Investor sentiment variation across regions: a Natural Language Processing analysis

by

Gabriel BARANES

Research project

Master of Computer Science with specialization in Finance

Illinois Institute of Technology

2023

Abstract

This research project offers an NLP model to analyze the investor sentiment in stock markets. Investor behavior, which depends in particular on market sentiment, may not be the same across markets. For a variety of reasons, information on listed companies may vary from one country to another. This may allow explaining why a company's stock price may differ across stock markets. This issue is particularly acute for multi-listed companies.

In the digital era, financial information is abundant and available in a wide variety of digital media. This makes difficult an exhaustive analysis of the information. This can be greatly facilitated by the use of Machine Learning and NLP tools. These tools enable information to be processed rapidly and made efficiently available to investors. For instance, traders can thus use the structured data provided by NLP tools and refine their trading strategies. The research project is part of this context and provides a tool for processing information and building a sentiment analysis.

After presenting some key concepts related to the analysis of financial markets, we give a brief overview of the sentiment analysis and the BERT model. We then build a model that assesses the overall sentiment expressed in news about a company in two different countries.

Acknowledgments

First and foremost, I would like to express my sincere gratitude to Professor Ricky Cooper, Professor Edward Chlebus, and Professor Benjamin Van Vliet for their invaluable support during the selection of my research project. Their insightful advice and constant encouragement have been of tremendous help throughout this crucial stage of my academic journey.

I would also like to extend my special thanks to Professor Ricky Cooper for closely monitoring the progress of my research project and providing enlightened guidance that significantly contributed to its success. His availability and expertise were essential assets that enriched the quality of my work.

My thanks also extend to all individuals who, directly or indirectly, contributed to the success of this research project.

Contents

1. Introduction

Information plays a fundamental role in the whole economy. Its importance comes from its ability to influence the decisions and behavior of individuals, companies and governments. A higher quality of information can lead to more optimal choices and a more efficient allocation of resources. Market functioning is therefore affected by the quality of information available to individuals and, more generally, the decision-makers. This is true not only for physical goods markets, but also for financial markets.

One of the main and much debated ideas in economics is that the price on a market reflects all the information available to economic agents. Full information increases market efficiency, and the price reflects the true value of the traded good. This is in line with the Efficient Market Hypothesis formulated by Fama in 1970 [1]. Since then, this hypothesis has been the subject of extensive debate among researchers. One of the main questions debated is whether information provides a good prediction of price levels and their movements in stock markets.

Traders, whether active in financial markets, commodity markets, foreign exchange, or other markets, heavily rely on information to make buying or selling decisions. In reality, information is imperfect. In other words, traders do not have access to all the information, and traders may have different information. This implies that investors who have better information can strategically use it to make their investment decisions. Acquiring information enables investors to improve their expectations regarding market trends, allowing them to make buying or selling decisions that optimize their monetary gains, taking into account their risk aversion.

In the digital era, online financial press has experienced significant growth, and information is abundant on social networks. Overall, this constitutes a vast amount of information with a wide diversity of opinions on financial market trends and sometimes divergent views on financial securities. In this context, traders find it challenging to effectively exploit this overwhelming amount of information. The choice of words and tone used by authors of online articles can also convey sentiments that directly impact investors' decisions. Today, things can be made easier with the development of Machine Learning models that provide powerful tools to analyze this extensive amount of financial information. Natural Language Processing (NLP) models can assess the collective sentiment that influences stock market movements [2]. NLP tools thus provide a set of structured data constructed from financial reports and news articles. Traders can utilize these structured data and refine their trading strategies.

Many companies are listed on multiple stock exchanges located in different countries or regions. These companies are called “multi-listed companies”, and are often multinational companies. For example, a multinational company based in the United States may choose to list its shares on the New York Stock Exchange (NYSE) but also on a European stock exchange such as Euronext or the London Stock Exchange (LSE). Being listed on multiple stock exchanges, these companies benefit from a number of advantages, such as access to a broader pool of investors, greater liquidity for their shares, or geographic diversification of their investors. Different market conditions or regulations may also motivate these companies to take such a decision.

Stock prices for a multi-listed company are not necessarily the same on the different stock markets. This can be explained by currency exchange rates, the trading volume and liquidity of each market, regulatory differences or different arbitrage opportunities. Stock prices can also differ because market sentiment can be different across countries or regions. Thus, different sentiments can result in price discrepancies across different marketplaces.

Market sentiment, and therefore investor behavior, may not be the same across regions. The reasons for this can be diverse: different economic conditions and different political, geopolitical, social or cultural factors. Another significant factor that can play a crucial role is related to the information delivered to investors. Indeed, information may not be presented in the same manner from one country to another and the quantity and quality of information may also vary. The type of information provided to investors may not be uniform, and this can lead to varied market sentiments across regions. A large literature in finance has focused on the market sentiment [3]. For instance, these researches study how investor sentiment affects market returns in different countries [4]. These studies also looked at the correlation between relative market sentiment and the relative stock prices of dual-listed companies, making the distinction between global and local sentiment depending on whether the sentiment comes from global or local sources. This article also examines the contagion between different local sentiments [5]. Researches in the field of computer science are also providing effective new methods for extracting potentially exploitable signals from the news [Mishev 2020].

In the following, we present a model that uses recent NLP algorithms to analyze market sentiment and its potential differences across regions. In section 2, we present some key useful concepts related to the analysis of financial markets and the main results of the literature on the relationships between information, sentiment and investor behavior in financial markets. Sections 3 and 4 present, respectively, the sentiment analysis and the BERT model (Bidirectional Encoder Representations from Transformers). Section 5 is devoted to the presentation of the model, and gives results and a discussion. Finally, the last section concludes.

2. Some key concepts of financial markets

2.1. The market efficiency: a brief overview

Information plays a crucial role in the functioning of markets. The main idea is that the price in a market reflects all the available information to economic agents. Thus, when all information is available, the market operates efficiently, and the price reflects the true value of the traded good. This general result applies particularly to financial markets, and therefore, movements in stock markets could be explained by publicly available information. This relates to the efficient market hypothesis formulated by Fama in 1970 [1].

The Efficient Market Hypothesis (EMH) forms the foundation of stock market analysis. Fama's hypothesis states that the price of stocks instantaneously reflects all publicly available information. This information is present in companies' annual reports but is also provided by traditional and specialized media through various channels (print newspapers, online publications, etc.). Moreover, today, this information is found in the numerous discussions on social media platforms.

This hypothesis strongly influences market analysis, particularly in financial markets. However, it is well-known that markets do not function as proposed by the Efficient Market Hypothesis. In a more recent paper, Fama [7] refines his analysis by specifying that there are three forms of markets based on the type of information available to investors. Thus, the state of the market can take three forms: strong, semi-strong, and weak.

The strong form of the EMH represents a situation where stock prices incorporate all types of information, ranging from privately held information to publicly available data. In this first form, investors also have access to all past information (availability of historical data). However, this extreme form likely does not exist, but it serves as a benchmark for theoretical analyses. The second form is the semi-strong form. The price of stocks incorporates public and historical information, but unlike the first form, prices do not reflect private information. This implies that private information does not affect prices and price movements in the stock markets. Lastly, the third form is the weak form. In this situation, prices only reflect historical information. Prices in this form do not incorporate private or current information, which, in turn, does not influence prices.

The Random Walk Hypothesis (RWH) is an alternative theory to the EMH that was first applied to financial market analysis by Samuelson [8]. According to this hypothesis, stock prices evolve randomly, making them impossible to predict. There is a certain resemblance between this hypothesis and the semi-strong form of the EMH. In fact, the Random Walk Hypothesis assumes that all public information is available to all investors.

Samuelson provides an interpretation of this random walk. Investors are constantly seeking to maximize their gains and, therefore, rely on all sorts of information to obtain excessive returns [8]. Each new piece of information is incorporated into investors' decisions as soon as it is generated. By instantly incorporating this information into stock prices, investors (traders) eliminate profit opportunities. This makes it impossible to forecast price movements since all information is already incorporated [9]. This idea is now well-integrated into theoretical works as well as among finance professionals.

Although both the Efficient Market Hypothesis and the Random Walk Hypothesis hold prominent positions in the world of modern finance, some question them by demonstrating that the stock market can still be predicted. Using a combination of multiple machine learning classifiers, Qian et al. [10] indicate that the forecast of the Dow Jones Industrial Average index can be achieved with an accuracy of 65%. Another example, the findings of Gallagher and Taylor [11] suggest a significant negative correlation between innovation-induced supply-side inflation and the actual returns of stocks.

2.2. Information, news and strategic approaches in investing

There is a growing literature in computer science and artificial intelligence that explores the relationship between information contained in textual sources and investor behavior in financial markets. These studies have been made possible by new NLP tools that enable sentiment analysis. Ashtiani and Raahemi [12] provide a systematic literature review of artificial intelligence research focusing on the links between market prediction and news. Some of these studies demonstrate that it is possible to predict the stock market by extracting valuable information from raw data even before it becomes publicly available. In the following, we present some of these research papers.

Unlike those who adhere to the Random Walk Hypothesis, some argue that it is possible to predict stock prices by deducing new information from a set of raw data, even before this information is disclosed by the market. This is where natural language processing (NLP) plays an important role. NLP tools have been used to predict the sales of physical products based on the frequency of mentions of these products in online discussions. The paper by Gruhl et al. [13] examines how discussions in online communities can predict book sales. Sentiment analysis conducted on blog data allows Mishne and Glance [14] to make sales forecasts for movies. Other research in economics demonstrates how it is possible to predict macroeconomic indicators using data collected from Google Trends [15].

These new NLP tools can also be used to assist investors in making decisions in the stock markets. These methods do not directly predict market movements but indirectly predict critical information that, if published, would impact stock prices. Hence, a substantial literature examines the correlation between news and stocks. For example, Veronesi [16] demonstrated that stock markets react excessively to negative news during prosperous periods and, conversely, underreact to positive news when the economic situation is unfavorable. Other studies, such as those by Chan [17], have explored how public news affects monthly stock returns, while Li [18] examines the relationships between "risk sentiment" using the annual reports filed by U.S.

companies. Finally, research studies investigate the impact of web media on stock markets [19].

Several other research studies have shown the direct prediction of price movements using publicly available textual data. For instance, Bollen et al. [20] demonstrate that sentiment indicators derived from Twitter data can serve as reliable predictors of daily stock price movements, achieving an accuracy rate of 87.6%. Another interesting research by Schumaker and Chen [21] utilizes machine learning algorithms on textual representations such as bag of words and noun phrases to predict stock movements. In a similar vein, Li et al. [22] combine sentiment analysis and news representation techniques to forecast fluctuations in stock prices. This potential short-term predictability enables investors equipped with these tools to anticipate the market's incorporation of news by several minutes, allowing them to make decisions to optimize their financial portfolios [23].

To conclude this section, let us give some comments about strategic approach in investing. The research on financial securities price forecasts leads to two schematic approaches to investment. In the following, we briefly give insights on these two approaches.

The first approach considers that the price of a stock can be inferred from the company's performance and strategic position in its market. The idea is to use public information such as Earnings Per Share (EPS), the company's profit margin, revenue growth rate, or the Price Earnings Ratio (PER) to determine the attractiveness of the stock and its price movement [24]. These pieces of information about the company's revenue and profitability allow the calculation of the intrinsic value of the stock, which can then be compared to the market price of the stock. Based on the ranking between the intrinsic value and the market value, investors make their buy or sell decisions. Therefore, when the intrinsic value is lower than the market value, it signals a positive indication, and investors deduce that there is an opportunity to buy the stock. In this first approach, NLP tools are useful for processing new information available on the company and using it to refine investment strategies well before the market incorporates the same information. These tools are currently used by private equity funds and hedge funds to optimize their portfolios of securities.

The second approach relies on the idea that there are trends in stock prices, and these trends can be used to predict future stock prices [21]. In this case, the principle adopted by investors contradicts the Random Walk Hypothesis. It involves considering that prices follow trends determined by collective psychology and the balance of supply and demand for underlying securities [25]. Investors who adhere to this approach reject the Efficient Market Hypothesis and assume that all information is already reflected in stock prices. In this view, intrinsic values contain no information about price movements in the stock markets and future prices of financial securities. Sentiment analysis, as it helps define the market's collective psychology, plays an important role in making predictions about stock prices.

3. Sentiment analysis

The general idea of sentiment analysis is to categorize the text into a certain emotional category, such as positive, negative, or neutral. The text can also be analyzed based on its tone. This involves choosing a more precise level of categorization that allows for a more nuanced evaluation of the text, for example, Negative, Positive, Uncertainty, Litigious, Strong Modal, or Weak Modal.

Sentiment analysis is a complex process that can be broken down into five distinct steps: Data collection, Text pre-processing, Sentiment detection, Sentiment classification, and Presentation of output. In the following, we will provide a brief overview of these different steps, simplifying the central two steps, i.e., sentiment detection and sentiment classification, for clarity.

3.1. Data collection

The first step involves collecting data generated by users. This data can come from various sources, such as news articles or social media. Initially, this textual data is disorganized and heterogeneous, with different languages, vocabularies, contexts, and formats. Because manual analysis of such data is impractical, Natural Language Processing (NLP) is one of the tools used to extract and classify sentiment from the text.

This data collection step needs to be carried out meticulously before implementing the NLP process and is often the most labor-intensive. To ensure success in this step, a substantial amount of raw data is required for training or evaluation purposes. To simplify the task, one can leverage API interfaces (Application Programming Interfaces). When APIs are not available for the relevant data, web scraping can be employed. Web scraping involves using coding tools to automatically extract, organize, and analyze data from the web.

3.2. Text pre-processing

This step involves preparing the text for the classification stage by cleaning it to reduce complexity and speed up the subsequent work. Texts often contain a lot of non-informative passages (HTML tags, scripts, advertisements), as well as noise. It may be necessary to eliminate words that have no impact on the overall sentiment orientation and remove non-textual and irrelevant content for analysis. Depending on the type of text, this can include numbers or dates written in different forms, as well as punctuation.

This second step helps to accelerate the classification stage if the data is effectively cleaned, and the resulting noise reduction helps improve the performance of the classifier. This is crucial if sentiment analysis needs to be performed in real-time, such as when analyzing financial news as it is published.

More specifically, it is common to distinguish at least seven preprocessing steps and techniques. The first, and the most basic, is white space removal. Next, it is necessary to eliminate all capital letters and replace them with lowercase letters to reduce text variability. The third step is punctuation filtering, by removing question marks or other punctuation marks to further reduce complexity. At this stage, certain symbols that have no utility in the language are also eliminated, such as "*", for example. However, when eliminating punctuation, caution must be exercised because punctuation sometimes contains additional information to exploit, as is the case with exclamation marks. The next step is the removal of stop words, which are frequently occurring words that do not provide useful information. These include words like "the, a, it, for," and so on. Next, normalization through replacement is performed to make the text more understandable for algorithms, which improves text processing. At this stage, it involves replacing incorrectly spelled words or words written entirely in uppercase with correct words written in lowercase. This step is often necessary when analyzing tweets or discussions on other social networks (e.g., emojis). Finally, the last step is tokenization, which involves dividing or fragmenting the text into its smallest possible form. Each element is called a token. These tokens can be words, subwords, or characters (n-gram characters).

All this text preprocessing work should be systematically performed, taking into account the intrinsic characteristics of the text. However, there is no clear evidence that this work significantly increases the performance of sentiment analysis [26] [27].

3.3. Sentiment detection and classification

The objective is to detect sentiment by examining sentences extracted from reviews and opinions. It involves separating sentences containing subjective expressions (opinions, viewpoints) from sentences containing objective elements (factual information). The latter are either rejected or identified as such, considering them as neutral sentiment.

Sentiment classification involves categorizing subjective sentences or words into predefined categories, such as positive, negative, good, bad, etc., as determined by the user. There are numerous classification methods that are increasingly effective but also more complex due to the introduction of neural networks. Older methods are based on lexicons, while more modern and high-performing methods are based on Transformer models.

The following briefly outlines sentiment classification, highlighting two approaches: the lexicon-based approach and the machine learning approach.

3.3.1 Algorithms based on lexicons

These algorithms involve focusing on the words themselves. They consider that certain words express a positive sentiment, while others strictly express a negative sentiment. By gathering enough words attributed to a positive, negative, or neutral sentiment, it becomes possible to classify an entire sentence or even the text. In reality, there are two lexicon-based methods: corpus-based and dictionary-based.

The corpus-based approach involves annotating the text with various tags. This allows for adding an indication of the intensity of the sentiment, thereby distinguishing, for example, "exemplary behavior" from "adequate behavior." The sentence containing the word "exemplary" would receive a higher score than the one containing the word "adequate."

The dictionary-based approach involves gathering a subset of words, called "seed words," from the text to be analyzed, assigning them a sentiment, and then expanding this list of words (dictionary) by adding their synonyms and antonyms. It is assumed that synonyms carry the same sentiment as the seed word, while antonyms represent the opposite sentiment. The dictionary thus represents a more specialized corpus for the text being analyzed.

The principle of these algorithms is then to identify and count the quantity of positive, negative, and neutral words. The more positive words there are, the more likely the text expresses a positive sentiment. However, this lexicon-based approach has limitations as it assigns a fixed sentiment orientation (positive, negative, or neutral) to a word. It does not take into account the subtleties of the texts.

3.3.2 Machine Learning algorithms

Machine learning (ML) approaches use supervised, unsupervised, or reinforcement learning to classify words, sentences, or entire documents into sentiment categories. In the case of supervised learning, the goal is to find a function f that produces an output value y given an input x . In other words, it involves creating a model that accurately predicts the outputs (labels) corresponding to new input data. The objective of unsupervised learning is to learn more about the features themselves, without making predictions like in supervised learning.

There are several supervised learning algorithms for sentiment classification. The most commonly used ones are Multinomial Naive Bayes (MNB), Maximum Entropy, Decision Trees, Support Vector Machines (SVM), and Artificial Neural Networks.

The MNB classifier operates based on conditional probabilities. It is built using the naive assumption that the features are independent of each other, which simplifies the probability calculations. During the training phase, MNB is fed with a training dataset containing examples of texts associated with their classification labels (e.g., positive/negative sentiment). It then calculates the conditional probabilities of different features (words) for each class. During the prediction phase, MNB uses these conditional probabilities to estimate the probability that a new text example belongs to each class. Using Bayes' theorem, it determines the most probable class for that example.

The Maximum Entropy classifier is also a probabilistic classifier but does not make the assumption of independence like MNB. It is based on the principle of maximizing conditional entropy and aims to find the most uniform probability distribution among all distributions that are consistent with the training data. It uses a log-linear model that assigns weights to each feature to capture their influences on the classification.

Decision Trees use a tree structure to make successive decisions based on the features of the data. The root of the tree represents the entire training data set, and each internal node of the tree corresponds to a particular feature. The branches emanating from a node represent the different possible values of that feature, and the terminal nodes correspond to predicted class labels. The process of constructing a decision tree involves recursively partitioning the training data set into smaller subsets, choosing the features that provide the best separation of classes.

The working principle of SVM relies on the concept of maximum separation between different classes. It aims to find an optimal hyperplane in a high-dimensional space that separates examples from different classes with a maximum margin. The examples closest to the hyperplane, called support vectors, are the ones that contribute the most to determining the hyperplane. SVM are generally computationally expensive for large data sets.

Finally, Artificial Neural Networks can also be used for classification. Their functioning simulates how the human brain analyzes and processes information. A neural network consists of many basic units called artificial neurons, which are organized into layers. Each neuron receives input signals, performs calculations on those signals, and produces an output. The outputs of neurons in one layer become the inputs of neurons in the next layer, allowing the propagation of information through the network. Neurons are connected by adjustable weights, which determine the strength of the influence of each neuron on neurons in the subsequent layers. Training the neural network involves iteratively adjusting these weights based on the training data to minimize a loss function.

3.4. Presentation of output

Let's remind that the main goal of sentiment analysis is to convert unstructured text into meaningful information. The last step is therefore the presentation of the results, which is done using graphs (pie charts, bar charts, line graphs). An overview of the results can be obtained using a confusion matrix.

To illustrate the performance measures presented later, it is easier to consider a binary classification (positive, negative). The confusion matrix is represented below:

		Predicted	
		True class	Negative class
Actual	True class	TP	FN
	Negative class	FP	TN

This matrix distinguishes true positives (TP) and false positives (FP) for each class. The four components of the matrix are:

True positives (TP): the number of results correctly predicted as "positive"

False positives (FP): the number of results falsely predicted as "positive"

True negatives (TN): the number of results correctly predicted as "negative"

False negatives (FN): the number of results falsely predicted as "negative"

From this confusion matrix, it is possible to calculate the main performance measures: Accuracy, F1-Score, and the Matthews Correlation Coefficient (MCC).

3.4.1 Accuracy

The Accuracy corresponds to the ratio between the number of samples correctly classified and the total number of samples. This ratio measures the rate of correct predictions. Using the notations from the confusion matrix, the ratio is:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy is easy to calculate. However, this performance measure has significant limitations when dealing with imbalanced data. Imbalanced data refers to a situation where one class is more frequent than the other. In such cases, it is possible to introduce weighting to correct the ratio (Weighted Balanced Accuracy).

3.4.2 The F1-Score

This measure combines the scores of Precision and Recall of the model. Precision measures how many "positive" predictions made by the model are correct. This measure is useful in cases where false positives are a greater concern than false negatives. This ratio is expressed as follows:

$$Precision = \frac{TP}{TP + FP}$$

Recall measures how many positive class samples present in the dataset are correctly identified by the model. This measure is therefore useful in cases where false negatives are more concerning than false positives. This ratio is expressed as follows:

$$Recall = \frac{TP}{TP + FN}$$

The *F1-Score* combines Precision and Recall by calculating their harmonic mean:

$$F1 - Score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

By directly using the confusion matrix, we obtain:

$$F1 - Score = \frac{2\,TP}{2TP + FP + FN}$$

When the dataset is imbalanced, the F1-Score may no longer be considered a reliable measure. The evaluation of the classifier's ability can be seen as overly optimistic in such cases. However, the F1-Score remains one of the most commonly used measures when it comes to assessing the performance of a model.

3.4.3 The Matthews correlation coefficient

The MCC (Matthews Correlation Coefficient) is a ratio that helps correct the issue of class imbalance. Its value is high only if the predictor has successfully predicted the majority of all data instances correctly. This measure takes a value between -1 and

+1, where extremely low values correspond to completely incorrect classification (-1) and perfect classification (+1). The value of this ratio is given by:

$$MCC = \frac{TP \ \times TN \ \ - FP \ \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Indeed, the MCC still has limitations, particularly when a confusion matrix has a row or column that is entirely null. In such cases, the ratio is undefined as its denominator is zero.

4. Transformers and BERT model

BERT (Bidirectional Encoder Representations from Transformers) is based on Transformer, which is an attention mechanism that learns contextual relationships between words in a text. Before introducing how BERT works, we will therefore briefly present the basic principles of the Transformer.

4.1. Transformers

Transformers are a type of neural network architecture that is mainly used in the field of natural language processing (NLP) and sequence understanding. Transformers were introduced by Vaswabi et al [28], from the Google team, in 2017. In their paper "Attention is all you need" they present an innovative model capable of outperforming the Long Short-Term Memory (LSTM) models hitherto used.

The main novelty in the transformer is that it is a model architecture based entirely on the attention mechanism. More precisely, it relies on self-attention. This enables models to capture long-range dependencies between all elements in a sequence. So, rather than sequentially processing elements one by one, transformers examine all elements in the sequence simultaneously, focusing on the relationships of importance between them.

A transformer is made up of several layers of attention and linear transformations, known as feed-forward layers. The principle is as follows. Each attention layer calculates attention weights for each pair of elements in the input sequence. The relative importance of each element to the other elements in the sequence is represented by these weights.

A transformer contains several layers of attention, and each layer of attention is made up of three main parts. The first part corresponds to multi-head attention. Its function is to divide attention calculations into several sub-spaces (called heads), each of which captures different relationships between the elements in the sequence. It is this mechanism that facilitates the processing of diverse and complex information. Secondly, the self-attention mechanism enables the model to take into account the relationships between all the elements in the sequence, using attention weights. This is an innovation, since it enables each element to attach itself to all the other elements, whether distant or nearby. In this way, the mechanism captures long-range dependencies. Finally, the third part corresponds to transformation operations. These include linear and non-linear operations (i.e. feed-forward layers). This enables the model to mix and transform information from different parts of the sequence.

To conclude, transformers enable models to transform sequences of raw data into very rich representations. Transformers have gained immense success in various applications such as machine translation, text generation, and natural language understanding, thanks to their remarkable ability in these domains. They far exceed the performance of traditional models based on recurrent neural networks (RNNs), and also enable greater computational efficiency.

4.2. The BERT model

BERT (Bidirectional Encoder Representations from Transformers) is a specific model based on the Transformer architecture. It was introduced by Google in 2018 [29] and has become extremely popular in the field of NLP. This model led to a publication by Delvin et al [30]. BERT is designed for natural language understanding and excels in tasks such as text classification, text comprehension and question-answering.

The main feature of BERT, which sets it apart from traditional models, is that it is a bidirectional model of context learning. Unlike previous language models, which used unidirectional architectures or limited context windows, BERT uses a pre-training task called "masked language modeling" (MLM). This involves randomly masking some words in a sentence and training the model to predict them based on the context of the surrounding words. This bidirectional approach enables BERT to capture rich contextual information on both sides of a word or sequence. This in turn allows it to better understand contextual relationships and gain a more precise understanding of the meaning of words in a given sequence.

Another distinctive feature of BERT is that it trains on huge quantities of unsupervised data, such as texts published on the web. The model thus learns general linguistic representations which can then be adapted to specific tasks with relatively less supervised data.

To conclude, the operation of BERT can be divided into two key steps: pre-training and fine-tuning. In the pre-training step, BERT is trained on vast quantities of unsupervised data, generally web texts. In this first stage, BERT uses an MLM approach. BERT also uses the "next sentence prediction" (NSP) which corresponds to an additional learning task. This involves providing the model with two consecutive sentences and training it to predict whether or not they are adjacent in the original text. This phase is important, as it allows BERT to understand the relationships between sentences in order to capture the overall context of the text.

In the second step, which is the fine-tuning step, BERT is adjusted on specific tasks with less supervised data. To clarify, fine-tuning consists of adjusting model weights using task-specific annotated data. So, during this phase, one or more additional layers are added to the pre-trained BERT model, forming a task-specific architecture. These new layers are randomly initialized and the model weights are adjusted using an optimization algorithm (such as gradient backpropagation) to minimize a task-specific loss function. BERT's fine-tuning is crucial as it allows the knowledge gained from prior training to be transferred to specific tasks, reducing the need for massive supervised data for each individual task. This model thus achieves high performance on various natural language processing tasks with a relatively smaller amount of supervised data.

5. Model methodology, analysis and discussion

This section describes the steps and methods in creating the sentiment analysis model. It also gives the main results obtained by using this model to compare sentiments in different countries for the same company.

During this research, I focused on developing Python computer code with the intention of leveraging my expertise in deep learning and natural language processing (NLP) to create a valuable program for the finance and investment domains.

Investors and traders dedicate significant time to reading the press, seeking information about a country's economic state, a company's performance, and other relevant factors to make informed investment decisions. Therefore, I embarked on the idea of designing a program that conducts sentiment analysis on press articles related to a specific company, comparing data from two different countries, namely Country A and Country B.

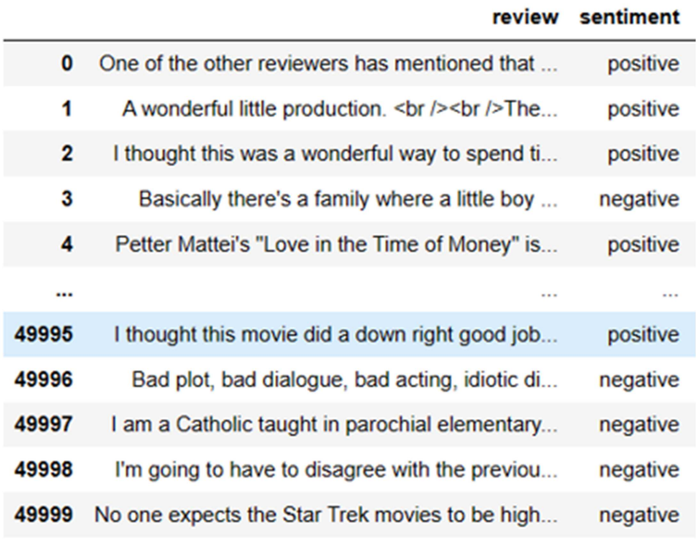
By employing sentiment analysis, the program can assess the overall sentiment expressed in online media articles about the company in each country. For example, Country A might generate more positive articles, while Country B might produce more negative articles regarding the same company. This approach allows for a unique perspective on the company's evolution, based on varying sentiments from different regions.

The rest of the section consists of several sections, each contributing to the development and evaluation of my sentiment analysis model. Section 5.1 introduces and explains the construction of the sentiment analysis model. Additionally, this section provides insights into the accuracy of the model's performance. Section 5.2 focuses on discussing the specific aspect of the code related to data scraping, which is used to gather relevant articles about various companies. Section 5.3 presents the complete code, incorporating both data scraping and sentiment analysis, and offer an illustrative example featuring a particular company. Section 5.4 compares the results of the sentiment analysis with the variation of stock prices. Lastly, Section 5.5 concludes the project by discussing the limitations of the code, acknowledging its potential constraints and areas for improvement.

5.1. The sentiment analysis model

5.1.1 Dataset

For training, validation, and testing my model, I worked with the IMDB database that I downloaded from Kaggle[1]. The IMDB database I used consists of 50,000 movie reviews, with each review labeled as either positive or negative sentiment. Figure 1 shows an extract from the IMDB Dataset by way of illustration.

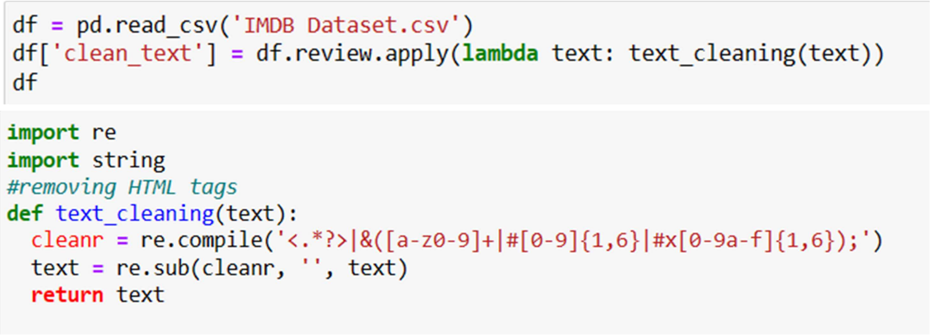


	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production. <br /><br />The...	positive
2	I thought this was a wonderful way to spend ti...	positive
3	Basically there's a family where a little boy ...	negative
4	Petter Mattei's "Love in the Time of Money" is...	positive
...	...	...
49995	I thought this movie did a down right good job...	positive
49996	Bad plot, bad dialogue, bad acting, idiotic di...	negative
49997	I am a Catholic taught in parochial elementary...	negative
49998	I'm going to have to disagree with the previou...	negative
49999	No one expects the Star Trek movies to be high...	negative

Figure 1: Extract from the IMDB Dataset

Before working with the IMDB dataset, the text has been clean up in order to use the BERT Tokenizer later. Accordingly, the implementation takes few lines of code as shown in Figure 2.

[1] The IMDB Dataset is downloaded from Kaggle: https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews

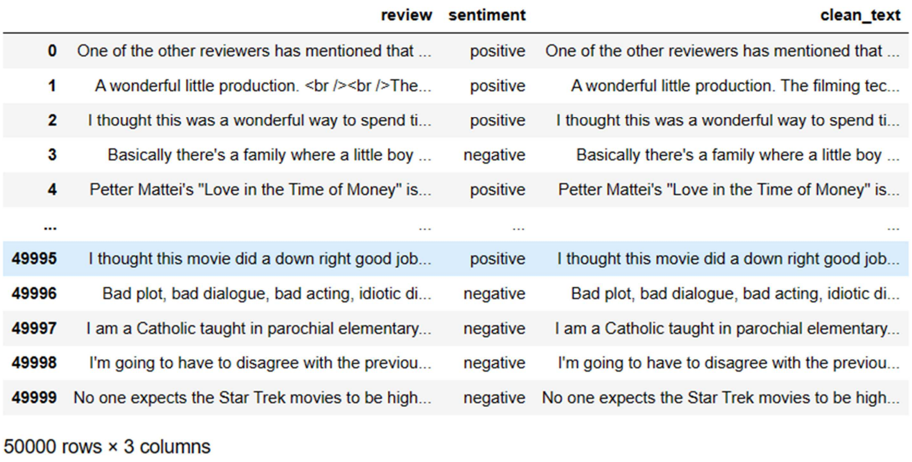


```
df = pd.read_csv('IMDB Dataset.csv')
df['clean_text'] = df.review.apply(lambda text: text_cleaning(text))
df
```

```
import re
import string
#removing HTML tags
def text_cleaning(text):
  cleanr = re.compile('<.*?>|&([a-z0-9]+|#[0-9]{1,6}|#x[0-9a-f]{1,6});')
  text = re.sub(cleanr, '', text)
  return text
```

Figure 2: Code required to clean up the text

After text cleaning, the dataset includes a 3rd column for *clean_text* as shown in Figure 3.



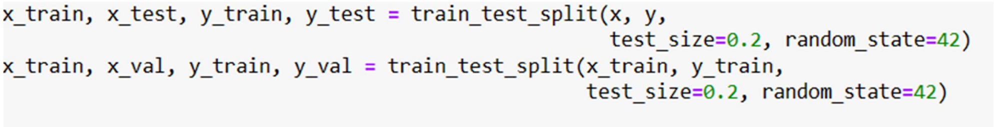
	review	sentiment	clean_text
0	One of the other reviewers has mentioned that ...	positive	One of the other reviewers has mentioned that ...
1	A wonderful little production. <br /><br />The...	positive	A wonderful little production. The filming tec...
2	I thought this was a wonderful way to spend ti...	positive	I thought this was a wonderful way to spend ti...
3	Basically there's a family where a little boy ...	negative	Basically there's a family where a little boy ...
4	Petter Mattei's "Love in the Time of Money" is...	positive	Petter Mattei's "Love in the Time of Money" is...
...	...	...	...
49995	I thought this movie did a down right good job...	positive	I thought this movie did a down right good job...
49996	Bad plot, bad dialogue, bad acting, idiotic di...	negative	Bad plot, bad dialogue, bad acting, idiotic di...
49997	I am a Catholic taught in parochial elementary...	negative	I am a Catholic taught in parochial elementary...
49998	I'm going to have to disagree with the previou...	negative	I'm going to have to disagree with the previou...
49999	No one expects the Star Trek movies to be high...	negative	No one expects the Star Trek movies to be high...

50000 rows × 3 columns

Figure 3: Extract from the IMDB Dataset after text cleaning

5.1.2 Data processing

After obtaining the dataset, I proceeded to split the data into distinct portions for training and testing the model. Specifically, I divided the dataset into three subsets: 32,000 samples for training, 8,000 samples for validation, and 10,000 samples for testing. This division allowed me to have separate data for training and fine-tuning the model's parameters, as well as data for evaluating its performance during testing. The following Figure 4 gives the data split.



```
x_train, x_test, y_train, y_test = train_test_split(x, y,
                                                    test_size=0.2, random_state=42)
x_train, x_val, y_train, y_val = train_test_split(x_train, y_train,
                                                  test_size=0.2, random_state=42)
```

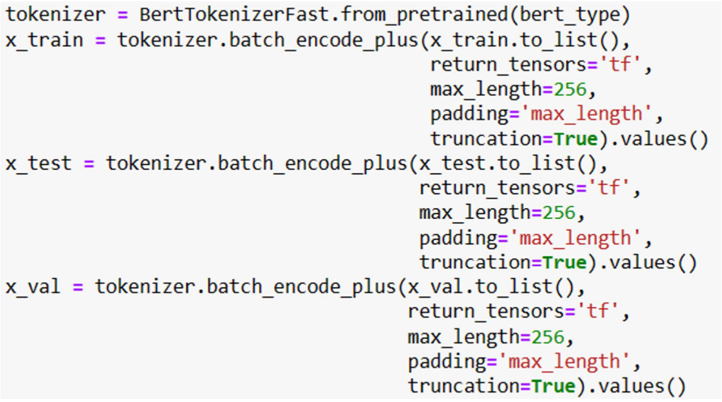
Figure 4: Code required for the data split

The next step is to tokenize and vectorize the data. First, I load a BERT model using the *TFBertModel.from_pretrained(bert_type)* function, where *bert_type* specifies the specific variant of BERT to be used. Here I chose to use "bert-base-cased". This pre-trained BERT model will be used for NLP tasks.

Then, I create a tokenizer using the *Bert Tokenizer Fast.from_pretrained (bert_type)* function. The tokenizer is responsible for converting raw text into numerical representations suitable for input to the BERT model.

The maximum length of the tokenized sequences is set to 256 (max_length = 256). Any text longer than this length will be truncated and shorter texts will be padded to reach the specified length. This ensures that all input sequences are of the same length, a requirement for most NLP models.

This way I tokenized then vectorized all the datas : *x_train* ; *x_test* and *x_val.* The code is defined as shown in Figure 5.



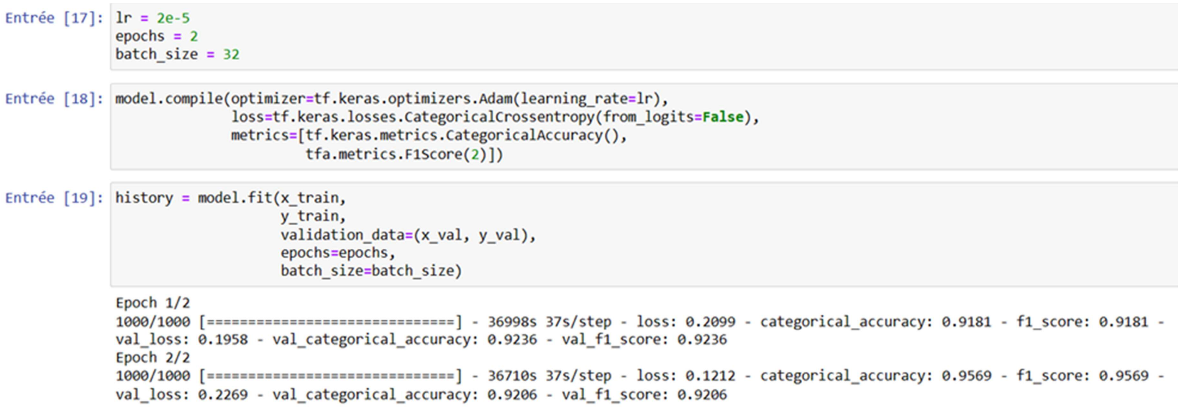
```
tokenizer = BertTokenizerFast.from_pretrained(bert_type)
x_train = tokenizer.batch_encode_plus(x_train.to_list(),
                                      return_tensors='tf',
                                      max_length=256,
                                      padding='max_length',
                                      truncation=True).values()
x_test = tokenizer.batch_encode_plus(x_test.to_list(),
                                     return_tensors='tf',
                                     max_length=256,
                                     padding='max_length',
                                     truncation=True).values()
x_val = tokenizer.batch_encode_plus(x_val.to_list(),
                                    return_tensors='tf',
                                    max_length=256,
                                    padding='max_length',
                                    truncation=True).values()
```

Figure 5: Code required for tokenization and vectorization

5.1.3 The model

To build the model I used the softmax activation function and categorical cross entropy loss. After fine tuning, I found that with a learning rate of 2.10^{-5}, a number of epoch=2 and a batch size = 32 the model was very good (validation loss = 0.22 and validation accuracy = F1-score = 0.92). These results are given in Figure 6.

The model's results could probably have been improved with a greater number of epochs, but the code was very time-consuming to execute, taking around 2 days for each run.



```
Entrée [17]: lr = 2e-5
             epochs = 2
             batch_size = 32

Entrée [18]: model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=lr),
                           loss=tf.keras.losses.CategoricalCrossentropy(from_logits=False),
                           metrics=[tf.keras.metrics.CategoricalAccuracy(),
                                    tfa.metrics.F1Score(2)])

Entrée [19]: history = model.fit(x_train,
                                 y_train,
                                 validation_data=(x_val, y_val),
                                 epochs=epochs,
                                 batch_size=batch_size)

             Epoch 1/2
             1000/1000 [==============================] - 36998s 37s/step - loss: 0.2099 - categorical_accuracy: 0.9181 - f1_score: 0.9181 -
             val_loss: 0.1958 - val_categorical_accuracy: 0.9236 - val_f1_score: 0.9236
             Epoch 2/2
             1000/1000 [==============================] - 36710s 37s/step - loss: 0.1212 - categorical_accuracy: 0.9569 - f1_score: 0.9569 -
             val_loss: 0.2269 - val_categorical_accuracy: 0.9206 - val_f1_score: 0.9206
```

Figure 6: Code for the model

The results from evaluating the model are presenting in the following.

The evolution of the training loss and the validation loss are given in Figure 7a. We can see that at epoch 1, training loss was at 0.21 and got down to 0.12 at epoch 2 while the validation loss went from 0.195 at epoch 1 to 0.225 at epoch 2.

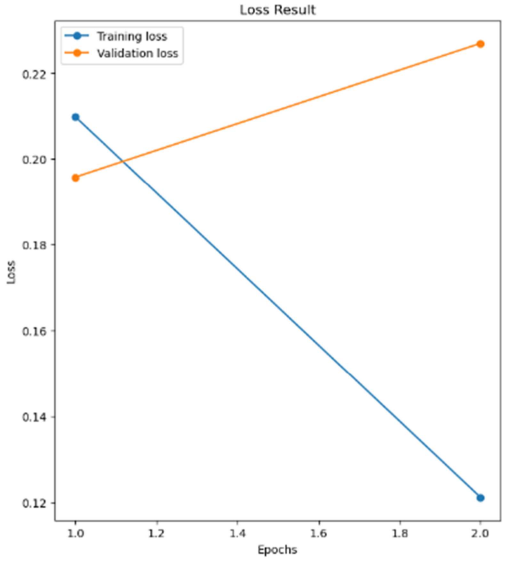


Figure 7a: Training loss and validation loss

The training accuracy and the validation accuracy are presented in Figure 7b. We can see that at epoch 1, training accuracy was at 0.918 and got up to 0.955 at epoch 2 while the validation accuracy went from about 0.924 at epoch 1 to 0.920 at epoch 2.

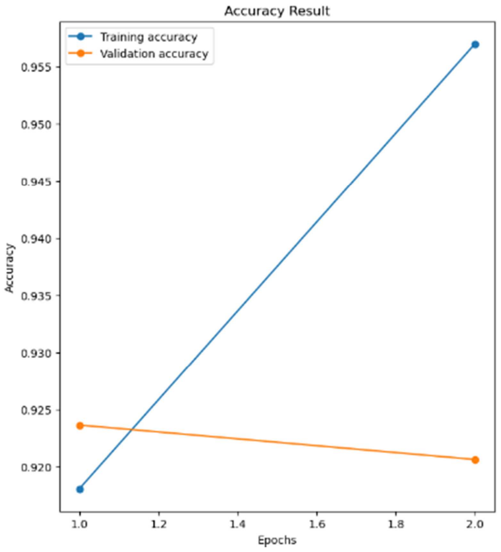


Figure 7b: Training accuracy and validation accuracy

The F1-Score evolves as shown in Figure 7c. This is very similar to the accuracy graph.

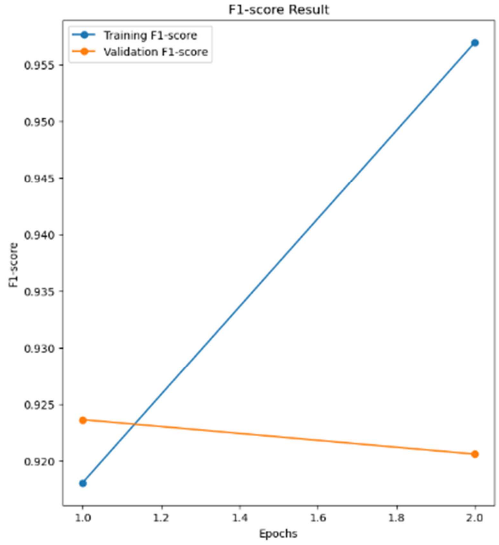
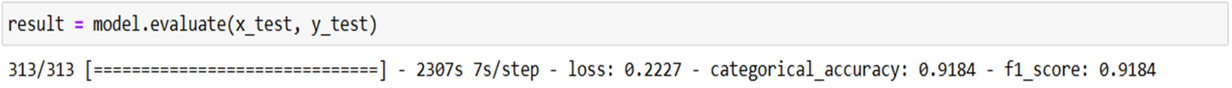


Figure 7c: The F1-score

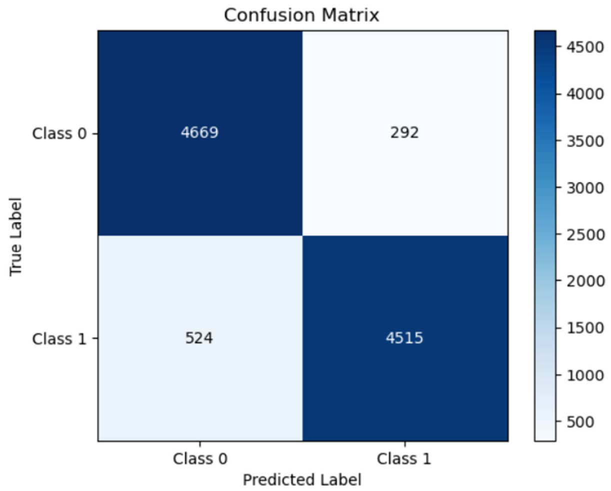
After training and tuning the hyper parameter, I finally tested my model on my « test » data and obtained a loss of 0.2227 and F1-score = accuracy = 0.9184 as shown in Figure 8.



```
result = model.evaluate(x_test, y_test)
313/313 [==============================] - 2307s 7s/step - loss: 0.2227 - categorical_accuracy: 0.9184 - f1_score: 0.9184
```

Figure 8: The test of the model

Finally, the Figure 9 shows the confusion matrix result while using the NLP model on the test data.



Class 0 = « Negative »

Class 1 = « Positive »

Figure 9: The confusion matrix

The confusion matrix represents the outcomes of testing my classification model with the *X_test*, *Y_test* data from the IMDB dataset. It shows that out of 4961 samples of class 0 (label « Negative »), 4669 were correctly predicted (True Negatives), but 292 were misclassified as class 1 (False Positives). Moreover, out of 5043 samples of class 1 (label « Positive »), 4515 were correctly predicted (True Positives), while 524 were misclassified as class 0 (False Negatives). Overall, the model appears to perform well.

This trained model can now be used to categorize any article into one of the two sentiment categories with a useful accuracy.

5.2. Data scrapping

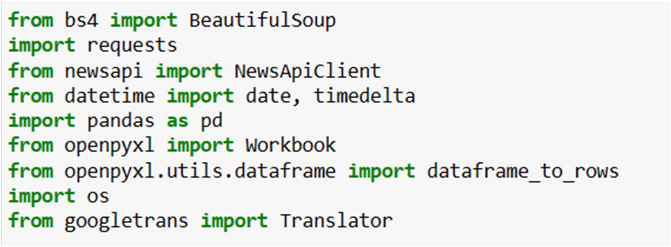
In the model, I undertook a comparative analysis by examining American articles in English from American sources and French articles in French from French sources, all pertaining to the same company.

To achieve this, I implemented two distinct specifications—one for scraping American articles and another for scraping French articles. While both specifications share similarities, the code for the French articles required additional complexity due to the necessity of using my sentiment analysis model, which operates exclusively on English text.

Consequently, to utilize the sentiment analysis model effectively, I had to first translate all the French articles into English. This translation process introduced intricacies and nuances, making the code for scraping French articles slightly more intricate than its American counterpart.

The following sections delve into the code for scraping French articles, providing a detailed explanation of the techniques employed for data retrieval, translation, and data preparation for the sentiment analysis model.

Figure 10 indicates the list of libraries I used for the data scrapping code for French articles:



```
from bs4 import BeautifulSoup
import requests
from newsapi import NewsApiClient
from datetime import date, timedelta
import pandas as pd
from openpyxl import Workbook
from openpyxl.utils.dataframe import dataframe_to_rows
import os
from googletrans import Translator
```

Figure 10: List of libraries for French articles

The crux of the code lies in retrieving articles that match a user-selected keyword. For instance, if the user inputs "*AAPL*" as the keyword, the code will fetch articles related to Apple.

To achieve this, I harnessed the power of the "*newsApi*" API, which allowed me to specify the publication date range. By setting the date range from the code execution day back to 30 days, I ensured that the articles retrieved were recent and relevant.

Furthermore, I meticulously curated a list of reliable and pertinent website sources. This selection process guaranteed that the articles gathered originated from trustworthy outlets, enhancing the quality and reliability of the data.

Once the articles were identified, the next step involved extracting their content from the HTML pages. This, however, posed a challenge, as HTML structures can vary significantly between different websites. Some articles were wrapped in *<p>* tags, while others were encapsulated within *<div>*, *<article>*, or *<section>* tags.

To navigate this complexity, I skillfully employed the *BeautifulSoup* library. Despite the varying HTML structures, I successfully extracted the article content, ensuring accurate and comprehensive data for subsequent analysis.

By adeptly handling the retrieval and extraction processes, my code empowers users to access pertinent articles based on their chosen keywords and enables them to gain valuable insights from reliable sources.

Once I retrieved the text of the articles in French, I had to translate it into English. For each article, I saved a .txt file containing the translated English text. To accomplish this translation process in my code, I used the library *Translator* from *googletrans*.

However, this library had a limitation where it could only translate 500 characters at a time. Consequently, I had to divide each of my articles into segments of 500 characters to ensure they could be fully translated.

5.3. Complete code

The complete code utilizes the Sentiment Analysis NLP model and the data scraping code.

First, the user provides a keyword as an argument to the main function. The code then uses the data scraping function to retrieve articles that match the indicated keyword. The Sentiment Analysis model is subsequently applied to each article, and the results are saved in an Excel file and a dataframe.

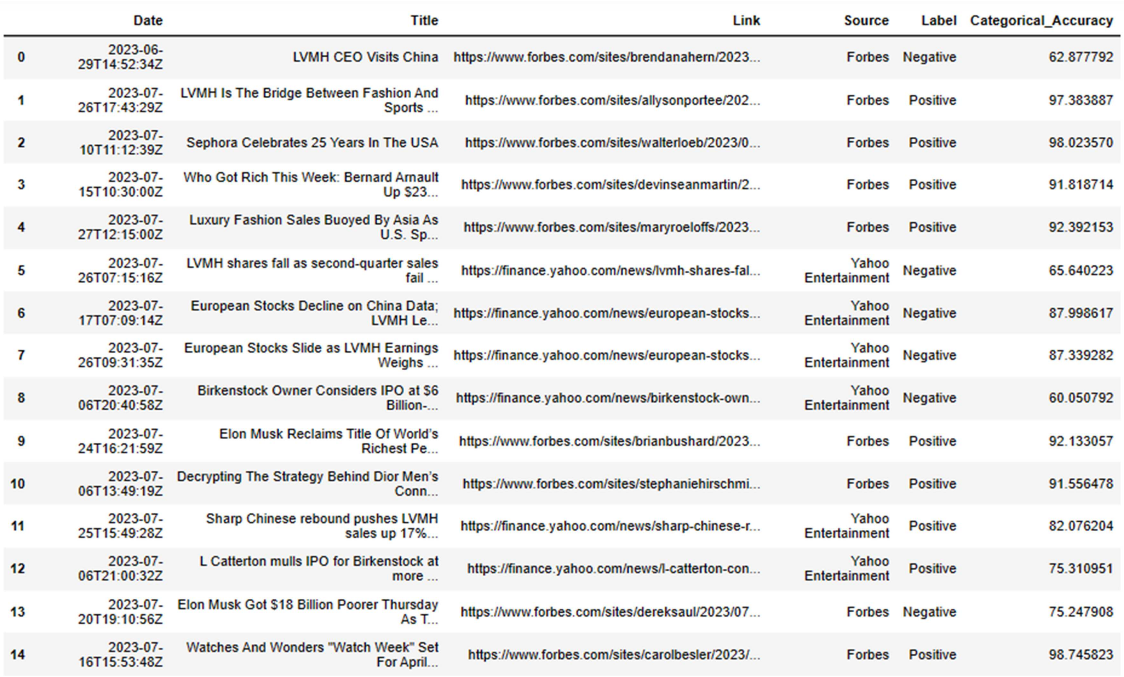
The dataframe contains four columns:

- Date (Date of publication of the article)
- Title (title of the article)
- Link (Link of the article)
- Source (webstite where the article comes from)
- Label (result of the sentiment analysis : « positive » or « negative »)
- Categorical_accuracy (accuracy of the sentiment analysis result)

The objective is to run the algorithm twice with the same keyword, once for American articles and once for French articles. The results for both sets are saved, allowing for a comparison between the French and American sentiments.

To facilitate a more meaningful comparison, the code calculates the average accuracy results for both "positive" and "negative" sentiments for each set. This approach enables a comprehensive evaluation of sentiment differences between the two regions

The results for the French company « LVMH » with only American articles are shown in Figure 11a.

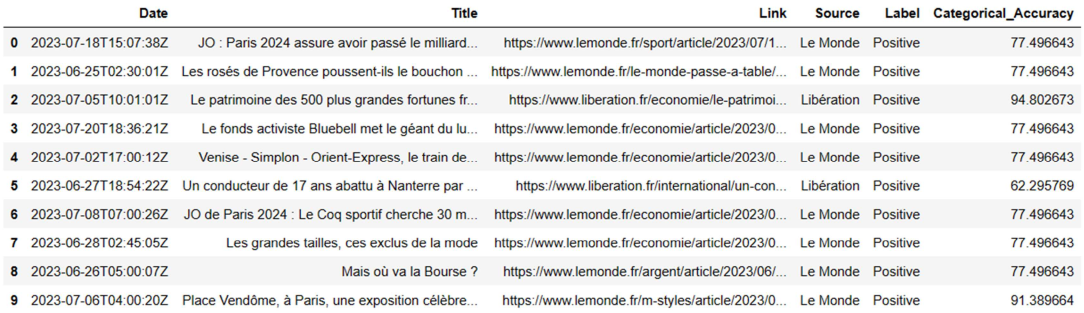


	Date	Title	Link	Source	Label	Categorical_Accuracy
0	2023-06-29T14:52:34Z	LVMH CEO Visits China	https://www.forbes.com/sites/brendanahern/2023...	Forbes	Negative	62.877792
1	2023-07-26T17:43:29Z	LVMH Is The Bridge Between Fashion And Sports ...	https://www.forbes.com/sites/allysonportee/202...	Forbes	Positive	97.383887
2	2023-07-10T11:12:39Z	Sephora Celebrates 25 Years In The USA	https://www.forbes.com/sites/walterloeb/2023/0...	Forbes	Positive	98.023570
3	2023-07-15T10:30:00Z	Who Got Rich This Week: Bernard Arnault Up $23...	https://www.forbes.com/sites/devinseanmartin/2...	Forbes	Positive	91.818714
4	2023-07-27T12:15:00Z	Luxury Fashion Sales Buoyed By Asia As U.S. Sp...	https://www.forbes.com/sites/maryroeloffs/2023...	Forbes	Positive	92.392153
5	2023-07-26T07:15:16Z	LVMH shares fall as second-quarter sales fail ...	https://finance.yahoo.com/news/lvmh-shares-fal...	Yahoo Entertainment	Negative	65.640223
6	2023-07-17T07:09:14Z	European Stocks Decline on China Data; LVMH Le...	https://finance.yahoo.com/news/european-stocks...	Yahoo Entertainment	Negative	87.998617
7	2023-07-26T09:31:35Z	European Stocks Slide as LVMH Earnings Weighs ...	https://finance.yahoo.com/news/european-stocks...	Yahoo Entertainment	Negative	87.339282
8	2023-07-06T20:40:58Z	Birkenstock Owner Considers IPO at $6 Billion-...	https://finance.yahoo.com/news/birkenstock-own...	Yahoo Entertainment	Negative	60.050792
9	2023-07-24T16:21:59Z	Elon Musk Reclaims Title Of World's Richest Pe...	https://www.forbes.com/sites/brianbushard/2023...	Forbes	Positive	92.133057
10	2023-07-06T13:49:19Z	Decrypting The Strategy Behind Dior Men's Conn...	https://www.forbes.com/sites/stephaniehirschmi...	Forbes	Positive	91.556478
11	2023-07-25T15:49:28Z	Sharp Chinese rebound pushes LVMH sales up 17%...	https://finance.yahoo.com/news/sharp-chinese-r...	Yahoo Entertainment	Positive	82.076204
12	2023-07-06T21:00:32Z	L Catterton mulls IPO for Birkenstock at more ...	https://finance.yahoo.com/news/l-catterton-con...	Yahoo Entertainment	Positive	75.310951
13	2023-07-20T19:10:56Z	Elon Musk Got $18 Billion Poorer Thursday As T...	https://www.forbes.com/sites/dereksaul/2023/07...	Forbes	Negative	75.247908
14	2023-07-16T15:53:48Z	Watches And Wonders "Watch Week" Set For April...	https://www.forbes.com/sites/carolbesler/2023/...	Forbes	Positive	98.745823

Number of ‘Positive’ labels: 9
Number of ‘Negative’ labels: 6

Figure 11a: Results obtained for the French company LVMH in American articles

The results for « LVMH » with only French articles are shown in Figure 11b.



	Date	Title	Link	Source	Label	Categorical_Accuracy
0	2023-07-18T15:07:38Z	JO : Paris 2024 assure avoir passé le milliard...	https://www.lemonde.fr/sport/article/2023/07/1...	Le Monde	Positive	77.496643
1	2023-06-25T02:30:01Z	Les rosés de Provence poussent-ils le bouchon ...	https://www.lemonde.fr/le-monde-passe-a-table/...	Le Monde	Positive	77.496643
2	2023-07-05T10:01:01Z	Le patrimoine des 500 plus grandes fortunes fr...	https://www.liberation.fr/economie/le-patrimoi...	Libération	Positive	94.802673
3	2023-07-20T18:36:21Z	Le fonds activiste Bluebell met le géant du lu...	https://www.lemonde.fr/economie/article/2023/0...	Le Monde	Positive	77.496643
4	2023-07-02T17:00:12Z	Venise - Simplon - Orient-Express, le train de...	https://www.lemonde.fr/economie/article/2023/0...	Le Monde	Positive	77.496643
5	2023-06-27T18:54:22Z	Un conducteur de 17 ans abattu à Nanterre par ...	https://www.liberation.fr/international/un-con...	Libération	Positive	62.295769
6	2023-07-08T07:00:26Z	JO de Paris 2024 : Le Coq sportif cherche 30 m...	https://www.lemonde.fr/economie/article/2023/0...	Le Monde	Positive	77.496643
7	2023-06-28T02:45:05Z	Les grandes tailles, ces exclus de la mode	https://www.lemonde.fr/economie/article/2023/0...	Le Monde	Positive	77.496643
8	2023-06-26T05:00:07Z	Mais où va la Bourse ?	https://www.lemonde.fr/argent/article/2023/06/...	Le Monde	Positive	77.496643
9	2023-07-06T04:00:20Z	Place Vendôme, à Paris, une exposition célèbre...	https://www.lemonde.fr/m-styles/article/2023/0...	Le Monde	Positive	91.389664

Number of ‘Positive’ labels: 10
Number of ‘Negative’ labels: 0

Figure 11b: Results obtained for LVMH in French articles

Figures 11a & 11b show the results of the model. For French articles, the mean of positive Categorical Accuracy is 79.1%. There are 10 “Positive” labels and 0 “Negative” labels. Results are different for American articles. Indeed, the mean of positive Categorical Accuracy for American articles is 91%, while for negative Categorical Accuracy, it is 73.2%. There are 9 “Positive” labels and 6 “Negative” labels.

We can make meaningful comparisons by examining the number of positive labels in both France and the United States, considering about the same number of studied articles, along with the respective average *categorical_accuracy* for each group.

The interpretation of results is as follow. American articles tend to have higher average accuracy for positive labels compared to French articles. This suggests that the predicted probabilities for positive labels are generally higher in American articles.

The number of positive labels is higher in French articles, which might indicate that positive events or topics are more prevalent in the news from France.

Conversely, the number of negative labels is higher in American articles, suggesting that negative events or topics may be more prominent in the news from the United States.

5.4. Discussion

This research suggests that investor sentiment can vary across regions. Different sentiments could therefore contribute to explain why, for the same company, the stock price and its variation could differ across stock markets.

To provide an illustration, it is intriguing to examine how the stock price of LVMH evolved over the chosen study period within the code execution. Figures 11a and 11b indicate that the publication dates of American articles fall between June 29 and July 27, while those of the French articles were published between June 25 and July 20.

A simple review of the price history of LVMH stock on Euronext Paris shows that on June 26th (the stock market was closed on June 25th), the stock was priced at €835, and on July 20th, it was trading at €854. Therefore, over this period, the stock increased by 2.27%. On the New York Stock Exchange, LVMH stock was trading at $186.58 on June 29th and $182.51 on July 27th, representing a decrease of 2.18% over this period. We observe that these variations are quite consistent with the results we have obtained in the sentiment analysis. However, the comparison concerning the stock's evolution on the New York Stock Exchange is more subject to interpretation.

The previous discussion should be approached with significant caution. Certainly, the price of a stock and its fluctuations are influenced by a multitude of factors. Indeed, the financial performance of the company and economic conditions (such as economic growth, inflation, interest rates, etc.) play a crucial role. Therefore, this discussion would require a rigorous analysis of the impact of investor sentiments on stock prices.

5.5. Limitations

While the code serves its purpose and effectively illustrates the research project, it does have certain limitations that offer opportunities for improvement. The main constraint arises from budget limitations, compelling me to rely on trial periods for APIs, especially "news API," which restricts the ability to scrape more than 100 articles per 24 hours. Consequently, in the examples and datasets provided, I have chosen to limit the number of articles to just 20 so that I can run the code several times in a single day. However, due to some articles being protected and inaccessible for scraping, the final count of articles was reduced to 10 for French articles and 15 for American articles.

In fact, another challenge I encountered was when attempting to scrape articles from certain web pages. Some websites implement systems that hinder or prevent access to article content through scraping. While some paid APIs could potentially overcome this obstacle, financial constraints prevent me from utilizing such solutions.

To enhance the code, I could focus on implementing a more efficient translation system. Currently, I divide the text into 500-character segments to avoid translation costs for lengthier texts. A more sophisticated translation approach would considerably improve the execution speed.

Furthermore, I recognize that utilizing more advanced paid APIs would significantly expand the capabilities of the code. These APIs could offer better article selection and higher accuracy. Additionally, subscribing to paid access from specific newspapers would provide valuable advantages during the scraping process, allowing me to access and gather articles that align precisely with the research interests.

In conclusion, while the code functions adequately, addressing its limitations by exploring alternative solutions and investing in more advanced tools would undoubtedly elevate its performance and enrich the research outcomes.

6. Conclusion

With the abundance of economic and financial information available in the market, amplified by online media platforms, investors have a profound interest in harnessing Machine Learning and Natural Language Processing (NLP) tools. The continuous development of innovations and services in this area has raised high expectations. This research project aligns perfectly with this context.

In this project, we begin by elucidating some crucial financial market concepts to facilitate a deeper insight. Over the past years, numerous studies have been conducted to grasp the impact of news on investors' decisions and the intricate relationship between news and stock prices. To this end, we outline the fundamental steps of sentiment analysis, along with the implementation of the Transformers principle and the BERT model.

In this study, we propose a model that utilizes recent NLP algorithms to conduct market sentiment analysis in two distinct countries. The underlying idea is that online media may offer varying information about companies across different countries, and investors can also be influenced by regional news. As a result, regional news can account for the diverse behaviors observed among investors in various regional markets, leading to variations in stock prices or fluctuations between these markets. Gaining a deeper comprehension of this phenomenon also empowers traders to fine-tune their investment strategies more effectively. To facilitate this comparative analysis of market sentiment between the two countries, we have developed a Python code as part of this project. The model's results demonstrate promising qualitative performance, yet there is room for improvement in information processing through various avenues. The work was executed within specific constraints, which could potentially be further narrowed down. As a result, this study presents exciting prospects for future research and exploration.

7. References

[1] Fama, E. F. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. Journal of Finance, 383-417.

[2] CFA Institute. (2019). Investment Professional of the future. CFA Institute.

[3] Baker, M. and Wurgler J. (2007). Investor Sentiment in the Stock Market. Journal of Economic Perspectives, vol.21(2), 129-151.

[4] Baker, Malcolm, and Jeffrey Wurgler. (2006). "Investor Sentiment and the Cross-Section of Stock Returns," Journal of Finance, 61(4): 1645–80.

[5] Baker, M., Wurgler J. and Yuan Y. (2012). Global, local, and contagious investor sentiment. Journal of Financial Economics, vol.104 (2), 272-287.

[6] Mishev, K., Gjorgjevikj, A., Vodenska, I., Chitkushev, L., & Trajanov, D. (2020). Evaluation of Sentiment Analysis in Finance: From Lexicons to Transformers. IEEE Access, 1 - 2.

[7] Fama, E. F. (1991). Efficient Capital Markets: II. The Journal of Finance, 46(5), 1575-1617.

[8] Samuelson, P. A. (1973). Proof that properly discounted present values of assets vibrate randomly. The Bell Journal of Economics and Management Science, 369-374.

[9] Lo, A. W., & MacKinlay, A. C. (2011). A non-random walk down Wall Street. Princeton University Press.

[10] Qian, B., & Rasheed, K. (2007). Stock market prediction with multiple classifiers. Applied Intelligence, 26(1), 25-33.

[11] Gallagher, L. A., & Taylor, M. P. (2002). The stock return–inflation puzzle revisited. Economics Letters, 75(2), 147-156.

[12] Ashtiani, M., Raahemi, B.. (2023). News-based intelligent prediction of financial markets using text mining and machine learning: A systematic literature review. Expert Systems with Applications, vol. 217, 119509.

[13] Gruhl, D., Guha, R., Kumar, R., Novak, J., & Tomkins, A. (2005). The predictive power of online chatter. In Proceedings of the eleventh ACM SIGKDD International conference on Knowledge discovery in data mining, 78-87.

[14] Mishne, G., & Glance, N. S. (2006). Predicting movie sales from blogger sentiment. In AAAI spring symposium: computational approaches to analyzing weblogs, 155-158.

[15] Choi, H., & Varian, H. (2012). Predicting the present with Google Trends. Economic Record, 88, 2-9.

[16] Veronesi P. (1999). Stock market overreactions to bad news in good times: a rational expectations equilibrium model. The Review of Financial Studies, vol. 12(5), 975-1007.

[17] Chan W.S. (2003). Stock price reaction to news and no-news: drift and reversal after headlines, Journal of Financial Economics, vol. 70(2), 223–260.

[18] Li F. (2006). Do stock market investors understand the risk sentiment of corporate annual reports?, Working paper.

[19] Li Q., Y. Chen, J. Wang, Y. Chen and H. Chen. (2017). Web Media and Stock Markets: A Survey and Future Directions from a Big Data Perspective, IEEE.

[20] Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. Journal of Computational Science, vol. 2(1), 1-8.

[21] Schumaker, R. P., & Chen, H. (2009). Textual analysis of stock market prediction using breaking financial news: The AZFin text system. ACM Transactions on Information Systems (TOIS), 27(2), 1-19.

[22] Li, Q., Wang, T., Li, P., Liu, L., Gong, Q., & Chen, Y. (2014). The effect of news and public mood on stock movements. Information Sciences, 278, 826-840.

[23] Gidofalvi, G., & Elkan, C. (2001). Using news articles to predict stock price movements. Department of Computer Science and Engineering, University of California, San Diego.

[24] Segal, T. (2020). Fundamental Analysis. Retrieved from https://www.investopedia.com/terms/f/fundamentalanalysis.asp.

[25] Hayes, A. (2020). Technical Analysis Definition. Retrieved from https://www.investopedia.com/terms/t/technicalanalysis.asp#citation-4.

[26] Angiani, G., Ferrari, L., Fontanini, T., Fornacciari, P., Iotti, E., Magliani, F., & Manciardi, S. (2016). A Comparison between Preprocessing Techniques for Sentiment

Analysis in Twitter. Proceedings of the 2nd International Workshop on Knowledge Discovery on the WEB, 8 – 10.

[27] Mat Zin, H., Mustapha, N., Murad, M., & Sharef, N. (2017). The effects of pre-processing strategies in sentiment analysis of online movie reviews. The 2nd International Conference on Applied Science and Technology, 1 – 8.

[28] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Polosukhin et al. (2017). Attention Is All You Need. 31st Conference on Neural Information Processing System, 1 - 15.

[29] Devlin, J., & Chang, M.-W. (2018). Open Sourcing BERT: State-of-the-Art Pre-training for Natural Language Processing. Retrieved from Google AI Blog: https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html

[30] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1, 4171–4186.