International Journal of New Computer Architectures and their Applications (IJNCAA) 7(2): 62-67
The Society of Digital Information and Wireless Communications, 2017 ISSN 2220-9085 (Online); ISSN 2412-3587 (Print)

Text Classification Using Time Windows Applied to Stock Exchange

Pavel Netolický, Jonáš Petrovský, František Dařena, Jan Žižka

Department of Informatics, Faculty of Business and Economics, Mendel University in Brno, Zemědělská 1, 613 00 Brno, Czech Republic

pavel.netolicky@mendelu.cz, jonas.petrovsky@mendelu.cz, frantisek.darena@mendelu.cz, jan.zizka@mendelu.cz

ABSTRACT

Each day, a lot of text data is generated. This data comes from various sources and may contain valuable information. In this article, we use text classification to discover if there is a connection between textual documents (specifically Facebook posts) and changes of the S&P 500 stock index. The index values and documents were divided into time windows according to the direction of the index value changes. In the first experiment, we used a batch processing approach to put the documents from all windows into one data set and a classification accuracy of 62% was achieved. In the second experiment, we used a data stream approach to divide documents into twelve data sets created from two neighboring windows and we achieved an accuracy of 68%. This indicates that posts, which companies write on their Facebook pages, are partially related to the performance of the stock index. Taking the concept change into account also enables better quantification of this relationship.

KEYWORDS

Machine Learning, Classification, Text Mining, Stock Exchange, Time Windows, Data Streams

1 INTRODUCTION

A huge amount of data is constantly being generated by people and organizations. The speed of data creation is rapidly growing and we use the term "data stream" for the constant flow of new data [1].

Data streams may be of various data types (text, image, numeric) and come from different application areas (computer networks monitoring, scientific experiments, internet search, social networks etc.). In comparison to batch processing (for which we have all data available at once), data streams processing needs a different approach, because classical approaches are not effective or even feasible [2].

In this article, we will focus on the connection between text documents published on the Internet and movements of stock prices (represented by a composite value of stock index). Some research in this area uses structured (quantitative) data to analyze the impact of data on stock prices [3]. Unstructured data (like text) may provide us with another complementary information with additional hard-to-quantify knowledge [4].

Behavioral finance theory says that emotions may deeply influence behavior and decision making of individuals as well as whole human societies [5]. This means that the prices on capital markets are (more or less) influenced by emotions, moods and opinions of market participants [6]. These attributes are often contained in text documents and therefore we decided to use text data for our research.

[7] examined the connection between the content of messages posted to a discussion board and movements of the Czech stock index. We will expand this approach further by focusing on the US stock market, using a larger number and another type (Facebook

posts) of text data and treating stock prices and related text documents as data streams divided into time windows as we suppose that the reasons of stock price changes evolve in time.

2 CURRENTLY USED METHODS

To model the behavior of a stock price with a relation to the content of text data we can use classification in a way that we examine the direction of the change of the stock price to create classes. This approach was used for example by [6]. The problem can be seen as text classification – given a text, decide its class (direction of the price movement). However, we must overcome two problems. The first problem is the definition of classes. [8] used a threshold value of 1% price change for the class determination. The second problem lies in choosing correct features. Many studies used just single words and this simple unigram bag-of-words model provided good results in [8].

There exist a wide range of supervised learning algorithm that can be uses for the text classification. An interesting approach is described in [9] – it focuses on sentence-level sentiment analysis of movie reviews. They used the cosine normalization, Term Presence, and Smoothed delta IDF as weighting schemes and the Recursive Neutral Tensor Network algorithm to achieve an accuracy of 87.60%. [10] used Naïve Bayes and SVM as algorithms and unigrams, bigrams, unigrams with bigrams, and unigrams with POS (Parts-of-speech) as features. The bigrams showed a lower accuracy then unigrams – the reason is that the resulting vectors were very sparse. All in all, the type of features used in the bag-of-words model has a little (maximal 2–3%) impact on the accuracy.

3 DATA AND METHODOLOGY

The goal of the work was to examine whether the content of text documents published on the Internet has any connection with stock price movements. We decided to use Facebook posts from company pages as the text data, because it has been a very rarely used data source for this area of research, we have lots of available data, and it might bring new interesting insights.

3.1 Stock prices

In our research, the values of the S&P 500 Index were used to represent stock prices. The index values reflect stock prices of the selected blue chip (large and famous) companies on the US stock market. The historical values of the index were downloaded from the website *investing.com*. For each trading day, we have a closing (end-of-day) numeric value of the S&P 500 Index available.

3.2 Text data

As the text data, posts from Facebook pages of the companies from the S&P 500 Index were used. In total, we examined 431 company pages. The company's Facebook page contains a sequence of documents arranged according to their publication time. These short postings are created by the company representatives. Figure 1 shows an example of a post on the Intel's page. A post may be commented by Facebook users. However, the comments were not used in the analysis.

In total, 138,713 Facebook posts published between 1. 1. 2015 and 15. 10. 2016 were used.

Figure 1. Example of a Facebook post

3.3 Classification methodology

We used text classification to predict whether the given document is connected with an upward or downward movement of the S&P 500 Index.

We examined the time series of the S&P 500 Index values between 1. 1. 2015 and 15. 10. 2016 and found time intervals (windows) in which the change (either positive or negative) of the index value between the first and the last day of the interval was at least 5%. In total, 24 such windows were found. In 12 of them, the index value grew and in 12 it declined. The length (a number of days) of the time windows varied between 4 and 30. Then, each document was, based on the time window in which it was published, assigned a class: 1 (up) for the positive index value change, 2 (down) for the negative one. Figure 2 shows an example of time windows between 1. 1. 2016 and 1. 4. 2016 with the assigned classes.

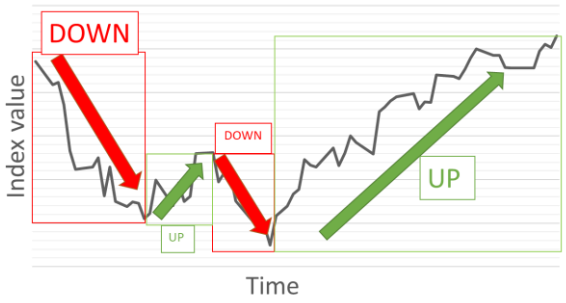

Figure 2. Classification classes identified in the time series of the stock index values

We decided to perform two types of experiments with different data sets used for the classification. In the first experiment, documents from all 24 windows were put into one data set. The results for this experiment are presented in section "Batch approach".

In the second experiment, we divided the documents into 12 data sets. Each data set consisted of the documents from two neighboring windows: one with an upward movement and one with a downward movement. The windows represented two classes for the classification. The results for this experiment are presented in section "Neighboring windows approach".

Text pre-processing and conversion

The raw text of each document was processed by a Python script as follows:

1. Remove all whitespace.
2. Lowercase all letters.
3. Tokenize the document – get words (using *TreebankWordTokenizer*).
4. Filter words – minimal length of three letters, exclude numbers.

The edited text was converted into a structured format by using a Python library called *scikit-learn* and its *Vectorizer* class. Only words that occurred at least 5 times in the whole document collection were included in the resulting vector representation.

The documents were converted to a bag-of-words representation using three different weighting schemes for the term-document matrix [11, p. 21–26]:

- Term Presence (TP): 1 if a term was present in a document, 0 if not.
- Term Frequency (TF): number of times a term was present in a document.
- TF-IDF: TF (local weight) multiplied by the IDF (global weight).

Classification

The converted data was split into the training (60%) and testing (40%) set. Each bag-of-words representation was processed by 10 classifiers (with default settings – no parameter optimization was made) in *scikit-learn*. The classifier's performance was evaluated by the achieved accuracy (proportion of the correctly classified instances on all examined instances [10, p. 268]) on the test set.

4 RESULTS AND DISCUSSION

One set of the text data (Facebook posts) together with the S&P 500 Index values was used to prepare the data for classification. The class-labelled data set was processed using the three weighting schemes (TP, TF, TF-

IDF) by 10 classification algorithms. In total, 30 classification results were obtained.

4.1 Batch approach

Tables 1, 2, 3 and 4 show the results for classification of the data set including all windows.

Table 1. Facebook posts – data statistics.

Total samples	Class 1 samples	Class 2 samples	Number of words
138,713	85,286	53,427	33,397

Table 2. Facebook posts – classification results.

Accuracy	Precision	Recall	F1 score
0.623	0.605	0.623	0.614

Table 1 shows the statistics about the data used for the classification. It is obvious that the data set was quite unbalanced (with more documents marked with index value going up). Table 2 shows the best classification results. The highest accuracy (62%) was achieved with the TF-IDF weighting scheme and the Multinomial Naïve Bayes classification algorithm.

Table 3 tells us that the used weighting scheme was not very important. However, we can see that the highest average accuracy was achieved by TF-IDF.

Table 3. The comparison of average accuracies achieved for each weighting scheme.

Weighting scheme	Average accuracy from all experiments
TF-IDF	0.598
TP	0.587
TF	0.586

Table 4 shows for each classifier the average accuracy from all experiments. We can see that the decision tree classifiers "ExtraTreesClassifier" and "RandomForestClassifier" performed the best with the accuracy around 61%.

Table 4. The comparison of average accuracies achieved by each classifier.

Classifier	Average accuracy
ExtraTreesClassifier	0.613
RandomForestClassifier	0.609
MultinomialNB	0.602
LogisticRegression	0.601
BernoulliNB	0.598
LinearSVC	0.597
MLPClassifier	0.596
DecisionTreeClassifier	0.564
NearestCentroid	0.533

4.2 Neighboring windows approach

Tables 5, 6, 7, and 8 show the results achieved for the 12 data sets consisting of two neighboring windows.

Table 5. Facebook posts – neighboring windows: data statistics.

Data set no.	Class 1 samples	Class 2 samples	Total samples	Data set balance ratio	Number of words
1	477	520	997	0.917	691
2	1,156	719	1,875	1.608	1,523
3	4,293	953	5,246	4.505	3,627
4	1,187	2,008	3,195	0.591	2,350
5	8,971	2,207	11,178	4.065	6,523
6	3,052	4,802	7,854	0.636	4,995
7	3,454	3,998	7,452	0.864	4,684
8	8,399	15,292	23,691	0.549	10,441
9	5,198	2,211	7,409	2.351	4,646
10	1,918	2,828	4,746	0.678	3,191
11	26,111	8,139	34,250	3.208	13,269
12	21,070	9,750	30,820	2.161	12,116

Table 5 shows the statistics about the data used for the classification. Because the length of the windows was variable, the numbers of documents greatly vary. It is also visible that most of the data sets are imbalanced. This should be taken into account when evaluating the results.

Table 6. Facebook posts – neighboring windows: classification results.

Data set no.	Accuracy	Precision	Recall	F1 score
1	0.584	0.602	0.584	0.593
2	0.615	0.598	0.615	0.606
3	0.808	0.653	0.808	0.722
4	0.639	0.733	0.639	0.683
5	0.803	0.807	0.803	0.805
6	0.646	0.653	0.646	0.650
7	0.554	0.553	0.554	0.553
8	0.674	0.669	0.674	0.672
9	0.721	0.727	0.721	0.724
10	0.618	0.614	0.618	0.616
11	0.800	0.782	0.800	0.791
12	0.698	0.702	0.698	0.700
Average	0.680	0.674	0.680	0.676

According to Table 6, the average accuracy (as well as the F1 score) was 68%. The best accuracy (as well as F1 score) was achieved for data sets 3 (72%), 5 (80%), and 11 (79%). The reason for this might be that they have a balance ratio around 4 (with more documents marked with index value going up).

Table 7. The comparison of average accuracies achieved with different weighting schemes applied to the neighboring windows of the Facebook posts.

Data set no.	TP	TF	TF-IDF
1	0.539	0.534	0.534
2	0.555	0.556	0.570
3	0.744	0.753	0.769
4	0.626	0.625	0.639
5	0.735	0.736	0.765
6	0.610	0.609	0.620
7	0.534	0.534	0.536
8	0.633	0.629	0.646
9	0.663	0.661	0.675
10	0.568	0.562	0.578
11	0.754	0.747	0.766
12	0.637	0.644	0.658
Average	0.633	0.633	0.646

From Table 7 can be seen that the highest average accuracy provided the TF-IDF weighting scheme (+1% in comparison to TP and TF).

Table 8. The comparison of average accuracies achieved by different classifiers applied to the neighboring windows of the Facebook posts.

Data set no.	Classifier	Avg. accuracy
1	NearestCentroid	0.587
2	LogisticRegressionCV	0.578
3	LogisticRegression	0.748
4	LogisticRegressionCV	0.633
5	LogisticRegression	0.784
6	MultinomialNB	0.630
7	ExtraTreesClassifier	0.553
8	SGDClassifier	0.656
9	MultinomialNB	0.701
10	LogisticRegressionCV	0.595
11	ExtraTreesClassifier	0.788
12	SGDClassifier	0.673

Table 8 shows the classifier that achieved the highest accuracy for each data set. We can see that most of the times the Logistic Regression (5 times) achieved the best result. Among the other classifiers, the Multinomial Naïve Bayes classifier, Extra Trees Classifier, and Stochastic Gradient Descent (SGD) Classifier were the most successful twice and the Nearest Centroid was the best only once.

5 CONCLUSION

The goal of the work was to examine whether the content of text documents published on the Internet (specifically Facebook posts) has any connection with stock price movements. We used the values of the S&P 500 Index and divided them into 24 time windows with either growing or decreasing index value trend. Subsequently, we examined (using the classification accuracy) the connection between the documents' content and the trend of the index value in the time window in which was the document published.

Two types of experiments were performed. In the first one, the documents from all 24 windows were put into one data set and we achieved an accuracy of 62%. The second experiment, in which we divided the documents into 12 data sets formed from two neighboring windows, provided better results – the average accuracy was 68%. Moreover,

for three data sets the accuracy was even higher – 72%, 79% and 80%. This means that classifying data from the neighboring windows brings on average better results than using only one data set. This might be related to the concept drift [12] phenomenon which requires a further investigation for this specific domain.

The achieved accuracy around 70% tells us that the posts which companies write on their Facebook pages are partially related to the performance of the whole stock index.

It must be noted that we did not optimize the parameters of used classification algorithms. By doing this, we might achieve a slightly higher accuracy.

This area could be further researched in various directions. Firstly, the analysis may be performed on more types of documents (e.g., newspaper articles). Secondly, the class assigning method may be enriched by using various thresholds of the index value changes (not only 5%). Thirdly, it might be interesting to examine not the whole stock index, but the stock prices of the individual companies instead.

ACKNOWLEDGEMENT

This research was supported by the Czech Science Foundation [grant No. 16-26353S ”Sentiment and its Impact on Stock Markets”] and Internal Grant Agency of Mendel University [No. PEF_DP_2017001 “Searching for semantic information and gaining knowledge from text data streams with new machine learning methods”] and Internal Grant Agency of Mendel University [No. PEF_DP_2017022 “Acquiring, filtering and analyzing of texts for stock markets”].

REFERENCES

[1] Aggarwal, C. C. Data Streams: Models and Algorithms. 2007. Springer

[2] Gama, J. Knowledge discovery from data streams. CRC Press, 2010.

[3] Petrovský, J., Netolický, P. and Dařena, F. Examining Stock Price Movements on Prague Stock Exchange Using Text Classification. International Journal of New Computer Architectures and their Applications (IJNCAA). Vol. 7 No. 1. (2017). pp. 8-13. ISSN 2412-3587.

[4] Sven S. Groth, Jan Muntermann. An intraday market risk management approach based on textual analysis. Decision Support Systems. Volume 50. Issue 4. March 2011. Pages 680-691

[5] Colm Kearney, Sha Liu. Textual sentiment in finance: A survey of methods and models. International Review of Financial Analysis. Volume 33. May 2014. Pages 171–185.

[6] Bollen, J., Mao, H. and Zeng, X. Twitter mood predicts the stock market. Journal of Computational Science. 2011. vol. 2. no. 1. p. 1–8.

[7] Kaplanski, G. and Levy, H. Sentiment and stock prices: The case of aviation disasters. Journal of Financial Economics. 2010. vol. 95. no. 2. p. 174–201.

[8] Lee, H., Surdeanu, M., MacCartney, B. and Jurafsky, D. On the Importance of Text Analysis for Stock Price Prediction. In: LREC. 2014. p. 1170-1175

[9] Maas, A. L., Daly, R. E., Pham, P. T., Huang, D. Learning word vectors for sentiment analysis. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. 2011. vol. 1. p. 142–150.

[10] Go, A., Bhayani, R. and Huang, L. Twitter sentiment classification using distant supervision. CS224N Project Report. Stanford. 2009. vol. 1. p. 12.

[11] Weiss, S. M., Indurkhya, N. and Zhang, T. Fundamentals of Predictive Text Mining. London: Springer. 2010. ISBN 978-1-84996-225-4.

[12] Lindstrom, P, Delany, S. J., Mac Namee, B. (2010) Handling Concept Drift in Text Data Stream Constrained by High Labelling Cost. Florida Artificial Intelligence Research Society Conference (FLAIRS). Florida, 19-21, May.