The Effect Of Context Removal In Sentiment Analysis On News Articles

PEPIJN MESIE, University of Twente, The Netherlands

Sentiment analysis can be used to help decision making on the stock market. By analyzing news articles, a computer can attempt to predict a change in stock price for the company that the news article is about. Through sentence removal in news articles, the influence context has on the performance of sentiment analysis on the articles is examined. The results show no significant change to the accuracy when the context is removed from the paragraphs.

Additional Key Words and Phrases: Sentiment-analysis, Financial news, Context removal, Machine learning, NLP

1 INTRODUCTION

The stock market is difficult to predict due to its instability and fluctuations. Having the ability to accurately predict stock price changes makes it so that an investor can make a lot of money through buying and selling companies' stock. Due to the possibilities and the complex nature of the stock market, there has been a lot of interest in the ability of computers to predict fluctuations in the stock market[3]. For a computer to make these kinds of predictions, a significant amount of data is required. A data source commonly used for this is news articles, as a lot of the value of a company is perceived from them, as shown by the research from Li et al. [9] which shows a correlation between investor sentiment and stock price change. A negative news article can "crash" a company's stock price, and the other way around. This makes news articles a great indicator of stock price changes, but since news articles are not directly interpretable for a computer, a sentiment analysis process is required.

In sentiment analysis, a computer algorithm attempts to identify the sentiment of a piece of text. The algorithm analyzes the text through a model based on a lot of textual information. The process of creating a model that consistently gives the correct sentiment requires a lot of effort and fine-tuning. This process includes the creation of a dataset, which is usually specifically designed around the target of the AI, as sentiment analysis algorithms have [12]. As described in the work by Birjali et al. [1], sentiment analysis can be split up into three main levels, they are the following:

- Aspect-level sentiment analysis
- Sentence-level sentiment analysis
- Document-level sentiment analysis

In this research, a closer look was taken at the effect of the removal of context from paragraphs in sentiment analysis. The research compared the accuracy of the full paragraphs from the news articles and the context-removed sentences from the same articles.

TScIT 37, July 8, 2022, Enschede, The Netherlands

2 PROBLEM STATEMENT

News articles contain a lot of information, but not all of this is relevant for use in the sentiment analysis process. Due to the complicated nature of sentiment analysis, it is difficult to determine what the optimal preprocessing steps are for the data. Research by Krouska et al. [8] has shown that preprocessing significantly influences the quality of sentiment analysis. It has been shown that news articles can be used to predict stock price changes, however, further research needs to be done on the preprocessing of the data to optimize predictions. News articles are not just a headline and contain a lot of information, this context information can be helpful for the reader to create a better understanding of the subject. This information, however, might not be relevant for the sentiment analysis. sentiment analysis algorithms significantly benefit from sentimentally consistent inputs, as these are easier for a computer to label. The extra sentences might be noise for the algorithm and lower the accuracy of the predictions. Therefore, it is important to look at how this "context information" influences the performance of different sentiment analysis methods.

2.1 Research Question

The research question that can be formed for the problem is: "To what extent does context affect the prediction of the sentiment of news articles?" To answer this question, the following sub-question needs to be answered: "What is the accuracy of sentiment analysis on news articles using sentence-level sentiment analysis in comparison to document-level sentiment analysis?"

3 RELATED WORK

News articles have been shown to influence the stock market. García[5] performed research on newspaper articles in the 20th century, his work showed that the articles could be used to predict stock price fluctuations based on the words used in them.

Artificial intelligence has been used to predict stock prices from varying information sources such as previous stock data [11] and news articles [6, 7, 10].

The method described in the work by Khedr et al.[6] utilized Naïve Bayes to predict whether the closing stock price would go up or down, in comparison to the closing price of the previous day. This method achieved high accuracies up to 89.80% by utilizing both previous stock data and financial news articles of a company on a given day, using multiple financial news websites for their dataset.

Li et al. [10] showed that high accuracy can be achieved using RBF kernel SVMs and three-category classification. Through this method, the data is split up on the stock price to either go up, down, or stay the same. By using financial news in combination with multiple sentiment dictionaries, high accuracies were achieved.

Feuerriegel and Prendinger[4] have also shown that applying sentiment analysis to support trading strategies is generally profitable (although not without risks).

Fang et al.[2] have researched the influence of sentence removal on product reviews. They achieved a significantly higher F1 score by filtering the reviews to only contain subjective sentences. A significant difference in this research compared to the research is the way the textual information is filtered, since the research filters on sentences containing company names, instead of the subjectivity of a sentence. Furthermore, the type of textual data is also significantly different, using news articles instead of product reviews. This is significant due to the low domain independence of sentiment analysis algorithms, as described in the research by Xhymshiti [12].

4 RESEARCH METHODS

4.1 Data Collection

Two data sources were used to create the dataset. The research has made use of news articles from The New York Times (NYT) in combination with historical stock prices from Yahoo Finance (YF). The companies were chosen based on their size in the USA, as this is where the news articles have been written. Furthermore, companies from various markets were used for the dataset.

For the news articles, The New York Times' developer API was used to collect as many news articles as possible. This API data contains, but is not limited by, the following information:

- Date
- Abstract
- Snippet
- Lead paragraph
- Section name
- Document type

For all the requests, a filter was applied to ensure that only articles where the "type of material" value was "News" to ensure that only news articles would be collected. The news articles can be filtered to only return the articles in which a certain word is mentioned. This function was used to gather news articles about specific companies. For each company on which data was collected, a Tab-Separated-Values (TSV) file was created containing all news articles' dates and lead paragraphs. To work with the bandwidth limitations set by the API, the code used to collect data was designed to precisely match the limitations set by the NYT. Their API allows for 4000 requests per day, with a maximum of 10 per minute. A delay of 7 seconds was added in between requests to prevent going over this limit. In total, over 200,000 news articles were collected. The companies used for the sentiment analysis can be found in Appendix A.

For the Yahoo Finance data, a Python library named YFinance[1] was used to collect data on the companies chosen for the dataset. This data contains the following information for each day the stock is traded:

- Date
- Opening price
- High
- Low
- Closing price
- Volume
- Dividends
- Stock splits

For this research, the focus was put on the volume (The amount that a company's stock is traded), next to the opening and closing stock price, while using the date to link financial data to the news articles.

4.2 Pre-Processing

4.2.1 Textual data. Two different datasets were created, the first dataset was used for document-level sentiment analysis, where the textual data consists of paragraphs in which news on the company is described. The second dataset contains data from the same news articles but filtered to just contain the specific sentence in which the company is directly mentioned for sentence-level sentiment analysis. For example, the following paragraph:

"For the first time in a decade, Netflix lost subscribers — 200,000 overall in the first three months of the year — the result of shifting economic forces, increasingly fierce competition from other streaming platforms, and the conflict in Ukraine. The Tuesday announcement, plus the company's warning that it expected to lose two million subscribers in the second quarter, sent the stock down 35 percent on Wednesday."[2]

This paragraph was directly used for sentiment analysis in the paragraph dataset. For the second dataset, further filtering was done to only leave the sentence containing the company name. This resulted in the following sentence:

"For the first time in a decade, Netflix lost subscribers — 200,000 overall in the first three months of the year — the result of shifting economic forces, increasingly fierce competition from other streaming platforms, and the conflict in Ukraine."[2]

Since the New York Times API does not return the entire article, only the lead paragraph can be used for the sentiment analysis. This is significant as the API is limited to filtering news articles containing words in the entire body. This means that after initial data collection, the news articles required further filtering to ensure that the dataset only contained paragraphs containing the company name in them. This filtering was done using a Regular Expression (RE). This allows for very strict filtering, for example ensuring the word "Bombshell" is not in the dataset under the assumption that the company "Shell" is mentioned.
The following Regular Expression was used:

```
([^a-zA-Z]|^){company}([^a-zA-Z]|$)}
```

Where {company} is replaced by the company name. This regular expression is only true if the exact company name is surrounded by anything that is not a letter.

4.2.2 Financial data. The financial data also requires some minor changes to better work for the sentiment analysis. Since different companies have significantly different values for the volume, the value is normalized to the minimum and maximum value for that company. This prevents major differences between companies on which the algorithm would overfit.
A field is created to calculate the ratio of the stock price, this is achieved through the following formula:

[1] https://pypi.org/project/yfinance/

[2] https://www.nytimes.com/2022/04/19/business/netflix-earnings-q1.html

$$R = \frac{Closing\ price - Opening\ price}{Opening\ price}$$

If this value is above 0, the stock has risen in price for a given day, if it is below 0, the stock price has gone down. The ratios from the previous 3 days were added to the stock price data as well, as this can show a trend in the stock price, improving the overall performance of the algorithm.

4.2.3 Merging data. Finally, the financial data was attached to the news articles. For each news article, it was checked whether the company's stock was traded on the day that the article is published, (if it is not, the news entry gets dropped from the dataset) and then merged on the date field. The result of this is the dataset containing all the paragraphs in which a company is mentioned. To create the database in which the context is removed which only contains the sentences in which the company is mentioned, more steps are required.
To split the paragraphs up into sentences, the NLTK toolkit was used. This library contains a function that can tokenize a piece of text into individual sentences. This function was applied to all the data points in the paragraph dataset, and from this result, the first sentence in which the company is mentioned was kept.
Another dataset was created that kept all the sentences in which the company was mentioned in the dataset, leading to a dataset with around 13% more data points.

4.3 Labeling

To be able to apply sentiment analysis to the dataset, labels had to be generated. The ratio calculated in the dataset generation was converted to a categorical label. Since The New York Times is a broad news website, a large amount of articles might not be significant financial news for a company. This resulted in a lot of articles that don't correlate to any significant stock price fluctuations. Because of this, a choice was made to split the data up into three labels. These labels represent the stock price going either up, down, or staying the same. To achieve this, a formula was applied to the ratio that determined to what label the data point belongs. The same formula was used as in the work by Li et al.[10].

$$label = \begin{cases} -1 & \text{if } ratio < -threshold \\ 0 & \text{if } threshold > ratio > -threshold \\ 1 & \text{if } ratio > threshold \end{cases}$$

Using this formula, it is possible to measure whether a different threshold changes the overall accuracy of the sentiment analysis. This does change the number of data points for each label, resulting in a very unbalanced dataset as the majority of the news data points have a ratio close to 0, as seen in Figure 1. To manage this issue, the RandomUnderSampler from the SKLearn extension called Imbalanced-Learn was used. This function (randomly) filters out data points to ensure that each label occurs equally in the final sentiment analysis, preventing major overfitting.

The size of the dataset decreases with the increase of the threshold, this is visualized in Figure 2. It was chosen to limit the threshold to below 8% as the size of the dataset would shrink to below 300 data points.

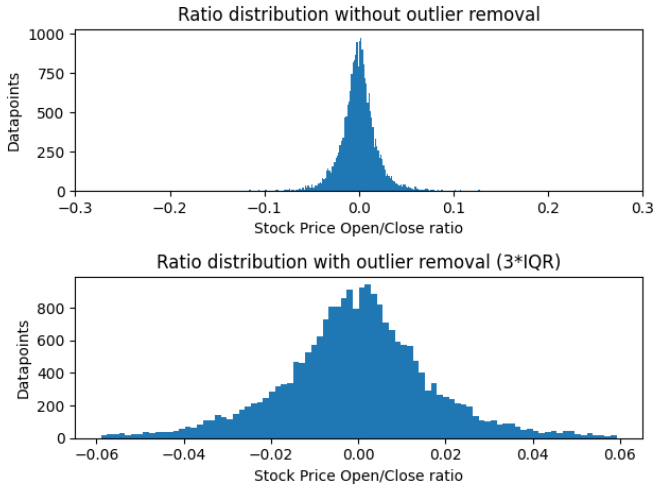

Fig. 1. The distribution of the open-to-close ratio

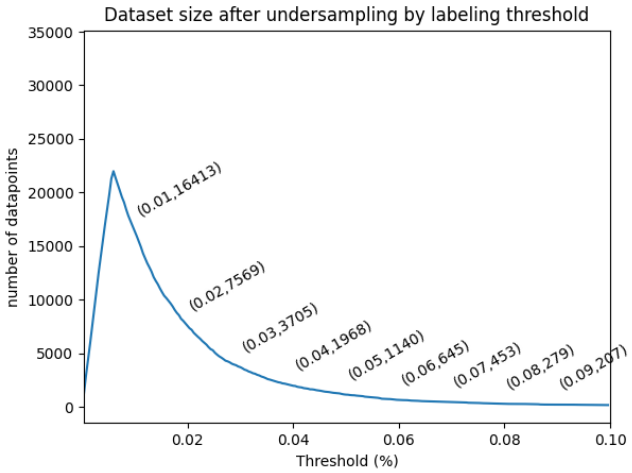

Fig. 2. The dataset size for a given threshold

4.4 Experiment Setup

The result of the previous steps is a 2D matrix containing the textual information, the volume, the stock price ratio of the previous three days, and the label. for the sentiment analysis, the textual information required further processing. For this experiment, the bag-of-words model was followed, converting words into tokens that the computer can process. The steps that were taken to allow for the results to be generated are extensively described in Appendix B.

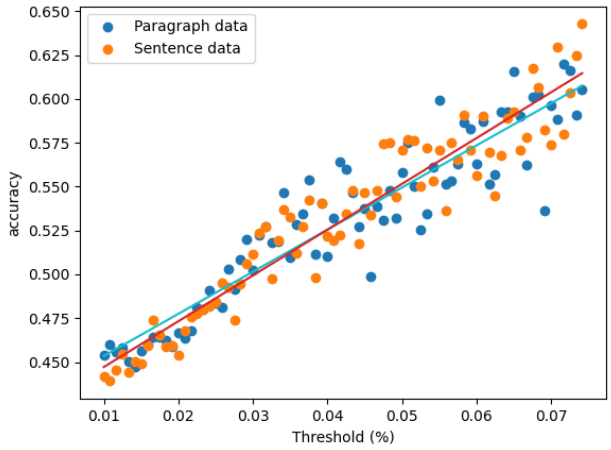

Fig. 3. The sentiment analysis accuracy with an equal amount of datapoints

4.5 Evaluation Methods

For the evaluation metric, accuracy was used. This method works well since, due to the undersampling, the dataset is guaranteed to be balanced.

The evaluation algorithm applies 5-Fold Cross Validation to verify the results and gives the mean of the accuracies from all the folds as the score. The analysis was run for multiple iterations for both datasets, using a variety of thresholds from 0.01 to 0.075.

5 RESULTS

In Figure 3, a scatter-plot is drawn of the accuracy of the datasets for a given threshold, with a best-fit line. The accuracy of the dataset increases when the threshold increases. This is because, with higher thresholds, only the bigger stock changes with their sentimentally clearer accompanying news articles remain labeled as either positive or negative. This should make it easier for the machine learning algorithm to differentiate between the different classes.

The increase in accuracy was similarly visible with higher thresholds, but it was chosen to discard these results, as the undersampling caused the dataset to shrink to a point where overfitting could not be ruled out.

5.1 Statistical Analysis

To find out whether there is any significant difference, the Wilcoxon signed-rank test was used. With a significance of 95%, the null hypothesis is: "Both the sentence and paragraph dataset come from the same distribution". The alternative hypothesis is: "The sentence and paragraph dataset comes from a different distribution". Using 75 paired data points, the probability that the null hypothesis is true is 0.024. This means that the null hypothesis can be discarded, and the alternative hypothesis is confirmed. Calculating the effect size using Cohen's d on this data results in a value of 0.087, from which it can be concluded that the effect size is small, which is also visible in Table 1.

Table 1. The Accuracy of both datasets for different threshold values

Threshold	Sentence Dataset	Paragraph Dataset
0.1	0.457	0.450
0.2	0.472	0.465
0.3	0.517	0.500
0.4	0.542	0.520
0.5	0.516	0.573
0.6	0.570	0.545
0.7	0.604	0.58
Average Acc.	0.533	0.529
Std	0.050	0.047

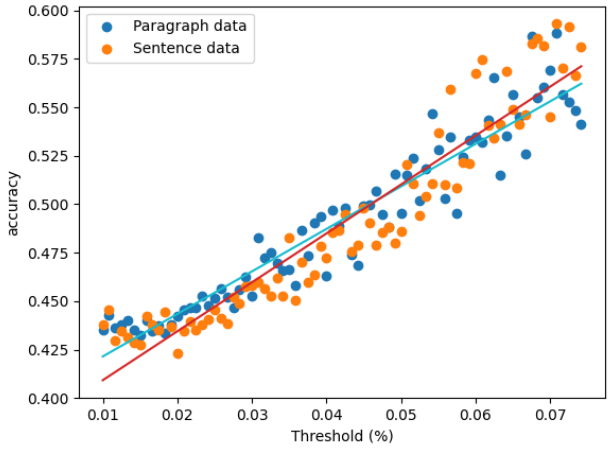

Fig. 4. The sentiment analysis accuracy where the trading volume was left out

5.2 Dataset Size

When splitting paragraphs into sentences for the sentence dataset, some paragraphs resulted in multiple sentences containing the company name. Both sentences were put in the dataset, resulting in more data points in the sentence dataset than in the paragraph dataset. To balance this out, only the first sentence in which a company is mentioned was kept in the final dataset. This was chosen as tests had shown that no significant improvements were made using multiple data points per paragraph.

5.3 Removing Features

In Figure 4, the datasets containing the same amount of data points were used, but the normalized volume and integer representation of the company were left out to ensure that the linear SVM was not overfitting on these features. This resulted in a slightly lower accuracy, but with a reduced spread. Both datasets show similar performance in comparison with each other.

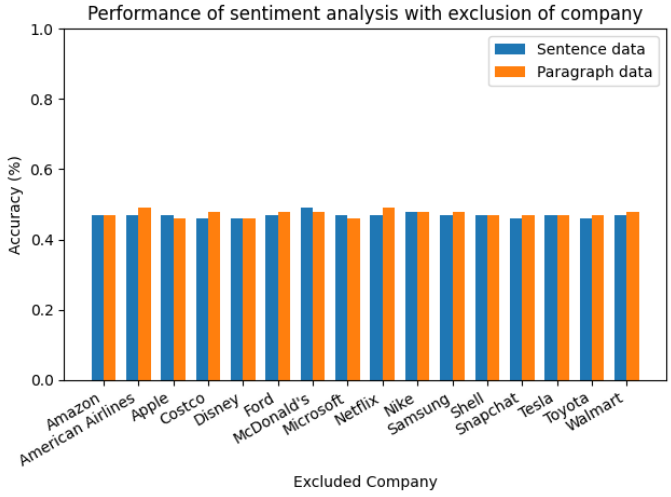

Fig. 5. The accuracy of the sentiment analysis with the exclusion of different companies

5.4 Company Exclusion

In Figure 5, the sentiment analysis was run using 10-Fold Cross-validation using an ngram_range of (1,2), an alpha value of 0.001, and tf-idf. Each bar represents the accuracy of the dataset excluding the company. This graph shows that the individual companies do not significantly affect the accuracy of the algorithm, as the visible variances are all within 1 standard deviation.

Since the dataset shrinks as the threshold increases, tests were run to ensure that there was no overfitting. At a threshold of 0.07, the dataset consists of 453 data points. To confirm that the increase in performance was not due to the dataset shrinking, the analysis was run with a threshold of 0.025, but limiting the dataset to only contain 450 data points. The result of this analysis, in comparison to analyzing the entire dataset, showed a decrease in accuracy of 3-5%, showing that the increased performance is most likely not caused by overfitting.

6 CONCLUSION

6.1 Sub Research Question

The performance of both the sentence-level sentiment analysis and the document-level sentiment analysis are very similar. Both datasets perform weakly on lower thresholds and perform better on news articles that are published on days when a company's stock price changes significantly.

The news articles collected from The New York Times in combination with financial stock data could achieve an accuracy between 45% and 60% for both of the datasets, depending on the threshold for the labeling.

6.2 Main Research Question

The removal of context has shown to have little effect on the accuracy of the sentiment analysis, and there is a significant spread around the best-fit line. Although the distribution of the accuracy of both datasets is (with 95% certainty) not the same, the Effect Size shows that this difference is very small.

7 DISCUSSION

Although the results show a significant improvement over arbitrary guessing, the results achieved with the described methods are a lot lower than that of the work by Li et al. [10]. This is expected, as the goal was not to achieve the highest possible accuracies, but to research the effect of context removal. In their work, they used a financial news website for their data, whereas in this research, a general news source was used. The financial news source only contained news articles of financial importance, whereas, in general news, it is possible for an article to mention a company without it being the main subject. Another difference between the works is that in this research undersampling was used to balance out the dataset, whereas, in Li et al.[10], this was not done as their dataset was better distributed.

The lower performance on small thresholds can be explained by the fact that stock prices always fluctuate. This means that news articles that are not financially significant might not be classified as neutral for lower thresholds. This lowers the ability of the algorithm to distinguish the different classes, resulting in a lower accuracy.

Finally, the data gathered from The New York Times only consisted of the lead paragraph for each article, instead of the full news article. Ideally, using the entire news article would be preferred, as it could lead to different results. It was not possible to research on this within the scope of the research but should be taken into account when interpreting the results.

8 FUTURE WORK

The results show no significant benefit from the removal of context in the described environment, as the performance was very similar. Further research could investigate whether it is possible to achieve a better time complexity and/or reduce the feature space without performance loss using the methods described in this paper.

Due to time constraints, the results were limited to only utilizing a single news source and algorithm. Future work could investigate whether similar results can be achieved using other sentiment analysis methods. Algorithms that use deep learning, such as Bert, might benefit significantly more from the context information in the textual information. These might therefore see more significant differences between the two datasets.

REFERENCES

[1] Marouane Birjali, Mohammed Kasri, and Abderrahim Beni-Hssane. 2021. A comprehensive survey on sentiment analysis: Approaches, challenges and trends. *Knowledge-Based Systems* 226 (8 2021), 107134. https://doi.org/10.1016/J.KNOSYS.2021.107134

[2] Xing Fang and Justin Zhan. 2015. Sentiment analysis using product review data. *Fang and Zhan Journal of Big Data* 2 (2015), 5. https://doi.org/10.1186/s40537-015-0015-2

[3] Fernando G.D.C. Ferreira, Amir H. Gandomi, and Rodrigo T.N. Cardoso. 2021. Artificial Intelligence Applied to Stock Market Trading: A Review. *IEEE Access* 9 (2021), 30898–30917. https://doi.org/10.1109/ACCESS.2021.3058133

[4] Stefan Feuerriegel and Helmut Prendinger. 2016. News-based trading strategies. *Decision Support Systems* 90 (10 2016), 65–74. https://doi.org/10.1016/J.DSS.2016.06.020

[5] Diego García. 2013. Sentiment during Recessions; Sentiment during Recessions. *THE JOURNAL OF FINANCE • LXVIII*, 3 (2013). https://doi.org/10.1111/jofi.12027
[6] Ayman E. Khedr, S.E.Salama, and Nagwa Yaseen. 2017. Predicting Stock Market Behavior using Data Mining Technique and News Sentiment Analysis. *International Journal of Intelligent Systems and Applications* 9, 7 (7 2017), 22–30. https://doi.org/10.5815/ijisa.2017.07.03
[7] Yoosin Kim, Seung Ryul Jeong, and Imran Ghani. 2014. Text opinion mining to analyze news for stock market prediction. *Int. J. Advance. Soft Comput. Appl* 6, 1 (2014), 2074–8523.
[8] Akrivi Krouska, Christos Troussas, and Maria Virvou. 2016. The effect of preprocessing techniques on Twitter sentiment analysis; The effect of preprocessing techniques on Twitter sentiment analysis. (2016). https://doi.org/10.1109/IISA.2016.7785373
[9] Rui Li, Dianzheng Fu, and Zeyu Zheng. 2017. An Analysis of the Correlation between Internet Public Opinion and Stock Market. (2017). https://doi.org/10.1109/ICISCE.2017.41
[10] Xiaodong Li, Haoran Xie, Li Chen, Jianping Wang, and Xiaotie Deng. 2014. News impact on stock price return via sentiment analysis. *Knowledge-Based Systems* 69, 1 (10 2014), 14–23. https://doi.org/10.1016/j.knosys.2014.04.022
[11] Mehar Vijh, Deeksha Chandola, Vinay Anand Tikkiwal, and Arun Kumar. 2020. Stock Closing Price Prediction using Machine Learning Techniques. *Procedia Computer Science* 167 (1 2020), 599–606. https://doi.org/10.1016/J.PROCS.2020.03.326
[12] Meriton Xhymshiti. 2020. Domain independence of Machine Learning and lexicon based methods in sentiment analysis. http://essay.utwente.nl/81995/

A COMPANIES USED IN THE DATASET

News articles and financial data of the following companies were collected:

- Amazon
- American Airlines
- Apple
- Costco
- Disney
- Ford
- McDonald's
- Microsoft
- Netflix
- Nike
- Samsung
- Shell
- Snapchat
- Tesla
- Toyota
- Walmart

B DATA-PROCESSING

The required steps to create the bag-of-words were performed using a SKLearn pipeline. The pipeline consisted of two steps, SKLearn's CountVectorizer turns a text document into a matrix of tokens, where each word is assigned a number. After which, a TfidfTransformer (also from SKLearn) was applied to turn the tokens into term frequencies, which helps to show the importance of a word in a document.
The pipeline was applied in a SKLearn ColumnTransformer, which allowed the pipeline to be applied to just the textual data, leaving the other features as they were.
The ColumnTransformer was then used in another pipeline that also contained the sentiment analysis function. For the sentiment analysis, a Linear Support Vector Machine algorithm was used as a predictor. This algorithm attempts to divide the values by drawing an imaginary line that splits all data points into their respective labels.

B.0.1 Optimization. To find the optimal parameters for the analysis, the sentiment analysis function is used in SKLearn's GridSearchCV. This function performs an exhaustive search over a group of predefined parameters to find the settings that result in the highest accuracy. For this research, 3 parameters were chosen:
The first parameter is the Ngram_range. This is a parameter for the CountVectorizer, which determines the number of words that are grouped. This can lead to clearer differentiation between positive and negative results, but can also result in the word-group occurrences being too low to accurately perform the sentiment analysis. The N-gram ranges chosen were (1,1) and (1,2), keeping all words separately and grouping them in pairs, respectively.
The second parameter is whether to use Inverse normalized Term Frequency (tf) or Term Frequency-Inverse Document Frequency (tf-idf) representation. Term Frequency converts the tokens into the number of occurrences of the word. Inverse Document Frequency is the number of occurrences of a token in the entire dataset.

Whenever a word is very frequent in the dataset, words such as "and" and "but", this number becomes smaller. By using tf-idf instead of just tf, it is possible to decrease the influence these commonly used words have on the sentiment analysis, as they give very little information compared to less frequently occurring words.
The final parameter used in the GridSearchCV is the alpha value. This value is used to determine how much regularization is to be applied, which can correct for outliers. The parameters used are 1e-3 down to 1e-9.