Impact of Sentiment analysis on Energy Sector Stock Prices : A FinBERT Approach

Conference Paper · June 2024

CITATIONS
0

READS
279

3 authors:

Sarra Ben Yahia
Hôpital Foch
1 PUBLICATION **0** CITATIONS
SEE PROFILE

José Ángel García Sánchez
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Rania Hentati kaffel
Université de Paris 1 Panthéon-Sorbonne
23 PUBLICATIONS **142** CITATIONS
SEE PROFILE

Impact of Sentiment analysis on Energy Sector Stock Prices : A FinBERT Approach

S. Ben Yahia[1], **J.A. García Sánchez**[1] **and R. Hentati-Kaffel**[2]

[1] *Centre d'Economie de la Sorbonne, Université Paris1 Panthéon-Sorbonne*

E–mail: *benyahiasarra9@gmail.com, jagarsanc@gmail.com*

[2] *Maison des Sciences Economiques, 106-112 Boulevard de l'Hôpital, 75013 Paris, France*

E–mail: *rania.kaffel@univ-paris1.fr*

SUMMARY: This study provides sentiment analysis model to enhance market return forecasts by considering investor sentiment from social media platforms like Twitter (X). We leverage advanced NLP techniques and large language models to analyze sentiment from financial tweets. We use a large web-scrapped data of selected energy stock daily returns spanning from 2018 to 2023. Sentiment scores derived from FinBERT are integrated into a novel predictive model (SIMDM) to evaluate autocorrelation structures within both the sentiment scores and stock returns data. Our findings reveal i) significant correlations between sentiment scores and stock prices. ii) Results are highly sensitive to data quality. iii) Our study reinforces the concept of market efficiency and offers empirical evidence regarding the delayed influence of emotional states on stock returns.

Key words. financial NLP, finBERT, information extraction, webscraping, sentiment analysis

1. Introduction

Incorporating sentiment analysis into behavioral finance significantly enhances market return forecasts by considering the broad spectrum of investor sentiment. This method extends traditional analysis, which typically focuses on financial asset prices, historical household consumption, and GDP values, to include the influence of media and social networks, such as X's posts.

Efficient Capital Markets states that markets are "informationally efficient" if prices at each moment incorporate all available information about future values. If there is a signal, not incorporated in market prices, that future values will be high, competitive traders will buy on that signal. Abnormal signals do not contradict classical theory. Fama asserts that the simple fact that there are so many cases of under- and over-reaction to specific signal information demonstrates the soundness of the theory of market efficiency, which converges towards equilibrium in the long term Fama (1970), Fama (1998).

The theoretical work of N Barberis and Vishny (1998) and that of Daniel et al. (1998) remain references for understanding the phenomena of over- and under-reaction to information and explaining the reasons for the phenomenon of overconfidence and its consequences on price.

The empirical results of Barberis et al. (1998) provide a better understanding of the phenomena of over- and under-reaction to information and explain the reasons for the phenomenon of overconfidence and its consequences for prices. The behavioral finance and sentiment analysis dimension of investors is a purely subjective approach that cannot be measured quantitatively, and it plays a crucial role in the evolution of prices and market trends.

In this sense, positive market sentiment can encourage people to buy assets, while negative sentiment can lead to selling behavior. Optimism about financial security is then influenced by various sources of information, such as economic news, tweets, and chats on social networks. And social media have become a real-time barometer of public sentiment,

with the instant dissemination of information, opinions and reactions from a large and diverse audience ((J Bollen and Zeng 2011), (W Zhang and Yu 2018), (Bissattini and Christodoulou 2013)).

Two main approaches can be used to measure sentiment analysis: the first is based on a specific composite financial ratios as a proxy. Qiu and Welch (2004) used the ratio of Long Shares to Total Volume. An increase in this ratio can be interpreted as an indication of bearish sentiment, as it implies that investors expect the future price of this share to fall. The second ratio is the "Put/Call" ratio, which was tested by Bandopadhyaya and Jones (2006). An increase in this ratio indicates pessimism about the underlying share, as the use of Puts reflects investors' desire to hedge against a possible fall in the share price.

Many other ratios have been tested in the financial literature (sometimes it is a composite indicator) such as in the paper by Baker and Wurgler (2006). They introduced several measures: the first on was the ratio Ratios of Long-Term Debt to Total Debt. An increase in this ratio is considered to be a negative signal about the company's future prospects. They also tested the ratio of Capital Expenditures to Total Assets. The second approach is based on:

1. Social media analysis, i.e. discussions on social networks, forums, blogs, publications on Twitter, messages on Yahoo! Finance, and other online platforms

2. Text and content analysis to assess the tone, emotions and opinions expressed in financial documents, news, etc.

3. Analysis of real-time data streams from various websites to instantly assess market sentiment (JR Piñeiro-Chousa and Pérez-Pico 2016); (Makrehchi and Liao 2013); (T Sprenger and Welpe 2014)).

Once the data has been extracted, deep learning methods and algorithms can be used to detect positive or negative sentiments. Then, machine learning models that predict future market value could be enriched with these behavioral variables. It is important, therefore, to understand the evolution of sentimental analysis methods.

Furthermore, although the analysis is informative and can provide buy or sell signals, confidence in these indicators must be strong. Trust can be measured in views and reactions to the post. The social network Twitter (X) has the advantage of having recently added the number of reproductions of the post, which allows the impact of this post to be quantified more accurately.

Sentiment analysis refers to the computational analysis focusing on the nuanced dimensions of sentiments, opinions, emotions, and appraisals. This analytical framework is employed to explore individuals' reactions and attitudes towards financial markets for example. Although feelings are not strictly positive or negative and include nuances, most models classify them as positive, negative or neutral. The use of NLP algorithms for sentiment analysis based on financial tweets for stock prediction has been intensively utilized recently. Furthermore, the most recent models show surprising results. For example, in the Amazon Review Polarity benchmark dataset, the sentiment prediction accuracy is 97.37% (Xie et al. 2019). This demonstrates on the one hand the effectiveness, on the other hand the interest in expanding its use to finance.

In this study, we aim to explore how we can model the relationship between sentiment derived from microblogs on stock market returns. We will examine the correlation and the presence of a lag effect, and investigate the accuracy of the method by analyzing how the frequency of positive and negative posts influences the quality of the predictions. In section 1 we will detail our methodology to extract the NLP score for each market's day. In section 2, we will present the results for the sentiment-informed market direction model (SIMDM) model and finally we will discuss these findings in the final section.

Before exploring into these results, it is essential to consider the role of Large Language Models (LLMs) in our analysis. Recent years have seen a proliferation of diverse approaches, including those utilizing LLMs, that can be categorized into three distinct groups :

1. The lexicon and ruled-based approaches which compiles sentiment words and are mainly concerning manual-building, corpus-based, and dictionary-based methods Alessia et al. (2018). Lexical methods often rely on pre-established word lists, where each word is assigned a particular sentiment. However, these methods are heavily reliant on the context in which they were developed (Ribeiro et al. 2015)). Some approaches, like the Valence Aware Dictionary and Sentiment Reasoner (VADER) (Hutto and Gilbert 2014), integrate lexicons with the analysis of sentence features to ascertain sentence polarity.

2. Word embeddings mark a major leap in understanding language, offering dense vector representations for words that capture semantic similarities. Models like Word2Vec (Mikolov et al. 2013) and GloVe (Jeffrey Pennington 2014) generate what are known as static word embeddings. These embeddings capture semantic information and relationships between words based on their co-occurrence in a corpus. However, each word is assigned a single, fixed vector, regardless of its context. This means that words with multiple meanings (homonyms or polysemous words) are represented by the same vector, regardless of the sentence they appear in.

3. Transformers represent a paradigm shift in sentiment analysis, introduced by Vaswani *et al.* in their seminal 2017 paper Vaswani et al. (2023). Unlike previous approaches, transformers leverage self-attention mechanisms to process entire sequences of text in parallel. This architectural innovation allows models to weigh the importance of different words within a sentence or document, making them highly effective at capturing context and understanding the nuanced meanings of words based on their use.

Including sentiment analysis data in a predictive model is often enriching for its predictive capacity. Historically, machine learning models have been used. R Ren and Liu (2019) used SVMs models to predict the direction of stock market movement and validated an accuracy rate of 89.93% of their SSE50 forecasts by adding their sentiment variables. P Koukaras and Tjortjis (2022) tested the effectiveness of a set of machine and deep learning algorithms (KNN, SVM, LR, MLP), coupled with the use of TextBlob and VADER. They used data directly from Twitter and StockTwits data, as well as Yahoo Finance. They evaluated price fluctuations based on financial and sentiment data and collected 90,000 tweets from Twitter and 7,440 tweets from StockTwits from July 16, 2020, to October Octobre 31, 2020. Their methods improve prediction results, with an F-score of 76.3% and an AUC value of 67% when using VADER for sentiment analysis and SVM for classification.

The vast majority of empirical studies focusing on analyzing the relationship between microblog sentiment and stock market returns are exclusively based on the daily impact, examining how daily aggregate sentiment influences daily returns (J Bollen and Zeng 2011); (T Sprenger and Welpe 2014); (W Zhang and Yu 2018)).

Given that the data flow is in real time, the granularity and size of the data seem to be significant for running a statistical learning model. Some other studies have focused on the impact of data frequency on the quality of the resulting sentiment analysis.

The construction of a sentiment indicator is a crucial step in understanding the relationship between investor sentiment and stock market returns. Testing this relationship allows researchers to explore the potential causal link between investor trends and market performance, thereby evaluating the validity of the market efficiency hypothesis. Numerous studies have delved into these causal relationships, highlighting the importance of sentiment analysis in financial research (Kim and Kim (2014); (Z Da and Gao 2015); (JR Piñeiro-Chousa and Pérez-Pico 2016); (W Zhang and Yu 2018); S Zaman and Saleem (2022)).

In this study, we aim to explore how we can model the relationship between sentiment derived from microblogs to stock market returns. We will examine the correlation and the presence of a lag effect and investigate the accuracy of the method by analyzing how the frequency of positive and negative posts influences the quality of the predictions.

Large Language Models (LLMs) have garnered significant attention in recent years due to their remarkable capabilities in understanding and generating human language. Their application in various domains, including market prediction, has shown promising results.

The advent of LLMs like BERT (Devlin et al. 2018a), GPT-1 (Brown et al. 2020), and their successors has revolutionized natural language processing (NLP) tasks. These models leverage vast amounts of data and sophisticated architectures to learn contextual embeddings of words, enabling them to capture nuances in language that are crucial for tasks such as sentiment analysis, text classification, and more, as evidenced by studies such as those by Sun, Huang, & Qiu (2019) Sun et al. (2019) and Xu, Liu, Shu, & Yu (2019) Sun et al. (2019).

Sentiment analysis, a key application of LLMs, involves extracting and quantifying sentiment from textual data. In the context of financial markets, sentiment analysis has been used to gauge investor sentiment from news articles, social media posts, and financial reports. By analyzing news articles and social media posts, LLMs can identify significant events, such as earnings announcements, mergers, and geopolitical developments. These events often lead to substantial market reactions, and timely detection can provide a competitive edge in trading strategies Ding et al. (2015).

2. Methodology

The methodology of this study is divided into three primary phases: Data Extraction, NLP Inference and Scoring Modelization, and Deep Learning Prediction and Analysis. The following sections provide a detailed description of each phase, as depicted in the methodology diagram.

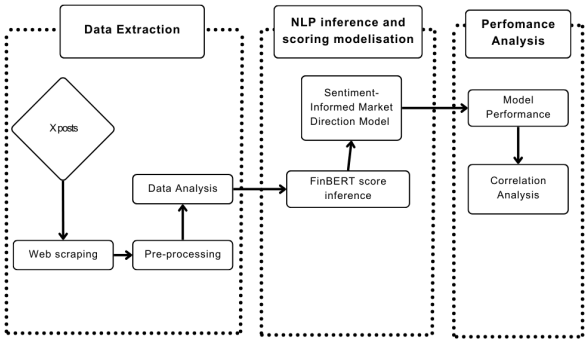


Figure 1: Overview of Methodology

2.1. Data Extraction

In this phase, we focus on the acquisition and preliminary processing of the raw data necessary for subsequent analysis.

- **X Posts**: Social media posts (referred to as "X posts", e.g., tweets) are the primary data source.
- **Web Scraping**: We employ web scraping techniques to automatically extract posts mentioning the selected companies.
- **Pre-processing**: The extracted data undergoes preprocessing to clean and structure it.
- **Data Analysis**: Initial data analysis is conducted to identify trends and patterns within the dataset, providing a foundation for the subsequent sentiment analysis. It is also a way to judge the quality of our data.

2.2. NLP Inference and Scoring Modelization

This phase involves the application of Natural Language Processing (NLP) techniques to infer sentiment from the pre-processed data and model the relationship between sentiment and market movements.

- **FinBERT Score Inference**: We utilize the FinBERT model, a financial domain-specific BERT variant, to infer sentiment scores from the text of the posts.
- **Sentiment-Informed Market Direction Model**: Based on the inferred sentiment scores, we develop a model to predict market direction. This model integrates sentiment data to provide insights into potential market movements, identifying whether the sentiment is indicative of a bullish or bearish trend.

2.3. Model Performance Analysis and Verification

The final phase focuses on the verification of our work, by looking at the model performance and its correlation analysis.

- **Model Performance**: We assess the overall performance of the predictive model. These metrics help in understanding how well the model is predicting the stock movements based on sentiment analysis.
- **Correlation Analysis**: We perform a correlation analysis to examine the relationship between the predicted stock movements and actual market data. This analysis helps in understanding the strength and direction of the relationship between sentiment-influenced predictions and real market trends.

2.4. NLP methodology

BERT: Pre-trained of Deep Bidirectional Model for Language Understanding

BERT, a transformer model developed by Google, represents a significant advancement in natural language processing (NLP). Pre-trained on extensive datasets, including the entire Wikipedia, BERT understands context bidirectionally, unlike traditional sequential models. This bidirectional approach enables BERT to capture the nuanced meanings of words based on their surrounding context within a sentence. As a result, it significantly advanced the state-of-the-art in NLP. Subsequent versions have been fine-tuned for various tasks, such as question answering, sentiment analysis, and named entity recognition.

Fine-tuning

BERT is fine-tuned for specific tasks by adding an additional output layer for each task.

2.4.1. FinBERT: Overview and Rationalization

FinBERT, developed by Prosus AI, is a derivative of the BERT model. However, given the specialized and often nuanced language used in financial texts, a generic BERT model can struggle to accurately interpret financial sentiment. FinBERT addresses this gap by being pre-trained on a corpus of financial texts, thereby imbuing it with an innate understanding of financial discourse Araci (2019).
Transitionning from BERT to FinBERT requires two main steps :

- The first step involves using a broad-based corpus, like BookCorpus and Wikipedia, to imbue the BERT model with a foundational comprehension of natural language across a wide spectrum of contexts. Following this, the team at ProsusAI re-trained BERT, focusing on a financial corpus to specifically tailor the model for financial applications. The TRC2-financial corpus, selected for this purpose, is a subset of Reuters' TRC2 dataset Reuters (2009). It is composed of 46,143 financial news articles, gathering over 29 million words for the period of 2008 to 2010.

- The second step in the fine-tuning phase is the integration of the Financial PhraseBank dataset, devised by Malo et al. (2013) This dataset is composed of approximately 5,000 sentences derived from financial news articles, each labeled for sentiment by a panel of 16 finance experts and master's students. Its uniqueness lies not only in the sentiment labels provided but also in the detailed documentation

of inter-annotator agreement levels.

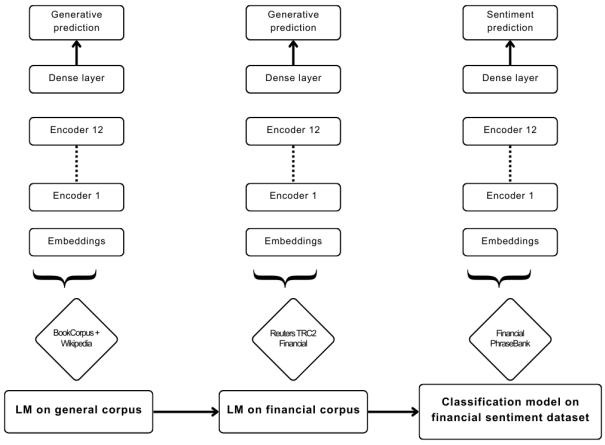


Figure 2: Overview of pre-training, further pre-training and classification fine-tuning

Fine-tuning a transformer-based model like FinBERT for a specific task such as sentiment classification involves a series of technical steps. Below is a comprehensive explanation on the process.

1. Model head modification The BERT model is initially designed to handle various NLP tasks. To tailor it for sentiment analysis, the model's final layer (head) must be modified to suit sentiment classification. Specifically, the original output layer is replaced with a new layer that has an output dimension matching the number of sentiment categories. Let $\mathbf{H}$ represent the hidden states of the model, then the new classification layer can be represented as:

$$\mathbf{y} = \text{softmax}(\mathbf{WH} + \mathbf{b}) \tag{1}$$

where:

- $\mathbf{W}$ is the weight matrix of the new classification layer,
- $\mathbf{b}$ is the bias vector,
- $\mathbf{y}$ is the predicted sentiment probability distribution.

2. Layer-wise Unfreezing The training begins with only the new classification layer unfrozen, meaning it is the only part of the model whose weights are updated initially. As training progresses, deeper layers of the model are incrementally unfrozen. This approach is known as layer-wise unfreezing and helps in stabilizing the training process by gradually incorporating more of the model's parameters. If L denotes the total number of layers, the unfreezing schedule can be formalized as:

$$\text{Unfreeze layer } l \text{ at epoch } e \quad \text{if} \quad e > \frac{k}{n}E \tag{2}$$

where:

- l is the layer index,
- e is the current epoch,
- k is a constant representing the fraction of total epochs E after which a new layer is unfrozen,
- n is the total number of unfreezing steps.

3. Discriminative Fine-Tuning Discriminative fine-tuning involves setting different learning rates for different layers of the network, with the learning rate decreasing for layers closer to the input. This is represented as:

$$\eta_l = \eta_0 \times \alpha^l \tag{3}$$

where:

- η_l is the learning rate for layer l,
- η_0 is the base learning rate,
- α is the discrimination rate (e.g., 0.85).

4. Training Phase The training phase involves optimizing the model's parameters using a labeled dataset such as the Financial PhraseBank. The optimization objective is to minimize the cross-entropy loss function, which measures the discrepancy between the predicted and true sentiment labels. The cross-entropy loss $\mathcal{L}$ is defined as:

$$\mathcal{L} = -\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(\hat{y}_{i,c}) \tag{4}$$

where:

- N is the number of training examples,
- C is the number of sentiment categories,
- $y_{i,c}$ is the true label (one-hot encoded) for the i-th example and c-th category,
- $\hat{y}_{i,c}$ is the predicted probability for the i-th example and c-th category.

By incorporating FinBERT into our analysis, we leverage a tool of exceptional accuracy (demonstrating 97% accuracy on Financial PhraseBank subsets with unanimous annotator agreement) for decoding the sentiment in financial news.

2.5. Data description

2.5.1. Market Returns

Daily market returns were extracted from Bloomberg, covering the period from October 1, 2017, to January 31, 2024.

2.5.2. Webscraping

We developed a custom webscraping tool to acquire data from X (formerly known as Twitter).. This section outlines the technical details and processes involved in data acquisition using Python and Selenium.

To facilitate data collection from X, we implemented a Python-based script utilizing Selenium along with the Chrome WebDriver. This setup automates the process of data extraction from the social media platform. The script initiates by launching an incognito Chrome browser session, with dimensions optimized for data extraction from the targeted platform. To ensure seamless data collection, it systematically suppresses pop-up notifications. The script then proceeds to authenticate by providing the necessary login credentials for X.

Upon successful authentication, the script extracts key data attributes from X tweets. These attributes include:

- Author's username
- Precise timestamp of the tweet
- Textual content of the tweet
- Reply/comment count

The extraction process is managed through specialized functions: one for parsing each tweet, another for configuring Chrome in incognito mode, and additional functions for handling notifications and the login process.

To ensure robustness, the script incorporates comprehensive exception handling mechanisms. These mechanisms allow the script to gracefully manage unexpected issues, thereby maintaining the integrity of the data collection process. Furthermore, the script actively monitors for any suspicious activity during data retrieval from X, ensuring compliance with the platform's data usage policies.

The described webscraping methodology enabled us to systematically gather valuable data, forming the empirical basis for our investigation. This approach ensures both the reliability and reproducibility of the collected data.

2.5.3. Data preprocessing

The preprocessing of data, especially when derived from web sources like social media, presents a foundation for any subsequent analysis. In our study, we employ a methodical approach to prepare and refine the dataset. This section delineates the steps undertaken to cleanse, normalize, and filter the dataset.

A sequence of transformations is implemented to standardize the dataset. This includes:

- Casting numerical columns to their appropriate data types and converting date fields into a consistent format.
- Cleaning the text data by removing extraneous elements—such as URLs, hashtags, mentions, and special characters—that could potentially distort sentiment analysis.
- Standardizing the text through lower casing and the application of regular expressions to eliminate noise.

Our methodology extends to include custom operations tailored to the nuances of social media text. This involves normalizing text by reducing character repetition and ensuring the preservation of meaningful words. These steps are crucial for extracting the essence of the textual data, ensuring it accurately reflects the intended sentiment without the distraction of stylistic embellishments common in online communication.

In dealing with missing values, a pragmatic approach is adopted, filling gaps in numerical columns and removing entries lacking text content. Furthermore, we implement a language filtering step to focus exclusively on English-language texts, aligning with FinBERT's linguistic training and capabilities.

2.5.4. Data description

The analysis period spans from January 2018 to February 2024, with daily stock data collected for five selected companies from sectors such as Energy, Basic Materials, and Utilities. The stocks included in this study are Total Energies, FMC Corp, BP PLC, Stora Enso, and BHP Group.

Table 1: Description of market returns for each stock in percentage

Stock	μ_R	σ_R	$\min_R$	$\max_R$
Total Energies	0.05	1.94	-15.55	15.06
FMC Corp	0.01	2.11	-19.35	13.70
BP PLC	0.04	2.24	-18.75	23.33
Stora Enso	0.03	2.17	-12.78	12.4
BHP Group	0.08	2.09	-16.48	14.94

Additionally, we gathered the number of tweets mentioning each company to assess market sentiment.

Table 2: Number of tweets webscraped for each company

Company	Number of tweets
Total Energies	191,292
FMC Corp	13,194
BP PLC	136,922
Stora Enso	37,268
BHP Group	36,517

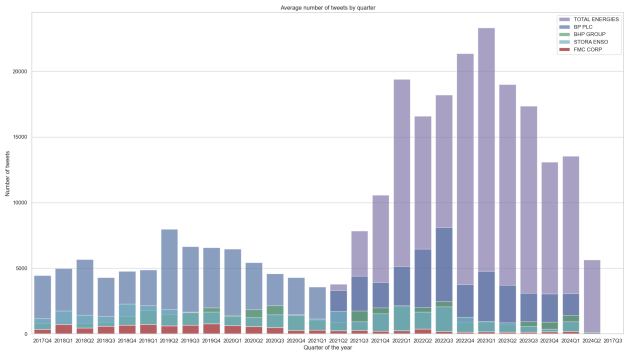


Figure 3: Average number of tweets by quarter

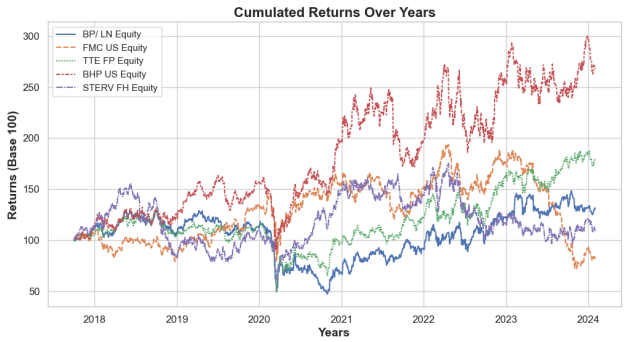


Figure 4: Cumulative returns of the selected stocks over the analyzed period.

The observed volatility in stock returns, driven by market sentiment, supports the viability of a daily buy/sell strategy based on tweet analysis. This approach allows for adapting to market conditions and potentially capitalizing on sentiment-driven price movements.

2.5.5. Topic modeling

We used topic modeling to better decipher and cluster tweets related to STORA ENSO to identify key themes and topics. The following methodology was adopted:

We utilized a BERT model for sentence embeddings, specifically the 'bert-base-nli-mean-tokens' variant from the Sentence Transformers library (Reimers and Gurevych 2019). The embeddings for each tweet were computed to transform the text data into a numerical format suitable for clustering.

To reduce the dimensionality of the embeddings, we employed UMAP (Uniform Manifold Approximation and Projection) with cosine metric (McInnes et al. 2020). This step transforms the high-dimensional embedding space into a two-dimensional space for easier visualization and clustering.

We determined the optimal number of clusters using the Elbow Method and Silhouette Score. The Within-Cluster Sum of Squares (WCSS) is calculated as:

$$\text{WCSS} = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2 \tag{5}$$

where C_i is the set of points in cluster i, and μ_i is the centroid of cluster i.

The Silhouette Score is computed as:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{6}$$

where $a(i)$ is the mean intra-cluster distance and $b(i)$ is the mean nearest-cluster distance for point i. The optimal number of clusters is chosen based on the maximum Silhouette Score.

We applied K-Means clustering to the UMAP embeddings using the optimal number of clusters determined in the previous step (Ding et al. 2024). This algorithm partitions the data into k clusters, minimizing the variance within each cluster.

To identify the top keywords for each cluster, we used TF-IDF (Term Frequency-Inverse Document Frequency) (Ramos 2003). The TF-IDF score for a term t in a document d is calculated as:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \log\left(\frac{N}{\text{DF}(t)}\right) \tag{7}$$

where $\text{TF}(t, d)$ is the term frequency of t in d, N is the total number of documents, and $\text{DF}(t)$ is the document frequency of t. This method helps in identifying the most relevant terms within each cluster.

As a result, the analysis determined that the optimal number of clusters is three. The UMAP projection and the optimal number of clusters determined

Figure 5: Topic Modeling UMAP project for STORA ENSO

by the Elbow and Silhouette methods validate the clustering results.

The top keywords for each cluster provide insight into the main topics discussed within each cluster.

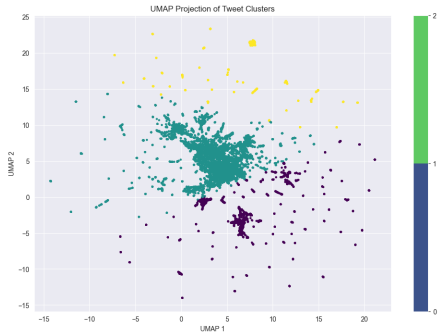
- Cluster 0: This cluster focuses on terms related to local and industrial aspects such as "oulu," "board," "seoay," "sijoittaminen" (investing), "tehtaan" (factory), "group," "yhti" (company), "packaging," "till," and "mets."

- Cluster 1: Keywords here emphasize sustainability and renewable materials, with terms like "finland," "sustainability," "sustainable," "materials," "renewable," "wood," "based," "paper," "packaging," and references to "vonderleyen."

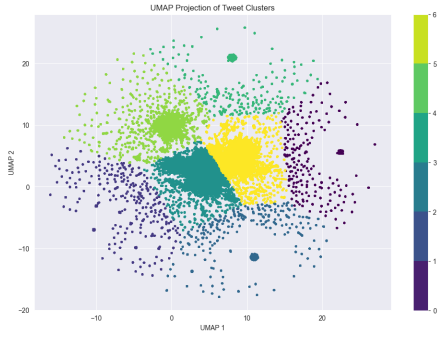
- Cluster 2: This cluster highlights investment and corporate activities with keywords such as "investment," "group," "nokia," "packaging," "transactions," "managers," "mets," "seoay," "finland," and "helsinki."

Topic modeling is a valuable metric for assessing data quality, as it helps determine if the emerging themes and clusters are relevant to the subject under investigation. Figure 3 illustrates the differences in data quality for each stock. Clear and distinct clusters in UMAP projections indicate high-quality data with easily identifiable themes, while overlapping or unclear clusters suggest noisy data or less distinct themes. For instance, BP PLC demonstrates high-quality data, whereas the graph for FMC CORP is less evident.

(a) BP PLC



(b) FMC CORP



(c) TOTALENERGIES SE

(d) BHP GROUP

Figure 6: Topic Modeling UMAP projection of tweet clusters for each stocks.

2.6. Sentiment-Informed Market Directional Model (SIMDM)

2.6.1. FinBERT inference

We used the FinBERT model described earlier in inference for each tweet webscraped. The model normally predicts the sentiment expressed in a piece of text, categorizing it as bearish or bullish based on the patterns it has learned during training.
In our case, we output as results the model's score rather than the Bear/Bull label. Each tweet will be processed to generate a corresponding score, which will then be integrated into our pipeline for a scoring model.

2.6.2. Scoring model for daily sentiment analysis and stock returns

The scoring model we have developed presents a comprehensive method to assess the influence of daily social media sentiment on stock market returns. Here, we detail the essential elements and functions of this model, highlighting its significance in connecting sentiment analysis with the dynamics of the financial market.

An important step in initializing and preparing the data for sentiment analysis and daily stock returns involves aligning the sentiment data with the dates of stock returns. It is vital to account for and omit non-trading days, such as weekends and holidays, from the analysis.

Our method analyzes sentiment ratios for each company, producing recommendations to either short (sell) or go long (buy) on a stock at a specific time, denoted as t. This approach is based on the hypothesis that substantial changes in public sentiment can forecast stock price movements, providing a strategic advantage when accurately interpreted and acted upon.

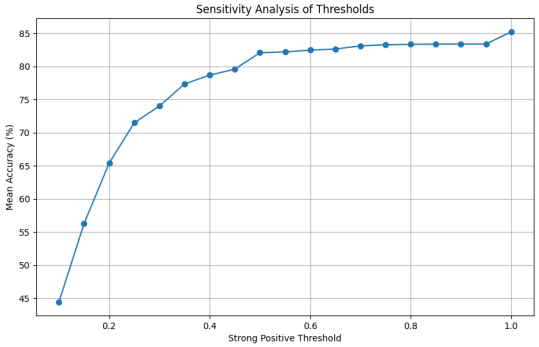
Signal calculation methodology: Our methodology analyzes sentiment ratios for each company, producing recommendations to either take a short (sell) or a long (buy) position on a stock at a specific time, denoted as t. Below, we outline the steps involved in this process:

For each company, the model:

1. Aggregates the sentiment ratios for one company each on a single day t. This aggregation can be represented as :

$$S_t = \frac{\sum_{i=1}^{N} w_i s_i}{\sum_{i=1}^{N} w_i} \tag{8}$$

where s_i is the ith sentiment ratio, and w_i is a weighting factor for each sentiment score. In this study, we operated under the assumption that each post contributes equally to the overall average sentiment score, implying a uniform contribution from all individual sentiments.



2. Then, we shift the aggregated sentiment ratio by one period, utilizing the previous day's aggregated sentiment as the basis for today's trading decision. This shifted sentiment ratio is then used to calculate the "buy/sell" signal :

$$B = \alpha \left(\frac{S_{t-1} - \mu}{\sigma} \right) \tag{9}$$

where α, β, μ, and σ are parameters that can be adjusted to fit the model. In our case, we set the parameters as $\alpha = 2$, $\sigma = 0.5$, and $\mu = 0.5$, for normalization and scaling of sentiment ratios.

Threshold-based decision making:
In the model, once the sentiment ratio for a given day is calculated, it is compared to the threshold :

- If the ratio indicates sufficiently strong positive sentiment, the model suggests a *"buy"* position, anticipating an uptick in the stock's price.
- Conversely, if the sentiment is strongly negative, crossing the threshold in the opposite direction, a *"sell"* position is recommended, forecasting a downturn.

We conducted a sensitivity analysis to understand how the strictness of the model in identifying strong positive signals impacts its overall predictive performance.

At the lowest thresholds, where the model is least selective, the mean accuracy starts at 40%.
As the threshold increases, the mean accuracy improves significantly. By around 0.6, the accuracy reaches almost 75%. This sharp rise suggests that moderate thresholds effectively filter out weaker signals, and enhance the model's prediction accuracy.
Beyond a threshold of 0.6, the increase in accuracy continues but at a slower rate, eventually plateauing around 75%. This flattening trend indicates that while higher thresholds do improve accuracy, the gains become marginal.

For practical applications, selecting a threshold at 0.5 appears optimal, balancing sensitivity and specificity to achieve high mean accuracy without being overly restrictive.

3. SIMDM results

In this section, we will present the results from the modelization presented in the last part. In the graphs below, we can already start to note different elements :

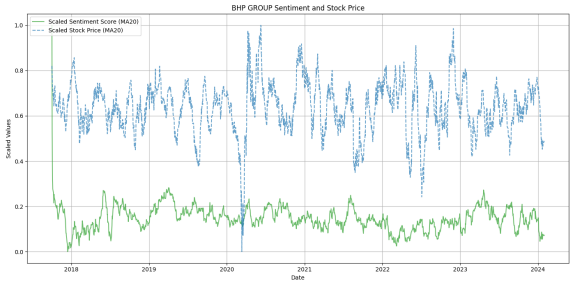


Figure 7: BHP GROUP. Sentiment scores and stock prices between January 1, 2018, and January 1, 2024. Sentiment scores are shown with a 20-day moving average (green line), and historical stock prices are depicted with a dashed blue line. Both metrics are scaled between [0,1] using Min-Max normalization.

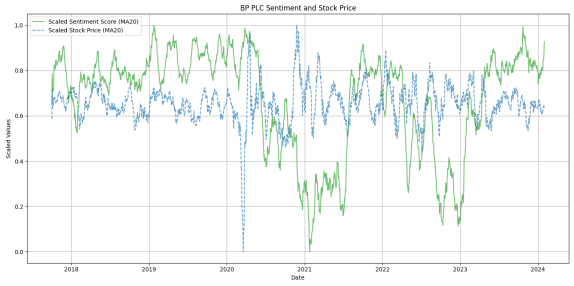


Figure 8: BP PLC. Sentiment scores and stock prices between January 1, 2018, and January 1, 2024. Sentiment scores are shown with a 20-day moving average (green line), and historical stock prices are depicted with a dashed blue line. Both metrics are scaled between [0,1] using Min-Max normalization.

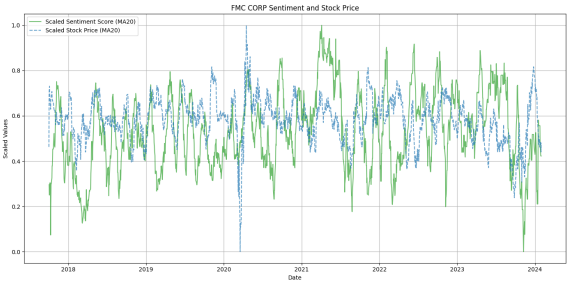


Figure 9: FMC CORP. Sentiment scores and stock prices between January 1, 2018, and January 1, 2024. Sentiment scores are shown with a 20-day moving average (green line), and historical stock prices are depicted with a dashed blue line. Both metrics are scaled between [0,1] using Min-Max normalization.

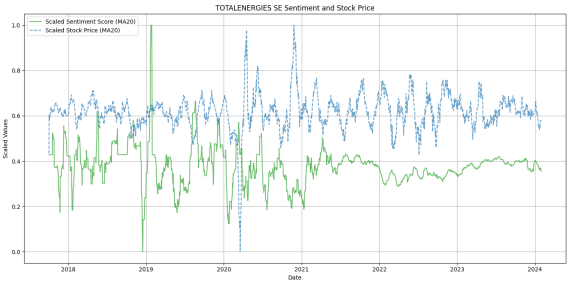


Figure 10: TOTALENERGIES SE. Sentiment scores and stock prices between January 1, 2018, and January 1, 2024. Sentiment scores are shown with a 20-day moving average (green line), and historical stock prices are depicted with a dashed blue line. Both metrics are scaled between [0,1] using Min-Max normalization.

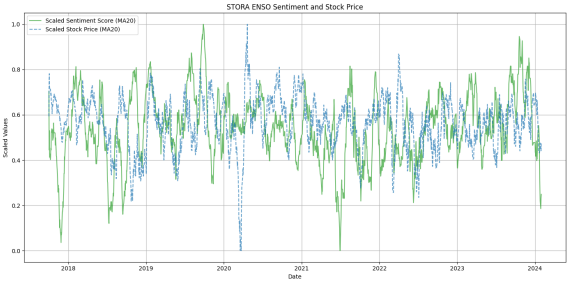


Figure 11: STORA ENSO. Sentiment scores and stock prices between January 1, 2018, and January 1, 2024. Sentiment scores are shown with a 20-day moving average (green line), and historical stock prices are depicted with a dashed blue line. Both metrics are scaled between [0,1] using Min-Max normalization.

- The sentiment scores across all companies exhibit considerable volatility, reflecting the dynamic and often rapid changes in public opinion and market sentiment. This volatility is also observed in stock prices, albeit to a slightly lesser extent. This notion is more or less noticeable depending on the company, however it is clearly visible for FMC CORP and STORA ENSO.
- While the general trend of correlation between sentiment scores and stock prices seems to be consistent across all companies, each company exhibits unique patterns in their sentiment and stock price movements. This finding underscores the importance of considering company-specific factors when analyzing sentiment and stock price relationships. For instance, company-specific events trends may influence sentiment and stock prices in unique ways.

3.1. Model Performance

Accuracy analysis:

The accuracy of the model is computed based on the alignment between the predicted signals and the actual market returns. Let N be the total number of predictions, B_i be the signal for the i-th prediction, R_i be the market return for the i-th prediction and μ_R as the mean of variation of R_i. The match function is defined as follows:

$$\text{match}(B_i, R_i) = \begin{cases} 1 & \text{if } (B_i > 0.5 \text{ and } R_i > \mu_R)), \\ 1 & \text{if } (B_i < -0.5 \text{ and } R_i < \mu_R)), \\ 1 & \text{if } (-0.5 \leq B_i \leq 0.5 \text{ and} \\ & -\mu_R) \leq R_i \leq \mu_R)), \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

The accuracy A is then computed as:

$$A = \frac{\sum_{i=1}^{N} \text{match}(B_i, R_i)}{N}$$

Table 3: Accuracy percentages for energy stocks.

Stock	Accuracy(%)
TOTAL ENERGIES	91.06
FMC CORP	44.39
BP PLC	93.56
STORA ENSO	83.75
BHP GROUP	97.66

3.2. Correlation analysis

3.2.1. Cross-correlation function

For correlation analysis between our home-made signal and the market returns, we used the **cross-correlation function**. The cross-correlation function $R_{xy}(k)$ between two signals $x(n)$ and $y(n)$ is defined, in its discrete form, as:

$$R_{xy}[k] = \sum_{n=-\infty}^{\infty} x[n]y[n+k] \tag{11}$$

We interpret the results of the cross correlation as:

- **Similarity measurement**: The cross-correlation function measures how similar the two signals x and y are for different shifts τ. A high value of $R_{xy}(\tau)$ indicates that the signals are similar when one is shifted by τ.
- **Lag/Lead relationships**: By examining the cross-correlation function, one can identify whether one signal leads or lags another. If $R_{xy}(\tau)$ is maximal at $\tau = k$, it suggests that $x(t)$ and $y(t+k)$ are highly correlated, implying a shift or delay of k.

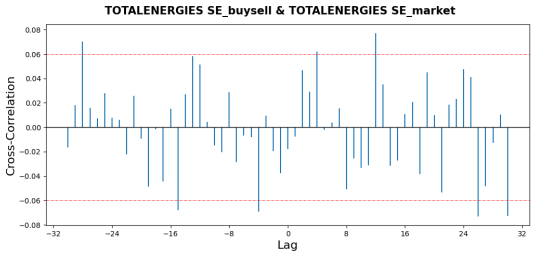


Figure 12: Cross-Correlation function for TOTALENERGIES SE.

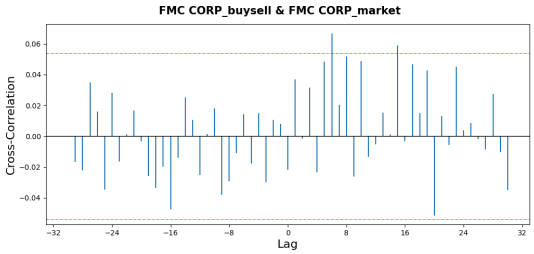


Figure 13: Cross-Correlation function for FMC CORP.

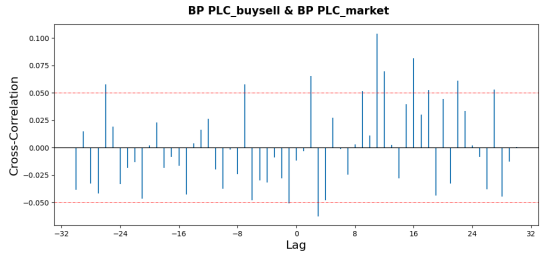


Figure 14: Cross-Correlation function for BP PLC.

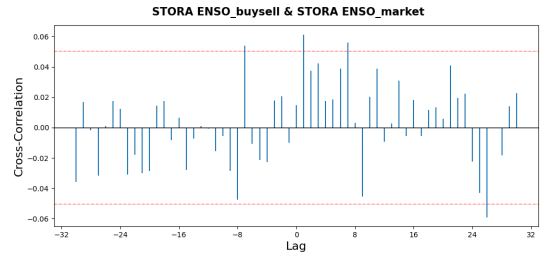


Figure 15: Cross-Correlation function for STORA ENSO.

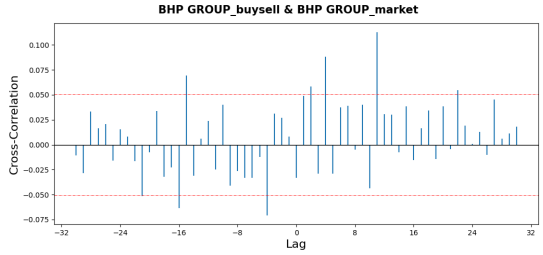


Figure 16: Cross-Correlation function for BHP GROUP.

Table 3 summarizes the predictive power of our signal for each company, based on the cross-correlation function analysis. Companies with significant peaks at positive lags indicate a stronger potential for using our signal to predict future market returns.

For instance, BP PLC shows very strong predictive power with multiple significant peaks, suggesting that our signal could be highly effective in forecasting its market movements. Conversely, companies like Total Energies exhibit low predictive power, indicating minimal usefulness of our signal in predicting future returns.

It is noteworthy that our accuracy and cross-correlation results show the poorest performance for stocks with both the fewest and the most tweets. This observation prompts us to consider not just the quantity of data but also its quality, highlighting the need for a balanced dataset in LLM analysis.

4. Discussion

Our study aimed to explore the complex relationship between tweet sentiment and stock market's returns movements within the energy sector using a sentiment-informed market direction model (SIMDM). The FinBERT model, tailored for financial sentiment analysis, demonstrated its effectiveness in extracting sentiment from social media data. This section delves into the critical insights drawn from our analysis, considering the implications, limitations, and potential future enhancements.

One of the foremost findings of our study is the paramount importance of data quality over sheer quantity in sentiment analysis. We observed that an excessive volume of data could potentially neutralize sentiment scores, rendering them less informative. Conversely, insufficient data failed to provide adequate information for reliable sentiment analysis. For instance, FMC CORP, which had the least number of tweets, showed lower predictive accuracy. In contrast, BP PLC, with a substantial and balanced dataset, exhibited very high predictive accuracy. This underscores the necessity of curating a balanced dataset that maintains the richness and relevance of information without overwhelming the model with noise.

The cross-correlation function analysis revealed interesting insights into the lag effect of sentiment on stock prices. For companies like BP PLC and BHP GROUP, significant peaks at positive lags suggest that sentiment scores can indeed predict future stock price movements with a certain lead time. This lag effect can be strategically leveraged for trading decisions, allowing investors to act on sentiment signals before they manifest in stock prices. However, this predictive power varied across companies, indicating that while the model is effective, its performance is influenced by company-specific factors and market dynamics.

A significant challenge in our study was the potential limitations of continual pre-training. While continual pre-training on domain-specific corpora, such as financial news articles, enhances the model's performance in those areas, it can also introduce biases. These biases arise from the model's exposure to a limited scope of language and context, which may not fully align with the diverse and informal language used in social media posts. Continual pre-training can inadvertently cause the model to overfit the specific style and terminology of the pre-training data, making it less adaptable to different data sources. Future research should explore other

paradigms and methods to mitigate these biases, such as incorporating more varied pre-training datasets and employing techniques that enhance the model's adaptability to new and diverse contexts.

Several limitations were encountered during this study. The primary limitation was the inherent noise and variability in social media data, which could impact the reliability of sentiment analysis. Moreover, the study focused solely on the energy sector, and the findings may not be directly transferable to other sectors without further validation. Future research could expand the scope to include multiple sectors, enhancing the generalizability of the model.

Additionally, incorporating more sophisticated sentiment analysis techniques, such as those accounting for nuanced sentiments and contextual dependencies, could further refine the model's accuracy. The integration of real-time data streams and the development of dynamic models capable of adapting to evolving market conditions would also be valuable enhancements.

In conclusion, this study underscores the potential of integrating sentiment analysis with financial models to enhance stock price predictions. The balance between data quality and quantity, the effectiveness of domain-specific models like FinBERT, and the importance of addressing training data biases are critical considerations for future advancements in this field. By addressing these challenges and building on the insights gained, we can develop more robust and accurate models for leveraging sentiment in financial markets.

References

Alessia, D., Ferri, F., Grifoni, P., and Guzzo, T. 2018, International Journal of Computer Applications, 125

Araci, D. 2019, FinBERT: Financial Sentiment Analysis with Pre-trained Language Models

Baker, M. and Wurgler, J. 2006, Journal of Finance, 61, 1645

Bandopadhyaya, A. and Jones, H. 2006, Journal Name, Volume Number, Page Numbers

Barberis, N., Shleifer, A., and Vishny, R. W. 1998, Journal of financial economics, 49, 307

Bissattini, C. and Christodoulou, K. 2013, Trends in Marketing and Financial Decisions,), july 4, 2013

Brown, T. B., Mann, B., Ryder, N., et al. 2020, arXiv:2005.14165

Daniel, K., Hirshleifer, D., and Subrahmanyam, A. 1998, Journal of finance, 53, 1839

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. 2018a, arXiv:1810.04805

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. 2018b, arXiv:1810.04805

Ding, L., Chen, Z., Wang, X., and Yin, W. 2024, Efficient Algorithms for Sum-of-Minimum Optimization

Ding, S., Lin, L., Wang, G., and Chao, H. 2015, Pattern Recognition,)

Fama, E. 1970, Journal of Finance, 25, 383

Fama, E. 1998, Journal of Financial Economics, 49, 283

Hutto, C. and Gilbert, E. 2014, in Proceedings of the Eighth International Conference on Weblogs and Social Media (ICWSM-14), 216–225

J Bollen, H. M. and Zeng, X. 2011, Journal of Computational Science, 2, 1

Jeffrey Pennington, Richard Socher, C. D. M. 2014, :10.3115/v1/D14-1162

JR Piñeiro-Chousa, M. L.-C. and Pérez-Pico. 2016, Journal of Business Research, 69, 2087

Kim, S. and Kim, D. 2014, Journal of Economic Behavior Organization, 107, 708–729

Makrehchi, M. and Liao, S. S. W. 2013, in Proceedings of IEEE/ACM International Conference on Web Intelligence

Malo, P., Sinha, A., Takala, P., Korhonen, P., and Wallenius, J. 2013, arXiv:1307.5336

McInnes, L., Healy, J., and Melville, J. 2020, UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction

Mikolov, T., Chen, K., Corrado, G., and Dean, J. 2013, arXiv:1301.3781

N Barberis, A. S. and Vishny, R. 1998, Journal of Financial Economics, 49, 307

P Koukaras, C. N. and Tjortjis, C. 2022, Telecom, 3, 358–378

Qiu, J. and Welch, I. 2004, Journal of Empirical Finance, 11, pp. 427

R Ren, D. W. and Liu, T. 2019, IEEE Systems Journal, 13, 760–770

Ramos, J. 2003, Using Tf-idf to Determine Word Relevance in Document Queries, Tech. rep., Department of Computer Science, Rutgers University

Reimers, N. and Gurevych, I. 2019, Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks

Reuters, T. 2009, Thomson Reuters Text Research Collection (TRC2), Web download, available from NIST upon request. Retrieved June 17, 2024, from https://trec.nist.gov/data/reuters/reuters.html

Ribeiro, F. B., Araújo, M., Gonçalves, P., Benevenuto, F., and Gonçalves, M. A. 2015, in Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI-15), 2903–2909

S Zaman, U. Y. and Saleem, T. 2022, Global Knowledge, Memory and Communication,)

Sun, C., Huang, L., and Qiu, X. 2019, arXiv preprint arXiv:1903.09588,)

T Sprenger, A Tumasjan, P. S. and Welpe, I. 2014, European Financial Management, 20, 926

Vaswani, A., Shazeer, N., Parmar, N., et al. 2023, arXiv:1706.03762

W Zhang, Z Deng, X. C. and Yu, W. 2018, Decision Support Systems, 114, 47

Xie, Q., Dai, Z., Hovy, E., Luong, M.-T., and Le, Q. V. 2019, arXiv:1904.12848

Z Da, J. E. and Gao, P. 2015, Review of Financial Studies, 28, 1–32

A. Appendix

Algorithm Description

A.1. Webscraping

Overview This document presents a concise algorithmic description of a Twitter scraper designed to extract tweets based on specific search criteria. The scraper operates through a series of steps, including initialization, data extraction, and data storage, leveraging a web automation tool (Selenium) for interaction with Twitter's web interface.

TwitterScrapper Class

Algorithm 1 Initialize the Twitter Scrapper

Input: research, link, output_path, start_date, end_date, mail, username, password, verbose
Output: An instance of the TwitterScrapper class
Initialize instance variables with input parameters

Algorithm 2 Extract Data from a Tweet Card

Input: card (HTML element representing a tweet)
Output: Extracted tweet data (username, handle, postdate, text, reply count, retweet count, like count)
Extract handle from the card element
Extract username from the card element
Extract postdate from the card element
Extract comment text from the card element
Extract reply count from the card element
Extract retweet count from the card element
Extract like count from the card element
return Extracted data as a tuple (username, handle, postdate, text, reply count, retweet count, like count)

Algorithm 3 Set Up and Open Chrome Browser

Output: Configured Chrome driver instance
Configure Chrome options for incognito mode
Open Chrome and navigate to Twitter login page
return Chrome driver instance

Algorithm 4 Perform Advanced Search

Input: driver, search parameters (research, start_date, end_date)
Construct search URL with parameters
Navigate driver to the constructed URL

Algorithm 5 Scroll and Extract Tweets

Input: driver
Output: Collected tweets
Initialize last position as None
Initialize end of scroll region as False
while not end of scroll region **do**
    Scroll the page
    Collect visible tweets
    Save tweets if not previously saved
    Update last position and end of scroll region status
end while
return Collected tweets

Algorithm 6 Save Tweet Data to CSV

Input: records (list of tweet data), output_path
Output: CSV file containing the tweet data
Write records to CSV at the specified output path

Algorithm 7 Main Scraping Process

Initialize the TwitterScrapper instance
Open and set up Chrome
Perform login and handle pop-ups if necessary
Perform advanced search
Scroll through the search results, extracting and saving data
Close the browser

Algorithm 8 Launch Scraper for Multiple Companies

Define list of companies
for each company in the list **do**
    Set output path based on company name
    Create and execute a TwitterScrapper instance
end for

A.2. BERT : Pre-trained of Deep Bidirectional Model for Language Understanding

BERT, developed by Devlin et al. Devlin et al. (2018b) at Google in 2019, has significantly advanced

state-of-the-art benchmarks in natural language processing. BERT uses L layers (Transformer blocks), with a hidden size of H and A self-attention heads. Two principal versions has been developed:

- **BERT$_{\mathbf{BASE}}$**: $L = 12$, $H = 768$, $A = 12$, Total Parameters=110M
- **BERT$_{\mathbf{LARGE}}$**: $L = 24$, $H = 1024$, $A = 16$, Total Parameters=340M

Input Representation

Input representation is a sum of token embeddings, segment embeddings, and position embeddings. Subsequently, for a given input sequence $x = [x_1, x_2, \ldots, x_n]$, the embeddings sequence is constructed as follows:

$$E_{\text{input}} = E_{\text{token}} + E_{\text{segment}} + E_{\text{position}} \tag{12}$$

Pre-training Tasks

Masked Language Model (MLM)

- Randomly masks 15% of the tokens in the input.
- The model predicts the masked tokens based on their context.
- Loss function: Cross-entropy loss over the masked tokens.

$$L_{\text{MLM}} = -\sum_{i \in \text{masked tokens}} \log P(t_i \mid \text{context}) \tag{13}$$

Next Sentence Prediction (NSP)

- Predicts if a given sentence B follows sentence A in the original text.
- Loss function: Cross-entropy loss for binary classification (IsNext vs. NotNext).

$$L_{\text{NSP}} = -\left[y \log \hat{y} + (1 - y) \log(1 - \hat{y})\right] \tag{14}$$

Self-Attention Mechanism

The core of the BERT model is its bidirectional encoder architecture, which utilizes a multi-head self-attention mechanism.

This design allows the model to consider the context from both directions (left and right) for each token in the input sequence. In each layer of the model, the self-attention mechanism computes the following:

$$\begin{aligned} Q &= H^{(l)} W_Q \\ K &= H^{(l)} W_K \\ V &= H^{(l)} W_V \end{aligned} \tag{15}$$

Here, Q (queries), K (keys), and V (values) are obtained by projecting the input representation $H^{(l)}$ using weight matrices W_Q, W_K, and W_V, respectively.

BERT uses scaled dot-product attention. For each head, the attention scores are computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \tag{16}$$

In this equation, the dot product of Q and the transpose of K is scaled by the square root of the dimensionality of the key vectors, $\sqrt{d_k}$, and subsequently passed through a softmax function to obtain the attention weights. These weights are then used to compute a weighted sum of the values V, thereby producing the attention output.

Following the computation of the attention scores, the outputs from the multi-head attention mechanism are concatenated and linearly transformed. Specifically, the multi-head attention mechanism is defined as:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{h}_1, \text{h}_2, \ldots, \text{h}_h) W_O$$

where each attention head is computed as:

$$\text{h}_i = \text{Attention}(QW_{Q_i}, KW_{K_i}, VW_{V_i})$$

Here, h is the number of attention heads, and W_{Q_i}, W_{K_i}, W_{V_i}, and W_O are the learned projection matrices for the i-th head and the output projection, respectively.

After the multi-head attention layer, the output undergoes a series of transformations:

1. Add & Norm: The output from the multi-head attention layer is added to the original input (residual connection) and normalized:

$$H^{(l)'} = \text{LayerNorm}(H^{(l)} + \text{MultiHead}(Q, K, V))$$

2. Feed-Forward Neural Network: The normalized output is then passed through a position-wise fully connected feed-forward network, which consists of two linear transformations with a ReLU activation in between:

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

Here, W_1, W_2, b_1, and b_2 are learned parameters. The output of the feed-forward network is then added to the input of the feed-forward network (residual connection) and normalized:

$$H^{(l+1)} = \text{LayerNorm}(H^{(l)'} + \text{FFN}(H^{(l)'}))$$

This process is repeated for each layer in the BERT model, with the output of each layer serving as the input to the next.

The final hidden states from the last layer of the BERT model can be used for various downstream tasks. For instance, for classification tasks, the hidden state corresponding to the [CLS] token is typically used. This hidden state is passed through a classification layer:

$$y = \text{softmax}(H_{[\text{CLS}]}W_C + b_C)$$

where W_C and b_C are learned parameters, and y is the predicted output.

Training Objective

The combined loss for pre-training is the sum of MLM and NSP losses:

$$L = L_{\text{MLM}} + L_{\text{NSP}} \tag{17}$$