion matrix.

As we can see, the number of dataset items assigned as a negative sentiment is similar in both ways; the difference is equal to the 291 dataset items. The bigger difference appears when we look at the numbers of the positive and neutral sentiments. The experts assigned positive sentiment more often than neutral compared to the automatic sentiment assignment. Using the automatic sentiment assignment dataset is more balanced than the manual way. To find the accuracy of the automatic sentiment assignment, the confusion matrix was calculated. Such kind of sentiment assignment model needs to be improved because, for now, the obtained accuracy is not high (just equal to 44%), so a better solution is needed. The best sentiment assignment is for the negative class, the worst for positive and neutral. However, it can be self-evident, because even for a human, sometimes it is hard to find the boundary between neutral and positive texts, each person’s interpretation varies.

### 3.3. Classification Algorithms and Their Quality Estimation

As above-mentioned, in this paper we chose two of the most commonly used traditional machine learning algorithms—Naive Bayes, a support vector machine (SVM), and one deep learning algorithm, a long short-term memory (LSTM). The related review

showed that these algorithms are suitable for sentiment analysis. Naive Bayes is a classification algorithm that applies density estimation to the data. The base of this algorithm is a Bayes theorem, where it naively assumes that the predictors are conditionally independent of the given class [27]. In our research, we used a multinomial Naive Bayes algorithm. The multinomial distribution is appropriate for a predictor whose observations are categorical. Naive Bayes classifiers assign dataset items to the most probable class using these steps: (1) estimate the densities of the predictors in each class; (2) model finds probabilities according to Bayes rule; (3) dataset item classification by estimating the probability for each class, and after, it assigns the dataset item to the class where the obtained probability is the highest.

A support vector machine is a supervised machine learning algorithm [28]. The main aim of the algorithm is to find an optimal boundary (hyperplane) between the possible outputs, it can help in the classification or regression tasks. Generally, the SVM does not support multiclass classification and it can deal just with binary classification, to separate dataset items into two classes. For multiclass tasks, the same principle is used, when the multiclass classification problem is split into multiple binary classification problems, the so-called “one-vs.-one” and “one-vs.-rest” methods can be used. Both methods are heuristic. Using the one-vs.-rest method, a binary classifier is trained on each binary classification problem and predictions are made using the model that is the most confident. The main issue of this method that the model has to be created for each class separately, so then the model can be slow and have problems when a dataset is large or have a very big number of classes. Unlike one-vs.-rest, the one-vs.-one approach splits the dataset into one dataset for each class versus every other class. Each binary classification model can predict one class and the model with the most predictions is predicted by the one-vs.-one method. The various parameters, such as kernel scale or function, can influence the SVM classification results.

A recurrent neural network (RNN) [29] is one of the artificial neural network (ANN) types that uses past information to improve the network performance of current and future inputs. The difference between RNN and other types of ANN, where the network contains a hidden state and loops, which allows the network to store past information in the hidden state and operate on sequences, makes the RNN unique. Over time, many algorithms based on RNN modifications showed up; one commonly used is long short-term memory [30]. The main problem of RNN, in practice, is with learning long-term dependencies, because recurrent neural networks are usually trained through the backpropagation method, so it can deal with vanishing or exploding gradient problems. In other words, the network weights can become very small or very high, so it limits and distorts the effectiveness of the learning process. The long short-term memory algorithm overcomes this issue by using the additional gates to control what information in the hidden cell makes it to the output and the next hidden state. The LSTM network consists of a cell, input gate, output gate, and forget gate, where the cell remembers values over some time intervals and the other gates regulate the flow of information into and out of the cell.

The long short-term memory network can be used in various fields and for different purposes; for example, speech recognition, image analysis, robot control, video analysis, sentiments analysis, etc. It can be used in classification, regression tasks, and in machine learning models as well. The core components of an LSTM network are a sequence input layer and an LSTM layer; each task requires a different LSTM architecture. In our paper, we used the long short-term memory as a classifier to assign the sentiment to the new input data. The network starts with a sequence input layer followed by an LSTM layer. To predict class labels, the network ends with a fully connected layer, a softmax layer, and a classification output layer. In the real world, the text data are naturally sequential, so the sentence or words in the text can be considered as a sequence, which is why it can be easily analyzed by LSTM. The all-training process can be done in four steps: (1) the text needs to be preprocessed; (2) the words need to be convert in the text to numeric sequences using a word encoding; (3) create and train the network with a word embedding layer;

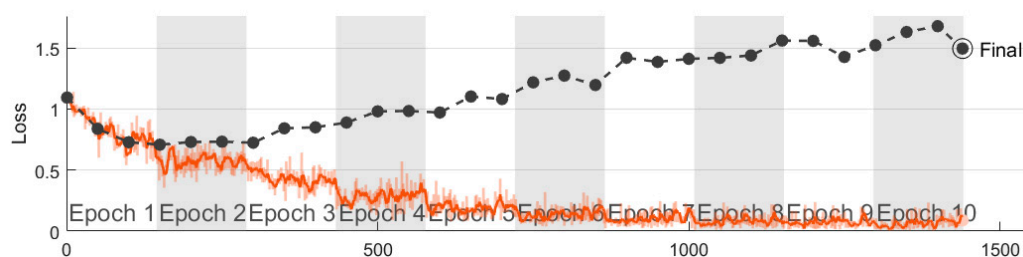


(4) use the trained LSTM network to classify the dataset. The quality of the trained LSTM network depends on various parameters starting from the text preprocessing part to the LSTM network parameters, so it is important to select the best parameters suitable for the analyzed dataset.

In our research, accuracy was used to evaluate the quality of trained classifiers. The classifier result could be compared to the actual result and summarized as follows: true positives (TP)—predicted positive and it is true; true negatives (TN)—predicted negative and it is true; false positives (FP)—predicted positive and it is false; false negatives (FN)—predicted negative and it is false. All of these values can be obtained and easily calculated from the confusion matrix. The formula of the accuracy measure is  $(TP + TN)/(TP + TN + FP + FN)$ .

#### 4. Experimental Investigation

To perform the experimental investigation to find the best classifier for the analyzed dataset, cross-validation and hyperparameters optimization with a grid search optimizer was used. Each experiment was repeated five times and the average accuracy was evaluated. For cross-validation, the commonly used 5-fold option was selected (80% training dataset, 20% testing dataset); moreover, the so-called stratify distribution was used, which helped to keep the same number of the class in each fold. Many parameters can influence the LSTM classifier results, but the main are sequence length (tokens number in the text), embedding dimension, hidden unit number, epoch, and of course, the LSTM architecture. In our research, we used the simple LSTM architecture because the analyzed dataset was quite small: sequence layer, word embedding layer, LSTM layer, dropout layer, fully connected layer (three class), softmax layer, and classifier layer. As we can see, one of the most complicated long short-term memory algorithms has many, various options; thus, first of all, the primary experiments were made to find the best epoch number for training. For this reason, all datasets were divided into three-part subsets: 80% training, 10% validation, and 10% testing. The related work review showed that, for LSTM models, the validation dataset was needed because it helped check the training process and avoid the underfit or overfit model problems. Two parameters of the LSTM algorithm were fixed regarding the primary experiments: dimension was equal to 50 and the epoch number was equal to 10. As we can see in Figure 5, the validation (black line) and training (orange line) loss curves became stable around epoch 2, meaning that further training does not make sense, the model does not learn.



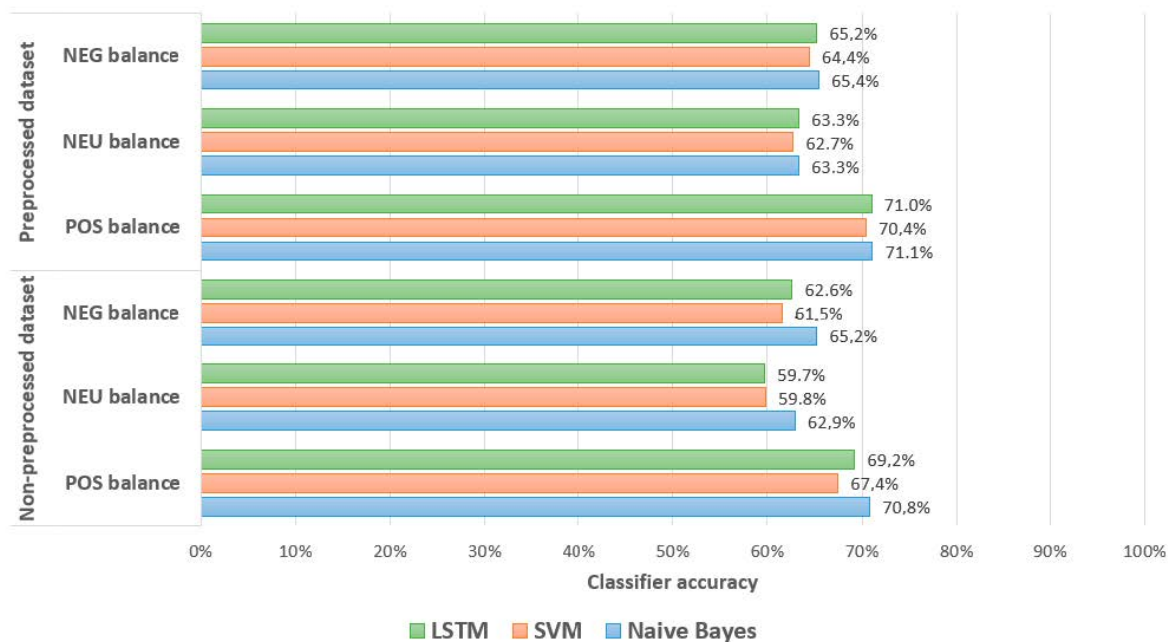
**Figure 5.** The LSTM training process of the analyzed dataset: training loss (orange), validation loss (black).

The hyperparameters optimization helps to find the best parameters of each classifier, which allows obtaining the highest model accuracy. The grid search was performed for two algorithms: a long short-term memory and a support vector machine. The multinomial Naive Bayes did not have a parameter to optimize, so for this model, the grid search was not used. All parameters, their bound ranges, obtained results, and all other important details of the experimental investigation, are presented in Table 3.

**Table 3.** Hyperparameters optimization using grid search optimizer.

Classification Algorithm	Dataset Preprocessing, Balance	Grid Search Parameter/Bounds	Best Parameters	Average Accuracy %	
Naive Bayes	Non-preprocessing	The grid search has not been performed for this algorithm.	POS balance	70.8	
			NEU balance	62.9	
			NEG balance	65.2	
	Preprocessed		POS balance	71.1	
			NEU balance	63.3	
			NEG balance	64.4	
Support vector machine	Non-preprocessing	Box constraint (BX): value log-scaled in the range [1e-3,1e3]. Kernel scale (KS): value log-scaled in the range [1e-3,1e3]. Kernel function (KF): Gaussian, linear, polynomial. Standardize (S): true, false.	BX: 0.4646; KS: -; KF: linear; S: false.	67.4	
			NEU balance	BX: 0.0215; KS: -; KF: linear; S: false.	59.8
			NEG balance	BX: 0.0215; KS: -; KF: linear; S: false.	61.5
	Preprocessed		POS balance	BX: 2.1544; KS: -; KF: linear; S: false.	70.4
			NEU balance	BX: 2.1544; KS: 10; KF: Gaussian; S: false.	62.7
			NEG balance	BX: 0.0215; KS: -; KF: linear; S: false.	64.4
Long short-term memory	Non-preprocessing	Number of tokens (NT): from 10 to 260 (step 10). Embedding dimension (ED): from 10 to 90 (step 20). Hidden units (HU): from 25 to 200 (step 25).	NT: 210; ED: 50; HU: 50.	69.2	
			NEU balance	NT: 50; ED: 30; HU: 30.	59.7
			NEG balance	NT: 150; ED: 70; HU: 75.	62.6
	Preprocessed		POS balance	NT: 70; ED: 50; HU: 50.	71.1
			NEU balance	NT: 30; ED: 60; HU: 50.	63.3
			NEG balance	NT: 30; ED: 50; HU: 75.	65.4

The comparison results, using a non-preprocessed and preprocessed dataset, with three balancing methods, are presented in Figure 6. As we can see, the accuracy results of the multinomial Naive Bayes algorithm almost, in all cases, were slightly higher, just in the cases of using the preprocessed dataset and NEU balance, the accuracy was the same as an LSTM algorithm (63.3%). Moreover, there was no significant difference between non-preprocessed and preprocessed dataset results. The highest accuracy was obtained using the preprocessed dataset and POS balance (71.1%), and the worst—NEU balance and non-preprocessed dataset (62.9%).



**Figure 6.** The algorithms comparison of financial news dataset.

Using the support vector machine algorithm accuracy, in all cases, was the lowest compared to the other analyzed classifiers. The hyperparameter optimization showed that the best kernel function in many cases was the linear, standardized and kernel scale were not used. The same as the multinomial Naive Bayes classifier, the SVM obtained the highest results with the preprocessed dataset and POS balance (70.4%), meanwhile the worst—non-preprocessed and NEU balance. We should mention that the difference between non-preprocessed and preprocessed dataset results was equal to approximately 2–3%, so this algorithm was more sensitive to the dataset preparation compared to the Naive Bayes where the accuracy difference was approximately 1%. The results obtained using the long short-term memory algorithm showed that the highest accuracy (71.1%) was when the dataset was preprocessed, the POS balance selected, and the parameters used were as follows: number of tokens—70, embedding dimensions—50, hidden units—50. The maximum token number of the analyzed preprocessed dataset was equal to 130, meaning that some dataset texts were truncated. The accuracy of the model using the maximum number of tokens was equal to 70.6%; the difference was small, meaning both ways were good to use. The same with the other classifier algorithms; the lowest accuracy was obtained using a non-preprocessed dataset and NEU balance (59.7%).

The main difference between our experimental investigation results compared with the results by Kapočiūtė-Dzikiene J. et al. [11], is the usage of the LSTM algorithm. In a related paper, the long short-term memory algorithm accuracy for a full non-balanced dataset is equal to approximately 50%; in our research we obtained 71.1%. In this paper, the experimental investigation showed that the long short-term memory accuracy was higher compared to the SVM, so it can be used in a financial context sentiment analysis similar to

other non-deep learning classifiers. The main result is the same as in the related paper's experimental research; the multinomial Naive Bayes algorithm, in all cases, obtains the highest model accuracy.

## 5. Conclusions and Future Work

Automated sentiment analysis for Lithuanian language texts is not an area that has been fully researched. Current research works in the area are mostly concentrated on comparative analysis of the traditional machine learning and deep learning methods for comments classification. However, another type of text specific and dataset balancing influence on sentiment analysis is missing. Therefore, our main contribution is the comparative analysis of the machine learning methods, solving the sentiment analysis task for the Lithuanian language with news text, published online, on the topic of finances.

Financial context filtering, based on a bag of positive and negative terms and bigrams, is not suitable for sentiment analysis of financial area news texts, as the achieved accuracy reaches approximately 44% only. It is not enough, as machine learning solutions increase the accuracy, almost in double; moreover, the number of manually-labeled and filtered items for each class significantly differs.

Traditional machine learning and deep learning methods for the sentiment analysis of Lithuanian-written financial news are more sensitive to lack of training data rather than an unbalance of classification classes. For all analyzed methods, the classification accuracy was highest (accuracy varies from 67.4% until 71.1%), with the biggest number of dataset records, which was the most unbalanced (dataset had 5780 records for positive class, 2598 records for negative class, and 1997 records for neutral class). Meanwhile, the fully balanced dataset (1997 records for each class) allowed achieving the maximum accuracy of 63.3% only.

As the sentiment analysis accuracy for Lithuanian financial news texts is dependent on the number of records in the dataset, future work should be oriented to extend the dataset and investigate the optimal size of the dataset. This knowledge would draw some guidelines for dataset requirements for sentiment analysis of the Lithuanian texts.

**Author Contributions:** Conceptualization, P.S. and R.Š.; methodology, P.S.; validation, P.S., R.Š., S.R. and A.S. formal analysis, R.Š. and P.S.; data curation, P.S., R.Š., S.R. and A.S.; writing—original draft preparation, R.Š. and P.S.; writing—review and editing, S.R. and A.S.; visualization, P.S.; supervision, P.S. and S.R. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research has been partly financed by the project “Extension of accounting and business management systems using the artificial intelligence methods” (No. 01.2.1-LVPA-K-856-01-0083).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Liu, B. Sentiment analysis and opinion mining. In *Synthesis Lectures on Human Language Technologies*; Morgan & Claypool Publishers: Chicago, IL, USA, 2012.
2. Sun, Q.; Niu, J.; Yao, Z.; Yan, H. Exploring eWOM in online customer reviews: Sentiment analysis at a fine-grained level. *Eng. Appl. Artif. Intell.* **2019**, *81*, 68–78. [[CrossRef](#)]
3. Bhuiyan, H.; Ara, J.; Bardhan, R.; Islam, D.M.R. Retrieving YouTube video by sentiment analysis on user comment. In Proceedings of the IEEE International Conference on Signal and Image Processing Applications (ICSIPA), Kuching, Malaysia, 12–14 September 2017; pp. 474–478.
4. Kharlamov, A.A.; Orekhov, A.V.; Bodrunova, S.S.; Lyudkevich, N.S. Social Network Sentiment Analysis and Message Clustering. In Proceedings of the 6th International Conference on Internet Science, Perpignan, France, 2–5 December 2019; Lecture Notes in Computer Science (LNCS). El Yacoubi, S., Bagnoli, F., Pacini, G., Eds.; Springer: Cham, Switzerland, 2019; Volume 11938, pp. 18–31.

5. Taj, S.; Shaikh, B.B.; Fatemah, M.A. Sentiment Analysis of News Articles: A Lexicon based Approach. In Proceedings of the 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 30–31 January 2019; pp. 1–5.
6. Taboada, M.; Brooke, J.; Tofiloski, M.; Voll, K.; Stede, M. Lexicon-based methods for sentiment analysis. *Comput. Linguist.* **2011**, *37*, 267–307. [[CrossRef](#)]
7. Hemmatian, F.; Sohrabi, M.K. A survey on classification techniques for opinion mining and sentiment analysis. *Artif. Intell. Rev.* **2019**, *52*, 1495–1545. [[CrossRef](#)]
8. Lithuanian Financial News Dataset and Bigrams. Available online: <https://www.kaggle.com/rokastrimaitis/lithuanian-financial-news-dataset-and-bigrams> (accessed on 16 March 2021).
9. Kourou, K.; Exarchos, T.P.; Exarchos, K.P.; Karamouzis, V.M.; Fotiadis, I.D. Machine learning applications in cancer prognosis and prediction. *Comput. Struct. Biotechnol. J.* **2015**, *13*, 8–17. [[CrossRef](#)] [[PubMed](#)]
10. Čeponis, D.; Goranin, N. Evaluation of deep learning methods efficiency for malicious and benign system calls classification on the AWSCTD. In *Security and Communication Networks*; Hindawi: London, UK, 2019; Volume 2019, pp. 1–12.
11. Rastenis, J.; Ramanauskaitė, S.; Suzdalev, I.; Tunaitytė, K.; Janulevičius, J.; Čenys, A. Multi-Language Spam/Phishing Classification by Email Body Text: Toward Automated Security Incident Investigation. *Electronics* **2021**, *10*, 668. [[CrossRef](#)]
12. Le, B.; Nguyen, H. Twitter Sentiment Analysis Using Machine Learning Techniques. In *Advanced Computational Methods for Knowledge Engineering; Advances in Intelligent Systems and Computing*; Le Thi, H., Nguyen, N., Do, T., Eds.; Springer: Cham, Switzerland, 2015; Volume 358.
13. Rahman, A.; Hossen, S.M. Sentiment Analysis on Movie Review Data Using Machine Learning Approach. In Proceedings of the 2019 International Conference on Bangla Speech and Language Processing (ICBSLP), Sylhet, Bangladesh, 27–28 September 2019; pp. 1–4.
14. Guo, X.; Li, J. A novel twitter sentiment analysis model with baseline correlation for financial market prediction with improved efficiency. In Proceedings of the 2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS), Granada, Spain, 22–25 October 2019; pp. 472–477.
15. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
16. Liu, Z.; Huang, D.; Huang, K.; Li, Z.; Zhao, J. Finbert: A pre-trained financial language representation model for financial text mining. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI, Yokohama, Japan, 7–15 January 2021; pp. 4513–4519.
17. Sohangir, S.; Wang, D.; Pomeranets, A.; Khoshgoftaar, T.M. Big Data: Deep Learning for financial sentiment analysis. *J. Big Data* **2018**, *5*, 3. [[CrossRef](#)]
18. Renault, T. Sentiment analysis and machine learning in finance: A comparison of methods and models on one million messages. *Digit. Financ.* **2020**, *2*, 1–13. [[CrossRef](#)]
19. Yadav, A.; Jha, C.K.; Sharan, A.; Vaish, V. Sentiment Analysis of Financial News Using Unsupervised and Supervised Approach. In *Pattern Recognition and Machine Intelligence; Lecture Notes in Computer Science*; Deka, B., Maji, P., Mitra, S., Bhattacharyya, D., Bora, P., Pal, S., Eds.; Springer: Cham, Switzerland, 2019; p. 11942.
20. Wan, X.; Yang, J.; Marinov, S.; Calliess, J.-P.; Zohren, S.; Dong, X. Sentiment correlation in financial news networks and associated market movements. *Sci. Rep.* **2021**, *11*, 3062. [[CrossRef](#)] [[PubMed](#)]
21. Krilavičius, T.; Medelis, Z.; Kapočiūtė-Dzikienė, J.; Žalandauskas, T. News Media Analysis Using Focused Crawl and Natural Language Processing: Case of Lithuanian News Websites. *Commun. Comput. Inf. Sci.* **2012**, *319*, 48–61.
22. Kapočiūtė-Dzikienė, J.; Davidsonas, A.; Vidugirienė, A. Character-Based Machine Learning vs. Language Modeling for Diacritics Restoration. In *Information Technology and Control*; Technologija: Kaunas, Lithuania, 2017; Volume 46.
23. Kapočiūtė-Dzikienė, J.; Damaševičius, R.; Woźniak, M. Sentiment Analysis of Lithuanian Texts Using Traditional and Deep Learning Approaches. *Computers* **2019**, *8*, 4. [[CrossRef](#)]
24. Lithuanian Stemming Algorithm. Available online: <https://snowballstem.org/algorithms/lithuanian/stemmer.html> (accessed on 29 January 2021).
25. Stefanovič, P.; Kurasova, O.; Štrimaitis, R. The n-grams based text similarity detection approach using self-organizing maps and similarity measures. *Appl. Sci.* **2019**, *9*, 1870. [[CrossRef](#)]
26. Kiritchenko, S.; Zhu, X.; Saif, M. Sentiment Analysis of Short Informal Text. *J. Artif. Intell. Res. (JAIR)* **2014**, *50*. [[CrossRef](#)]
27. Trevor, H.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed.; Springer Series in Statistics; Springer: New York, NY, USA, 2009.
28. Shmilovici, A. Support Vector Machines. In *Data Mining and Knowledge Discovery Handbook*; Maimon, O., Rokach, L., Eds.; Springer: Boston, MA, USA, 2005.
29. Williams, R.J.; Hinton, G.E.; Rumelhart, D.E. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536.
30. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]