BACHELOR'S DEGREE THESIS

# Degree in Data Science and Engineering

**Title: Sentiment Analysis in Finance**

**Author: Inés Broto Clemente**

**Advisors:** Ariel Duarte-López, PhD. and Argimiro Arratia Quesada, PhD.

**Month and year: January, 2023**

**Facultat d'Informàtica de Barcelona**
**Facultat de Matemàtiques i Estadística**
**Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona**

Universitat Politècnica de Catalunya
Facultat d'Informàtica de Barcelona
Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona
Facultat de Matemàtiques i Estadística

Degree in Data Science and Engineering
Bachelor's Degree Thesis

# Sentiment Analysis in Finance

**Inés Broto Clemente**

Supervised by
Ariel Duarte-López, PhD. and
Argimiro Arratia Quesada, PhD.

January, 2023

En primer lloc, gràcies a totes aquelles persones que han compartit amb mi la creació d'aquest treball. En especial:

Gràcies al meu pare, la meva mare i al meu germà pel recolzament incondicional en tot el que faig.

Gràcies a la Cande per haver fet un esforç quan hem conviscut juntes estant en moments vitals diferents i als amics de la uni per acompanyar-me durant tota aquesta etapa.

Gracias a Ariel Duarte López por todos los consejos, el apoyo y la confianza durante todo el desarrollo de este proyecto; por haberme dado la oportunidad de escribirlo a su lado y por haber hecho de mi primer trabajo un lugar agradable y cómodo donde aprender y encontrar mi camino. Gracias también a Argimiro Arratia por el seguimiento, las correcciones y por estar abierto a cualquier propuesta que hemos realizado.

Finally, thanks to the company *Acuity Trading S. L.* for allowing me to work with them and providing me with the data and part of the infrastructure necessary for project development. Also, thanks for affecting me with an innovative spirit and an exciting view of the work. A special mention to the Signal Center department, who have had enormous patience in introducing me to the world of technical analysis and with whom working has been a great pleasure.

## Abstract

Data runs the world. Anyone who is even slightly interested in having an impact and generating a valuable product should take this into account. The recent developments in Natural Language Processing have attracted the attention of a large number of practitioners. One of the most relevant applications in this field is the sentiment analysis of a given text. It is a valuable and challenging task that many companies in various sectors include as part of their data science pipeline. For example, in the field of Finance, the most innovative companies work to obtain sentiment indicators by analyzing textual data, either to predict market movements, guide trading strategies, or sell that valuable information to third parties. Using data from 20 well-known equities with a large capitalization margin from the U.S. market, this work aims to be a proof of concept of two different Sentiment Analysis methodologies in the financial sector. The first methodology computes sentiment scores from article text conditioned to the stocks' price return values. The second methodology studies the semantic and syntactic relationships between words to calculate the sentiment linked to a term of interest. Both methods represent a valuable and opposite starting point for sentiment score computation. Finally, a numerical study of the results is carried out considering correlation values and causality test performance. Also, this work introduces a framework for the simulation of trading strategies guided by the acquired scores.



### Keywords

Sentiment Analysis, Finance, Machine Learning, Natural Language Processing

## Resumen

Los datos gobiernan el mundo. Cualquier persona que quiera diseñar un producto valioso y tener impacto debe tener esto en cuenta. Los recientes avances en el campo del Procesamiento de Lenguaje Natural han llamado la atención de un gran número de investigadores. Una de las aplicaciones más relevantes en este campo es el análisis de sentimientos a partir de un texto determinado. Esta es una tarea desafiante que muchas empresas de diversos sectores incluyen en sus líneas de trabajo. Por ejemplo, en el sector financiero, las empresas más innovadoras trabajan para obtener indicadores de sentimiento mediante el análisis de datos textuales, ya sea para predecir los movimientos del mercado, guiar estrategias de inversión o vender esta valiosa información a terceros. Utilizando datos de 20 conocidas compañías con un amplio margen de capitalización del mercado estadounidense, este trabajo pretende ser una prueba de concepto de dos metodologías distintas de análisis de sentimientos en el sector financiero. La primera metodología calcula las puntuaciones de sentimiento a partir del texto de un artículo condicionado a los valores de retorno del precio de las acciones. La segunda metodología estudia las relaciones semánticas y sintácticas entre palabras para calcular el sentimiento vinculado a un término de interés. Ambos métodos representan un punto de partida interesante y opuesto para el cálculo de sentimiento. Por último, se realiza un estudio numérico de los resultados considerando los valores de correlación y un test de causalidad. Asimismo, este trabajo introduce un marco para la simulación de estrategias de inversión guiadas por las puntuaciones de sentimiento calculadas.

### Palabras clave

Análisis de sentimiento, Finanzas, Aprendizaje Automático, Procesamiento de Lenguaje Natural

## Resum

Les dades regeixen el món. Qualsevol persona interessada a dissenyar un producte valuós i tenir impacte ha de tenir això en compte. Els avenços recents en el camp del Processament de Llenguatge Natural han cridat l'atenció d'un gran nombre d'investigadors. Una de les aplicacions més rellevants en aquest camp és l'anàlisi de sentiments a partir d'un text determinat. Aquesta és una tasca desafiant que moltes empreses de diversos sectors inclouen com a part de les seves línees de treball en ciència de dades. Per exemple, en el sector financer, les empreses més innovadores treballen per obtenir indicadors de sentiment mitjançant l'anàlisi de dades textuals, ja sigui per predir els moviments del mercat, guiar estratègies d'inversió o vendre aquesta valuosa informació a tercers. Utilitzant dades de 20 conegudes companyies amb un ampli marge de capitalització del mercat nord-americà, aquest treball pretén ser una prova de concepte de dues metodologies diferents d'anàlisi de sentiments en el sector financer. La primera metodologia calcula les puntuacions de sentiment a partir del text d'un article condicionat als valors de retorn del preu de les accions. La segona metodologia estudia les relacions semàntiques i sintàctiques entre paraules per calcular el sentiment vinculat a un terme d'interès. Tots dos mètodes representen un punt de partida interessant i oposat per al càlcul de sentiment. Finalment, es realitza un estudi numèric dels resultats considerant els valors de correlació i un test de causalitat. Així mateix, aquest treball introdueix un marc per a la simulació d'estratègies d'inversió guiades per les puntuacions de sentiment calculades.



### Paraules clau

Anàlisi de sentiment, Finances, Aprenentatge Automàtic, Processament de Llenguatge Natural

# Contents

# 1. Introduction

Natural language processing (NLP) is an area of research that explores how computers can be used to understand and manipulate natural language. As a sub-field of computer science, it combines Linguistics and Machine Learning (ML) to provide computers with the ability to process and analyze textual data sets. Its main goal is to make a computer capable of understanding the contents of documents, including the contextual nuances of the language within them. This technology can accurately extract information and insights from textual data and categorize and organize such information.

One of the most popular and actively researched tasks within the NLP field is Sentiment Analysis (SA) (or Opinion Mining (OM)). The work by Medhat et al. [2014] defines SA as "the computational study of people's opinions, attitudes and emotions toward an entity where the entity can represent anything an opinion can be formed on". The two expressions SA or OM are usually used interchangeably; however, there exists a slight difference in between them. OM extracts and analyzes people's opinion about an entity while SA identifies the sentiment expressed in a text and then analyzes it. This work focuses on the latter task.

In general terms, SA can be considered a classification process. When talking about sentiment classification techniques, one can first distinguish between two main approaches: ML and lexicon-based. The former aims to be able to understand statistical patterns that can be found in text data-building classifiers. It relies on many different ML algorithms and defines the problem as having a set of records (text units) where each record should be labeled to a class. Depending on the existence of labeled training documents there are supervised ML approaches and unsupervised ML approaches. Dridi et al. [2019] proposes a supervised approach that is learned by using several feature sets, consisting of lexical features, semantic features, and a combination of lexical and semantic features directly extracted from text. Playing with annotated data, Akhtar et al. [2020] uses a multi-layer perceptron for predicting the degree of intensity for emotion and sentiment by combining the outputs of deep learning and classical feature-based models. Chunli et al. [2022] presents a novel semantic and syntactic-enhanced neural model that aims to predict sentiment intensity concerning a target (company or stock symbol) in a text considering its context. Their model incorporates a self-attention mechanism to capture semantic contextual information and an edge-enhanced graph convolutional network to aggregate node-to-node features.

On the other hand, a lexicon-based approach involves computing sentiment orientation on text units directly from the semantic orientations of the words in it, so they rely on dictionaries of words or phrases (the opinion lexicon) annotated with the words' semantic orientation or polarity. Dictionaries for lexicon-based approaches can be created manually or automatically. Even though there are dictionaries for general domains (*SentiWordNet* [Baccianella et al., 2010] or *SenticNet* [Cambria et al., 2010]), it is common to find use cases defining dictionaries in specific domains. For instance, Consoli et al. [2022a] introduces *SentiBigNomics*, which provides users with a sentiment lexicon for the economic and financial domains. Section 2.2 gives a more detailed analysis of this work. In Hota et al. [2021], the authors tune a general lexicon and rule-based SA tool for sentiment extraction in social media. The system computes sentiment scores for COVID-19 tweets by summing the valence scores of each word in the lexicon, adjusting them according to the rules, and applying a normalization.

There are three main granularity levels in SA: document-level, sentence-level, and entity/aspect-level. Since sentences can be thought of as short documents, there is no fundamental difference between document-level and sentence-level SA. Choosing between one or the other will be conditioned on the methodology applied during the analysis. Both granularity levels aim to classify an opinion information unit (either a complete document or a sentence) as expressing a positive or negative opinion, and both of them are considered coarse-grained sentiment analysis. As an example, the work by Ke et al. [2019] sticks to this granularity level.

Nevertheless, classifying text at the document or sentence-level does not provide the detailed opinions on an entity or its characteristics which is needed in many applications. For instance, "The voice quality of this phone is not good, but the battery life is long" turns out to be a pretty challenging sentence for a coarse-grained system that focuses on the sentiment related to the entity *phone*. To get more accurate results, we need to jump into the aspect-level feature selection to perform fine-grained sentiment analysis. Fine-grained SA aims to classify the sentiment concerning specific aspects of entities. The first step is to identify the entities and their aspect(s), then to categorize the polarity of opinion expressions for the detected aspect(s), and finally (and optionally), to aggregate the sentiments extracted for each of the aspects linked to an entity. The already mentioned works by Akhtar et al. [2020], Dridi et al. [2019] and Consoli et al. [2022a] are good examples of the applicability of fine-grained SA.

In the financial domain, applying such methodologies opens the door to a wide set of applications. Most of the research focuses on the stock return prediction problem, which speaks for his name: taking advantage of the assumed correlation between sentiment scores and the upcoming market moves to predict stock values. However, SA in the finance domain can also potentially bring opinion indicators that can guide investment strategies attempting to replicate human decision-making behavior, which always includes an emotional component.

Due to the explosion of available textual data, an increase in computing power, and the huge research and rapid advances in the NLP and text mining field, many companies related to the analysis of financial news have emerged in recent years. An example of it is Acuity Trading S. L., a news technology company that focuses on news analytic technology and NLP to offer data through visualizations and integrated flows with which investors can take advantage of this valuable data source.

The main goal of this work is to perform a comparative analysis of two different SA approaches for sentiment extraction from financial news. The work by Ke et al. [2019] introduces a supervised text-mining methodology that extracts sentiment information from news articles to predict asset returns. They use their document-based (coarse-grained) supervised learning framework over Dow Jones Newswires (DJN) articles to compute sentiment scores for the entities or assets that the articles talk about. Their proposal constructs sentiment scores tailored to the return prediction problem using the returns data itself to design a dictionary on which to base the scoring of new articles. This methodology has already been studied in Marcel Portas' Final Master's Thesis, supervised by Prof. Argimiro Arratia, and a sentence-based modification of their implementation will be the first methodology this project will be focusing on. On the other hand, Consoli et al. [2022a] proposes an unsupervised Fine-Grained Aspect-Based SA system that aims to identify the sentiment associated with a specific topic of interest in each sentence of a document. It relies on a set of semantic polarity rules that allow understanding the origin of sentiment, and aggregating it accordingly to the syntax relationships between words. An adapted version of its implementation is the second main contribution of this work.

To compare both methodologies, we compute sentiment scores related to 20 well-known companies with a large capitalization margin from the U.S. market. As for the textual data, we use a corpus of financial news published by DJN in the period from 2017 to 2020. Besides, real market prices are needed for the construction of the first model as well as for the trading strategy implementation and comparison metrics. In this analysis, we have used daily price data provided by *Dukascopy Swiss Banking Group* [SA, 2022].

From the statistical point of view, this work puts in place a descriptive and correlation study framework to analyze the relationships between the obtained sentiment scores and the real market prices. Supporting the obtained results with meaningful correlation patterns or at least being able to find some explainability or causality between different data time series is expected. Furthermore, to prove the value, applicability, and usefulness of the obtained sentiment scores, a trading strategy based on the obtained indicators is implemented and simulated using BackTrader [2022], a Python library for backtesting purposes.

The document is structured the following way: Section 2 presents an extended description of the two methodologies studied within this project and the variations concerning previous work. Section 3 details the methodology with a system diagram and a discussion of the steps that have been carried out, as well as the specifics of software and resources. Section 4 evaluates the obtained results. Finally, Section 5 concludes this work and outlines possible future research lines. At the end, an Appendix includes extended results.

# 2. Previous works

In this section, we introduce the two different SA approaches we are aiming to test in this work. As discussed, the first one proposes a supervised ML model that scores news articles as positive or negative conditioned on stock price returns without the use of any pre-computed dictionary [Ke et al., 2019]. As already mentioned in Porta Vallés [2020] Master's Thesis, this method proceeds in three steps: 1) isolating a list of sentiment-charged terms via predictive screening, 2) assigning sentiment weights to these words via topic modeling, and 3) aggregating terms into an article-level sentiment score via penalized likelihood. On the other hand, the second approach takes advantage of the semantic and syntactic relationships among words to retrieve sentiment scores [Consoli et al., 2022a]. It is an unsupervised approach that relies on a detailed set of semantic polarity rules to understand the origin of the sentiment. The scores are aggregated using a rule of thumb that propagates sentiment scores within a sentence taking into account semantic connotations.

Both methodologies aim to collect and automate the extraction of sentiment from a text to predict sentiment scores that correlate to the sentiment they would produce to an expert reader. The chosen labeling data (return values) in the first case, and the use of a specialized lexicon in the second case, place both methodologies in SA for economic and financial applications. However, keep in mind that both implementations are flexible enough to be used in other fields.

## 2.1 SESTM: Sentiment Extraction via Screening and Topic Modeling

In Ke et al. [2019] the authors propose a model-based approach that aims to understand the sentimental structure of a text corpus without the need for a pre-existing dictionary. This method is based on the assumption that the sentiment-charged words in news stories and returns of mentioned assets are linked. Unlike other modeling approaches, this methodology uses this assumption to guide the modeling step. Then, price return values become labels for text data that are classified into positive sentiment-charged or negative sentiment-charged text data depending on the return's polarity. At that point, one can get a dictionary of positive and negative words and use that dictionary to estimate sentiment scores for new articles. The proposed model is named SESTM (Sentiment Extraction via Screening and Topic Modeling) and consists of three steps: 1) screening for the set of sentiment-charged words. 2) estimating the positive and negative sentiment topics weights and, 3) using penalized maximum likelihood to estimate sentiment scores for new articles.

For the sake of comprehension, we will follow the same notation as Ke et al. [2019]. Consider a collection of $n$ news articles and a dictionary of $m$ words. $D$ is the document-term matrix with dimensions $m \times n$ where each vector $d_i \in \mathbb{R}_+^m$ stores the word counts in article $i$, so that $d_{j,i}$ is the number of times word $j$ appears in article $i$. We define $D_{[s]}$ as a subset of rows from $D$ that contains the rows whose indices are listed in the set $S$ (sentiment-charged words). Then, $d_{[s],i}$ denotes the column vector corresponding to the $i^{th}$ column of $D_{[s]}$. For each article (or sentence) $i$, we have an associated stock return $y_i$ which is a 3-day return of the mentioned stock price on the publication date of the article.

Each article possesses a sentiment score $p_i \in [0,1]$, with $p_i = 1$ when the document sentiment is maximally positive, $p_i = 0$ when maximally negative and $p_i = 0.5$ when it is neutral. Assuming that sentiment score $p_i$ links returns $y_i$ with the word vector $d_i$, which means that $p_i$ serves as a sufficient statistic for the influence of the article on the stock return ($d_i$ and $y_i$ are independent given $p_i$), we need at least two components to fully specify the data generating process: one that governs the distribution of the stock return $y_i$ given $p_i$, and another one that governs the article word count vector $d_i$ given $p_i$.

For the conditional distribution of returns given sentiment scores we assume:

$$P(sgn(y_i) = 1) = g(p_i), \text{for a monotone increasing function g}(\cdot), \tag{1}$$

where $sgn(x)$ is the sign function that returns 1 if $x > 0$ and -1 otherwise. With this, we state that the higher the sentiment score, the higher the probability of realizing a positive return.

For the conditional distribution of word counts in an article we define the following partition:

$$\{1, 2, ..., m\} = S \cup N, \tag{2}$$

where $S$ is the set of indices for sentiment-charged words, $N$ is the set of indices for sentiment-neutral words, and $1, 2, ..., m$ is the set of indices for all words in the vocabulary. $S$ and $N$ have dimensions $|S|$ and $m - |S|$ respectively and that will be threshed in the screening (first) step. Since $d_{[N],i}$ only contain neutral words we will leave it unmodeled and assume it is independent of the vector of interest $d_{[S],i}$ for which we assume a mixture multinomial distribution generation of the form:

$$d_{[S],i} \sim Multinomial(s_i, p_i O_+ + (1 - p_i) O_-), \tag{3}$$

where $s_i$ is the total count of sentiment-charged words in article $i$ (determines the scale of the multinomial) and $O_\pm$ are a probability distributions over words. $O_+$ is a "positive sentiment topic" and describes expected word frequencies in maximally positive sentiment article ($p_i = 1$). Likewise, $O_-$ is a "negative sentiment topic" and describes expected word frequencies in a maximally negative articles ($p_i = 0$). Then, at word level, a word $j$ is "positive" if the $j^{th}$ entry of $O_+$-$O_-$ is positive i.e. if the word has a larger weight in the positive sentiment topic than in the negative sentiment topic and, on the contrary, a word $j$ is "negative" if the $j^{th}$ entry of $O_+$-$O_-$ is negative.

In the following sections, we give a detailed description of the three steps that conform to this methodology.

### 2.1.1 Screening for Sentiment-Charged Words

The first step selects the vocabulary from which we will learn the sentiment topics. Note that estimating a topic model for the entire vocabulary (sentiment-charged and sentiment-neutral terms) is not only a very challenging task but also would deal with high computational costs and a high signal-noise ratio since, in this context, sentiment-neutral words act as noise. Instead, the authors propose to isolate a set of sentiment-charged words and then estimate a topic model only taking into account this subset. To do this, they use a supervised approach based on the assumption that sentiment-charged words will appear when no-neutral stock returns happen. Intuitively, if a word frequently occurs in articles related to positive stock return values, this word is likely to contribute as a positive word. The screening procedure calculates the frequency with which word $j$ occurs with positive returns using what in statistical literature is known as marginal screening, defined as:

$$f_j = \frac{\text{count of word } j \text{ in article with } sgn(\mathrm{y}) = +1}{\text{count of word } j \text{ in all articles}}. \tag{4}$$

Now, proper thresholds need to be set to thresh sentiment-charged terms from neutral-sentiment terms. Let $\hat{\pi}$ be the fraction of articles tagged with a positive return that we expect to be around $\frac{1}{2}$. For a neutral-sentiment word, we expect to see $f_j \approx \hat{\pi}$, since the assumption is that its occurrence is uncorrelated with the sign of returns. Therefore, we need to set an upper and lower threshold ($\alpha_+$ and $\alpha_-$ respectively) to define sentiment-charged words based on $f_j$:

- terms with $f_j > \hat{\pi} + \alpha_+$ are positive sentiment terms,
- terms with $f_j < \hat{\pi} - \alpha_-$ are negative sentiment terms.

Finally, a third threshold is necessary to prevent the screening from including very infrequent terms, since these include very noisy information about their relevance to sentiment. In Porta Vallés [2020], the author adds another boundary to very frequent words, since they also do not provide valuable information. Let $k_j$ be the count of word $j$ in all articles (i.e. the denominator of $f_j$, which is denoted as $k_j$), $k_{lower}$ and $k_{upper}$ are the restrictions on the count of articles including word $j$. Words are restricted to the ones that comply $k_j > k_{lower} \cap k_j < k_{upper}$.

All in all, the final estimate set $S$ is defined by:

$$\widehat{S} = \{j : f_j \geq \hat{\pi} + \alpha_+ \cup f_j \leq \hat{\pi} - \alpha_-\} \cap \{j : k_j > k_{lower} \cap k_j < k_{upper}\}, \tag{5}$$

where threshold $\alpha_+$, $\alpha_-$, $k_{lower}$ and $k_{upper}$ are hyper-parameters that can be tuned via cross-validation.

### 2.1.2 Learning Sentiment Topics

The next step once we threshed the relevant-words list $S$ is to fit a two-topic model (positive $+$ and negative $-$) to the sentiment-charged word counts. We can gather the two topic models in a matrix $O = [O_+, O_-]$, which determines the expected counts of the sentiment-charged words in each article. Since $O_\pm$ captures information on both the frequency of words and their sentiment, it is helpful to reorganize the topic vectors in frequency ($F$) and tone ($T$) vectors:

$$F = \frac{1}{2}(O_+ + O_-) \text{ and } T = \frac{1}{2}(O_+ - O_-). \tag{6}$$

If a word has a larger value in $F$, it appears more frequently overall and if a word has a larger value in $T$, its sentiment is more positive.

In this context, each document is associated with a return value and the assumption is that it contains information about the document's sentiment. Hence, returns serve as training labels and we can take a supervised learning approach to estimate $O$ (or equivalently, to estimate F and T). Considering $p_i$ ($i^{th}$

article's sentiment score), $d_{i,[S]}$ ($i^{th}$ article's sentiment-charged words counts) and $s_i$ (total $i^{th}$ article's sentiment-charged words), one can define a vector of sentiment-charged words frequencies as:

$$E\tilde{d}_{i,[S]} = E\frac{d_{i,[S]}}{s_i} = p_i O_+ + (1 - p_i)\, O_-, \tag{7}$$

or, in matrix form:

$$E\widetilde{D} = OW, \quad \text{where} \quad W = \begin{bmatrix} p_1 & \cdots & p_n \\ 1 - p_1 & \cdots & 1 - p_n \end{bmatrix}, \quad \text{and} \quad \widetilde{D} = \left[\tilde{d}_1, \tilde{d}_2, \ldots, \tilde{d}_n\right]. \tag{8}$$

Ke et al. [2019] proposes a simple approach to estimate $O$ via regression of $\widetilde{D}$ on $W$. To estimate $W$, one uses the standardized ranks of returns as sentiment scores for all articles in the training sample. For each article $i$ in the training sample, where $y$ is the vector of returns sorted in ascending order:

$$\hat{p}_i = \frac{\text{rank of } y_i \text{ in } \{y_l\}_{l=1}^{n}}{n}. \tag{9}$$

Given $\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n$ one obtain the estimator of $O$, $\widehat{O}$, as:

$$\widehat{O} = \widehat{D}\widehat{W}' \left(\widehat{W}\widehat{W}'\right)^{-1}, \quad \text{where} \quad \widehat{W} = \begin{bmatrix} \widehat{p}_1 & \widehat{p}_2 & \cdots & \widehat{p}_n \\ 1 - \widehat{p}_1 & 1 - \widehat{p}_2 & \cdots & 1 - \widehat{p}_n \end{bmatrix}. \tag{10}$$

Since $\widehat{O}$ may have negative entries, it is necessary to set them to zero and re-normalize each column to have a unit $\ell^1$-norm.

### 2.1.3 Scoring New Articles

Considering the definition in equation (3), a sentiment score for a new article ($\hat{p}$) is estimated using maximum likelihood estimation:

$$\widehat{p} = \arg\max_{p \in [0,1]} \left\{ \widehat{s}^{-1} \sum_{j=1}^{\widehat{s}} d_j \log\left(p\widehat{O}_{+,j} + (1 - p)\widehat{O}_{-,j}\right) + \lambda \log(p(1 - p)) \right\} \tag{11}$$

The author adds a penalty term to help cope with the limited number of observations and the low signal-to-noise ratio inherent to sentiment learning. The penalty term is $\lambda \log(p_i(1 - p_i))$. Imposing a penalty shrinks the estimate toward a neutral sentiment score of $1/2$, where the amount of shrinkage depends on the magnitude of $\lambda$ tuning parameter ($\lambda > 0$).

### 2.1.4 Project variation

The original work restricts the number of articles used in the modeling phase to those that only speak about one company. Otherwise, it would not be clear which return to use when labeling an article that speaks about more than one company. However, a huge amount of data is lost when applying this restriction. The implemetation used in this work jumps from the article-level to the sentence-level when pre-processing the data. If we split the articles into sentences, we get smaller documents that can be fed into the modeling pipeline in the same way as before. This idea allows more data can be taken into account since sentences are less likely to speak about different companies than the complete news story. We have used spaCy's Named Entity Recognition (NER) component to thresh the sentences where entities appeared and to get each entities' name. Linking spaCy's predicted entities to companies' identifiers is a challenging task that will be discussed in Section 3.3. Once the sentiment scores for each sentence are available, we aggregate them to the document-level granularity we shall use to compute daily scores. Choosing the proper aggregation metric is not trivial and will be discussed in Section 3.6.

## 2.2 FiGAS: Fine-Grained, Aspect-Based Sentiment Analysis

Even though Ke et al. [2019] shows promising results, it has some limitations due to its methodology. First of all, although the amount of data has increased when linking companies to sentences instead of to complete articles, the whole sentence structure is lost when building the Bag of Words model. The words that appear next to the company name have the same relevance as the ones that appear three or four words away and any semantic relationship is considered. Moreover, words preceded by valence shifters [1] do not have a special treatment. For example, when analyzing the sentences "Apple's new product is good" and "Apple's new product is not good", both *good* apparitions fall into the same word count although they clearly show opposite sentiments and may be related to opposite return values. Valence shifters work similarly but not necessarily negating the meaning of the following word but also emphasizing it or exaggerating it. Still, "Apple's new product is good" and "Apple's new product is very good" have noticeably different emotional meanings, but our system is not making any difference between the first and the second *good*.

The approach proposed by Consoli et al. [2022a] performs a fine-grained SA allowing the system to track content based on the syntactic and semantic relationship among the words. It also considers the existing of valence shifters when computing the scores. This method focus on a specific Term (or Token) of Interest (ToI) and computes the sentiment score of the sentence using the words related to a specific ToI, rather than the overall sentence sentiment. This task can be split in two parts:

1. Select the words that are related to the selected topic or company.
2. Aggregate all the sentiment scores of the selected words using a sentiment dictionary.

### 2.2.1 Methodology

In the original methodology, the ToI focuses on specific economic concepts, for instance, gross domestic product. In the first place, a set of synonyms are derived from the proposed economic concept and after that, a rule based-procedure extracts information from all the synonym concepts identified in the text. The sentiment extraction is based on applying the set of rules that are discussed in this report. These rules intrinsically define a hierarchy among the words: the ToI remains in the first position and it is followed by a path of words that comply with the semantic rules. Each of the words in the path is susceptible to adding an emotional meaningful score to the sentiment related to the ToI, that's why each word lemma will be consulted to the sentiBigNomics dictionary [Consoli et al., 2022b] to retrieve its sentiment score. After that, sentiment scores are aggregated following the words path backward: from the last matching-rule word to the ToI, to compute its final sentiment score.

[1]**Valence shifter:** Linguistic contextual item that can increase, reduce or neutralise the prior polarity of a word. For instance *very*, *more*, *barely*,...See that negations are usually included within this class of words.

The algorithm was developed in Python, v3.7.6, and will run in Python, v3.9.7. The source code is accessible online on the authors' GitHub page [Consoli et al., 2022c]. The extraction process is based on the linguistic features of the spaCy Python library [Honnibal and Montani, 2017]. In particular, the pipeline relies on spaCy's en_core_web_lg language model, which represents an English multi-task Convolutional Neural Network trained on OntoNotes and with GloVe vectors trained on Common Crawl. The library also assigns word vectors, context-specific token vectors, part-of-speech (POS) tags, Dependency Parsing (DEP), and recognizes named entities. Details on the methodology are summarised below:

– *Tokenisation:* The input text is split into words (spaCy tokens). Only when one of the economic concepts of interest (ToI and synonyms) is found in the text, rule checking is triggered. In this way, the authors can isolate the sentences and the portions of text that are of interest.

– *Lemmatisation:* The uninflected forms of the words (lemmas) are used to retrieve sentiment from the BigNomics dictionary. SpaCy includes a Lemmatizer component that matches each token with its lemma.

– *Location detection:* (optional) This feature can be useful when the user is interested in exploring the sentiment expressed for an economic concept in a specific location. It assigns the most frequently named entity location detected in the text as the location the text is talking about. Since we are not interested in this feature, it won't be used in the main pipeline.

– *Named Entity Recognition (NER):* NER is an information extraction procedure that identifies and labels named entities present in unstructured text. This algorithm's NER procedure is based on the model trained by spaCy v3.4 that provides an accuracy of around 86%. In the original version, NER is used for Location (or eventually other entity labels) detection for filtering results related to the same economic concept but to different locations or persons. However, this project goes one step further and directly uses it for detecting Organizations and Companies as the first step of the main pipeline. So, only text related to entities and potentially sentiment-charged is processed. Details on this will be presented in Section 3.3.

– *POS tagging:* After the tokenization step, POS parsing and tagging are needed since the semantic rules will be partly based on these tags. SpaCy offers a POS component that assigns classes to each term based on the Universal POS tags categories [UD, 2022b] proposed by the Universal Dependencies (UD) framework [UD, 2022a]. The statistical model used by spaCy to predict the POS tags shows a really good level of accuracy for the task (declared accuracy in v3.4 of 97.06%).

– *DEP:* DEP establishes the relationship between a word and the other terms in the sentence that modifiy its meaning. The semantic rules are also based on the dependency relations between words so the algorithm needs to have access to these tags. SpaCy v3.4 includes a DEP component that shows accuracy levels of 89.76%, and also relies on the UD framework to classify the parsed dependencies [UD, 2022a]. After a ToI is found in the input text, the proposed rules are checked to construct the set of words that will be taken into account when computing the sentiment score (only the words that match one of the rules will be taken into account). Details on the syntax rules are provided in Section 2.2.2.

– *Tense detection:* (optional) This feature can be useful when the user is interested in exploring the sentiment expressed for an economic concept in a specific verbal tense (i.e., past, present, future). SpaCy provides further tags than the basic POS tagging we all learn at school, as it will be shown

| POS | TAG | Description |
|---|---|---|
| CONJ | CC | Conjunction, coordinating |
| ADP | IN | Conjunction, subordinating or preposition |
| ADJ | JJ | Adjective |
| ADJ | JJR | Adjective, comparative |
| ADJ | JJS | Adjective, superlative |
| VERB | MD | Verb, modal auxiliary |
| NOUN | NN | Noun, singular or mass |
| PROPN | NNP | Noun, proper singular |
| PROPN | NNPS | Noun, proper plural |
| NOUN | NNS | Noun, plural |
| ADV | RBR | Adverb, comparative |
| ADV | RBS | Adverb, superlative |
| VERB | VB | Verb, base form |
| VERB | VBD | Verb, past tense |
| VERB | VBG | Verb, gerund or present participle |
| VERB | VBN | Verb, past participle |
| VERB | VBP | Verb, non-3rd person singular present |
| VERB | VBZ | Verb, 3rd person singular present |

Table 1: Main spaCy POS tags used in the algorithm. The first column includes spaCy's coarse-grained POS tags from UD [2022b]. The second column specifies spaCy's the fine-grained version of these tags. The third column gives a short description of each POS category.

after stating the implementation results. The authors propose a logic to guess the tense related to the ToI within the sentence, however, this is proposed as an experimental procedure and will be deprecated in this work.

– *Negation handling:* Whenever a negation term is detected in a chuck of text, the sentiment score is adjusted (i.e., the score is multiplied by $-1$).

### 2.2.2 Rule Scheme:

In order to tackle the words related to our ToI and aggregate its sentiment, a set of syntax rules based on the POS and DEP tags are applied. Both features, extracted using spaCy components (see Section 13, are based on the UD framework. Tables 1 and 2 introduce the notation that will be used later to describe the semantic rules.

As it has been stated in the implementation details, when a matching rule is detected, FiGAS assigns a sentiment score to each of the tokens associated with the ToI from the SentiBigNomics lexicon. The overall sentiment of the chunk is then computed regarding the hierarchical intrinsic structure created by the rules (see Section 2.2.4). The final sentiment polarity for the whole chunk is reversed (negation handling) in case the chunk contains a negation or a term with a negative connotation (i.e., DEP = neg). The heuristic rules are based on both the syntax and semantics of the text and this allows for explanations of how the algorithm reaches its final sentiment polarity score. This is a convenient feature since it provides transparency and interpretability to the analysis. In this project, the rules have been classified into two

| DEP | Description |
|---|---|
| acl | Clausal modifier of noun (adjectival clause) |
| advcl | Adverbial clause modifier |
| advmod | Adverbial modifier |
| amod | Adjectival modifier |
| attr | Attribute |
| dobj | Direct object |
| neg | Negation modifier |
| oprd | Object predicate |
| pcomp | Complement of preposition |
| pobj | Object of preposition |
| prep | Prepositional modifier |
| xcomp | Open clausal complement |

Table 2: Main spaCy DEP tags used in the algorithm. The first column includes spaCy's syntactic dependency relation tags and the second column gives a short description of each.

types: the ones that parse words directly linked to the ToI and the ones that go one step further and jump two steps away from the ToI to look for sentiment-charged terms. Note that the arrows in the figures included in this section do not determine the directionality in which the rules are applied.

One-step rules:

1. **ToI followed by an adjectival modifier (amod)**
   Adjectival modifiers are usually phrases that serve to modify the meaning of a noun. The token of interest is linked by an adjectival modifier dependency relation (DEP = amod) to a term that can either be an adjective or a verb (see Figures 1 and 2).

   - *"The* **new** *<u>Apple watch</u> went on sale on Friday"*

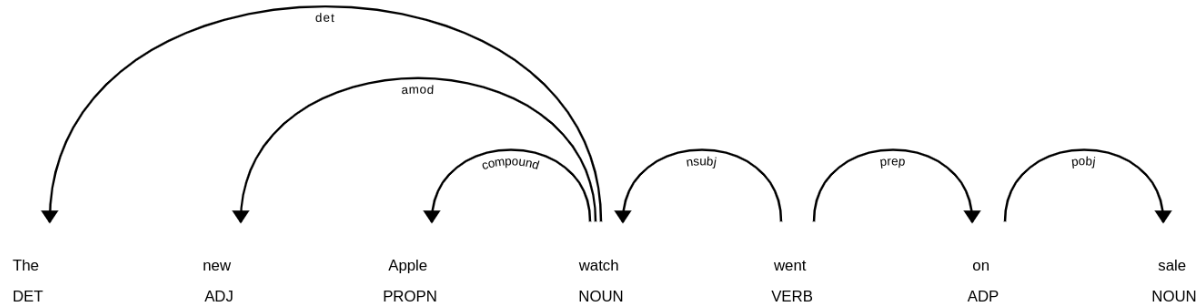


Figure 1: Adjective as adjectival modifier.

   - *"That facts not lost on* **unionized** *<u>AT&T workers</u>"*

2. **ToI followed by an adjectival clause (acl)**
   Adjectival clauses are finite or non-finite clauses that usually work to describe a noun in a sentence.

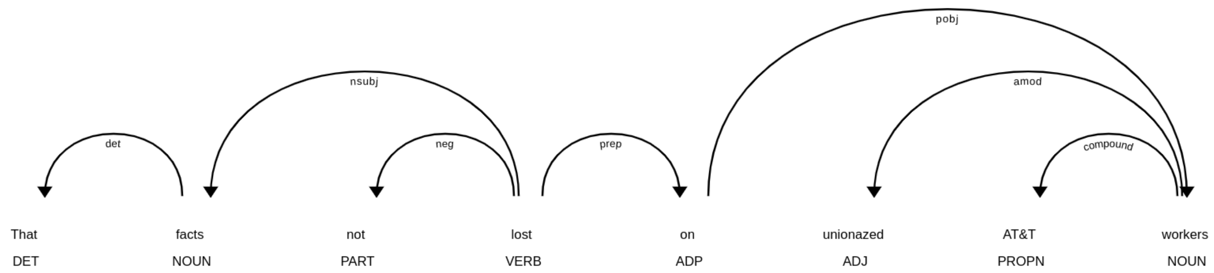


Figure 2: Verb as adjectival modifier.

The token of interest is linked by an adjectival clause dependency relation (DEP = acl) to a verb, which is the root of the adjectival clause (see Figure 3).

- *"...especially with <u>Apple</u>* **expected** *to introduce its..."*

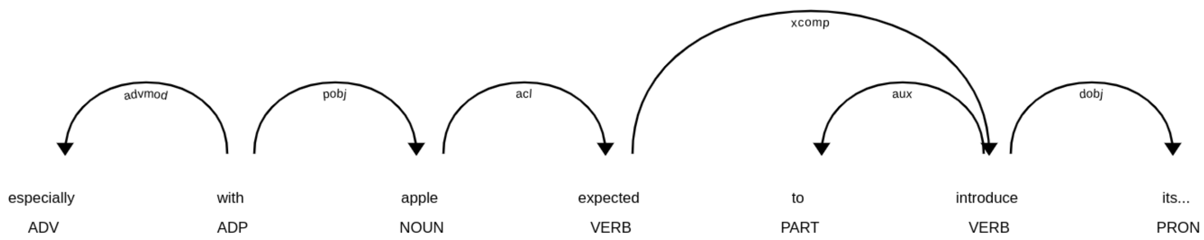


Figure 3: Verb introducing an adjectival clause.

More-than-one-step Rules:

3. **ToI linked to a verb followed by a direct object(dobj) or an attribute(attr)**
A direct object is the noun or noun phrase that's receiving the action of the verb. An attribute can be described the same way but it receives this name when the involved verb is a copulative verb. In this case our ToI is directly connected to a verb that in turn connects with a noun as a direct object or an attribute (see Figure 4)

    - *" <u>Apple Music</u>* **accelerates subscriptions***"*

4. **ToI linked to a verb followed by an adjectival complement (acomp)**
An adjectival complement is a standard adjective that provides additional information about a verb. The ToI can be linked to a verb that in time connects with an adjective which adds emotional information through an adjectival complement dependency (see Figure 5).

    - *"<u>QCOMP</u>* **remains independent***"*

5. **ToI linked to a verb followed by an adverbial modifier (advmod)**
An adverbial modifier can be either a non-clausal adverb or adverbial phrase that modifies the predicate. The ToI is linked to a verb that is modified by an adverb through an adverbial modifier dependency (see Figure 6).

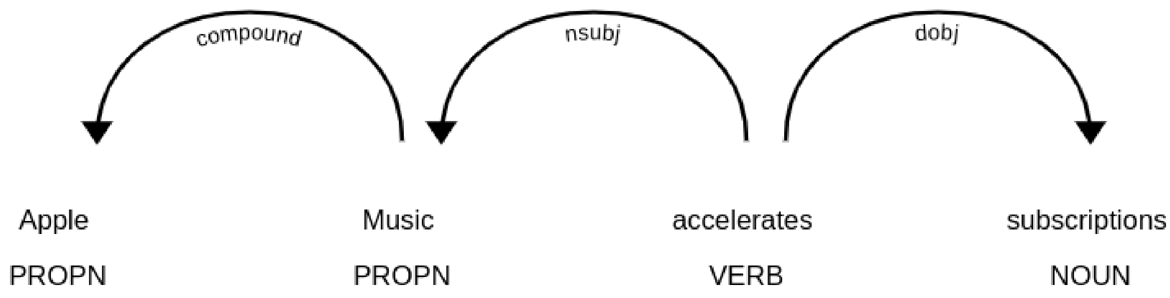


Figure 4: Noun as direct object.

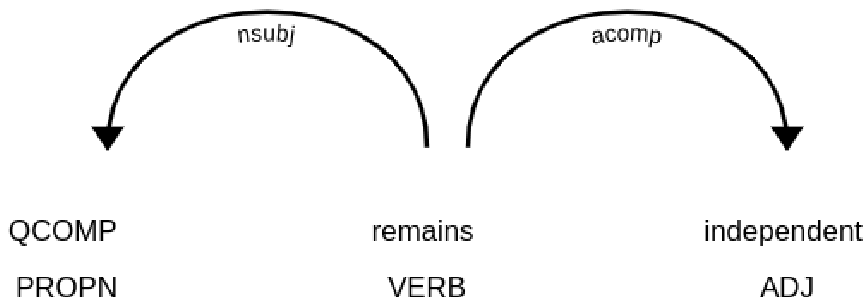


Figure 5: Adjective as adjectival complement.

- *"Amazon* **recently acquiring** *a package of NFL games..."*

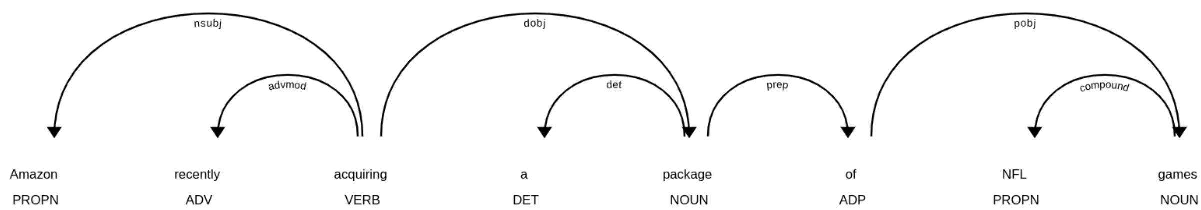


Figure 6: Adverb as adverbial modifier.

6. **ToI linked to a verb followed by an object predicate (oprd)**
   An object predicate is a word that provides information about the direct object of a verb. In this case the ToI will be linked to a verb that in turn is linked to an adjective (of any type) through an object predicate dependency (see Figure 7).

   - *"Dow Jones* **turns positive***"*

7. **ToI linked to a verb followed by an open clausal complement or an adverbial clause modifier (xcomp or advcl)**

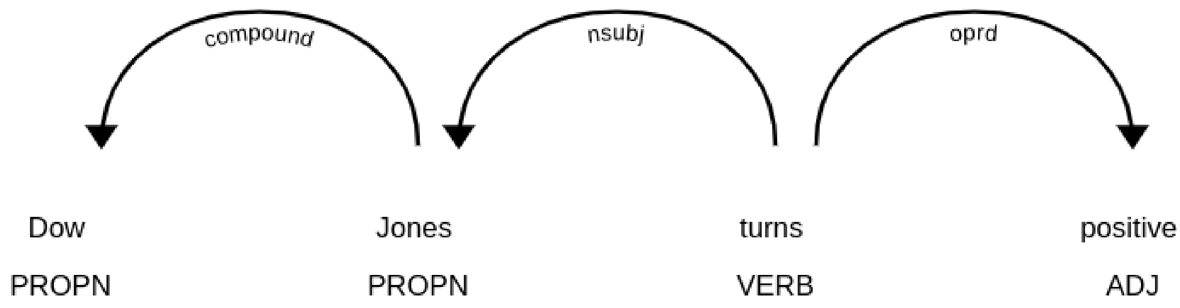


Figure 7: Adjective as object predicate.

Complete sentences can also be complements of other sentences. An open clausal complement is a predicative or clausal complement without its own subject. Oppositely, an adverbial clause modifies a verb or another predicate as a modifier, not as a core complement, so the subject is present in the sentence. In this case the ToI is linked to a verb that in turn is connected to another verb, the core of the modifier clause (see Figure 8).

- *"Amazon* **moving** *to* **buy** *WFM whacked stocks..."*

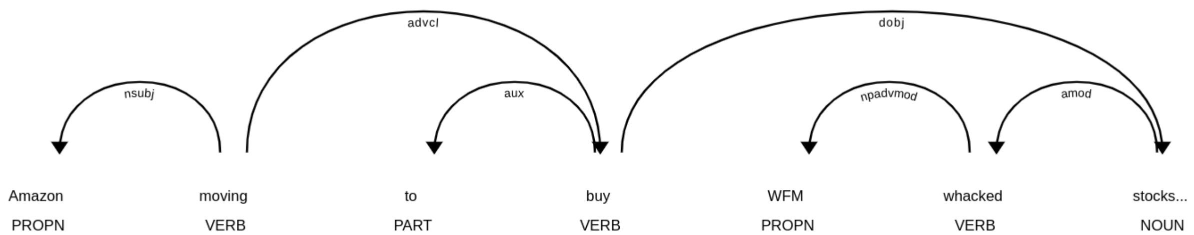


Figure 8: Verb introducing an adverbial clause.

8. **ToI linked to a verb followed by prepositional modifier (pobj or pcomp)**
   Finally, complete sentences and other semi-complex structures can be introduced by prepositions and become prepositional modifiers. When the preposition takes you to a noun, the last becomes a prepositional object (pobj) and when it takes you to a verb that introduces a new sentence, it becomes a prepositional complement (pcomp) (see Figures9 and 10).

   - *"Apple* **works on** *original* **content***"*
   - *"Tom Enders* **accused** *Boeing* **of supporting** *the policy."*

### 2.2.3 Dictionary

To quantify the sentiment score of the words, the authors designed a new fine-grained sentiment lexicon for the economics and financial domains named SentiBigNomics. Examples of commonly used dictionaries for dictionary-based SA approaches are SentiWordNet [Baccianella et al., 2010] and SenticNet [Cambria et al.,

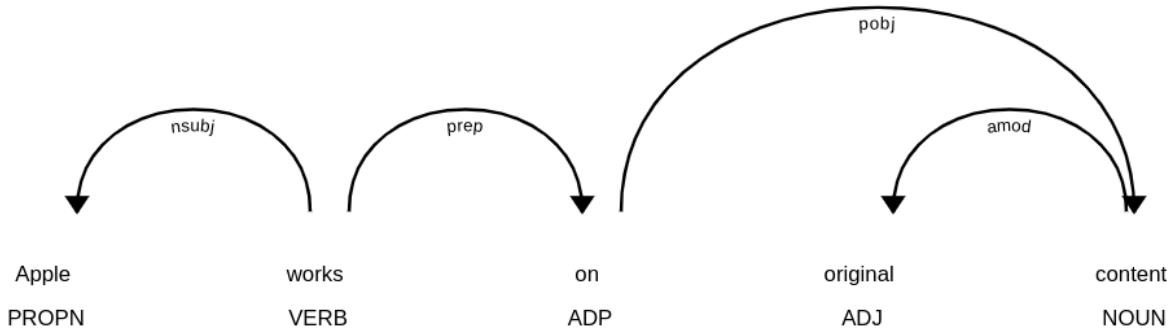


Figure 9: Noun as prepositional object.

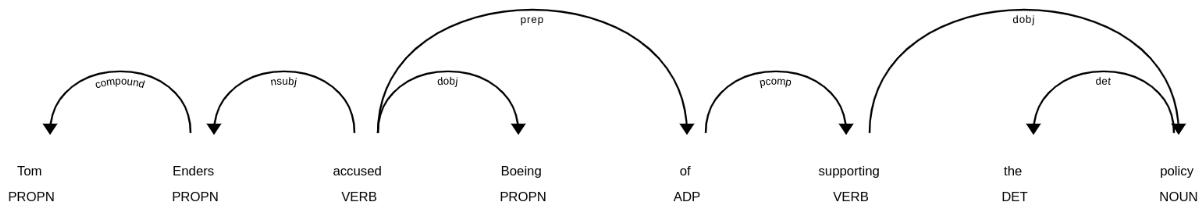


Figure 10: Verb as prepositional complement.

2010], which were developed for general-purpose, and LMD [Loughran and McDonald, 2011], which is a domain-specific dictionary but all the words are equally weighted. A more extended comparison between those dictionaries and SentiBigNomics can be found in Consoli et al. [2022a], as well as further details on its construction.

### 2.2.4 Score aggregation

Semantic rules are the instrument that the authors use to isolate the sentiment connotation of a ToI. The outcome of the rules consists of a set of words that are semantically related to the ToI. These words represent the vocabulary considered when computing the sentiment. The definition of the rules itself carries an intrinsic directionality that establishes a logic for the aggregation of sentiment scores retrieved from SentBigNomics. Sentiment polarity is propagated backward, from the last word involved in a matching rule to the ToI. Assume that we have two consecutive terms, namely $y$ and $x$, with their sentiment scores denoted by $s(y)$ and $s(x)$, respectively. We propagate the sentiment score from the last actor identified by the rule, say $x$, to its previous linked term, $y$, to obtain the propagated sentiment score $\widehat{s}(y)$ as follows:

$$s(\mathrm{x}) \rightarrow s(\mathrm{y}) \Longrightarrow \widehat{s}(\mathrm{y}) \tag{12}$$

The rationale behind this logic is to be able to understand the origin of the sentiment and to propagate it through the semantic relationships of a sentence. More specifically, the magnitude of the propagated score $\widehat{s}(\mathrm{y})$ is defined as:

$$\widehat{s}(\mathrm{y}) = sign(s(y)) * sign(s(x)) * (|s(y)| + (1 - |s(y)|) * |s(x)|) \tag{13}$$

The authors state that this formula modifies the sentiment score for $y$ proportionally to the sentiment score of the subsequent linked term $x$. The proposed formula becomes the rule of thumb for calculating and propagating the sentiment polarities within the selected words. In case of a concatenation of more rules within the same text fragment, the overall sentiment would be computed by using the same logic: first, associating the corresponding score polarity to each term that was involved in a matching rule, and second, propagating the scores backward until the ToI is reached. Note that the sentiment score for the ToI does not enter into the overall computation, but only its sign is taken into consideration. In particular, if the ToI has a negative connotation, then the final chunk sentiment is inverted.

### 2.2.5 Project variation

The main difference between the original implementation and this project is that we focus on the SA of the companies included in financial news instead of economical concepts. An easy way to tailor the discussed methodology to this task is to define the entities mentioned in the text as the ToI around which to perform the analysis. SpaCy's NER component is used during the pre-processing step to filter the entities that appear in each document and to select only the sentences associated with a given stock. This reduces the required computational resources because only the necessary content is processed.

Due to the multiple ways we can use to refer to a single company, linking spaCy's recognized entities to company identifiers becomes a challenging task. Further details on the matching procedure can be found in Section 3.3. Once the rules are applied, an aggregation procedure is executed to obtain the document (daily) level sentiment scores. Choosing the proper aggregation procedure is not trivial and is discussed in Section 3.6.

# 3. Methods

The current section focuses on specific details regarding the implementation stage. Figure 11 provides a wide overview of the main procedures involved when calculating the sentiment scores. The following bullet points explain a general description of each phase of the implementation.

Input data:

- Financial news provided by DJN in *NML* format, which is Dow Jones standard *XML* format.

Initial pre-processing:

- Using spaCy, we link the sentences with the stocks that appear in the text. The output of this stage is a dictionary where each entity is linked to the set of sentences that talk about it.

Pipeline splitting:

- **SESTM:** A pre-processing step is performed to remove non-sentiment-charged content. The cleaned text related to each stock is associated with their price return computed from daily stock prices. Then, we build the SESTM model which is used to compute daily scores for each equity.
- **FiGAS:** A set of syntactic and semantic rules is applied over the raw text to extract sentiment scores for each sentence linked to an entity within the document. FiGAS sentiment scores need to be aggregated to get daily sentiment scores. To jump from sentence to article-level scores we consider

two possible ways of aggregating the data, the addition and the arithmetic mean of the scores. The daily scores are calculated as the weighted average of the article-level scores computed for a given entity on a given day, where the weights are the number of words that contributed to computing each article-entity score.

## 3.1 Textual data

Thanks to the partnership between Acuity Trading S. L. and Prof. A. Arratia, we are able to use the content published by DJN. This content is released as *NML* files, which is DJN's *XML* extension with special tags for financial textual data. Apart from the textual information, *NML* files include other relevant metadata that simplified the work during the implementation. For example, publishing dates, stocks referenced in the story, etc. Figure 12 shows an example of a *NML* file. We use the content provided in the DJN's products DN and PB, which include news published in The Wall Street Journal, Barron's, Smart Money, etc. from 2017, 2018, 2019, and 2020.

## 3.2 Price data

The model proposed by Ke et al. [2019] requires price data for calculating the sentiment scores. To build up the model, the SESTM methodology uses stock returns data as labels. First, let us define what is a return in the financial domain: let $P_t$ be the price of an asset at time $t$. Given a time scale $\bar{t}$, the $\bar{t}$-period simple return at time $t$, $R(t)$, is the rate of change in the price obtained from holding the asset from time $t - \bar{t}$ to time $t$ [Arratia, 2014]:

$$R_t(\bar{t}) = \frac{P_t - P_{t-\bar{t}}}{P_{t-\bar{t}}} = \frac{P_t}{P_{t-\bar{t}}} - 1 \tag{14}$$

A priori, it is difficult to determine how long it takes for sentiment-charged news to have a noticeable impact on prices. If prices adjust slowly, then it makes sense to align articles not only with contemporaneous returns but also with future returns. However, it could be the case that news is a restatement of recently revealed information, in that case it is better to align the content with prior returns. Without better guidance on timing choice, given an article at time $t$ that talks about a stock $S$ with price $P_t$ we define the adjusted linked return as:

$$\bar{R}_t = \frac{P_{t+1}}{P_{t-2}} - 1 \tag{15}$$

Note that this timing is for sentiment training purposes only. Indeed, when testing the model we would not allow a look-ahead into future prices $P_{t+1}$.

This data set is also required for the statistical analysis conducted in Section 4.1 and in Section 4.3 when backtesting the financial strategies, which requires real market Open, High, Low and Close (OHLC) prices[2].

The price data used in this work is provided by *Dukascopy Swiss Banking Group*, a Swiss online bank that offers Internet-based and mobile trading services, banking, and other financial services. Their Historical Data Feed [SA, 2022] includes historical price data for a variety of financial instruments (e.g. Forex, Commodities, and Indices). The data consists of 4 time series for each instrument with daily records

[2]An opening price is the first price a stock trades at when the market opens at 9:30 a.m., and a closing price is the last price traded at when the market closes at 4:00 p.m., In the same way, the high and low prices are the highest and lowest price at which a financial instrument has traded during the day. The high and low prices can exceed opening and closing prices.

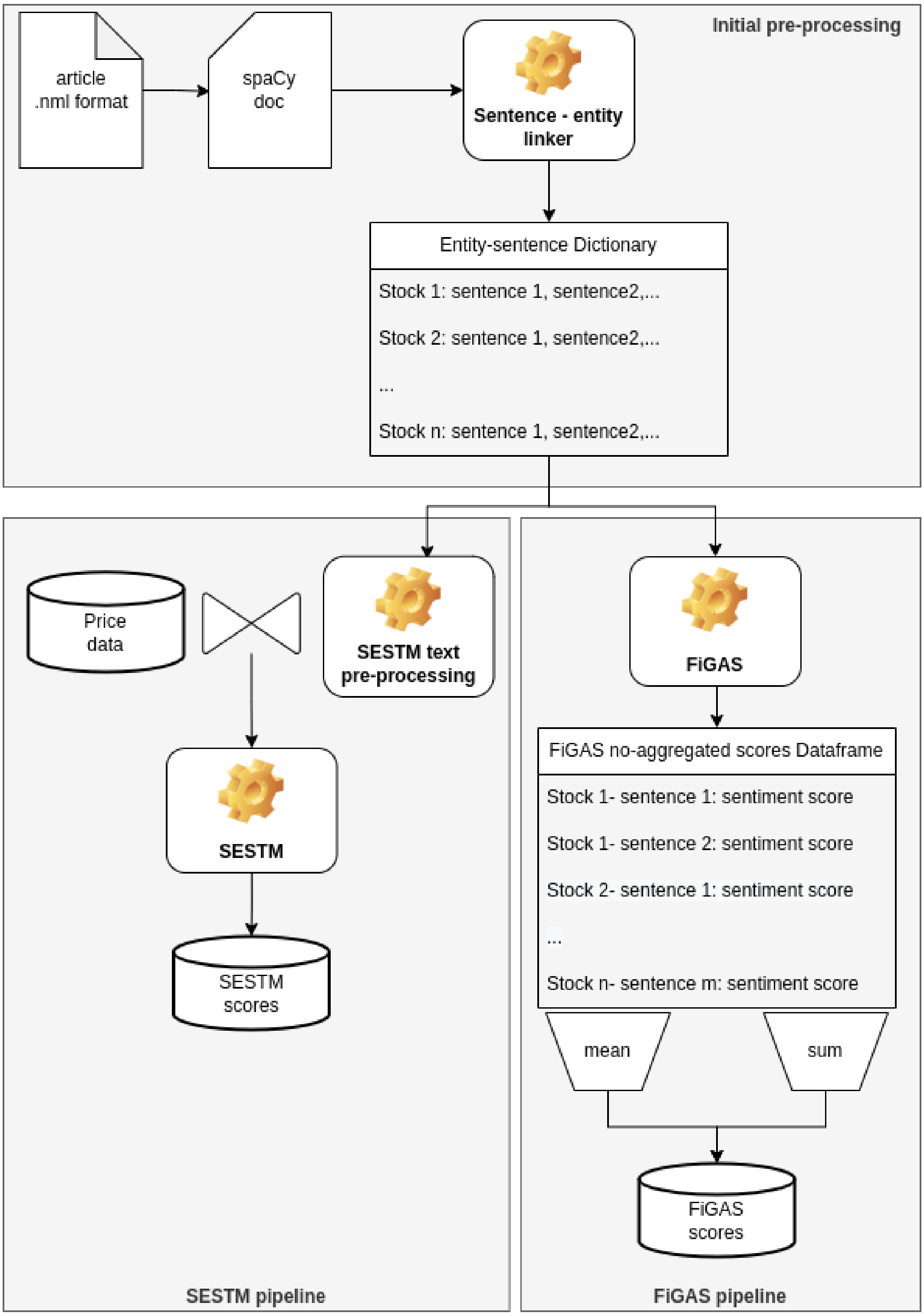


Figure 11: Global diagram of the implemented solution for computing the sentiment scores.

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE doc SYSTEM "djnml-1.0b.dtd">
<doc msize="000006783" md5="c18bcdb9ef2931e2c287b5ce5c260f14" sysId="secnwsdcmn3p2" destination="AW" distId="JLP9" transmission-date="20220803T060013Z" >
<djnml publisher="DJN" docdate="20220803" product="PB" seq="7" xml:lang="en-us" >
<head>
<copyright year="2022" holder="Dow Jones &amp; Company, Inc." ></copyright>
<docdata>
<djn>
<djn-newswires news-source="DJPB" origin="DJ" service-id="CO" >
<djn-press-cutout/>
<djn-urgency>0</djn-urgency>
<djn-mdata brand="DJ" temp-perm="P" retention="N" hot="N" original-source="WJ" accession-number="20220803000007" page-citation="A9" display-date="20220803T0600Z
<djn-coding>
<djn-subject>
<c>N/PB</c>
<c>N/WSJ</c>
<c>N/DJWI</c>
<c>N/GEN</c>
<c>N/HLT</c>
<c>N/LIF</c>
<c>N/RND</c>
</djn-subject>
<djn-market>
<c>M/NND</c>
</djn-market>
<djn-product>
</djn-product>
<djn-geo>
<c>R/NME</c>
<c>R/US</c>
</djn-geo>
<djn-journal>
<c>J/PSJ</c>
</djn-journal>
</djn-coding>
<byline name="Andrea Petersen and Alex Janin" org="Dow Jones &amp; Company, Inc." ></byline>
</djn-mdata>
</djn-newswires>
</djn>
</docdata>
</head>
<body>
<headline>
PERSONAL JOURNAL  ---  Tame Your Stress Level  Walking With a Friend  ---  The combination of exercise, getting out in nature and connecting with  other people
Petersen and Alex Janin
<hl1>PERSONAL JOURNAL </hl1>
<hl2>--- </hl2>
<hl2>Tame Your Stress Level </hl2>
<hl2>Walking With a Friend </hl2>
<hl2>--- </hl2>
<hl2>The combination of exercise, getting out in nature and connecting with </hl2>
<hl2>other people improves hormonal balance, boosts coping mechanisms </hl2>
<hl2>---- </hl2>
<hl2>By Andrea Petersen and Alex Janin</hl2>
</headline>
<text>
<p>
   If reducing stress has been on your summer to-do list, there&apos;s one powerful thing you can still do before the season ends: get in the habit of taking a
<p>
  Stress is battering us on many fronts. About 87% of adults said rising prices due to inflation are a significant source of stress, according to a March survey
```


Figure 12: Example of a DJN news in *NML* format.

for OHLC prices. Returns are computed using *pct_change* pandas function [pandas development team, 2020] over close prices with a 3-days window. The return date needs to be shifted one day to ensure the linkage defined in equation (15). See that during the weekend markets are closed so we need to take special care when computing Fridays or Mondays return values.

## 3.3 SpaCy

SpaCy [Honnibal and Montani, 2017] [3] is an open-source software library for advanced NLP written in Python and Cython that focuses on providing software for production usage. The central data structures in spaCy are the *Language*, the *Doc*, and the *Vocab* objects. The *Language* class is used to process a text, and turn it into a *Doc* object (spaCy's containers for accessing linguistic annotations). A *Doc* owns the sequence of tokens and all their annotations. The *Vocab* stores the strings, word vectors, and lexical attributes avoiding the storage of multiple copies of this data. This saves memory space and ensures there's a single source of truth. To turn plain text into a *Doc* object we use what is usually referred to as the

[3] **SpaCy:** All the mentions of spaCy along this document refer to spaCy v3.4.

processing pipeline. It consists of one or more components that are called sequentially on the *Doc*. SpaCy provides a range of built-in pipeline components for different language processing tasks (e.g. *DEP parsing*, *POS tagging*, *NER*, *lemmatizing*, *sentence recognition*,etc.) and also allows adding custom components. Figure 13 shows a standard spaCy pipeline. The *Language* object coordinates the pipeline components taking raw text and sending it through the pipeline. Each component needs to be fed with a *Doc* and returns the processed *Doc*, which is then passed on to the next component.

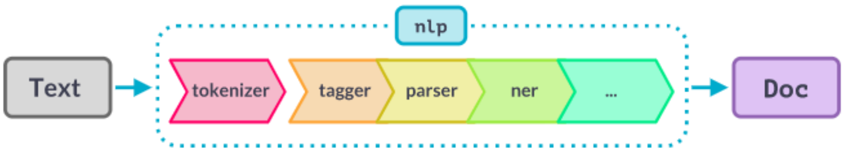


Figure 13: SpaCy's standard pipeline [Honnibal and Montani, 2017].

The first component always has to be a tokenizer. Its task is to split the raw text into meaningful segments and to convert them to the *Doc* format. This is done by applying specific rules to each language, for example, punctuation at the end of a sentence should be split off – whereas "U.K." should remain one token. After this step, each *Doc* consists of individual tokens over which we can iterate. Following the spaCy notation, a set of consecutive tokens is called a *span*.

The second component of our pipeline is a POS tagger. Using SpaCy's trained component, we predict POS tags for all the tokens in the *Doc* with a 97% of accuracy. The tagging procedure analyzes each token's context to choose the most likely tag, for instance, a word following "the" in English is most likely a noun (i.e. POS = NOUN). A list of the main POS tags used in this work has already been introduced in Section 2.2.2.

In a similar way, the DEP parser assigns DEP tags to each token according to their syntactic relationships and its POS tag. SpaCy's trained pipeline shows an accuracy level of 90%. A list of the main DEP tags used in this work has already been introduced in Section 2.2.2.

The fourth component needed in our pipeline is a lemmatizer. This component sets the base forms for tokens using rules based on POS tags and lookup tables. We use these lemmas to query the SentiBigNomics dictionary to get the sentiment score of each token.

Finaly, the NER component stores the list of named detected entities in the *Doc.ents* object, and gives to each token two new attributes: *ent_type* and *ent_iob*. The first one includes the named entity type information (e.g. organization as *ORG*, geopolitical entities as *GPE*, persons as *PERSON*, etc). The second attribute is an IOB[4] code that indicates whether a token belongs to an entity. SpaCy's NER component shows an accuracy of 85%. In this work, we focus on the named entities tagged as Organizations (ORG).

The entities detected by spaCy do not always have a direct match to the stock identifiers. To overcome this issue we define a disambiguation procedure, which also considers the existence of different ways to refer to the same company. Our approach requires a set of keywords used to refer to each stock. For example, "Apple Inc.", "Apple", "APPL", etc. refer to *Apple*. At this point, we would like to acknowledge Acuity Trading S.L. for allowing us to use the stock's identifiers list used in the company for business purposes. For confidentiality reasons, this list won't be included nor discussed in this work.

[4] **IOB:** IOB is a common tagging format in computational linguistics for tagging tokens in a chunking task. It is short for Inside, Outside, Beginning [Ramshaw and Marcus, 1995].

Another key component of our procedure involves string comparison. To that aim, we have chosen *Levenshtein distance* [Levenshtein, 1966]. String matching is a very active research field and there is no consensus on which similarity metrics produce better or more accurate results. It usually depends on the application and the available data. As mentioned, we decided to use the *Levenshtein distance*, which, given two strings $a, b$ (of length $|a|$ and $|b|$ respectively) is defined as:

$$\operatorname{lev}(a,b) = \begin{cases} |a| & \text{if } |b| = 0, \\ |b| & \text{if } |a| = 0, \\ \operatorname{lev}(\operatorname{tail}(a), \operatorname{tail}(b)) & \text{if } a[0] = b[0], \\ 1 + \min \begin{cases} \operatorname{lev}(\operatorname{tail}(a), b) \\ \operatorname{lev}(a, \operatorname{tail}(b)) \\ \operatorname{lev}(\operatorname{tail}(a), \operatorname{tail}(b)) \end{cases} & \text{otherwise}, \end{cases} \tag{16}$$

where the tail of some string $x$ is a string of all but the first character of $x$, and $x[n]$ is the $n^{th}$ character of the string $x$, counting from 0. Note that the first element in the minimum corresponds to deletion (from $a$ to $b$), the second to insertion and the third to replacement. This definition corresponds directly to the naive recursive implementation. This distance can be seen as a standard in the string-matching research field. Performing a complete string-matching metrics study falls outside the scope of this work. At the beginning of this work, some other distances were tested to find the best approach and *Levenshtein* showed the most accurate results.

Our disambiguation procedure works as follow: given a *Doc*, let $E = [e_1, e_2, ...e_m]$ be the list of entities predicted by spaCy and $S = [s_1, s_2, ...s_n]$ the list of stocks found in the *NML*'s metadata. $Doc_{[e_i]}$ is the set of sentences where the $i^{th}$ entity appears. We select the filters $F = [f_{1,1}, f_{1,2}...f_{2,1}, f_{2,2}, ...f_{n,q}]$ that refer to the stocks in $S$. Note that a single stock $s_j$ can correspond to multiple filters $[f_{j,1}, f_{j,2}, f_{j,3}, ...]$. First, we look for exact matches between $F$ and $E$. If any exists (i.e. $e_i = f_{j,k}$), we add the {key, value} pair $\{s_j, Doc_{[e_i]}\}$ to the entity-sentence dictionary (see Figure 11). Second, we compute the *Levenshtein*'s distance between all the entities left in $E$ (not exact matches) and the stocks list $S$ to look for similar strings. For each entity $e_i$, we select the filter $f_{j,k}$ that shows the smaller distance. If the distance between $e_i$ and $f_{j,k}$ is smaller than a certain threshold $t_{sim}$, we add $\{s_j, Doc_{[e_i]}\}$ to the entity-sentence dictionary. $t_{sim}$ is a parameter, and the intuition behind it is to base it on the length of the words we're computing similarity on i.e. the shorter the words, the easier it is to find two similar words by chance, so the higher $t_{sim}$ should be.

At this stage the initial pre-processing is done and its outcome is the entity-sentence dictionary.

## 3.4 FiGAS pipeline

The FiGAS implementation has been slightly modified for the sake of computational resource optimization. In syntactic terms, economic concepts usually act as the sentence object. On the contrary, entities normally act as the active or passive subject of a sentence. This difference has been considered when applying the set of rules defined in Section 2.2.2.

For each entry of the entity-sentence dictionary, every sentence is an input for the FiGAS methodology, which returns a score (see the FiGAS no-aggregated scores dataframe in Figure 11). For each stock, the scores retrieved from different sentences are aggregated using the sum and the arithmetic mean, using different aggregation metrics will be discussed later in this report.

## 3.5 SESTM pipeline

Before feeding the text into the SESTM modeling step, a pre-processing filtering is applied to clean the text and get rid of the non-informative parts. We lowercase the text and apply a set of regular expressions to remove URLs, websites, stop words [5] in English, and other non-alphanumeric characters providing no relevant information to the SA. After this, the text is labeled with $\bar{R}_t$ return values to jump into the Screening stage within the SESTM pipeline. Its outcome is the vocabulary, which contains a set of positive and negative words and their emotional scores. Then, as it has been discussed in section 2.1, this vocabulary is used to score the daily text related to each stock.

## 3.6 Database

Another challenge of this work is to efficiently manage the data storage. The advantages of designing a good database are noteworthy, specially when dealing with large amounts of data. Otherwise, the system would rapidly run out of space and execution times of the aggregation and modeling stages would increase a lot. See the database schema in Figure 14.

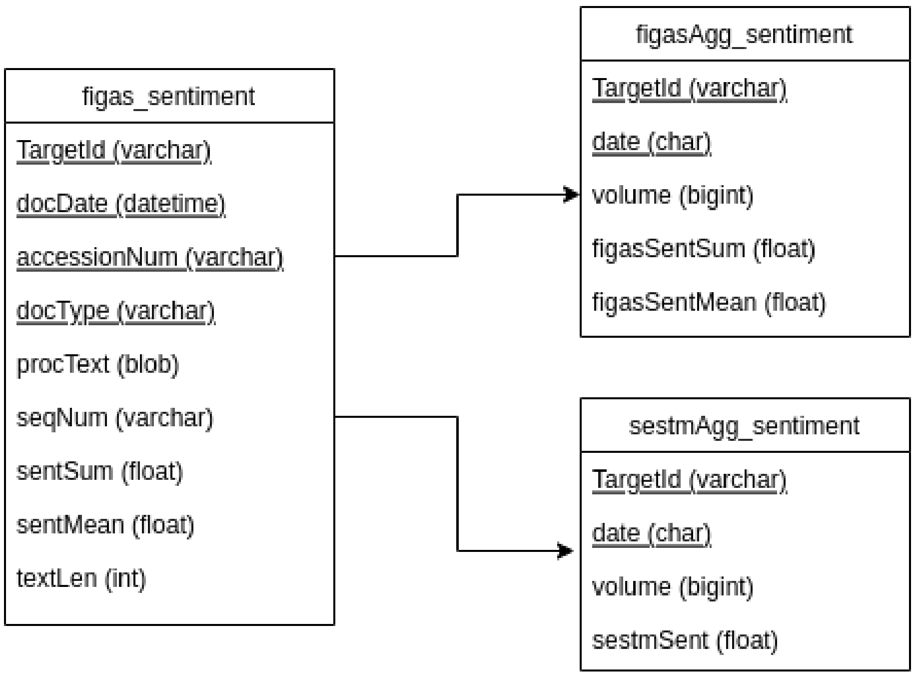


Figure 14: Database schema.

There are two different aggregation levels. The *figas_sentiment* table stores the document-level FiGAS sentiment scores for each stock (*TargetId*). Remember that, for this methodology, we keep the summation and the arithmetic mean as different document-level aggregations. The text that was involved in the computation of sentiment scores (*procText*) is compressed and stored in the same table using MySQL [MySQL, 2022]. The *figasAgg_sentiment* and *sestmAgg_sentiment* tables store daily-aggregated sentiment

[5] **Stop words:** Stopwords are those words that do not provide any useful information because they don't have any meaning (prepositions, conjunctions, etc.)

scores. The first one is built by aggregating the fields *sentSum* and *sentMean* from *figas_sentiment* using a weighted average where weights are the number of words in *procText*. This aggregation gives more weight to sentiment scores extracted from larger texts, which are likely to be more reliable. The second one contains SESTM sentiment scores, which are computed using all the available text linked to an equity in a specific date.

## 3.7 Computation details

The complete project has been developed in Digital Ocean [LLC., 2022] cloud servers. Everything but the news processing for the score computation has been developed in a server with 4 cores and 8 GB of RAM. To accelerate the score computation, a 32-cores and 64 GB RAM server was used in the initial pre-processing step. On average, it takes 0.3225 seconds to process one news article. The pre-processing within the SESTM pipeline is also performed in parallel.

## 3.8 Correlation and Granger causality

It's a common practice to evaluate SA methodologies in the financial domain computing the correlation coefficients between sentiment scores and market prices. In this work, we study the relationship between these time series comparing how well SESTM and FIGAS scores correlate to market return values. In this case, return values are defined as:

$$R_t = \frac{P_t}{P_{t-3}} - 1 \tag{17}$$

Also, it is interesting to see how correlated are both types of sentiment scores.

There are several measures for calculating the degree of correlation. The most common one is the *Pearson correlation coefficient* [Benesty et al., 2009], which is sensitive only to a linear relationship between two variables. Considering the existing variability in stock markets, to assume a linear relationship between our sentiment scores and market returns is too strong an assumption. For this reason, we decided to use *Spearman's rank correlation coefficient* [Spearman, 1961], which is more sensitive to nonlinear relationships.

We also use the Granger Causality test [Granger, 1969] to determine whether one time series can be used to forecast another. In our case, we would like to see that the sentiment series Granger-causes market prices. However, it is reasonable to think that Granger causality can happen in both directions: either sentiment scores will have an impact on the stock prices or the market prices will guide the writing of articles. The implementation of the Granger causality test appears in the *statsmodel* package [Seabold and Perktold, 2010]. Given two time series $a$ and $b$, the hypothesis are defined as follows:

- $H_0$: $a$ is not the cause of $b$ with a certain delay.
- $H_1$: $a$ is the cause of $b$ with a certain delay.

Thus, if the p-value is below the significance level, the null hypothesis can be rejected, concluding that $a$ Granger-causes $b$. We have set a significance level of $\alpha = 0.1$.

When modeling market values, it is very unlikely to have a causal relationship that holds for large periods like the one we are analyzing (4 years). Due to this fact, we test the Granger causality using a moving window of 21 days. The window size has been chosen according to the 21 monthly-average trading days. For the delay, we use a maximum lag of 5 days. This is aligned with the word by Yu [2014], who points out that several studies have noticed a decay in the impact of news on asset prices and the complete disappearance of news effects within 2-5 days. The smallest lag in which we can reject the null hypothesis will be kept as the lag that presents causality for that window.

## 3.9 Backtesting

Technical Analysis (TA) is a trading discipline employed to evaluate investments and identify trading opportunities by analyzing statistical. Unlike fundamental analysis, which attempts to evaluate a security's value based on business results such as sales and earnings, TA focuses on the study of price and volume. TA operates from the assumption that past trading activity and price changes of a security can be valuable indicators of the security's future price movements when paired with appropriate investing or trading rules. Taking advantage of this analysis, people define signals, which are the triggers for actions, either to buy or sell a security or other asset. These triggers can be human-generated using technical indicators or automatically generated using data usually based on market action. The triggered actions are commonly known as trades, which usually consist of paired actions, either buy and then sell or vice versa. There are two main trade actions: Long trades, linked to upward trends and therefore perform a buying action, and short trades, related to downward patterns and consist of selling action.

There are several ways to create trading signals. They are usually based on different indicators and rules over the market's situation that define when to enter the market (taking long positions) or when to leave it (taking short positions). One of the most common practice is the Dual Moving Average Crossover. The concept is fairly straightforward, calculate two different-period moving averages over the price series of a security and look for crossover points, which are the points where the values of the two moving averages may be equal and/or cross one another. TA suggests that when the Short Term Moving Average (STMA) moves above the Long Term Moving Average (LTMA), that represents a Buy (or Long) signal. Conversely, when the STMA moves below the LTMA, that indicates a Sell (or Short) signal. The intuition behind this strategy can be explained in terms of momentum. The principle of momentum states that a price that is moving up (or down) during period t is likely to continue to move up (or down) in period t+1 unless evidence exists to the contrary [Achelis, 2001]. When the STMA moves above the LTMA, this provides a lagged indicator that the price is moving upward relative to the historical price and vice versa.

It is reasonable to think that using sentiment indicators in the definition of the trading rules can improve the performance of the strategies. Our goal is to integrate the sentiment scores with TA common practices to guide trading strategies. This work will perform a Dual Moving Average Crossover strategy over the sentiment data computed with both methodologies. The strategy is defined as follows:

– **Instances**

1. STMA over $s$ days up to time $t$:

$$MA(s)_t = \sum_{i=t-s}^{t} \frac{S_i}{s} \text{ where } S_i \text{ is the sentiment score at time } i$$

2. LTMA over $l$ days up to time $t$:

$$MA(l)_t = \sum_{i=t-l}^{t} \frac{S_i}{l} \text{ where } S_i \text{ is the sentiment score at time } i$$

– **Trading Rules**

1. Long trade at $O_{t+1}$ if $MA(s)_t > MA(l)_t$ where $O_{t+1}$ is the close price at time $t+1$
2. Short trade at $O_{t+1}$ if $MA(s)_t < MA(l)_t$ where $O_{t+1}$ is the close price at time $t+1$

As a baseline, we chose the same Dual Moving Average Crossover strategy but this time over price data to see if following sentiment-based signals improves performance.

– **Baseline Instances**

1. STMA over $s$ days up to time $t$:

$$MA(s)_t = \sum_{i=t-s}^{t} \frac{C_i}{s} \text{ where } C_i \text{ is the close price at time } i$$

2. LTMA over $l$ days up to time $t$:

$$MA(l)_t = \sum_{i=t-l}^{t} \frac{C_i}{l} \text{ where } C_i \text{ is the close price at time } i$$

– **Baseline Trading Rules**

1. Long trade at $O_{t+1}$ if $MA(s)_t > MA(l)_t$ where $O_{t+1}$ is the close price at time $t+1$
2. Short trade at $O_{t+1}$ if $MA(s)_t < MA(l)_t$ where $O_{t+1}$ is the close price at time $t+1$

Common values for $s$ and $l$ are $s$=*20* and $l$=*50*, however, we'll be running the strategy for $\left(\ s, l\ \right) = \left\{\ 5, 10, 15, 20\ \right\} X \left\{\ 25, 50, 100\ \right\}$

## 3.10 Backtrader

Until now we have the rules that will guide our investment strategy but we need an environment able to manage all this information and perform a trading strategy simulation. BackTrader [2022] is a feature-rich Python framework for backtesting purposes. It allows the user to focus on writing reusable trading strategies, indicators, and analyzers instead of having to spend time building infrastructure. Backtrader is composed of a set of classes and its parameters, which can be easily tuned and the **Cerebro** class. The latter is the cornerstone of Backtrader and orchestrates everything. Besides it, we'll be using the following classes:

- **Data Feeds:** to gather input data
- **Strategies:** to define the logic behind the strategy
- **Analyzers:** to get some data while the strategy is running and be able to analyze it afterwards.

Since Backtrader is a pure Python library, it's quite easy to use it once you understand the economic concepts. Further details on the implementation will be omitted since they would only mean repeating backtrader's documentation[6].

[6]**Backtrader:** https://www.backtrader.com/

For evaluating our strategies we'll focus on two different trading indicators. The first one is the profit and loss balance (PnL), which stands for the percentage of profit regarding the initial cash. The second one is the *Sharpe Ratio* [Sharpe, 1994], which is an indicator that compares the return of an investment with its risk. To get an idea, a *Sharpe Ratio* less than 1 is considered bad, between 1 and 1.99 it is considered adequate, between 2 and 2.99 it is considered very good and greater than 3 it is considered excellent [7].

# 4. Results

This section presents the results obtained after analyzing the computed sentiment scores. We start presenting the new's volume for each of the 20 well-known equities on which we focus our study. Then, we analyze the correlation between sentiment scores and market prices. After that, we introduce the results of the Granger causality test. We also backtest the trading strategies defined in Section 3.9 and perform an optimization procedure that chooses their most suitable parameters. Finally, we investigate the similarity between the SentiBigNomics dictionary, the automatically generated SESTM vocabulary, and Loughran and McDonald [2011] dictionary. The latter is a commonly used SA lexicon in the financial domain. For the sake of illustration we provide full plots for one company in this section, and further detailed plots for other two companies in the Appendix A.1.

## 4.1 Descriptive analysis and correlation study

In this work we have analyzed a total of 304124 news published by DJN from 2017 to 2020 and related to 20 well-known companies with a large capitalization margin from the U.S. market. Table 3 shows the list of companies on which we focus our analysis alongside their market capitalization. Figure 15 shows the distribution of the number of news associated with each one of the equities classified by DJN's product type. Note that there are way more articles from the DN source than from PB. The most popular company in terms of article appearances is *Morgan Stanley* and the less popular one is *Walt Disney*.

Table 4 contains the correlation coefficients obtained using the returns and the sentiment scores of each company. The $FiGAS_m - Ret$ combination for *Albhabet Inc.* is the value showing a grater correlation coefficient between sentiment scores and returns. *Boeing* is the company that presents a most notable value in terms of the correlation between the sentiment series, specifically in the relationship $FiGAS_m - SESTM$. We can state that the difference between the FiGAS sum-aggregated scores ($FiGAS_s$) and FiGAS mean-aggregated scores ($FiGAS_m$) is negligible regarding the correlation with return values. We keep both approaches in the following steps but it seems that their difference is not relevant. A detailed study on the selection of the aggregation approach is left as part of the future work. It's also interesting to see that FiGAS and SESTM sentiment scores show almost no correlation. This fact suggests that both methodologies are capturing different parts of the sentiment-related information contained in the text. It opens the door to a further analysis on the applicability of a compound sentiment score.

As it has been said in Section 3.4, the rules proposed by Consoli et al. [2022a] have some limitations when the ToI is a stock instead of an economic concept. In the syntactic structure of a sentence, economic concepts usually play the role of the object while entities are usually the subject. This difficulties the information extraction process when following the rules defined in Section 2.2.2. Figure 25 shows the computed sentiment scores for *Apple*. Observe that the FiGAS methodology seems to generate a less noisy signal than the SESTM methodology.

[7] **Sharpe ratio range:** https://www.investopedia.com/ask/answers/010815/what-good-sharpe-ratio.asp

| Company | Cap. margin ($B) |
|---|---:|
| Apple | 2092 |
| Amazon | 847.96 |
| Boeing | 122.17 |
| Bank of America | 273.32 |
| Walt Disney | 167.62 |
| Ford | 49.04 |
| Facebook | 562.19 |
| General Electric | 77.84 |
| General Motors | 49.72 |
| Alphabet Inc. | 1107 |
| Google | 1107 |
| Goldman Sachs | 116.40 |
| JP Morgan | 397.00 |
| Morgan Stanley | 145.21 |
| Microsoft | 1658 |
| Netflix | 137.82 |
| AT&T Corp | 136.90 |
| Tesla | 397.03 |
| Wells Fargo | 161.64 |
| Walmart Stores | 386.37 |

Table 3: List of the 20 equities analyzed in this work alongside their capitalization margin in Billions of U.S. dollars.

## 4.2 Granger causality

To look for causality patterns between the stock's sentiment scores and its return values we have applied the Granger causality test. The bidirectional relationship (i.e. causality testing in both directions) is studied to determine the existence of a *contemporaneous correlation* between the target return and it's stock sentiment series. Details on the test definition have already been mentioned in Section 3.8. Figures 16, 17 and 18 show Granger test results for the SESTM, the FiGAS mean-aggregated, and FiGAS sum-aggregated scores respectively. A dot is placed on the date when the null hypothesis has been rejected with a significance level of 0.1.

The three plots show that causality is not constant over the time. However, a causal relationship is observed in several periods; either bidirectionally, sentiment Granger-causing price, or price Granger-causing sentiment. Figures 26, 27, 28 are an alternative visualization of the performance on the Granger causality test for *Apple*. The plots show the p-value obtained in each time window. The horizontal dashed line establishes the threshold below which the null hypothesis is rejected. Figure 26 shows a dominance of the relationship SESTM sentiment Granger-causes price during February and April 2018. Figure 27 reveals that the relationship $FiGAS_m$ sentiment Granger-causes price predominates during the first quarter of 2018 and the second quarter of 2020. In Figure 28, we observe a similar behaviour in April-May 2018 and during July 2019. It seems that the opposed methodologies and even the different aggregation methods carry singular sentiment information. This supports the idea of computing a single sentiment indicator from the indicators obtained with the different methodologies.

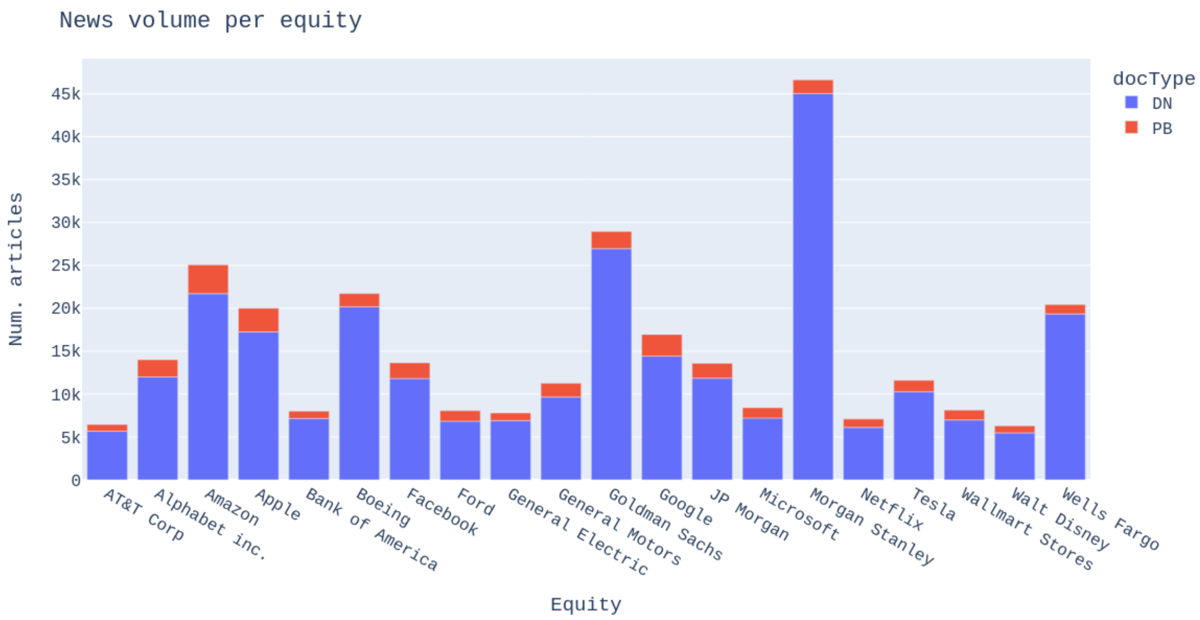


Figure 15: Distribution of the number of news associated with each one of the equities classified by DJN product types.

Figures 19, 21 and 23 show the minimum-lag distribution when the null hypothesis for the Granger causality test in the sentiment-price direction has been rejected. This is interesting to determine how long it takes for news to have a noticeable impact on market prices. The results agree with the ones stated in Yu [2014], which says that the impact of news in the market occurs within 2-5 days. In most of the cases the null hypothesis is rejected with a minimum lag of two days. *Apple*'s refusing lag average is between one and two days. Figures 20, 22 and 24 show the same but considering the relationship price Granger-causes sentiment.

## 4.3 Backtesting strategy

Using the sentiment scores, we have backtested the trading strategy detailed in Section 3.9. The price-based version of the same strategy is used as the baseline. Figures 29, 27 and 31 show the sentiment-based trading strategy performance for Apple. On top, the first plot shows price data with its corresponding STMA and LTMA moving averages. The strategy shown in this plot corresponds to the one using the pair of window lengths that maximizes the Sharpe Ratio. Green and red triangles stand for buying and selling actions respectively. The second plot shows sentiment scores and its corresponding LTMA and STMA moving averages. For sentiment-based strategies, crossovers between moving averages on the second plot trigger buying and selling actions. The two plots at the bottom show the heat maps used for representing the windows sizes that optimize the PnL and the Sharpe ratio. Figure 32 displays the content for the baseline strategy. The only difference on the visual summary is that in Figure 32, the crossovers that drive the performance are the ones seen in the first plot.

Taking *Apple* as an example, the sentiment-based strategies outperform considerably the baseline in terms of Sharpe Ratio an PnL. We observe an increasing of the triggered actions when sentiment

| Company | $FiGAS_s - Ret$ | $FiGAS_m - Ret$ | $SESTM - Ret$ | $SESTM - FiGAS_s$ | $SESTM - FiGAS_m$ |
|---|---|---|---|---|---|
| Apple | -0.023 | -0.024 | -0.015 | 0.037 | 0.032 |
| Amazon | 0.064 | 0.060 | 0.056 | 0.031 | 0.005 |
| Boeing | 0.0534 | 0.061 | 0.087 | 0.083 | **0.123** |
| Bank of America | -0.002 | -0.036 | 0.0216 | -0.044 | -0.045 |
| Walt Disney | -0.064 | -0.073 | 0.050 | 0.034 | 0.027 |
| Ford | -0.005 | 0.036 | 0.057 | 0.042 | 0.052 |
| Facebook | 0.025 | 0.033 | 0.016 | 0.029 | 0.033 |
| General Electric | 0.051 | 0.052 | 0.009 | -0.012 | -0.011 |
| General Motors | -0.028 | -0.024 | -0.065 | -0.048 | -0.063 |
| Alphabet Inc. | 0.095 | **0.103** | -0.021 | 0.029 | 0.026 |
| Google | 0.073 | 0.085 | -0.022 | 0.019 | 0.001 |
| Goldman Sachs | -0.005 | -0.037 | 0.009 | -0.006 | 0.009 |
| JP Morgan | -0.030 | -0.051 | 0.044 | 0.036 | 0.016 |
| Morgan Stanley | -0.035 | -0.065 | 0.060 | -0.040 | -0.068 |
| Microsoft | 0.069 | 0.06 | -0.015 | 0.002 | -0.002 |
| Netflix | -0.003 | -0.014 | 0.038 | 0.056 | 0.046 |
| AT&T Corp | -0.045 | -0.046 | 0.028 | -0.086 | -0.089 |
| Tesla | 0.005 | -0.001 | 0.007 | 0.023 | 0.052 |
| Wells Fargo | 0.034 | 0.051 | -0.058 | 0.031 | -0.046 |
| Wallmart Stores | -0.024 | -0.025 | -0.075 | 0.057 | 0.054 |

Table 4: Spearman correlation coefficients for market returns and sentiment scores for each company. Columns $FiGAS_s - Ret$, $FiGAS_m - Ret$ and $SESTM - Ret$ show the Spearman correlation coefficients between the market return values and the sum-aggregated FiGAS ($FiGAS_s$), mean-aggregated FiGAS ($FiGAS_m$), and SESTM sentiment scores. Columns $SESTM - FiGAS_s$ and $SESTM - FiGAS_m$ include the Spearman correlation coefficients between the SESTM and FiGAS (sum-aggregated and mean-aggregated) sentiment scores.

orchestrates the strategy. Among the different methodologies, the SESTM scores show the maximum PnL percentage and the mean-aggregtaed FiGAS scores reveal a considerably higher sharpe ratio. For now, *Apple*' results of the window size optimization procedure display a huge variability that does not allow us to draw conclusions.

Tables 5, 6 and 7 contain the average PnL, Sharpe Ratio and trades count respectively for different window sizes. The averages have been computed over the 20 well-known companies on which this study focuses. The first two columns indicate the window sizes used for the STMA and LTMA. Alongside, we find the average indicators for the SESTM, mean-aggregated FiGAS ($FiGAS_m$), sum-aggregated FiGAS ($FiGAS_s$) and price-based strategies. The general trend shows a slightly better performance when back-testing the sentiment-based strategies. On average, we also see an increase in the number of transactions when using sentiment scores, which achieves better results even with transaction costs. It seems that, depending on the window size, different SA methodologies outperform the others. However, the optimal values for the different sentiment scores are not similar enough to opt for one pair of values.

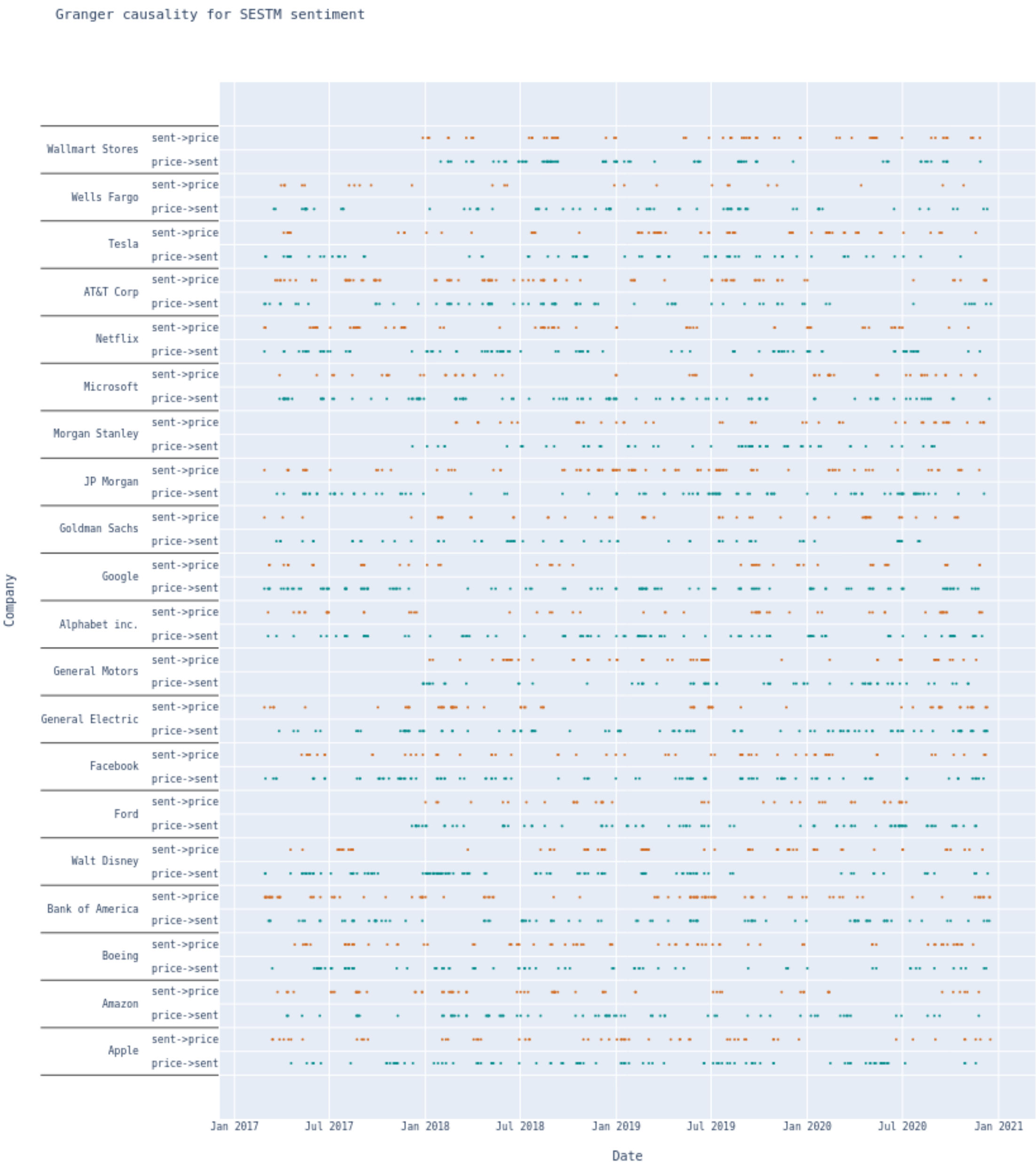


Figure 16: Granger causality test performance for the SESTM sentiment scores in both directions for all the companies. A dot is placed on the date when the null hypothesis has been rejected with a significance level of 0.1.

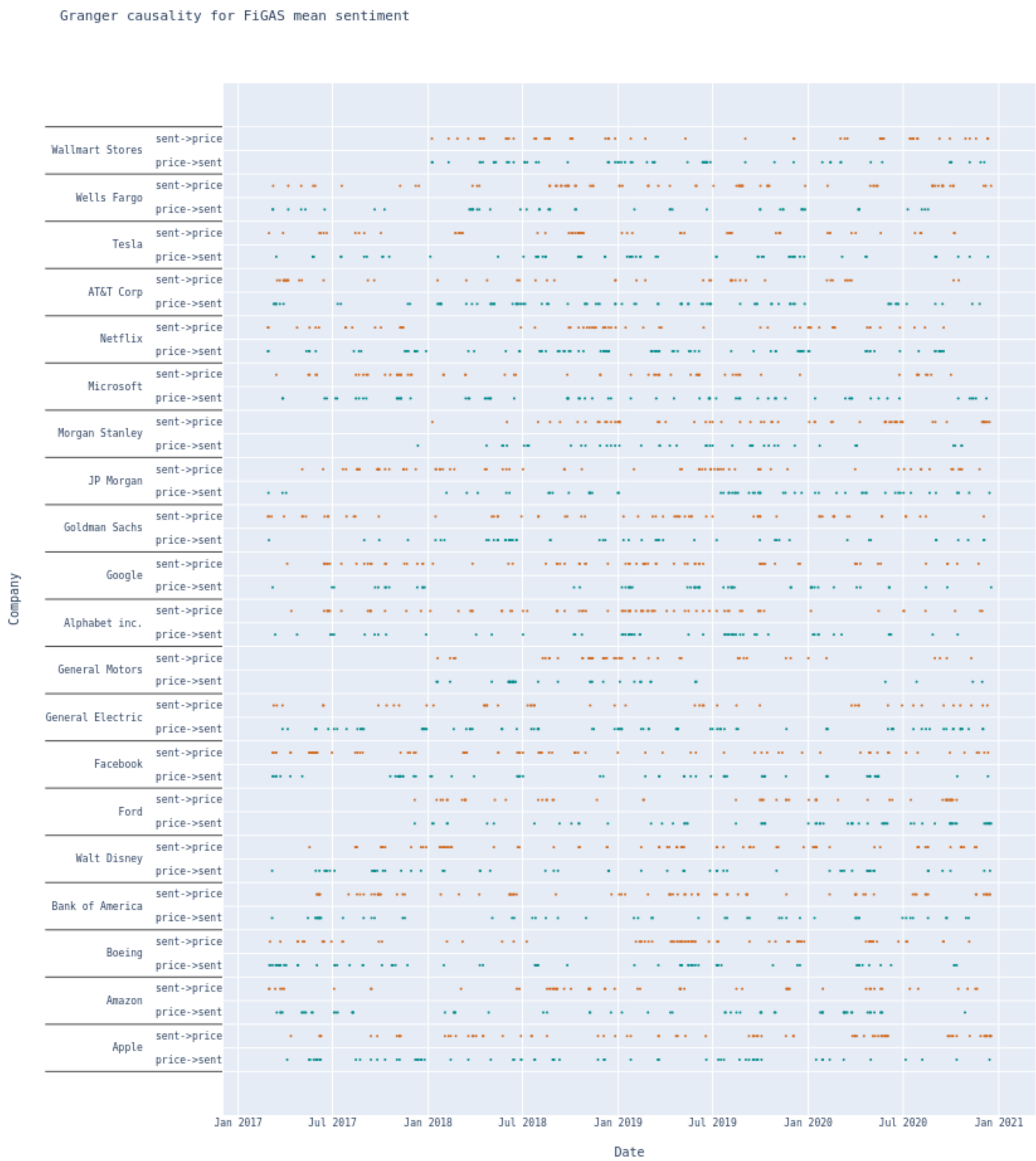


Figure 17: Granger causality test performance for the mean-aggregated FiGAS sentiment scores in both directions for all the companies. A dot is placed on the date when the null hypothesis has been rejected with a significance level of 0.1.

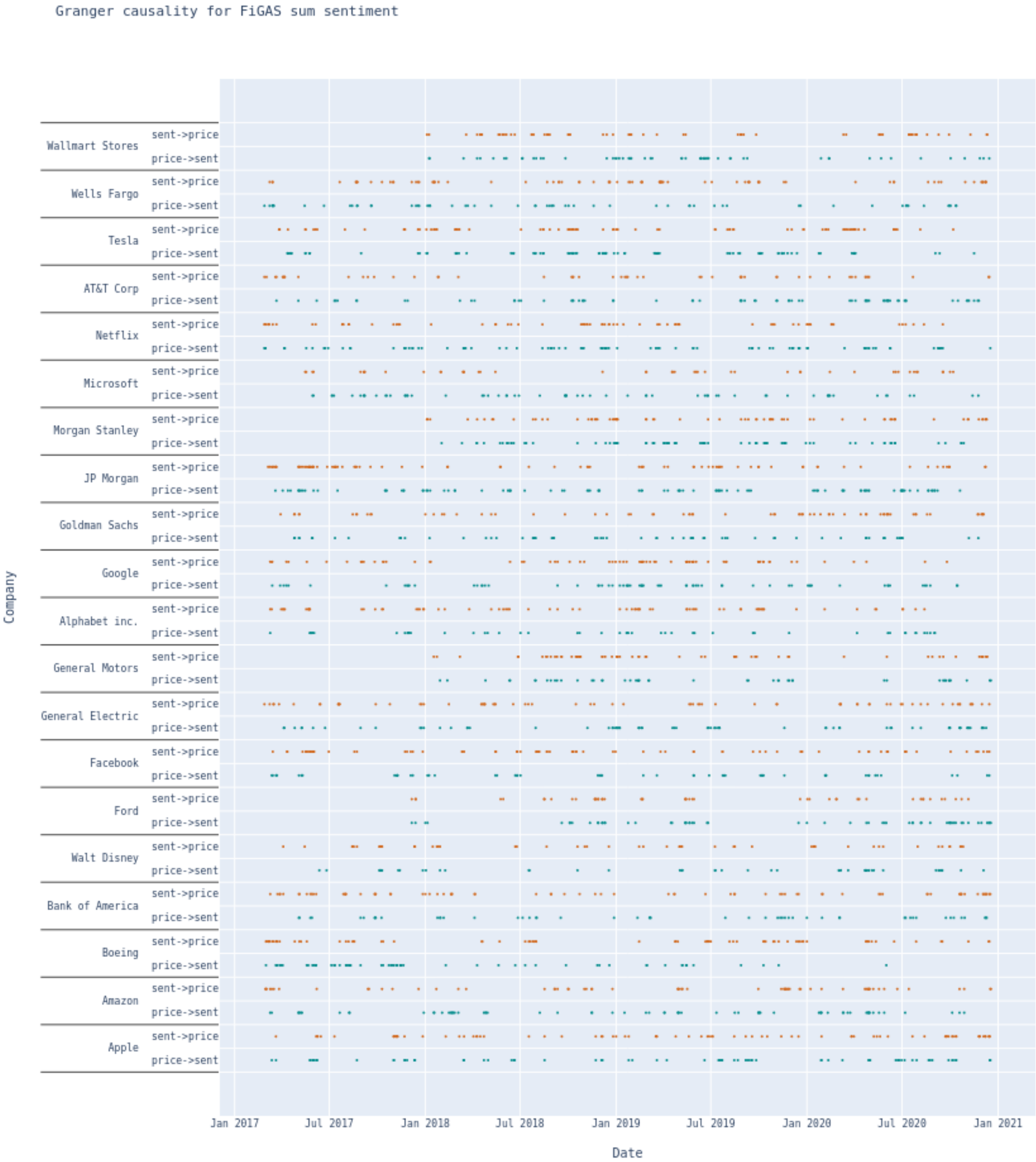


Figure 18: Granger causality test performance for the sum-aggregated FiGAS sentiment scores in both directions for all the companies. A dot is placed on the date when the null hypothesis has been rejected with a significance level of 0.1.

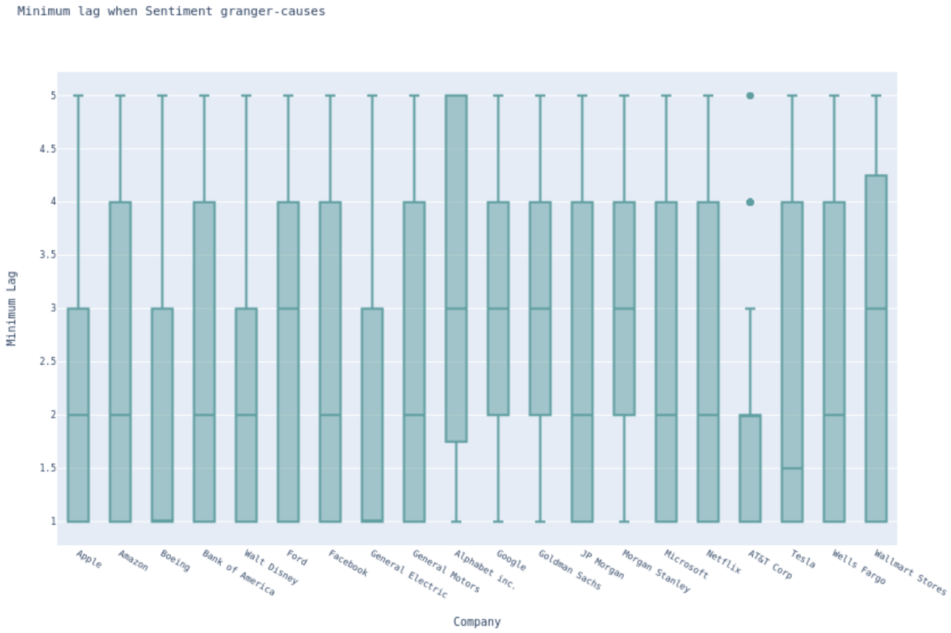


Figure 19: Minimum-lag distribution obtained from the Granger causality test on the relationship: SESTM sentiment scores Granger-cause price.

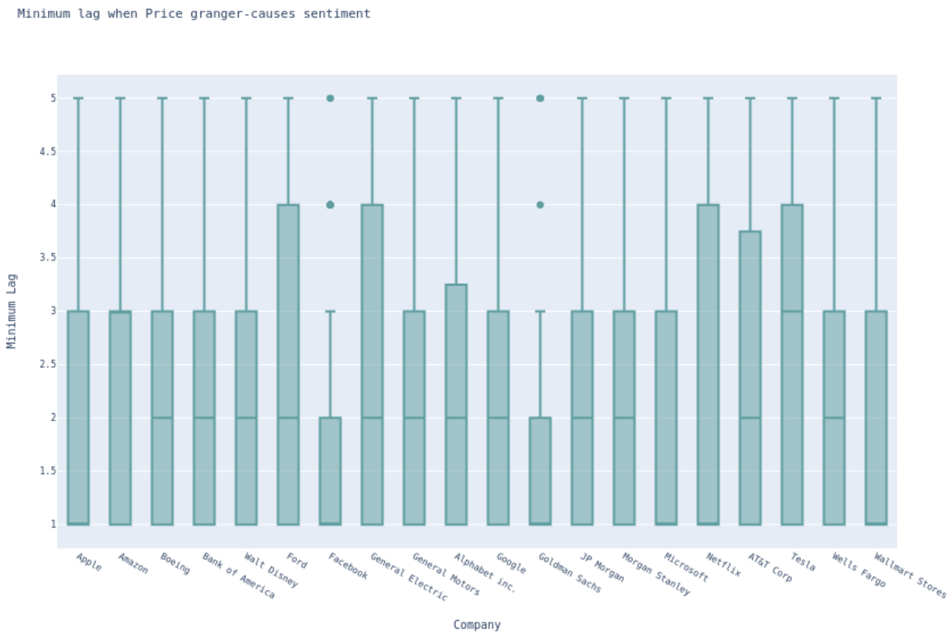


Figure 20: Minimum-lag distribution obtained from the Granger causality test on the relationship: price Granger-causes SESTM sentiment scores.

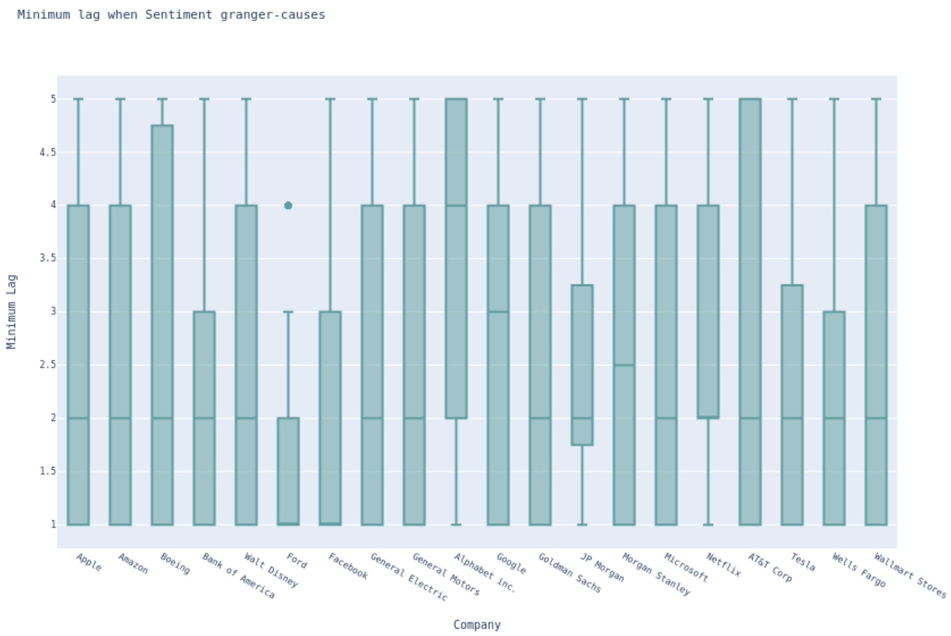


Figure 21: Minimum-lag distribution obtained from the Granger causality test on the relationship: FiGAS mean-aggregated sentiment scores Granger-cause price.

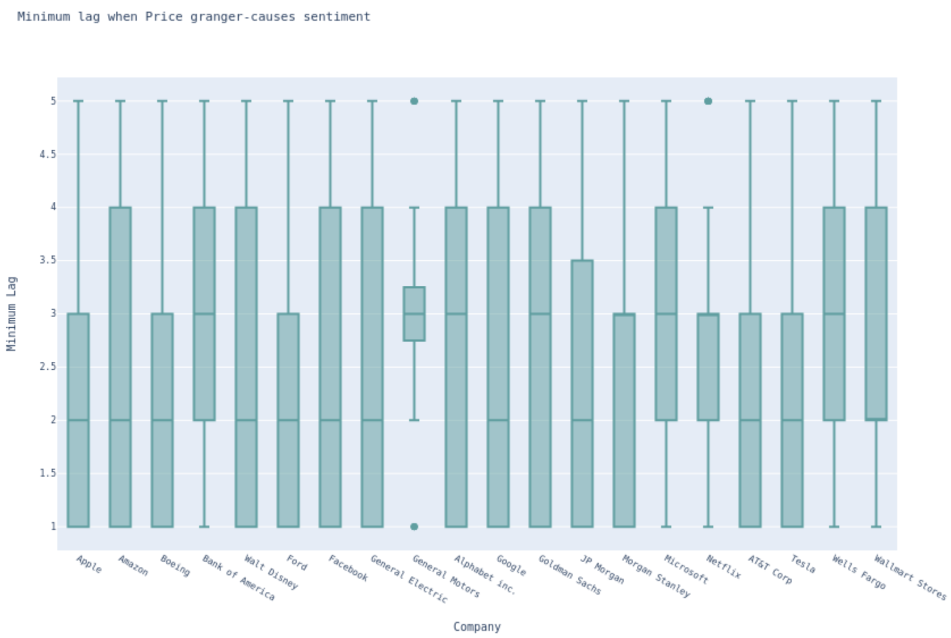


Figure 22: Minimum-lag distribution obtained from the Granger causality test on the relationship: price Granger-causes FiGAS mean-aggregated sentiment scores.

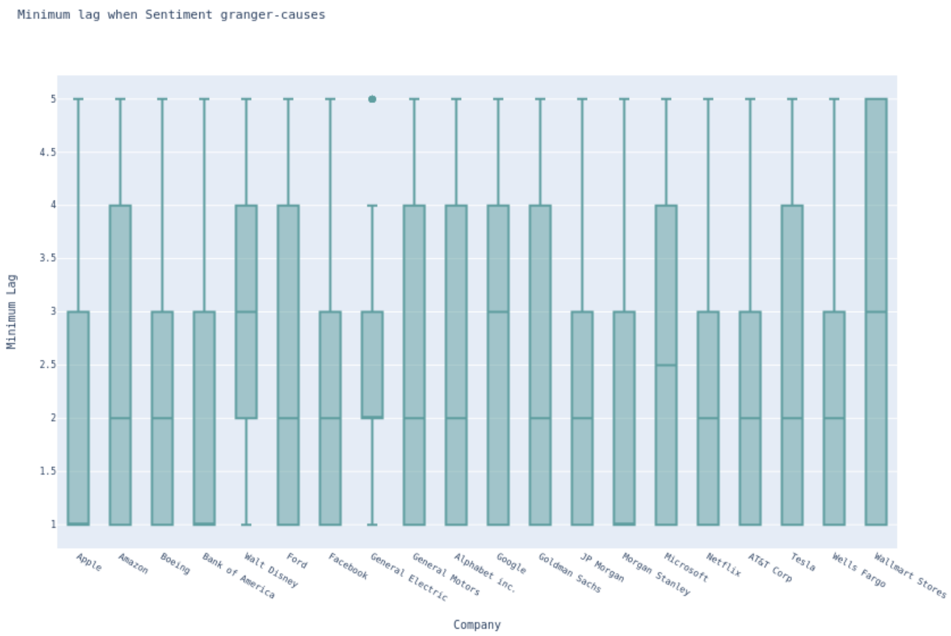


Figure 23: Minimum-lag distribution obtained from the Granger causality test on the relationship: FiGAS sum-aggregated sentiment scores Granger-cause price.

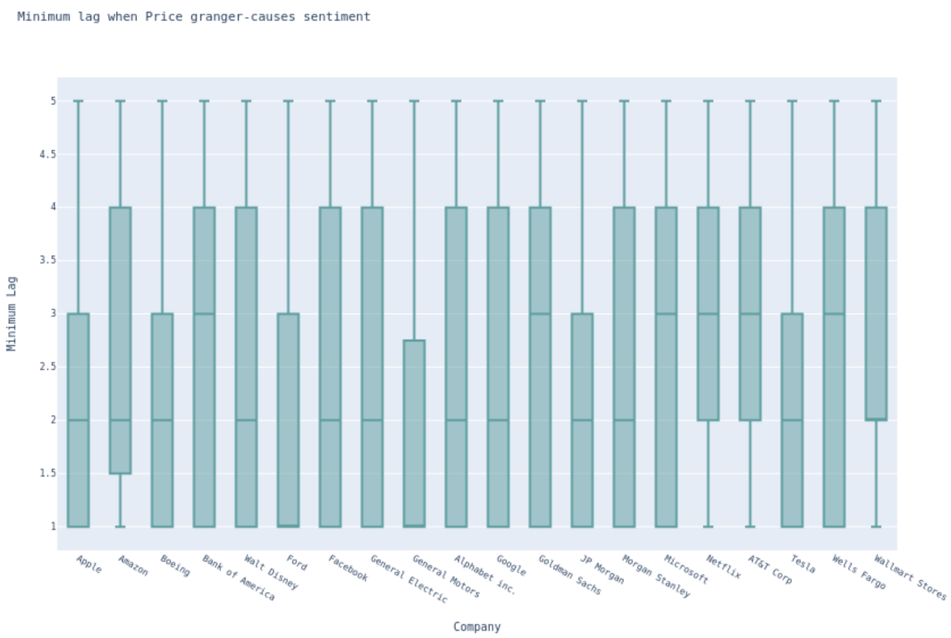


Figure 24: Minimum-lag distribution obtained from the Granger causality test on the relationship: price Granger-causes FiGAS sum-aggregated sentiment scores.

| MA window size | | PnL | | | |
|---|---|---|---|---|---|
| short window | long window | SESTM | $FiGAS_m$ | $FiGAS_s$ | Price-based |
| 5 | 25 | 0.231 | 0.105 | 0.129 | 0.146 |
| 5 | 50 | 0.157 | 0.143 | 0.164 | 0.172 |
| 5 | 100 | 0.088 | 0.151 | 0.185 | 0.129 |
| 10 | 25 | 0.208 | 0.048 | 0.121 | 0.111 |
| 10 | 50 | 0.087 | 0.033 | 0.032 | 0.134 |
| 10 | 100 | 0.077 | 0.089 | 0.068 | 0.124 |
| 15 | 25 | 0.129 | 0.102 | 0.172 | 0.106 |
| 15 | 50 | 0.123 | 0.063 | 0.135 | 0.116 |
| 15 | 100 | 0.093 | 0.133 | 0.079 | 0.097 |
| 20 | 25 | 0.135 | -0.014 | 0.140 | 0.134 |
| 20 | 50 | 0.117 | 0.056 | 0.139 | 0.119 |
| 20 | 100 | 0.128 | 0.184 | 0.115 | 0.100 |

Table 5: Average PnL values for different window sizes over the 20 companies considered in this study for the different backtested strategies.

| MA window size | | Sharpe ratio | | | |
|---|---|---|---|---|---|
| short window | long window | SESTM | $FiGAS_m$ | $FiGAS_s$ | Price-based |
| 5 | 25 | 0.354 | 0.243 | 0.191 | 0.042 |
| 5 | 50 | 0.180 | 0.295 | 0.231 | 0.133 |
| 5 | 100 | 0.116 | 0.285 | 0.279 | 0.121 |
| 10 | 25 | 0.422 | 0.076 | 0.113 | 0.053 |
| 10 | 50 | 0.157 | 0.106 | 0.068 | 0.189 |
| 10 | 100 | 0.186 | 0.199 | 0.164 | 0.135 |
| 15 | 25 | 0.328 | 0.069 | 0.218 | 0.105 |
| 15 | 50 | 0.137 | 0.129 | 0.079 | 0.156 |
| 15 | 100 | 0.132 | 0.304 | 0.201 | 0.089 |
| 20 | 25 | 0.114 | 0.069 | 0.099 | 0.124 |
| 20 | 50 | 0.098 | 0.043 | 0.155 | 0.178 |
| 20 | 100 | 0.199 | 0.310 | 0.268 | 0.091 |

Table 6: Average Sharpe Ratio values for different window sizes over the 20 companies considered in this study for the different backtested strategies

| MA window size | | Trades count | | | |
|---|---|---|---|---|---|
| short window | long window | SESTM | $FiGAS_m$ | $FiGAS_s$ | Price-based |
| 5 | 25 | 74 | 78 | 77 | 26 |
| 5 | 50 | 68 | 75 | 73 | 16 |
| 5 | 100 | 62 | 71 | 70 | 11 |
| 10 | 25 | 57 | 58 | 60 | 21 |
| 10 | 50 | 49 | 51 | 50 | 12 |
| 10 | 100 | 41 | 47 | 47 | 8 |
| 15 | 25 | 55 | 59 | 57 | 22 |
| 15 | 50 | 40 | 44 | 41 | 10 |
| 15 | 100 | 34 | 38 | 37 | 6 |
| 20 | 25 | 73 | 80 | 77 | 26 |
| 20 | 50 | 35 | 40 | 37 | 10 |
| 20 | 100 | 31 | 33 | 33 | 6 |

Table 7: Average trades for different window sizes over the 20 companies considered in this study for the different backtested strategies

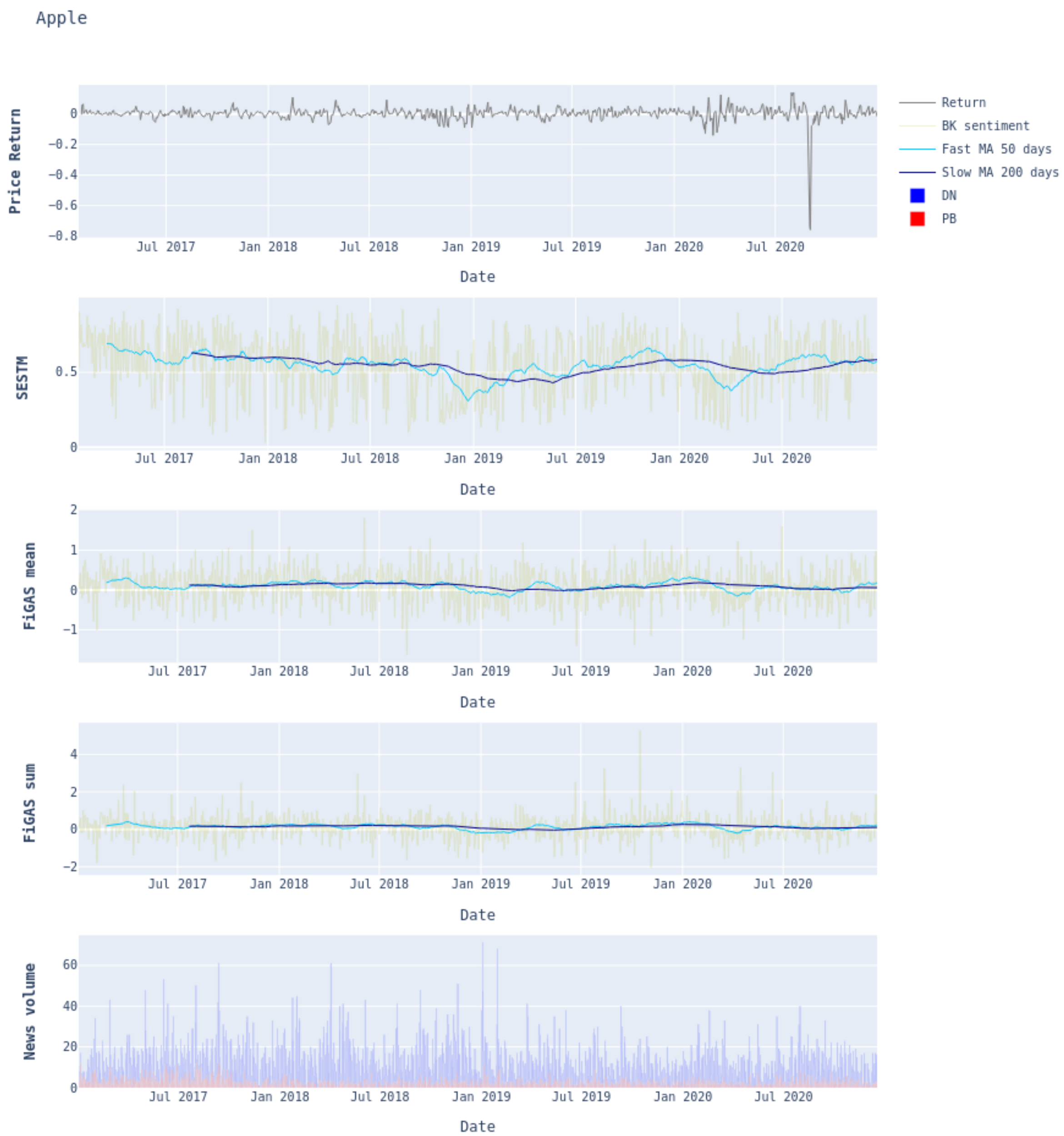


Figure 25: Obtained scores for *Apple*. The first line-plot shows market returns for *Apple* from 2017 to 2020. Below we can find the sentiment scores obtained using the three different procedures: the SESTM methodology, the sum-aggregated FiGAS methodology, and the mean-aggregated FiGAS methodology. On top, a 50-days LTMA and a 20-day STMA are drawn in light and dark blue, respectively. The final bar-plot shows the daily volume of news for *Apple* depending on the DJN product type.

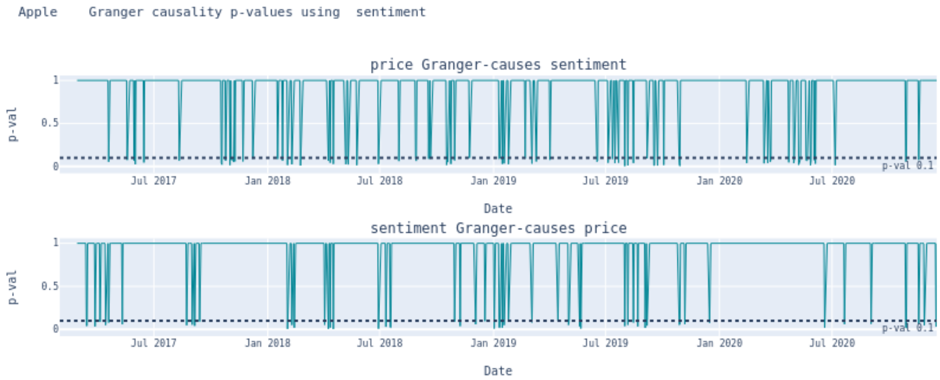


Figure 26: P-values for Granger causality test using SESTM sentiment scores computed for Apple.

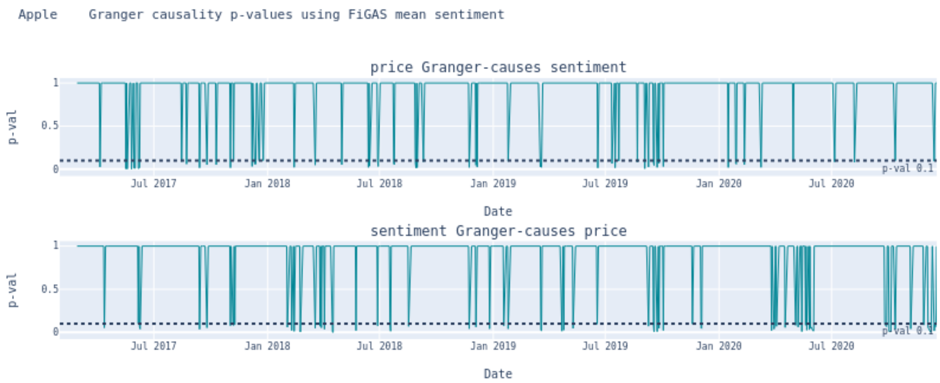


Figure 27: P-values for Granger causality test using mean-aggregated FiGAS sentiment scores computed for Apple.

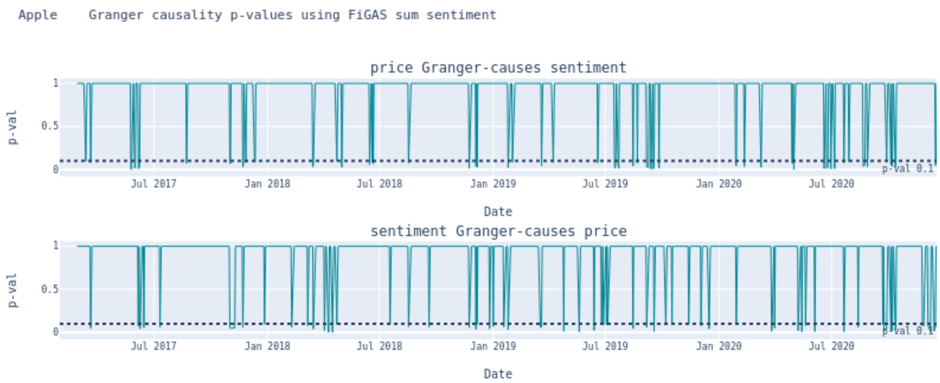


Figure 28: P-values for Granger causality test using sum-aggregated FiGAS sentiment scores computed for Apple.


Figure 29: Trading strategy for *Apple* guided by the SESTM sentiment scores. The first line plot shows price market values for *Apple* with its corresponding LTMA and STMA (25-day and 5-day respectively). Green and red triangles point out buying and selling actions. The second line plot shows the sentiment scores and its corresponding LTMA and STMA (25-day and 5-day respectively). The two heat maps at the bottom show the PnL and Sharpe Ratio values for different window sizes. The strategy shown in the line plots corresponds to the one using the paired values of window lengths that maximizes the Sharpe Ratio.


Figure 30: Trading strategy for *Apple* guided by the mean-aggregated FiGAS sentiment scores. The first line plot shows price market values for *Apple* with its corresponding LTMA and STMA (100-day and 15-day respectively). Green and red triangles point out buying and selling actions. The second line plot shows the sentiment scores and its corresponding LTMA and STMA (100-day and 15-day respectively). The two heat maps at the bottom show the PnL and Sharpe Ratio values for different window sizes. The strategy shown in the line plots corresponds to the one using the paired values of window lengths that maximizes the Sharpe Ratio.


Figure 31: Trading strategy for *Apple* guided by the sum-aggregated FiGAS sentiment scores. The first line plot shows price market values for *Apple* with its corresponding LTMA and STMA (25-day and 15-day respectively). Green and red triangles point out buying and selling actions. The second line plot shows the sentiment scores and its corresponding LTMA and STMA (25-day and 15-day respectively). The two heat maps at the bottom show the PnL and Sharpe Ratio values for different window sizes. The strategy shown in the line plots corresponds to the one using the paired values of window lengths that maximizes the Sharpe Ratio.

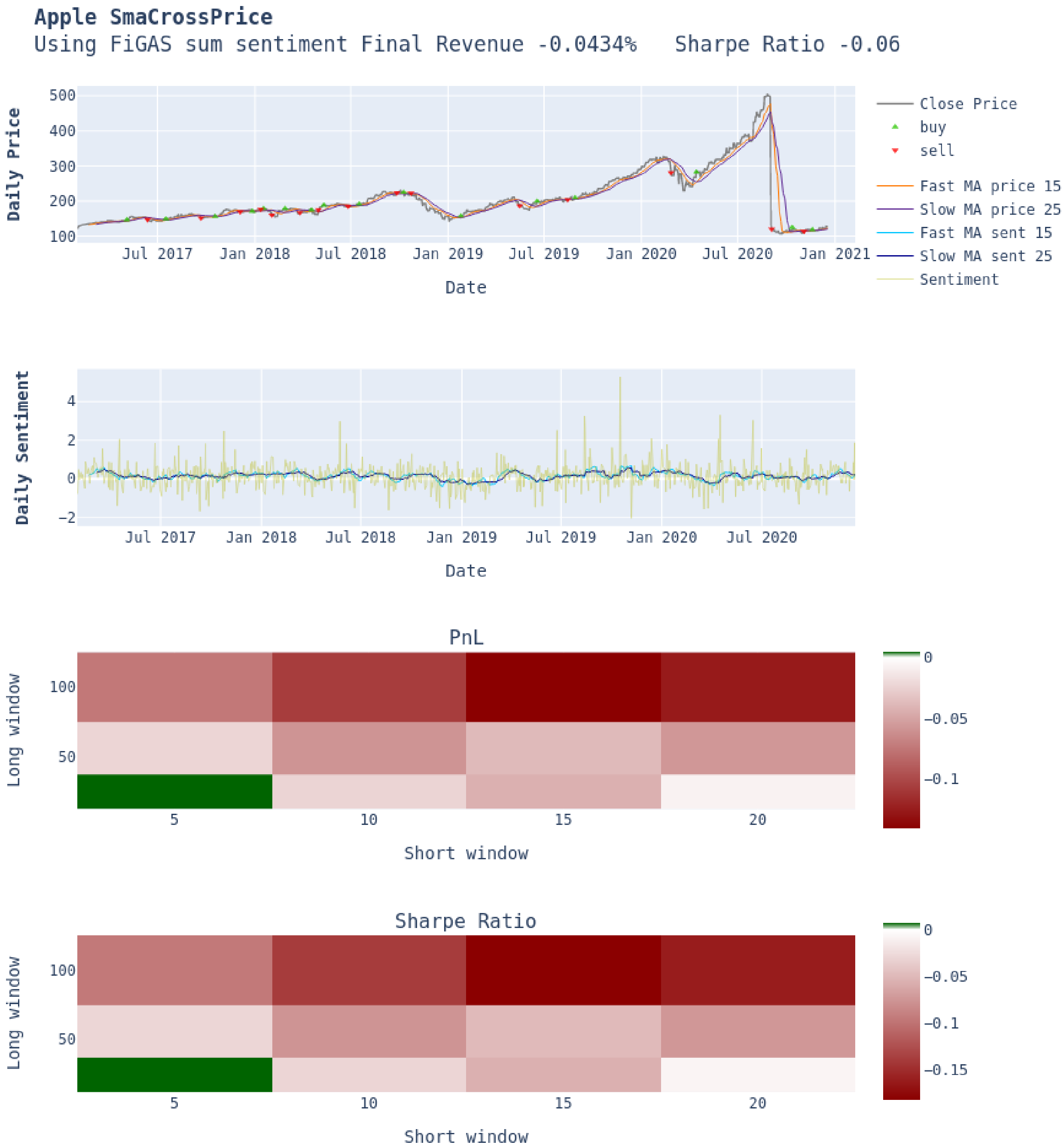


Figure 32: Price-based trading strategy for *Apple*. The first line plot shows price market values for *Apple* with its corresponding LTMA and STMA (25-day and 15-day respectively). Green and red triangles point out buying and selling actions. The second line plot shows the FiGAS sum-aggregated sentiment scores and its corresponding LTMA and STMA (25-day and 15-day respectively). The two heat maps at the bottom show the PnL and Sharpe Ratio values for different window sizes. The strategy shown in the line plots corresponds to the one using the paired values of window lengths that maximizes the Sharpe Ratio.

## 4.4 Dictionary comparison

As a final qualitative comparison between both methodologies, we retrieved the vocabulary from SESTM modeling step [Ke et al., 2019] and compared it to the SentiBigNomics [Consoli et al., 2022a] and LMD [Loughran and McDonald, 2011] dictionaries. This gives us an idea of the robustness of the former approach regarding the sentiment-charged word selection. Table 8 provides the positive, negative and neutral word distribution of each dictionary.

| | Total words | Negative words | Positive words | Neutral words |
|---|---|---|---|---|
| SESTM dict | 97 | 47 | 50 | 0 |
| SentiBigNomics (used in FiGAS) | 7296 | 3554 | 2582 | 1160 |
| LMD | 86531 | 2352 | 357 | 83822 |

Table 8: Total, positive, negative and neutral words distribution of the SESTM, SentiBigNomics and LMD dictionaries.

The dimensions and the distribution are quite different for each dictionary. However, the data is enough to count how many matches there are between the dictionaries to get a qualitative idea of the differences and similarities.

- **SESTM vocabulary vs. LMD** (8 matches)

  Negative matching words: *abuse, anticompetitive, bad, criticize, loss*

  Positive matching words: *outperform, critical, profitable*

- **SentiBigNomics vs. LMD** (1277 matches)

  Negative matching words: *abandon, aberrant, slowing, suffering, threatening, worsening,...*

  Positive matching words: *able, abundance, abundant, exciting, impressed, greatness,...*

- **SESTM vs. SentiBigNomics** (20 matches)

  Negative matching words: *abuse, anticompetitive, bad, criticize, hard, loss, shed, slide, slip, slump, tumble*

  Positive matching words: *climb, double, net, outperform, profitable, protection, rally, settle, surge*

# 5. Conclusions and future work

SA is a very active field of research. Therefore, defining a sentiment extraction strategy is not a trivial matter. From time to time new approaches appear, and this means that the development of these technologies must constantly update. In general, these methodologies have to consider other external factors, such as the volume of information or the scalability of the implementations. Above all, the content where they will be applied plays a key factor when designing the methodology. This work integrates the study of linguistics within NLP, economics, and data modeling. These three fields are already in themselves broad fields of study and research. In this work, we have understood the complexity of each one of them developing a product that is up to the task in each field.

The two SA methodologies analyzed in this work are a very valuable proof of concept for SA in finance. In this work, we have used a restricted amount of data to mitigate the computational cost. Sticking to DJN also reduces the amount of information we are dealing with. Nowadays, there are millions of alternative sentiment-laden text sources that would be worth including in any SA pipeline for finance, for instance Twitter or online blogs.

The main limitation of the SESTM model is its low adaptability over time. Syntax rules depend only on the text itself and can be easily updated and explained. On the contrary, the SESTM methodology builds up a model that eventually needs to be updated because the vocabulary composing it is no longer significant for the news. This forces the practitioners to update the model from time to time. However, the robustness which supports the methodology regarding statistics and optimization evokes more than significant results. On the other side, FiGAS scores face the challenge of interpreting the raw text similarly to how a human would do it through a syntactic and semantic analysis. This implies a high degree of complexity when working with financial articles, where we find technicalities, sentences in active and passive voice, elliptical subjects, double negatives, and other complex structures. In the case of FiGAS, it is worth noting the importance of applying the extended set of rules. The most valuable contribution of the new set of rules is that it does not focus on one type of pattern (such as the surroundings of named entities in a text), but it applies the rules recursively, which makes it a highly flexible methodology. Using the first version of the rules, the results included lots of zero-scored articles. All in all, it is hard to choose one method over the other. The possibility of combining the scores obtained with both methodologies, besides proposing different aggregation metrics for different granularity levels, remains open to study.

In terms of the TA application, it seems that basing a trading strategy solely on sentiment series is a pretty presumptuous technique. It would be interesting to carry out a study with more complex trading strategies that combine indicators based on price with those based on sentiment and include certain limits to reduce the risk of investments and large losses (this is often achieved by defining the target and stop values based on the price series that trigger closing trade actions when they are reached). Nevertheless, the fact that the presented sentiment-based strategies show similar results to the price-based strategies used as the baseline suggests that sentiment includes information about price trends. This induces us to think that more complex strategies combining both series would outperform the strategies presented in this work.

## 5.1 Future work

The following lines introduce some ideas that are considered important to improve the quality of the sentiment indicators.

There is a general rule of thumb in data science that asserts: the more data, the better. Focusing on data from 20 stocks from 2017 to 2020 limits the generalization capabilities of the SESTM model. Also, it reduces the robustness of the drawn conclusions. Extending the data coverage for a higher amount of companies and going further in time is recommended to build more reliable indicators.

In the text pre-processing step, many filtering options have been discarded. When matching stock symbols with recognized entities, the literature encouraged us to opt for the *Levenshtein distance*, but a comparative study of different string similarity metrics is very interesting open research line in this context. Also, there is a huge variability on possible aggregation procedures for the sentiment scores. An in-depth study of the different aggregation procedures could be very valuable to deepen the use of these indicators in trading strategies, which are often orchestrated at different time frequencies (weekly, daily, hourly,...).

Regarding the SESTM modeling, we would like to analyze the impact of adding a weighting score to the words in the text. For instance, to give more importance to the words that appear in the article's caption. The best approximation we though of is to double-count the words that appear in captions when counting appearances in the frequency matrix, but that is a pretty scruffy solution. It turns out that the main limitation of this approach is that one can not ponder words since it works with how frequently they appear but not with their context.

For the FiGAS implementation, it would be very convenient to study the syntactic rules in depth by a team of linguistic experts. Surely this could help refine the rules used and even add new rules.

A deeper study on correlation and Granger causality tests would add value to the computed scores. The results presented in this work could be considered preliminary, but a more exhaustive analysis would be needed to find higher correlation values and general lag patterns. It would also be interesting to work with a version of the Granger causality test that does not assume linearity, and distribution-free.

We suggest backtesting the sentiment scores with other trading strategies. In particular to combine price-based indicators with sentiment-based indicators to obtain better results. It is probably a good idea to develop this work hand in hand with TA specialists.

We recommend performing an analysis of the usability of the sentiment scores in other use cases beyond guiding a trading strategy. For example, it could be interesting to test their performance to predict the asset's market price, or to quantify financial risk towards a financial instrument.

# References

Achelis, S. B. (2001). Technical analysis from a to z.

Akhtar, M. S., Ekbal, A., and Cambria, E. (2020). How intense are you? predicting intensities of emotions and sentiments using stacked ensemble [application notes]. *IEEE Computational Intelligence Magazine*, 15(1):64–75.

Arratia, A. (2014). *Computational finance an introductory course with R*. Atlantis Press.

Baccianella, S., Esuli, A., and Sebastiani, F. (2010). Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*.

BackTrader (2022). Welcome to backtrader!

Benesty, J., Chen, J., Huang, Y., and Cohen, I. (2009). Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer.

Cambria, E., Speer, R., Havasi, C., and Hussain, A. (2010). Senticnet: A publicly available semantic resource for opinion mining. In *AAAI Fall Symposium: Commonsense Knowledge*.

Chunli, X., Zhang, J., Li, F., Fei, H., and Ji, D. (2022). A semantic and syntactic enhanced neural model for financial sentiment analysis. *Information Processing & Management*, 59:102943.

Consoli, S., Barbaglia, L., and Manzan, S. (2022a). Fine-grained, aspect-based sentiment analysis on economic and financial lexicon. *Knowledge-Based Systems*, 247:108781.

Consoli, S., Barbaglia, L., and Manzan, S. (2022b). Sentibignomics.

Consoli, S., Barbaglia, L., and Manzan, S. (2022c). Sentibignomics.

Dridi, A., Atzeni, M., and Reforgiato Recupero, D. (2019). Finenews: fine-grained semantic sentiment analysis on financial microblogs and news. *International Journal of Machine Learning and Cybernetics*, 10(8):2199–2207.

Granger, C. W. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society*, pages 424–438.

Honnibal, M. and Montani, I. (2017). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Hota, H., Sharma, D. K., and Verma, N. (2021). 14 - lexicon-based sentiment analysis using twitter data: a case of covid-19 outbreak in india and abroad. In Kose, U., Gupta, D., de Albuquerque, V. H. C., and Khanna, A., editors, *Data Science for COVID-19*, pages 275–295. Academic Press.

Ke, Z. T., Kelly, B. T., and Xiu, D. (2019). Predicting returns with text data. Technical report, National Bureau of Economic Research.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.

LLC., D. (2022). Digital ocean.

Loughran, T. and McDonald, B. (2011). When is a liability not a liability? textual analysis, dictionaries, and 10-ks. *The Journal of Finance*, 66(1):35–65.

Medhat, W., Hassan, A., and Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, 5(4):1093–1113.

MySQL (2022). Mysql developer zone.

pandas development team, T. (2020). pandas-dev/pandas: Pandas.

Porta Vallés, M. (2020). A supervised learning framework for news-based sentiment scores conditioned to financial returns. Master's thesis, Universitat Politècnica de Catalunya.

Ramshaw, L. A. and Marcus, M. P. (1995). Text chunking using transformation-based learning.

SA, D. B. (2022). Historical data feed :: Dukascopy bank sa: Swiss forex bank: Ecn broker: Managed accounts: Swiss fx trading platform.

Seabold, S. and Perktold, J. (2010). statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*.

Sharpe, W. F. (1994). The sharpe ratio. *The Journal of Portfolio Management*, 21(1):49–58.

Spearman, C. (1961). The proof and measurement of association between two things.

UD, U. D. (2022a). Universal dependencies.

UD, U. D. (2022b). Universal pos tags.

Yu, X. (2014). *Analysis of new sentiment and its application to finance*. PhD thesis, Brunel University.

# A. Appendix

## A.1 Results

Due to space limitations, only two other companies have been selected to show the detailed plots of the results of their analysis in this Appendix. The elected equities are Bank of America and Tesla. Readers interested in visualizing the plots of results of the the analysis for all the 20 companies, can request these directly from the authors.

### A.1.1 Descriptive analysis

Data overview for Bank of America and Tesla. The first plot shows market returns for the corresponding equity from 2017 to 2020. Below we can find the sentiment scores obtained using the three different procedures: the SESTM methodology, the sum-aggregated FiGAS methodology, and the mean-aggregated FiGAS methodology. On top, a 50-days LTMA and a 20-day STMA are drawn in light and dark blue, respectively. The final bar-plot shows the daily volume of news for the corresponding equity depending on the DJN product type.


Figure 33: Obtained scores for Bank of America. The first line-plot shows market returns for Bank of America from 2017 to 2020. Below we can find the sentiment scores obtained using the three different procedures: the SESTM methodology, the sum-aggregated FiGAS methodology, and the mean-aggregated FiGAS methodology. On top, a 50-days LTMA and a 20-day STMA are drawn in light and dark blue, respectively. The final bar-plot shows the daily volume of news for Bank of America depending on the DJN product type.

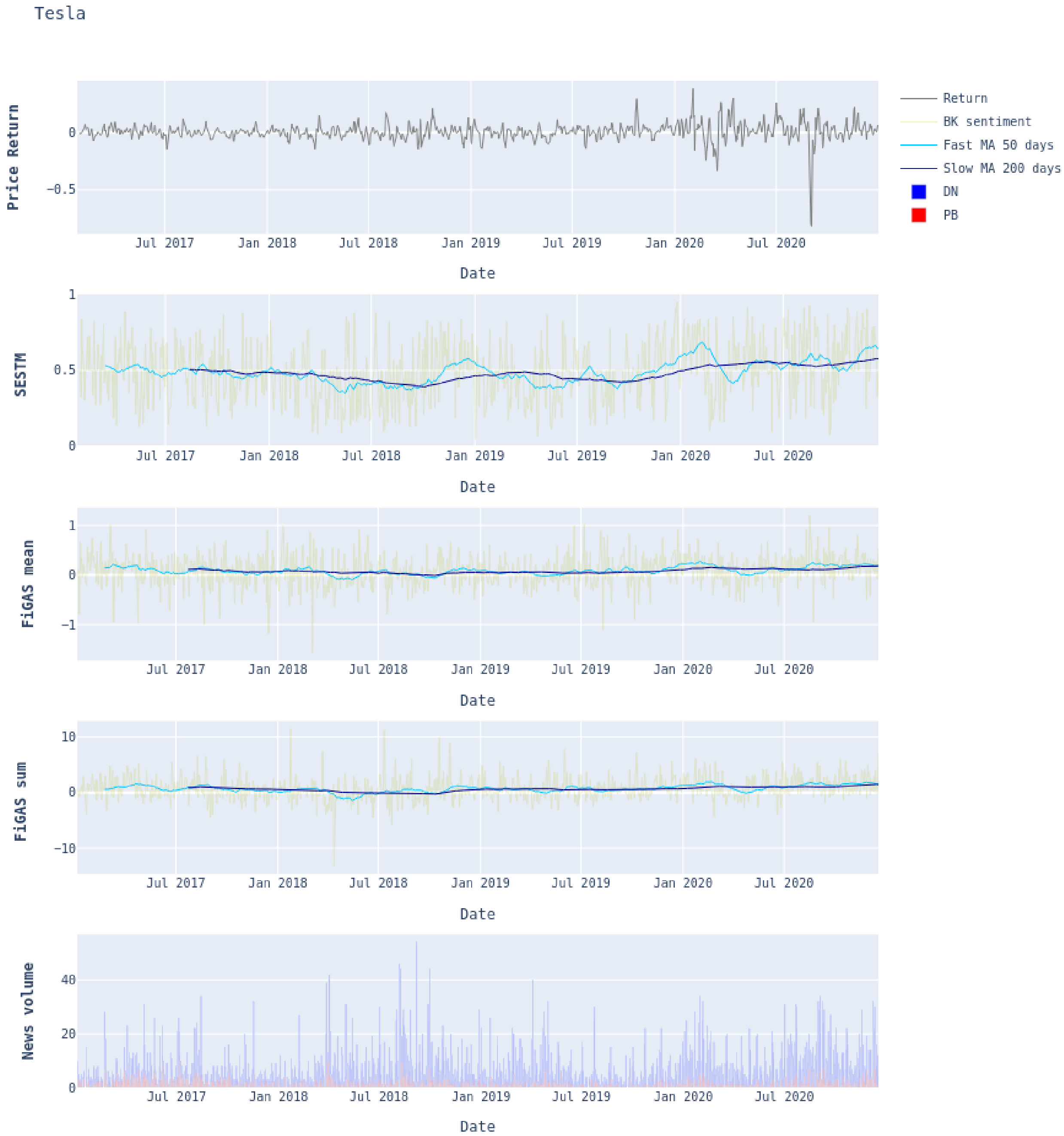


Figure 34: Obtained scores for Tesla. The first line-plot shows market returns for Tesla from 2017 to 2020. Below we can find the sentiment scores obtained using the three different procedures: the SESTM methodology, the sum-aggregated FiGAS methodology, and the mean-aggregated FiGAS methodology. On top, a 50-days LTMA and a 20-day STMA are drawn in light and dark blue, respectively. The final bar-plot shows the daily volume of news for Tesla depending on the DJN product type.

### A.1.2 Granger causality

Alternative visualization of the performance on the Granger causality test for Bank of America and Tesla. The plots show the p-value obtained in each time window. The horizontal dashed line establishes the threshold below which the null hypothesis is rejected.

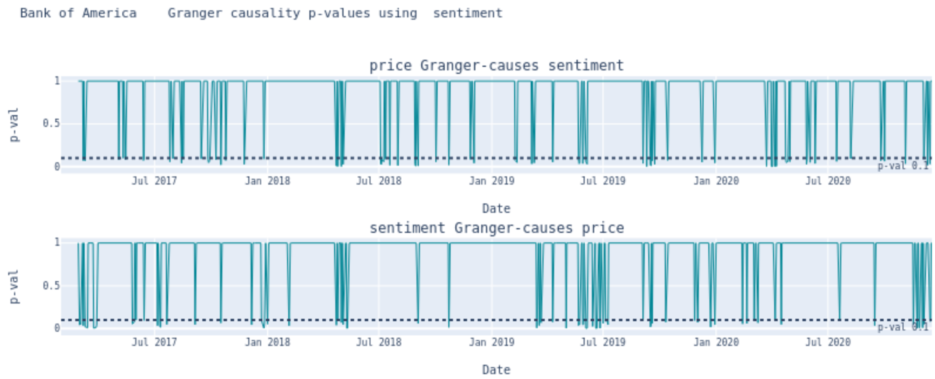


Figure 35: P-values for Granger causality test using SESTM sentiment scores computed for Bank of America.

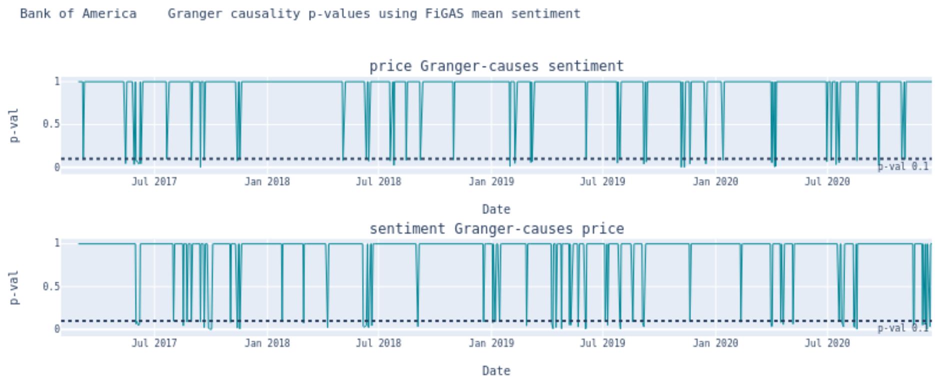


Figure 36: P-values for Granger causality test using mean-aggregated FiGAS sentiment scores computed for Bank of America.

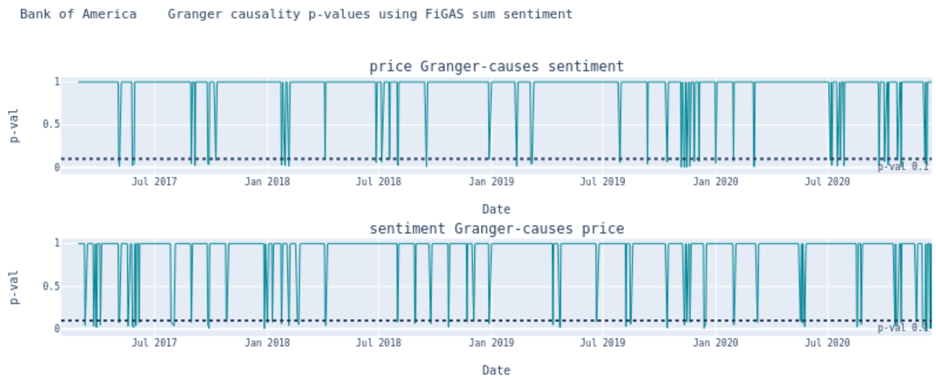


Figure 37: P-values for Granger causality test using sum-aggregated FiGAS sentiment scores computed for Bank of America.

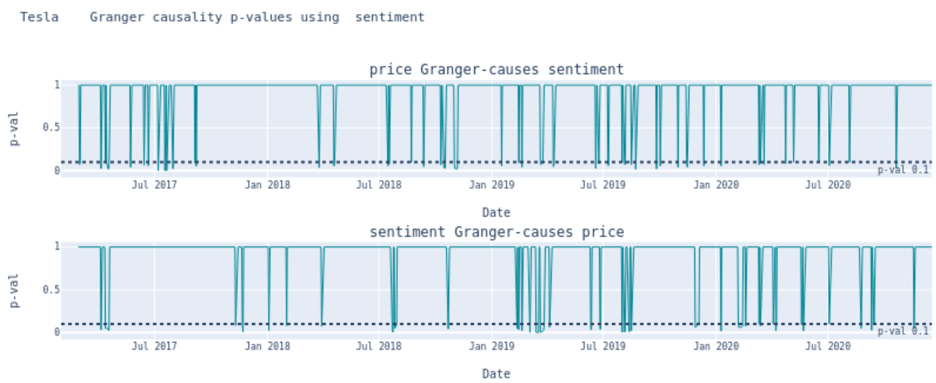


Figure 38: P-values for Granger causality test using SESTM sentiment scores computed for Tesla.

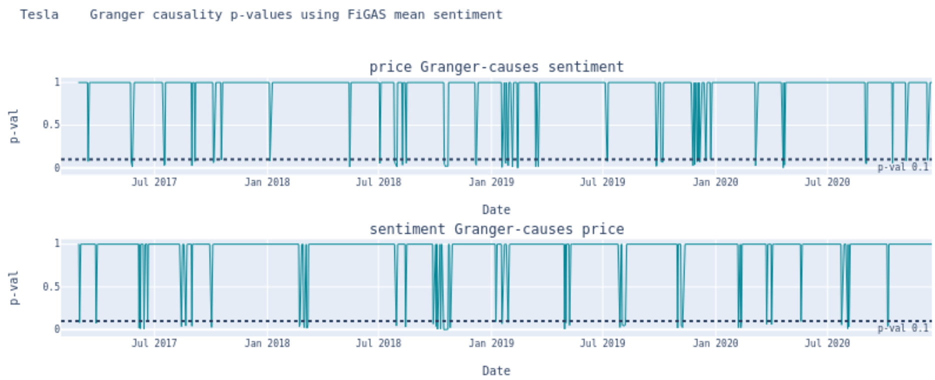


Figure 39: P-values for Granger causality test using mean-aggregated FiGAS sentiment scores computed for Tesla.

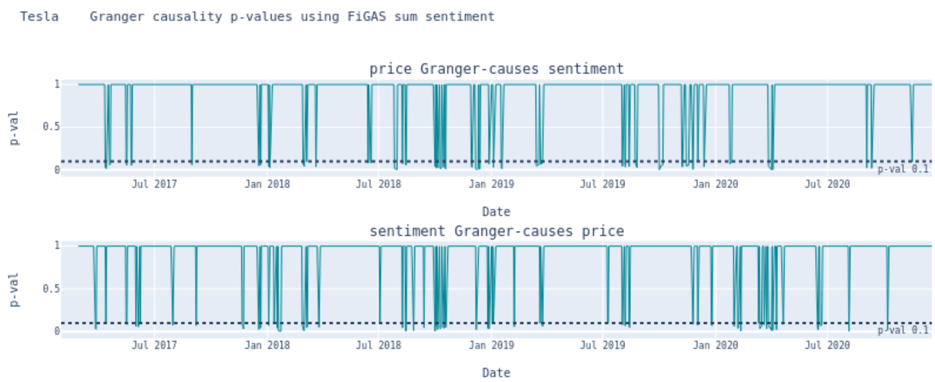


Figure 40: P-values for Granger causality test using sum-aggregated FiGAS sentiment scores computed for Tesla.

### A.1.3 Backtesting strategies

Sentiment-based and price-based strategies visual summary. On top, the first plot shows price data with its corresponding STMA and LTMA moving averages. The strategy shown in this plot corresponds to the one using the pair of window lengths that maximizes the Sharpe Ratio. Green and red triangles stand for buying and selling actions respectively. The second plot shows sentiment scores and its corresponding LTMA and STMA moving averages. For sentiment-based strategies, crossovers between moving averages on the second plot trigger buying and selling actions. The two plots at the bottom show the heat maps used for representing the windows sizes that optimize the PnL and the Sharpe ratio. Note that in price-based strategies, the crossovers that drive the performance are the ones seen in the first plot.

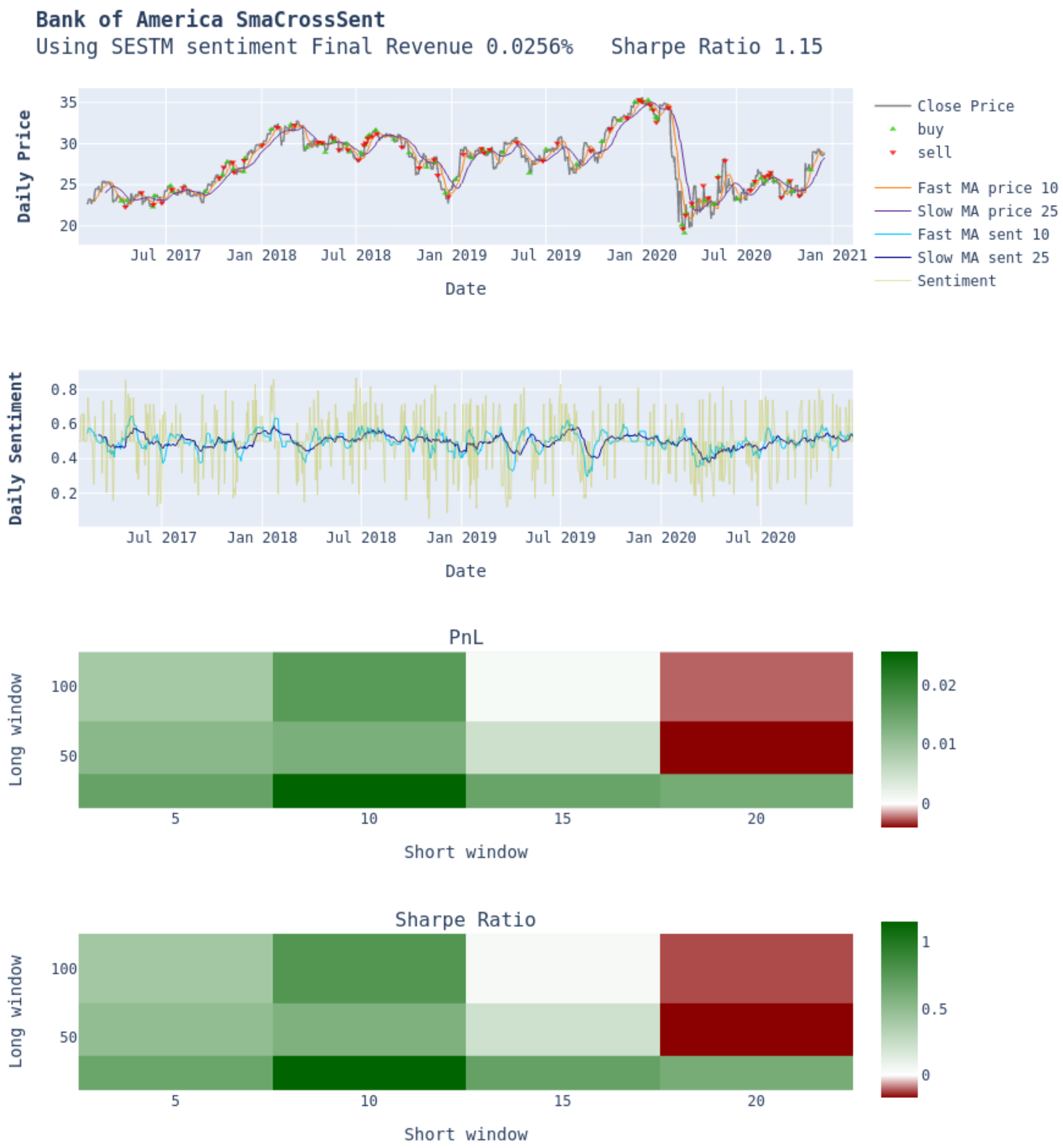


Figure 41: Trading strategy for Bank of America guided by the SESTM sentiment scores. The first line plot shows price market values for Bank of America with its corresponding LTMA and STMA (25-day and 10-day respectively). Green and red triangles point out buying and selling actions. The second line plot shows the sentiment scores and its corresponding LTMA and STMA (25-day and 10-day respectively). The two heat maps at the bottom show the PnL and Sharpe Ratio values for different window sizes. The strategy shown in the line plots corresponds to the one using the paired values of window lengths that maximizes the Sharpe Ratio.


Figure 42: Trading strategy for Bank of America guided by the mean-aggregated FiGAS sentiment scores. The first line plot shows price market values for Bank of America with its corresponding LTMA and STMA (100-day and 5-day respectively). Green and red triangles point out buying and selling actions. The second line plot shows the sentiment scores and its corresponding LTMA and STMA (100-day and 5-day respectively). The two heat maps at the bottom show the PnL and Sharpe Ratio values for different window sizes. The strategy shown in the line plots corresponds to the one using the paired values of window lengths that maximizes the Sharpe Ratio.

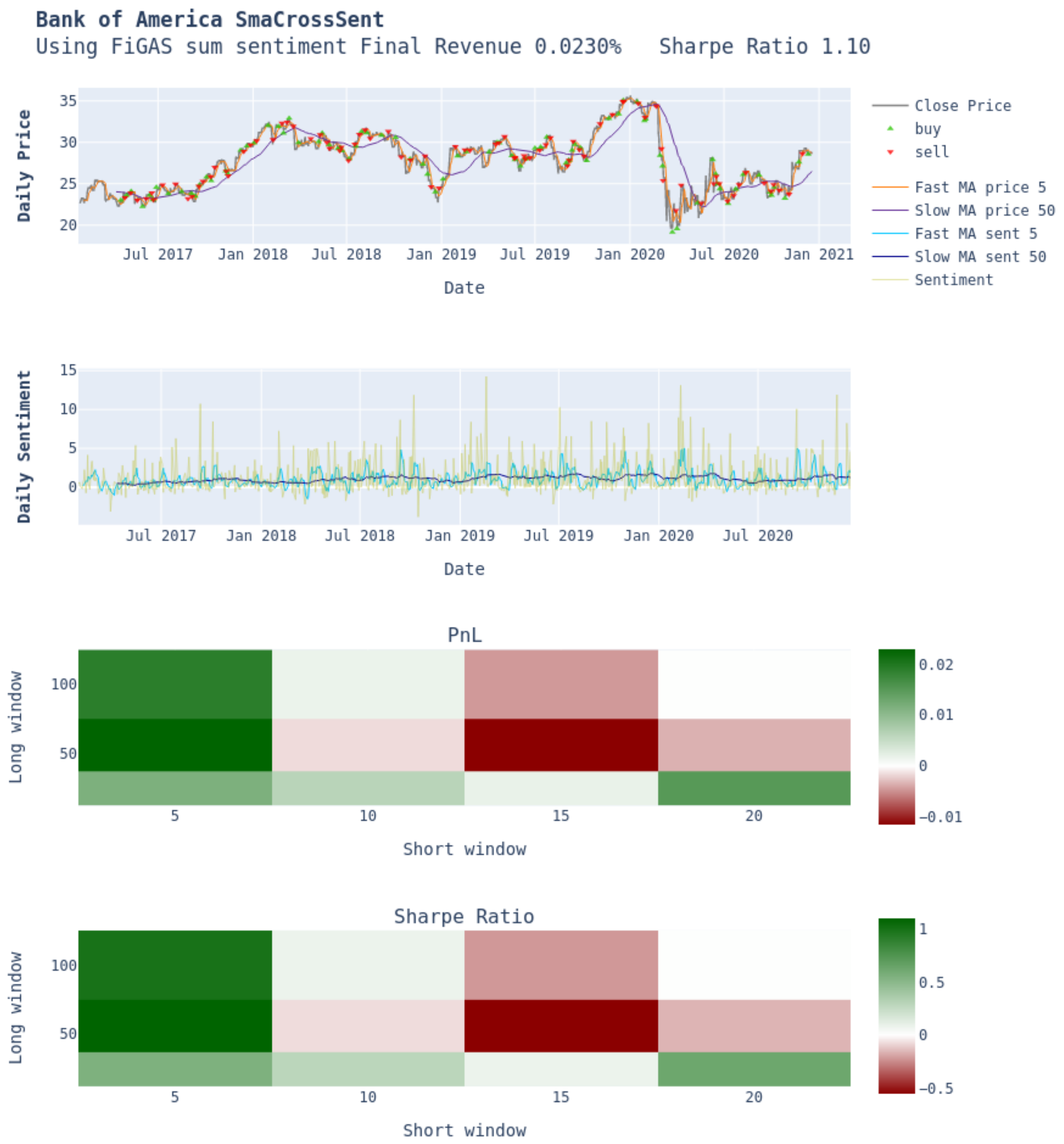


Figure 43: Trading strategy for Bank of America guided by the sum-aggregated FiGAS sentiment scores. The first line plot shows price market values for Bank of America with its corresponding LTMA and STMA (50-day and 5-day respectively). Green and red triangles point out buying and selling actions. The second line plot shows the sentiment scores and its corresponding LTMA and STMA (50-day and 5-day respectively). The two heat maps at the bottom show the PnL and Sharpe Ratio values for different window sizes. The strategy shown in the line plots corresponds to the one using the paired values of window lengths that maximizes the Sharpe Ratio.


Figure 44: Price-based trading strategy for Bank of America. The first line plot shows price market values for Bank of America with its corresponding LTMA and STMA (50-day and 5-day respectively). Green and red triangles point out buying and selling actions. The second line plot shows the sentiment scores and its corresponding LTMA and STMA (50-day and 5-day respectively). The two heat maps at the bottom show the PnL and Sharpe Ratio values for different window sizes. The strategy shown in the line plots corresponds to the one using the paired values of window lengths that maximizes the Sharpe Ratio.

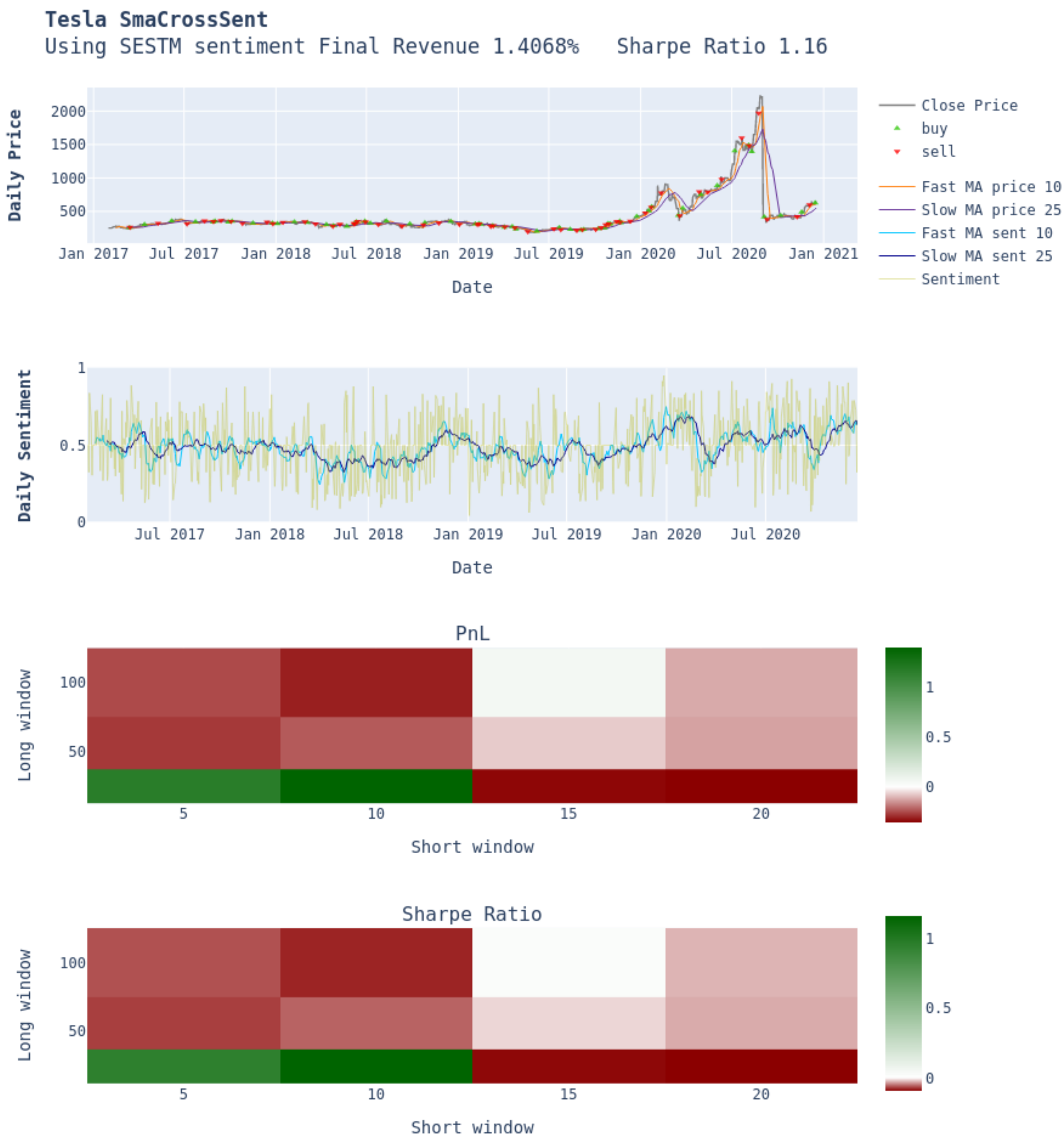


Figure 45: Trading strategy for Tesla guided by the SESTM sentiment scores. The first line plot shows price market values for Tesla with its corresponding LTMA and STMA (25-day and 10-day respectively). Green and red triangles point out buying and selling actions. The second line plot shows the sentiment scores and its corresponding LTMA and STMA (25-day and 10-day respectively). The two heat maps at the bottom show the PnL and Sharpe Ratio values for different window sizes. The strategy shown in the line plots corresponds to the one using the paired values of window lengths that maximizes the Sharpe Ratio.

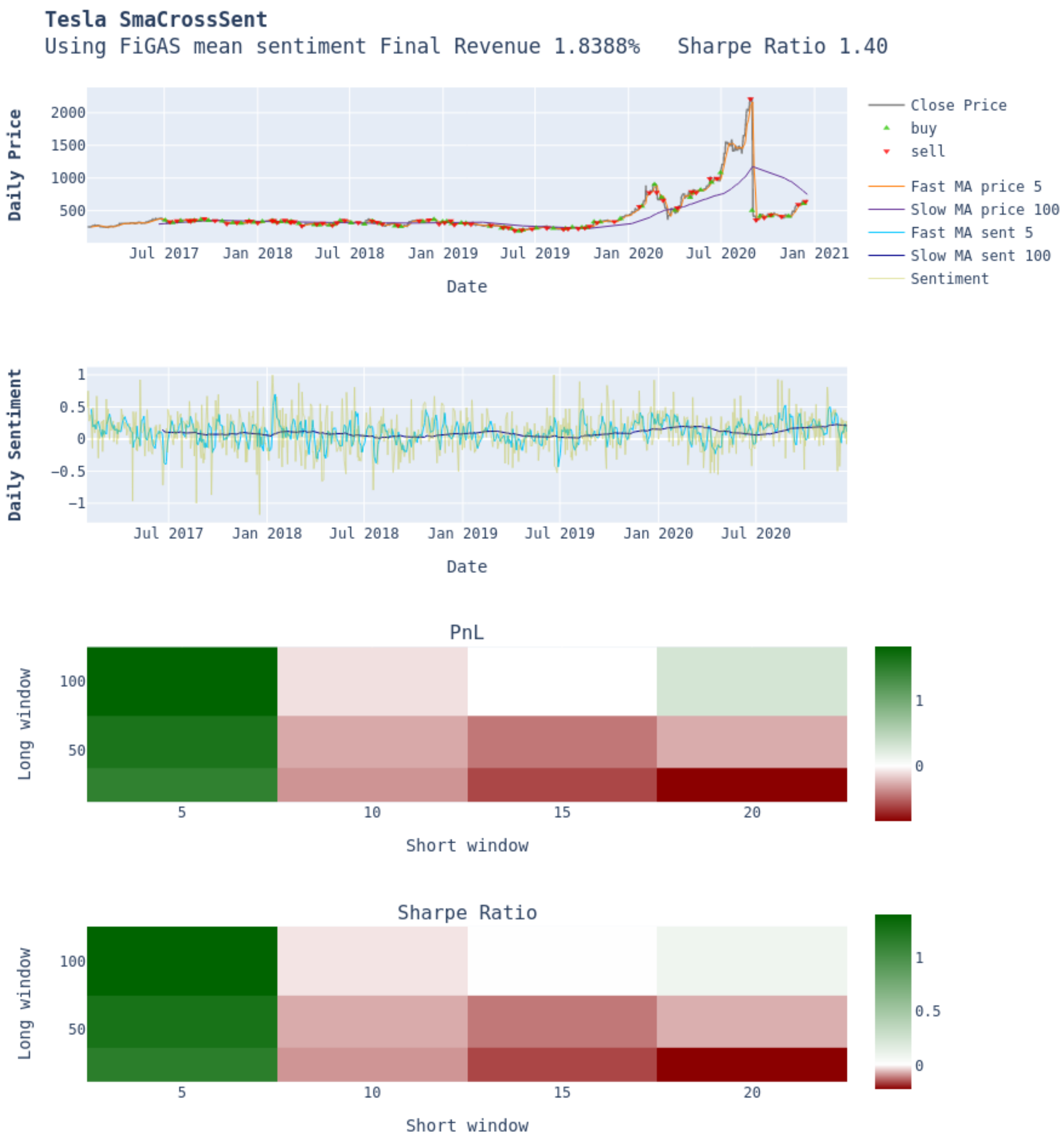


Figure 46: Trading strategy for Tesla guided by the mean-aggregated FiGAS sentiment scores. The first line plot shows price market values for Tesla with its corresponding LTMA and STMA (100-day and 5-day respectively). Green and red triangles point out buying and selling actions. The second line plot shows the sentiment scores and its corresponding LTMA and STMA (100-day and 5-day respectively). The two heat maps at the bottom show the PnL and Sharpe Ratio values for different window sizes. The strategy shown in the line plots corresponds to the one using the paired values of window lengths that maximizes the Sharpe Ratio.


Figure 47: Trading strategy for Tesla guided by the sum-aggregated FiGAS sentiment scores. The first line plot shows price market values for Tesla with its corresponding LTMA and STMA (25-day and 20-day respectively). Green and red triangles point out buying and selling actions. The second line plot shows the sentiment scores and its corresponding LTMA and STMA (25-day and 20-day respectively). The two heat maps at the bottom show the PnL and Sharpe Ratio values for different window sizes. The strategy shown in the line plots corresponds to the one using the paired values of window lengths that maximizes the Sharpe Ratio.


Figure 48: Price-based stratgey for Tesla. The first line plot shows price market values for Tesla with its corresponding LTMA and STMA (100-day and 5-day respectively). Green and red triangles point out buying and selling actions. The second line plot shows the sentiment scores and its corresponding LTMA and STMA (100-day and 5-day respectively). The two heat maps at the bottom show the PnL and Sharpe Ratio values for different window sizes. The strategy shown in the line plots corresponds to the one using the paired values of window lengths that maximizes the Sharpe Ratio.