Analysis of sentiment in financial news articles using deep neural networks

Business Project submitted for the degree of Master of Science in

Business Analytics

By,

Prasanna Venkatesan, BEng (Hons.), MSc

Aston Business School

I, Prasanna Venkatesan, confirm that the work presented in this project is my own. Where information has been derived from other sources, I confirm that this has been indicated in the project.

January 2022

Acknowledgement

I would first like to thank my supervisor, Dr. Viktor Pekar, for giving me this opportunity to conduct this project and his invaluable expertise in formulating the thesis topic and methodology in particular. I would also like to thank my family and friends for their continuous support and assistance throughout the writing of this project.

Table of Contents

List of Figures

List of Tables

Abstract

Automated analysis of sentiment expressed in financial news with respect to different financial instruments (stocks, commodities, currencies) is an important part of the decision making and risk assessment of modern investors. A common approach to automatically analyze sentiment expressed in text is to use machine-learning-based Natural Language Processing methods which would assign a categorical label, such as “positive”, “negative” or “neutral”. In the context of financial sentiment analysis, the goal is to interpret a piece of financial text as expressing bullish or bearish opinions toward certain financial instruments. Recent research has shown that deep learning methods, such as USE and BERT language models are able to provide superior results compared to traditional methods such as bag-of-words (BOW) when used with machine learning methods. Therefore, the aim of this project is to develop and compare financial Sentiment Analysis models using traditional text representation models with new models such as USE and BERT. A Financial Phrasebank containing labelled news was used to build Sentiment Analysis models using BOW, USE and BERT. The results indicated that SVM classifiers using BERT outperformed both BOW and USE models. A comparison between SVM and MLP Neural Networks using BERT revealed that SVM outperformed MLP. Therefore, the results mainly suggest that SVM classifiers using BERT can be used for real world Sentiment Analysis while taking into consideration the limitations discussed in the project. The dataset, Machine Learning Models and the Jupyter Notebook containing the codes for this project can be found in this link.

Chapter 1

1. Introduction

Alan Turing's work "Computing Machinery and Intelligence" sparked interest in Natural Language Processing in 1950. Turing claimed that a computer could be called intelligent if it could converse with a human without the human recognizing they were conversing with a machine [1]. Since then, NLP has come a long way, with significant advances in the field, particularly in the previous decade. The Global NLP Market was worth USD 10.72 billion in 2020 and is predicted to be worth USD 48.46 billion by 2026, growing at a CAGR of 26.84 percent over the forecast period (2021-2026) [2]. This is due to the widespread success of NLP in our daily lives. NLP is used in Information Retrieval (IR) systems, as well as Machine Translation (MT), Question-Answering, Sentiment Analysis (SA) and other applications. NLP is able to do achieve this by being able to distinguish between the seven interrelated layers (Section 2.1.) that people use to draw meaning from written or spoken language [3], [4]. Sentiment Analysis (SA), the main subject of this business project, is a prime example of this. Because SA uses multiple interconnected layers in tandem to successfully classify sentiments.

SA systematically identifies, extracts, quantifies, and studies subjective information using NLP, text analysis, and computational methods. Subjective information are mainly opinions of people's attitudes, judgements, or feelings towards entities, events, and their attributes. For example, SA is used to analyze thousands of reviews about a product to determine whether buyers are satisfied with the cost, quality and different aspects of the product. Therefore, it's basic goal is to find opinions, determine the sentiments expressed in them, and then classify people's positive, negative and neutral feelings. Furthermore, the rise of the internet and social media has resulted in an abundance of opinionated texts. This has led to SA being one of the most active fields in computer science, with numerous articles being published on the topic. SA solutions are being developed by hundreds of startups, and major statistical packages such as SAS and SPSS contain dedicated SA modules [5]–[9]. Businesses are adopting SA to gain a competitive advantage and market insights, entirely due to SA's many practical and future uses.

In recent years, the most notable uses of SA has been in financial markets. The automated analysis of sentiment represented in financial news in relation to various financial indicators such as stocks, commodities, currencies and so on is an important part of modern investors' decision making and risk assessment. Traditionally, analysts, investors, and institutional traders relied extensively on technical indicators such as moving averages (MAs) and relative strength indices (RSIs) to forecast stock trends [10]. However, stock prices are influenced by qualitative information from a range of sources, including social media, business disclosures, third-party news items, and analyst reports, in addition to financial figures. These qualitative financial information about a company influence market participants' expectations as well. The increase in GameStop stock price in January 2021, which was largely executed by Redditors, is an excellent example of this [11].

Therefore, a typical method to automatically analyze sentiment expressed in news is to utilize machine-learning-based NLP methods that assign either a category label, such as "positive," "negative," or "neutral," or a numerical score indicating the sentiment's strength and polarity. The sole purpose of this is to interpret if the financial news is indicating bullish or bearish thoughts about certain financial indicators. In recent years, there has been a lot of research in this field. A study conducted by Lee et al. [12] that evaluated the importance of SA for stock prediction is an excellent example of this. They devised a technique that forecasts changes in company stock prices in reaction to financial events reported in 8-K documents. According to the findings of their study, utilizing this technique improves their prediction accuracy by more than 10 percent, and the effect is most noticeable in the near term but lasts for up to five days. Similar studies [13]–[19] have been published highlighting the importance of SA in the financial sector.

Support Vector Machines (SVM), Naïve Bayes, or Maximum Entropy machine learning methods are employed for SA in all or most of the studies described above. Among these machine learning methods, SVM has the best performance in terms of SA [20]–[27]. However, text representation is critical for machine learning approaches because the performance of these models is dependent on it [28]. Machine learning approaches necessitate the transformation of texts into numerical forms that are clearly defined and have consistent lengths, vectors in particular. Furthermore, the majority of the experiments listed above use traditional methods,

such as the bag-of-words (BOW) model, to represent text for use in machine learning methods for SA. The difficulty with traditional models is that they ignore any information pertaining to the sequence or structure of words in the document. Therefore, doing a poor job in making sense of the text data.

As a result, based on recent improvements in the field, it is possible to argue that pre-trained deep neural network language models such as Universal Sentence Encoder (USE) [29] and Bidirectional Encoder Representations from Transformers (BERT) [30] can overcome this constraint. And, when used with machine learning methods, they can provide superior results compared to machine learning methods using traditional methods such as BOW. Little to no studies have been conducted by researchers to compare these models in terms of SA in the financial domain. Resulting in a clear knowledge gap in this field. Therefore, the aim of this project is to develop and compare financial SA models using traditional text representation models with new models such as USE and BERT.

Chapter 2

2. Literature Review

2.1. Natural Language Processing (NLP)

Before diving into Sentiment Analysis, it's important to first understand Natural Language Processing. NLP is defined as a set of computing approaches for analyzing and modelling naturally occurring texts at one or more levels of linguistic analysis in order to achieve human like language processing for a wide range of applications [4]. In simple terms, NLP is the field of research and application that investigates how computers may be used to interpret and manipulate natural language text or speech in order to perform meaningful tasks [3]. NLP has become increasingly common and effective in our daily lives over the last decade: automatic machine translation is prevalent on the web and in social media; text classification keeps our email inboxes from collapsing under a deluge of spam; search engines have advanced beyond string matching and network analysis to a high level of linguistic sophistication; and dialogue systems are an increasingly common and effective way to get and share information. These various applications are built on a common set of ideas, which include algorithms, languages, logic, statistics, and other fields [31].

While the whole pedigree of NLP is dependent on a number of other disciplines, it is reasonable to consider NLP an AI discipline because it strives for human-like performance. Except for AI researchers, NLP is not typically seen as a goal in and of itself. Others see NLP as a tool for completing a certain task. As a result, there are Information Retrieval (IR) systems that use NLP, as well as Machine Translation (MT), Question-Answering, and so on. There are numerous approaches or procedures from which to pick when performing a certain type of language analysis using NLP. The only criterion is that they are in a language that humans use to communicate. The text being analyzed should not be generated particularly for the purpose of the study, but rather obtained from real usage. Furthermore, when people produce or interpret language, multiple levels of language processing are known to be at work, and an NLP-based

system is one that employs any subset of these levels of analysis. This causes a lot of uncertainty among non-specialists about what NLP is and whether it is "weak" or "strong" NLP [4].

The key problem of natural language understanding is at the heart of any NLP application. There are three primary issues to consider when developing complex algorithms that interpret natural language. The first is concerned with thought processes, the second with the representation and interpretation of linguistic input, and the third with world knowledge. Thus, an NLP system may start at the word level to determine the morphological structure and nature of the word, then move on to the sentence level to determine the word order, syntax, and meaning of the complete phrase, and finally to the context and broader environment or domain. In a given context or domain, a given word or sentence may have a certain meaning, and it may be connected to many other words and/or sentences [3]. To understand natural languages, it is necessary to be able to differentiate between the seven interconnected levels that humans utilize to derive meaning from written or spoken languages [3], [4]:

- Phonetic or phonological level that is concerned with pronunciation
- Morphological level that is concerned with the tiniest components of words that contain meaning, as well as suffixes and prefixes
- Lexical level that is concerned with the lexical meaning of words and the analysis of components of speech
- Syntactic level that is concerned with sentence grammar and structure
- Semantic level that is concerned with the meaning of words and sentences
- Discourse level that is concerned with the structure of various kinds of text through the use of document structures
- Pragmatic level that is concerned with knowledge derived from the outside world, i.e. knowledge derived from sources other than the document's content

In NLP applications, all or some of these levels of analysis may be applied. An example of NLP application that uses these levels of analysis is Sentiment Analysis. Given that basic understanding of NLP has been formed, the following section will discuss the project's chosen topic, Sentiment Analysis.

2.2. Sentiment Analysis

Textual information in the world can be broadly classified into two types: facts and opinions. Facts are objective expressions of entities, events, and their qualities. Opinions are usually subjective expressions that describe people's attitudes, assessments, or feelings towards entities, events, and their attributes [5], [9]. The concept of opinion is extremely broad and this project solely concentrates on opinion expressions that represent people's positive, negative or neutral feelings. Opinions are so important that we want to know what others think whenever we need to make a decision. This applies not only to individuals but also to businesses. For example, when an individual needs to make a decision, he or she often seeks advice from friends and family, whereas when a business wants to learn about the general public's opinions about its products and services, it conducts opinion polls, surveys, and focus groups [7], [9].

Until recently, little research had been conducted on the processing of opinions and one of the reasons for that is due to the fact there was little opinionated texts available before the internet. People can now post product reviews on merchant websites and share their opinions on nearly anything in Internet forums, discussion groups, and blogs. This online word-of-mouth behavior provides new and quantitative information sources with several practical uses. It may no longer be required for a corporation to conduct surveys, establish focus groups, or hire external consultants to gather consumer feedback. However, finding opinion sources and monitoring them on the internet can be a formidable task because there are a great number of diverse sources. Each source may also contain a large amount of opiniated texts. It is difficult for a human reader to locate relevant sources, extract connected phrases containing opinions, read them, summarize them, and organize them into useful forms [7], [9]. As a result, automated opinion discovery and summarization are needed, and SA emerges from this need.

SA falls under the linguistic application of NLP and is one of the most difficult NLP problems. SA uses text analysis and computational techniques in combination with NLP to automate the classification of sentiment from various sources such as emails, support tickets, chats, social media, blogs, surveys, articles, documents, etc. [32]. SA purely focuses on the polarity (positive, negative, neutral) but also on feelings and emotions (angry, happy, sad, etc.),

urgency (urgent, not urgent) and even intentions (interested, not interested) depending on what needs to be interpreted. For example, using SA to automatically analyze thousands of reviews about a product could help discover if customers are happy about the pricing plans and customer service. It's target is to find opinions, identify the sentiments they express and then classify their polarity. SA is currently one of the most active study fields in computer science, with over 7000 articles published on the subject. SA solutions are being developed by hundreds of startups, and major statistical packages such as SAS and SPSS already contain dedicated SA modules [5]–[8].

As previously discussed, the availability of opinionated texts from numerous sources on the internet has substantially risen, resulting in a surge of interest in SA by both academics and businesses. Businesses are embracing SA in order to acquire a competitive advantage and market information. This is entirely owing to SA's many practical and future uses. Business analytics, politics, recommender systems, expert finding, summarization, government intelligence, market and FOREX rate prediction, box office prediction, marketing intelligence, and other critical applications are among them [20], [33]. The most prevalent use is in the field of consumer product and service reviews, with many websites now giving automated summaries of reviews of certain goods and attributes [5]. "Google Product Search" is an example of such a website. However, one of the most significant applications of SA in recent years has been in the financial markets, which is the primary focus of this project. The section that follows will go into depth on the use of SA in the Financial Sector and past work done in this field.

2.2.1. Sentiment Analysis in the Financial Sector

SA has played a key role in the financial industry, where financial news, texts, tweets, and other communications regarding public firms are evaluated to investigate the relationship between their sentiment and financial indicators such as stock returns and volatility. Previously, analysts, investors, and institutional traders relied extensively on technical indicators such as moving averages (MAs) and relative strength indices (RSIs) to forecast stock trends [10]. However, stock prices are influenced not just by financial numbers but also by qualitative information from a variety of sources, including social media, business disclosures, third-party news items, and analyst reports. These qualitative financial information regarding a company

impacts market participants' expectations too. Positive news about a company, for example, may encourage investors to acquire shares, pushing up the company's share price, whilst bad news has the reverse effect [10]. As a result, SA enables different companies to monitor numerous sources in real time, enabling them to respond appropriately and provide better results. However, conducting SA is not a simple process. It is estimated that 90 percent of the world's data is unstructured, or disorganized. As a result, in recent years, researchers have published various papers related to SA in the financial sector, which are discussed in detail below.

Lee et al. [12] studied the role of text analytics in stock price prediction by creating a corpus of 8-K reports describing financial events. The findings demonstrated that adding textual information is certainly significant, particularly in the short term. When text was included in the studies, forecasting the following day's price movement increased by 10 percent. Hence, the study clearly demonstrates the importance of SA in stock price prediction. Similarly, Yu et al. [13] investigated the impact of social and traditional media on firm equity value. The study utilized a novel and large-scale dataset that includes daily media content from multiple traditional media and social media sites for 824 publicly traded firms across six industries, with stock return and risk serving as indicators of companies' short-term performance. According to the findings, social media has a stronger association with company stock performance than traditional media, and social and traditional media have a high interaction effect on firm performance. It was also shown that the impact of various types of social media differs greatly.

Li et al. [14] used SA to investigate the impact of news on stock price returns. The news articles were represented in the study as vector space models that were multiplied by a sentiment word matrix. The dataset included news articles gathered from the Hong Kong financial news archive between January 2003 and March 2008, as well as daily stock quotes gathered from Yahoo! Finance over the same time period. When evaluated, SA on this dataset produced an accuracy of 69.68 percent, indicating that the relationship between news impact and intra-day stock price return can be studied in the future. Likewise, Chan et al. [15] looked into a financial text stream consisting of twelve million words that aligned to a stock market index. The study's findings revealed that there was substantial evidence of a long persistence in the mood time series created. When text streams were used to express sentiments, it was

discovered that these sentiments were useful for analyzing the trends in the stock market index, despite the fact that sentiment and stock market indexes are normally regarded to be completely uncorrelated. Another study, undertaken by Chowdhury et al. [16], displayed the sentiments of 15 companies over a 4-week period, with SA scores indicating an average accuracy score of roughly 70.1 percent. Finally, a comparison of the positive sentiment curve and stock price trends reveal a co-relationship of 67 percent, indicating a semi-strong to strong association.

Bollen et al. [17] gathered approximately 9 million public tweets from 2.7 million people between February 28th and December 19th, 2008 for SA to forecast stock market. The research matched the resultant public mood from SA to the Dow Jones Industrial Average (DIJA) in order to forecast changes in DIJA over time. The findings revealed that tweets are corelated or even predictive of DIJA values. Another study, undertaken by Li et al. [18], tries to examine whether the price of a sample of 30 businesses listed on the NASDAQ and the New York Stock Exchange can be predicted by the given 15 million tweets. Patterns between public sentiment and real stock price movement were established using NLP and data mining approaches. The stock price of select companies could be forecasted with an average accuracy of 76.12 percent. Similarly, Pagolu et al. [19] looked into tweets to see if there was a correlation between stock market fluctuations and sentiment in tweets. The findings revealed that there is a high correlation between the rise and fall of the stock market and popular sentiments expressed in tweets.

According to the studies discussed above, SA can or does play a key role in the financial sector. If retail investors or businesses can take advantage of SA, it can lead to significant stock market returns. However, these studies make a variety of assumptions, thus this should not be considered a complete trading strategy. Given that the benefits of SA in the financial industry have been shown through the studies stated above, it is crucial to explore the various methodologies used to conduct SA in these studies. Not all of the studies mentioned above utilize the same methodology, and different methodologies result in varying SA accuracy scores. As a result, the following section will go over the various levels and methods for SA.

2.3. Classification Levels and Methods for Sentiment Analysis

SA is thought to be a classification process, with three basic classification levels: document-level, sentence-level, and aspect-level SA. Document-level SA attempts to classify an opinion document as having a positive or negative sentiment. It regards the entire document as a basic information unit. Sentence-level SA attempts to classify the sentiment expressed in each sentence. The initial step is to determine whether the sentence is subjective or objective. Sentence-level SA will assess whether the sentence expresses positive or negative sentiment if the sentence is subjective. Furthermore, because sentences are merely small documents, there is no fundamental distinction between document and sentence level classifications. However, classifying text at the document or sentence level does not provide the necessary detail needed on all elements of the entity, which is required in many applications; one must travel to the aspect level to gain these details [5], [9], [32].

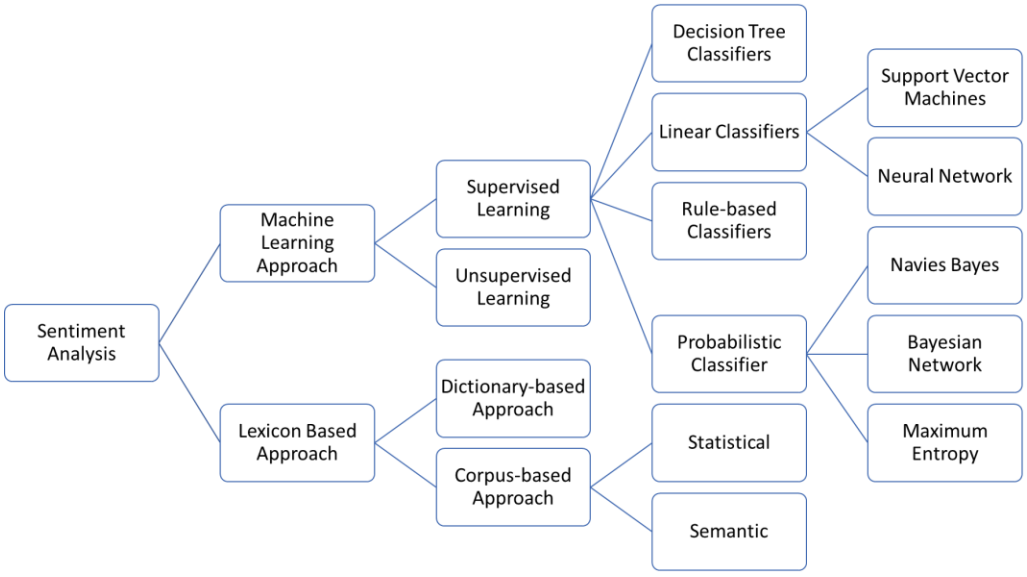

Figure 1: Sentiment Analysis Methods

However, the focus of this project is limited to document-level SA, and the topics that follow will be built upon it. It has been demonstrated that specific domain-oriented SA has attained remarkable accuracy in document-level SA. The feature vector employed in these tasks commonly contains traditional bag-of-words model, which should be domain-specific and constrained [34]. At the document level, the SA does not go into depth, and the analysis is done

in an abstract and generic manner, which speeds up the analysis. Initially, the majority of studies were undertaken at the document level and focused on datasets such as news and product evaluations. Because of the increased popularity of social media, several types of datasets were developed, resulting in a further rise in studies at this level [35]. Ravi et al. [20] conducted a SA survey and also found that the majority of SA studies were done at document-level. Following that, various methodologies for SA have been developed over time, with machine learning and lexicon-based approaches being the two most common [36]. Figure 1 displays a flow chart of the methodologies that can be utilized for SA, and because machine learning approaches have already been determined to be used in this project, the lexicon-based approach will not be reviewed further.

2.3.1. Machine Learning Approach

Machine Learning approach predicts the polarity of sentiments based on trained as well as test dataset while taking advantage of various machine learning methods and using language models. Given this, machine learning approach can be classified as either supervised or unsupervised based on the training data. Supervised methods make use of a large number of labelled training data, whereas unsupervised methods are utilized when finding these labelled training data is difficult [36]. Because the datasets used in this project are labelled with specified input and output, they are used with supervised machine learning methods, which will be reviewed further. In a supervised machine learning based classification, a training dataset is used by an automatic classifier to learn the various features of the dataset, and a test dataset is used to assess the automatic classifier's performance [37]. There are several supervised machine learning methods, as illustrated in Figure 1, including Decision Tree, Support Vector Machines (SVM), Neural Network, Naïve Bayes, and others, that can be used as an automatic classifier.

Ravi et al. [20] reported in their survey of SA that it was glaringly evident that SVM is the most often applied classifier for various SA applications and also yielding the best accuracy in most studies. Likewise, a document-level SA survey conducted by Behdenna et al. [21] reported that in machine learning approach the main classifiers used were SVM and Naive Bayes with majority of the studies also using “Bag of Words” for text representation. Furthermore, a

comparison study of Naïve Bayes and SVM on SA was conducted by Rahat et al. [22] using airlines reviews. The results revealed that SVM performed better than Naïve Bayes with an accuracy score of 82.48 percent and 76.56 percent, respectively. Similar results between SVM and Naïve Bayes performance were found in a study by Alves et al. [23] where the main focus was on tweets written in Portuguese during the 2013 FIFA confederations Cup. In another study, Tan et al. [24] conducted an empirical study of SA for Chinese documents using four feature selection methods and five machine learning methods: centroid classifier, K-nearest neighbor, winnow classifier, Naïve Bayes and SVM. The dataset used was Chinese sentiment corpus with a size of 1021 documents that contains three domains: education, movie and house. The results suggested that among the five machine learning methods, SVM performed the best with micro-averaged f-score of 0.8685.

Al Amrani et al. [25] examined the performance of Random Forest and SVM methods for SA in another study. The study utilized a dataset of 1000 positive and negative reviews that were evenly split. According to the results, SVM performed slightly better than Random Forest, with a accuracy of 82.4 percent and 81 percent, respectively. However, the study also used a hybrid method for SA, combining SVM and Random Forest, which achieved the greatest accuracy of 83.4 percent when compared to SVM and Random Forest methods. Lu et al. [26] used sentiment dictionary and SVM to perform SA on film review texts. At first, the study created four different sentiment dictionary using the film review dataset and secondly, the SVM model training set is constructed by computing the combination of sentiment weight and user scoring. Finally, the test dataset results revealed that SVM outperformed basic sentiment dictionary methods in terms of accuracy.

In addition, SVM were reported to be the best available machine learning method for text classification tasks and financial prediction by Hagenau et al. [27]. They also compared the performance of artificial neural networks, NB, and SVMs and discovered that SVMs performed the best. The studies discussed above are also in accordance to most studies discussed in [20], [35], [38] where SVM performed better than other machine learning methods available in most cases. An example of a machine learning method outperforming SVM is Naïve Bayes achieving comparable or better performance than SVM in a study reported by Zhang et al. [39].

The superior performance of SVM is due to its suitability for solving high dimensional problems compared to other techniques. Because of the sparse nature of text, few features are unimportant, but they tend to be connected with one another and often structured into linearly separable categories, making text data perfectly suited for SVM classification [38]. The section that follows builds on the theory of SVM and reviews SVM methodology.

2.3.1.1. Support Vector Machines (SVM)

SVMs have shown to be the most effective and widely used supervised machine learning approach for traditional text classification, with a strong theoretical foundation that outperforms machine learning methods in most cases for SA. It is a relatively new type of machine learning method introduced by Vapnik [40]. SVM seeks a decision surface to split the training dataset into two classes based on the structural risk minimization principle from computational learning theory and makes decisions based on the support vectors that are picked as the only effective elements in the training dataset [24]. SVMs seek a hyperplane represented by vector $\vec{w}$ that has the maximum margin of separation between the positive and negative training vectors of documents, as seen in Figure 2.

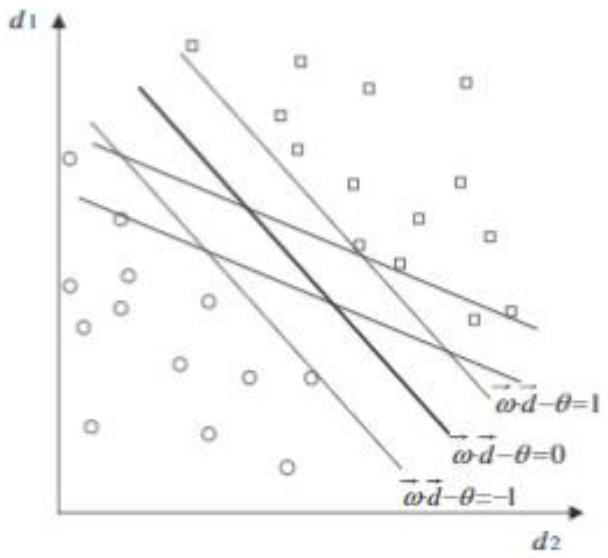

Figure 2: Maximum margin classifier of SVM

This hyperplane's findings can then be turned into a constrained optimization problem. If document d_i belongs to the class +/-, let $y_i$equal +1/-1. The solution is as follows:

$$\vec{w} = \sum_{i=1}^{n} \propto^{*} y_i \overrightarrow{d_\iota}, \propto_i \geq 0$$

Where $\propto_i$ are the results of a dual optimization problem. The preceding equation demonstrates that the resulting weight vector of the hyperplane is a linear combination of $\overrightarrow{d_i}$. Only those cases that contribute and have a coefficient $\overrightarrow{d_\iota}$ larger than zero are counted. These vectors are known as support vectors because they are only document vectors that contribute to $\vec{w}$ [39], [41]. The goal of SVM optimization is to minimize [24]:

$$\overrightarrow{\propto^{*}} = \arg\min \left\{ -\sum_{i=1}^{n} \propto_i + \sum_{i=1}^{n} \sum_{j=1}^{n} \propto_i \propto_j y_i y_j \langle \overrightarrow{x_\iota}, \overrightarrow{x_j} \rangle \right\}$$

$$subject\ to: \sum_{i=1}^{n} \propto_i y_i = 0; 0 \leq \propto_i \leq C$$

Tan et al.'s [24] investigation of SA using Chinese documents showed an average macro f-score of 0.8664 using SVM and discovered that sentiment classifiers are highly dependent on domains or subjects. Al Amrani et al.'s [25] study to classify positive and negative Amazon reviews using SVM yielded an accuracy of 82.4 percent. In another study, Aftab et al. [42] applied SVM to tweet SA and measured the outcomes in terms of precision, recall, and f-score. The study made use of two datasets comprising tweets regarding self-driving cars and Apple devices. The average precision, recall, and F-score for the dataset of self-driving cars were 55.8 percent, 59.9 percent, and 57.2 percent, respectively. The average precision, recall, and f-score for the Apple products dataset were 70.2 percent, 71.2 percent, and 69.9 percent, respectively. The results revealed that SVM performance is also dependent on the training dataset. To summarize, the majority of the research covered in the previous sections used SVM for SA and achieved high performance scores. It should be highlighted, however, that SVM performance is domain specific and strongly dependent on the training dataset.

2.4. Text Representation

Text modelling is extremely difficult and complex, and machine learning methods prefer inputs and outputs that are well defined and have constant lengths. Consequently, machine learning methods are unable to work directly with raw text, and the text must be transformed into a numerical format before being used. In particular, a vector of numbers, which is frequently referred to as feature extraction or feature encoding [43]. The bag-of-words (BOW) model is a popular and straightforward approach of extracting features from text data. However, because NLP is a diverse field with many distinct tasks, most task-specific datasets only contain a few thousand or a few hundred thousand human-labeled training examples. Modern deep learning-based NLP models, on the other hand, benefit from far larger volumes of data, and improve when trained on millions, if not billions, of annotated training examples. Researchers have developed a range of methods for training general purpose text representation models using the massive volume of unannotated text on the web to help close this data gap (known as pre-training). The pre-trained model can then be fine-tuned on small-data NLP tasks like SA, resulting in significant improvement in performance over training from scratch on these datasets [29], [44].

As a result, the sections that follow discusses the most commonly used text representation method, BOW, as well as two novel text representation models, Universal Sentence Encoder (USE) and Bidirectional Encoder Representations from Transformers (BERT).

2.4.1. Bag of Words (BOW)

The BOW model is easy to understand and use, and it has had considerable success in a variety of tasks, including language modelling and document classification, among others. It converts arbitrary text into fixed-length vectors by counting the number of times each word appears in the text. This is commonly referred to as vectorization. It involves two main components: a vocabulary of well-known words and a measure of the presence of well-known words in the vocabulary. Any information in the document that relates to the sequence or structure of words is discarded without further consideration. The model is just concerned with whether or not known words appear in the document; it is not concerned with where in the document they appear. The underlying assumption is that documents with similar content are

considered to be similar. BOW can be as basic or as complex as the user desires it to be. The difficulty arises from the decision on how to build the vocabulary of known words (or tokens) as well as how to score the presence of known words [43]. Most of the earlier studies and surveys discussed previously used machine learning methods based on the BOW model. The BOW model is not further reviewed because it is simply used as a baseline in this project, and how to use a BOW model for machine learning purposes using Python will be described in the methodology.

2.4.2. Universal Sentence Encoder (USE)

It has already been determined that, while approaches like BOW convert text into vectors, they do not account for where the word appears in the document. Capturing the context of the entire sentence in that vector while embedding a sentence, on the other hand, tends to result in a large gain in performance. This is where the Universal Sentence Encoder (USE) steps in. Textual data is encoded into high-dimensional vectors by USE, which may then be utilized for text classification, semantic similarity, clustering, and other natural language applications. Cer et al. [29] first proposed it in 2018 and made the several versions of USE models trained with different goals including size/performance, multilingual, and fine-grained question answer retrieval publicly available in [45]. USE includes two types: one trained with a Transformer encoder and one trained with a Deep Averaging Network (DAN).

The Transformer encoder model creates sentence embeddings by using the transformer architecture's coding sub-graph. The subgraph generates a context-aware representation of the words in the input sentence. It also takes into account the identity and sequence of all other words. At each word position, the element-wise sum of that representation is computed and turned into a fixed-length sentence encoding vector. In the DAN variation, input embeddings for words and bi-grams are averaged and fed into a feedforward DNN (Deep Neural Network), yielding sentence embeddings. It has been discovered that such DANs perform well on text classification tasks. The accuracy and processing resource requirements of the two are trade-offs. While the one with the Transformer encoder is more accurate, it is also more computationally intensive. The one with DNA encoding is less expensive computationally and has slightly worse accuracy. For more information on USE, see [37]. There are no research that have been

conducted in SA that have used USE; however, there are two excellent examples where USE has been used in tweet classification [46] and the detection of fake news [47]. These two examples can be used as a starting point for understanding how SA can be achieved through the usage of USE.

2.4.3. Bidirectional Encoder Representations from Transformers (BERT)

The material in this section is primarily drawn from the article "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" by Devlin et al. [30] and [48], [49]. Transformer, an attention mechanism that learns contextual relationships between words in a text, is used by BERT. Transformer, in its most basic version, consists of two distinct mechanisms: an encoder that reads the text input and a decoder that generates a prediction for the task. Because the purpose of BERT is to construct a language model, only the encoder mechanism is required. Therefore, BERT is a transformer encoder stack where BERT is pre-trained to understand language and fine-tuned to learn a specific task. The Transformer encoder reads the complete sequence of words at once, as contrast to directional models, which reads the text input sequentially (left-to-right or right-to-left). As a result, it is regarded as bidirectional, however it would be more accurate to describe it as non-directional. This feature enables the model to learn the context of a word based on all of its surroundings (left and right of the word). A study conducted by Vaswani et al. [50] describes the detailed operation of Transformer.

Given that many directional models predict the next word in a sequence, it inherently limits context learning. To overcome this challenge, BERT uses two training strategies: Masked LM (MLM) and Next Sentence Prediction (NSP). When training a BERT model, MLM and NSP are trained together with the goal of minimizing the combined loss function of the two strategies.

- MLM masks some of the tokens in the input at random, and the goal is to guess the original vocabulary id of the masked word based solely on its context. For example, the [MASK] brown fox leapfrogged the [MASK] dog. The MLM objective permits the representation to integrate the left and right contexts, allowing us to pre-train a deep bidirectional Transformer. For prediction, the training data generator selects 15 percent of the token positions at random. If the i-th token is picked, we replace it with (1) the

[MASK] token 80 percent of the time, (2) a random token 10 percent of the time, and (3) the unchanged i-th token 10 percent of the time.

- In NSP, two sentences are chosen and a binary classification is performed to determine whether the two sentences are sequential or not.

Following that, in terms of applying BERT for classification tasks like SA. They are accomplished in the same way that Next Sentence Classification is accomplished by putting a classification layer on top of the transformer out for the [CLS] token. Here's [51] an example of text classification using BERT and the different BERT models with their appropriate weights can be found in [52]. The appropriate preprocessing model for each BERT model can be used to feed plain text into these models.

In contrast to USE, there have been studies in which BERT has been used for SA. BERT for stock market SA was employed on 582 documents from a variety of financial news sources by De Sousa et al. [53]. On this dataset, the BERT model was fine-tuned, and it achieved a f-score of 0.725 as a result. When Chiorrini et al. [54] investigated the SA of tweets using BERT, they discovered that the model had an accuracy of 92 percent. However, the fact that the BERT model itself is fine-tuned to operate as a classifier in the vast majority of studies employing BERT should be recognized. As a result, this approach is not used in this project because it would be unfair to compare BOW and USE text representations with BERT in this manner. Because of this, only pretrained text representations will be taken from the BERT model, which will then be employed in machine learning methods for SA.

2.5. Summary

Over the last decade, NLP has grown dramatically, becoming more prevalent and successful in our daily lives. NLP is being used for a variety of applications, one of which is SA. SA automates sentiment classification from diverse sources by combining text analysis and computational approaches with NLP. SA simply aims to classify polarity, feelings, emotions, urgency, and even intents. It has been one of the most investigated issues in the field of NLP due to its benefits and usefulness. Its applications include politics, product reviews, government intelligence, market and FOREX rate prediction, and so on. SA has been a game changer in the

financial sector, allowing researchers to forecast market movements based on conventional financial news and social media. A common strategy for SA is to anticipate the sentiment of a news using machine learning approaches. Machine learning methods such as Naïve Bayes, SVM, Maximum Entropy, and others have been employed for SA. SVM, on the other hand, has been the most common machine learning approach for SA and has also demonstrated the best performance when compared to other machine learning methods. Furthermore, the representation of text in a right manner is critical for successful SA as text cannot be supplied directly into machine learning methods; instead, it must be transformed into vectors using methods such as BOW. Traditional text representation methods, such as BOW, do not, however, capture the context of the text. As a result, performance suffers. However, new methods like USE and BERT can capture the context of the text, resulting in much greater performance than traditional models. Following on, the methodology for analyzing sentiment in financial news using these text representations is discussed in the following chapter.

Chapter 3

3. Methodology

3.1. Applied Technology

The main application used for this project was Jupyter notebook [55]. The Jupyter Notebook is a web-based application that allows you to create and share documents with code, visualizations, and text. It can be used for data science, statistical modelling, machine learning, and a variety of other tasks. Over 40 programming languages are supported, including Python, R, Julia, and Scala. Given that, Python is the programming language utilized in this project. Python is a high-level general-purpose programming language that is interpreted. The use of considerable indentation in its design emphasizes code readability. Its language elements and object-oriented approach are intended to assist programmers in writing clear, logical code for small and large-scale projects [56]. Furthermore, the project takes advantage of open-source libraries specifically intended for data analysis and machine learning. NumPy, Pandas, and Seaborn are data analysis and visualization libraries, whereas Scikit-Learn and TensorFlow are machine learning libraries. Refer to [57]–[61] for further information on the libraries.

3.2. Dataset

The dataset named “Financial Phrase Bank” used in this project has already been collated by [62]. The initial dataset was created using English news from all OMX Helsinki listed firms. An automated web scraper was used to download the news from the LexisNexis database. A random sample of 10,000 items was chosen from this news database to ensure enough coverage among small and large enterprises, companies in various industries, and news sources. All sentences that did not contain any of the lexical entities, were rejected. This decreased the overall sample size to 53,400 sentences, each of which contains at least one recognized lexical entity. The phrases were then categorized based on the sorts of entity sequences that were observed. Finally, a sample of 5,000 sentences was chosen at random to reflect the entire news database.

Then, using only the information directly available in the given sentence, a phrase level annotation exercise was performed to characterize each sample sentence as positive, negative,

or neutral. Because the study was limited to financial and economic topics, the annotators group comprised of 16 people who were well-versed in financial markets. The annotators were asked to analyze the sentences solely from the perspective of an investor, i.e. whether the news could have a positive or negative impact on the stock price, and statements with sentiments that are irrelevant from an economic or financial standpoint are labelled as neutral.

The final release of the financial phrase bank dataset includes 4,846 sentences divided into two columns: "sentiment" and "news." The "news" column contains the company's financial news, while the "sentiment" column contains the news's accompanying label as positive, negative, or neutral. There is no train, validation, or test split in the dataset. The dataset, on the other hand, is provided in four different configurations based on the percentage of agreement among annotators. This project uses the dataset where the configuration was the number of phrases with higher than 50 percent annotator agreement, which included 4,846 sentences. Refer to [62] for further information on the annotation task and various configurations of the dataset.

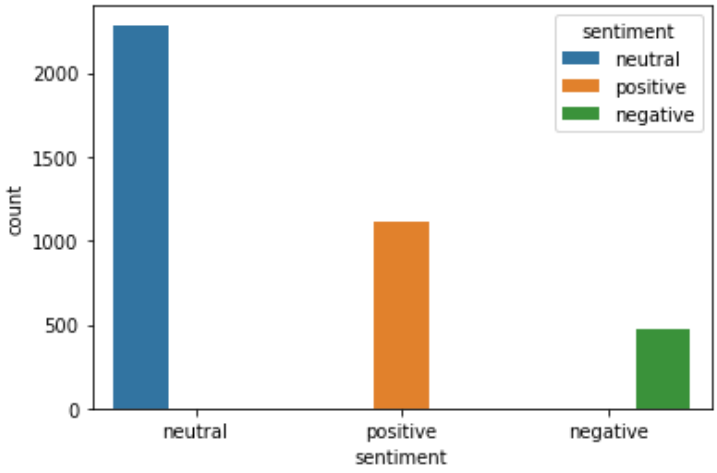

Figure 3: Distribution of sentiment in the train dataset

3.3. Preparation & Analysis of the Dataset

The dataset Financial Phrase Bank with 4,846 rows and the columns "sentiment" and "news" mentioned above was loaded into the Jupyter notebook. Then, the dataset was split into train (fntrain) and test (fntest) datasets, with the test dataset set aside until it was needed to avoid data snooping. Data snooping refers to statistical judgements made by the researcher after

reviewing the data. There were 3,876 and 970 rows in the new train and test datasets, respectively. After inspection, there were no null values in either the train or test datasets. However, there were only 3,872 unique values in the train dataset, indicating that there are four duplicates in the dataset. These duplicates were deleted, leaving the train dataset with only 3,782 rows. Figure 3 depicts the distribution of positive, negative, and neutral sentiment in the train dataset, with values of 1,116, 474, and 2,286, respectively. This step was not done for the test dataset as this would lead to data snooping. Finally, because the datasets have been cleaned and prepped, they are now ready to be encoded, as detailed in the following section.

3.4. Text Representation using BOW, USE & BERT models

This section discusses the text representation of the sentences in both the train and test datasets using Bag of Words (BOW), Universal Sentence Encoder (USE), and Bidirectional Encoder Representation from Transformers (BERT) models. It is important to note that each sentence in the dataset is considered a document, and only the news sentences are encoded, leaving the sentiments alone.

3.4.1. Bag of Words (BOW)

A BOW model, as previously discussed, is a method of extracting features from text for use in machine learning models. It includes two components: a vocabulary of known words and a measure of the presence of known words. As a result, the **"CountVectorizer"** from the scikit-learn library was initially imported to construct BOW representations of sentences in a NumPy array. The following were the basic linguistic preprocessing parameters that were applied to the **"CountVectorizer"**: strip accents as **“Unicode”** which converts accented characters to non-accented characters and stop words as **"English"** where a built-in stop word list for English is used. All other parameters of the **"CountVectorizer"** were left as default and for more information on the **"CountVectorizer"** parameters refer to [63]. The **"fit transform"** method was invoked after creating the **"CountVectorizer"** instance. As a result, this first fits on the data, i.e., extracts individual features from each document as specified by the arguments to the constructors method (e.g., extract bigrams), and then **"transforms"** the data, i.e., creates NumPy arrays with rows representing documents, columns representing features extracted from the

documents, and values in the cells representing counts of each feature in each document. As a result of the transformation, the train dataset now has 3,872 rows, one for each new sentence, and 8779 columns, one for each feature. Due to the large number of zeros in each row, the dataset is stored in a particular data structure known as a **"sparse matrix"**. The sparse matrix stores data more efficiently and may thus represent reasonably big text collections without causing memory issues.

The vocabulary of the text collection on which it was fitted was saved by the **"CountVectorizer"**. The vocabulary is simply a dictionary, with keys representing original words and values representing their indices in the created matrix. The vocabulary will be employed internally by the fitted vectorizer when the test data is transformed, resulting in a document-by-feature matrix with the same columns and order as the training data. The fitted vectorizer was now applied to the test dataset as well. Because the vectorizer has previously been fitted, the **"transform"** method was invoked rather than the **"fit"** or **"fit transform"** methods. The test dataset now includes 970 rows, one for each sentence, but the same amount of columns, 8,779.

The observed counts were then translated into TF-IDF weights to reflect the value of each word in this document. The **"TfidfTransformer"** was used for this. The transformer was first fitted on the training data, and then it was used to convert both the train and test datasets. TF-IDF was used to overcome a specific problem. That is, highly common words begin to dominate the document, yet they may not provide as much informational content to the model as uncommon but possibly domain-specific words. As a result, TF-IDF rescales the frequency of words based on how frequently they appear in all documents, penalizing scores for common words like "the" that are also frequent across all documents [43]. Finally, on the training dataset, a scaler is fitted and utilized to transform both the train and test datasets. The scaler employed was a **"MaxAbsScaler"**, which can handle sparse matrices. The datasets for training (bow_fntrain) and testing (bow_fntest) were now ready for usage in SA.

3.4.2. Universal Sentence Encoder (USE)

As discussed before, USE encodes text into high-dimensional vectors and are available in several versions with different goals in mind including size, performance multilingual, and fine-

grained question answer retrieval. The USE large model [64] was used for this project, which is an Transformer encoder model that creates sentence embeddings by using the transformer architecture's coding sub-graph. This model was chosen due to its high performance. This model has been trained and optimized for text that is larger than a word, such as sentences, phrases, or short paragraphs. As a result, the USE large model was loaded into the Jupyter Notebook from TensorFlow Hub, and datasets (train & test) were encoded. The output was a 512 dimensional vector for each sentence in the dataset. This high-dimensional vector output was then transformed into NumPy arrays. The final train (use_fntrain) and test (use_fntest) datasets, which were 3,872 rows and 512 columns and 970 rows and 512 columns, were now ready to be utilized for SA.

3.4.3. Bidirectional Encoder Representations From Transformers (BERT)

BERT uses a deep, pre-trained neural network with the transformer architecture to create dense vector representations for natural language; for more information on BERT, check preceding sections. BERT, like USE, provides a multitude of models; for this project, the BERT English Uncased model with 12 hidden layers (L), 768 hidden size (H), and 12 attention heads (A) trained on the Wikipedia and BooksCorpus datasets was used [65]. In BERT, model size is important since larger models perform better. Therefore, the above mentioned, second largest BERT model was picked keeping the computation time in mind while also not making significant sacrifice in performance.

However, before being fed into BERT, the dataset must be modified using the appropriate preprocessing model as specified in the BERT documentation. The preprocessing model for the BERT used in this study can be found in [66]. As a result, the preprocessing model was loaded into the Jupyter Notebook from TensorFlow Hub, and the datasets (train & test) were converted to numeric token ids and grouped into numerous tensors for BERT input. As input, the preprocessed dataset was now fed into the BERT model loaded from TensorFlow Hub. The BERT model returned a map that contained three key values: **"pooled_output"**, **"sequence_output"**, and **"encoder_outputs"**. The **"pooled_output"** represented each input sequence as a whole and was used as the input for the SA. The number of rows in the initial datasets and the BERT model's

hidden size (H) dictates its shape. However, because **"pooled_output"** was in the form of tensors, it was converted to NumPy arrays. Finally, the train dataset (bert_fntrain) comprised 3,872 rows and 768 columns, while the test dataset (bert_fntest) had 970 rows and 768 columns and was ready for SA.

3.5. Training and Testing of Machine Learning Models for Sentiment Analysis

According to literature review, SVM exhibits the highest performance in terms of SA. Following that multiple variations of SVM exists, however, due to the popularity and superior performance in text categorization, this project employs linear SVM classifier. Therefore, Linear SVM classifier was imported from the Scikit-library, together with GridSearchCV, for hyperparameter tuning. To obtain more accurate results, hyperparameter tuning is a key task in SVM. In Grid-Search, different models having different parameter values are trained and then evaluated using cross validation. For the SVM classifier, default parameters were used, except for the regularization constant C. Because the value of C for a best performing model cannot be evaluated in advance. Therefore, hyperparameter tuning using different C values were done utilizing Grid-Search. The different C values used were $C \in [0.1, 0.5, 1, 2, 4, 6, 8, 10]$. In addition, five-fold cross validation to prevent overfitting and a random state of seven to control randomness was used. Finally, macro averaging f1-score was used to evaluate the performance of the models. Also, given that this is a multiclass classification problem using linear SVM, Scikit-library by default uses the One-vs-The-Rest approach. Then, the models with different C values were fitted on the train datasets resulting in a total of 24 models, with each text representation having 8 models. The best performing model for each text representation was saved as a result of this. The saved models were then evaluated using the appropriate test datasets. Lastly, the model with BOW text representation was treated as the baseline.

3.6. Performance Measures

The performance of the method proposed were evaluated for all models using macro-averaged Precision, Recall and F-Score, and Confusion Matrix (Table 1). Macro-averaging approach was used due to the major imbalances in the classes as seen in Figure 3. Macro-averaging balances all classes. Given a set of confusion tables, macro-averaging generates a set

of values. Each value represents the precision or recall of an automatic classifier for each category. These measures are used to calculate the average performance of an automatic classifier in terms of precision, recall, and F-Score [67].

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$F - score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Table 1: Confusion Matrix

	Predicted Positives	**Predicted Negatives**
Actual Positives	True Positive	False Negative
Actual Negative	False Positive	True Negative

Chapter 4

4. Results and Discussion

As a result of hyperparameter tuning, Table 2 presents the macro-averaged validation and train f-scores of the 24 SVM classifier models trained on BOW, USE, and BERT text representations for a variety of C values. Each text representation consists of eight models, which are organized in decreasing order of the validation score for each text representation in Table 2, as this was used to evaluate their performance. The models performance using BOW is between 0.669 and 0.639. The performance of the models is influenced by the values of C, with lower values of C resulting in better performance. However, evidence of overfitting can be seen in all models because training scores are near perfect while validation scores are not. There are no significant differences in model performance for different values of C for models using USE. However, evidence of overfitting is seen yet again, but this time it is better than BOW models. There is a significant difference in performance varying between 0.706 and 0.633 for models utilizing BERT. Similar to BOW models, the BERT models perform better with smaller C values. Overfitting is evident; nevertheless, as compared to BOW and USE models, overfitting in BERT models is the least.

Table 2: Macro-averaged f-scores for the linear SVM classifier models using BOW, USE and BERT text representations

BOW			USE			BERT		
C	**Validation Score**	**Train Score**	**C**	**Validation Score**	**Train Score**	**C**	**Validation Score**	**Train Score**
0.5	0.669	0.993	2	0.716	0.840	0.5	0.706	0.823
1	0.664	0.996	1	0.715	0.825	2	0.693	0.819
2	0.658	0.998	10	0.712	0.869	0.1	0.693	0.766
4	0.650	0.999	0.5	0.712	0.807	4	0.692	0.838
6	0.646	0.999	8	0.710	0.866	1	0.691	0.831
0.1	0.645	0.968	4	0.710	0.855	8	0.684	0.814
8	0.644	0.999	6	0.710	0.862	6	0.671	0.816
10	0.639	0.999	0.1	0.691	0.747	10	0.633	0.763

The existence of high quantity of noise in the training data, as well as a large number of variables, could be the cause of overfitting. In the future, this could be overcome by lowering the amount of variables using feature selection, regularizing the training data, or gathering more training data. However, for the time being, the highest performing models for each text representation is utilized to evaluate the test datasets, leaving the issue of overfitting aside. For BOW, USE, and BERT text representations, the highest performing models have C values of 0.5, 2 and 0.5, respectively. Unless otherwise stated, the following discussion is limited to these three models and test datasets.

Table 3: Macro-averaged Precision, Recall and F-Score of the test datasets using SVM classifier model for different text representations

Text Representation	Precision	Recall	F-Score
BOW	0.675	0.637	0.652
USE	0.700	0.674	0.686
BERT	0.742	0.706	0.722

Table 3 shows the macro-averaged Precision, Recall, and F-score of the test datasets as evaluated by the appropriate SVM classifier model for each text representation. According to the model using BOW, the macro-averaged Precision, Recall, and F-Score are 0.675, 0.637, and 0.652, respectively. The macro-averaged Precision, Recall, and F-Score for the model utilizing USE are 0.700, 0.674, and 0.686, respectively. The macro-averaged Precision, Recall, and F-Score for the model employing BERT are 0.742, 0.706, and 0.722, respectively. The SVM classifiers' performance is comparable to the studies reported in the literature review. Some studies, however, have found superior performance, which is mostly owing to the diverse datasets used since the performance of SVM classifiers depend on the dataset used and its domain. Furthermore, comparing the f-scores of the train and test datasets reveal that there are signs of overfitting as expected.

Following on, SVM classifiers using USE and BERT outperform the baseline BOW model. And, of the three, the BERT text representation is the most effective. Given the overfitting, the models were further evaluated using a confusion matrix to select the best model for SA

of financial news. This is done from the perspective of an investor, with the idea that when the prediction is positive, negative, or neutral, the investor chooses to buy, sell, or hold a company's stocks, respectively. The confusion matrix for the BOW, USE, and BERT models is depicted in Figures 4(a), 4(b), and 4(c), respectively. Figure 4(a) shows that the BOW model predicts neutral sentiment with an accuracy of 83 percent, whereas positive and negative feelings are predicted

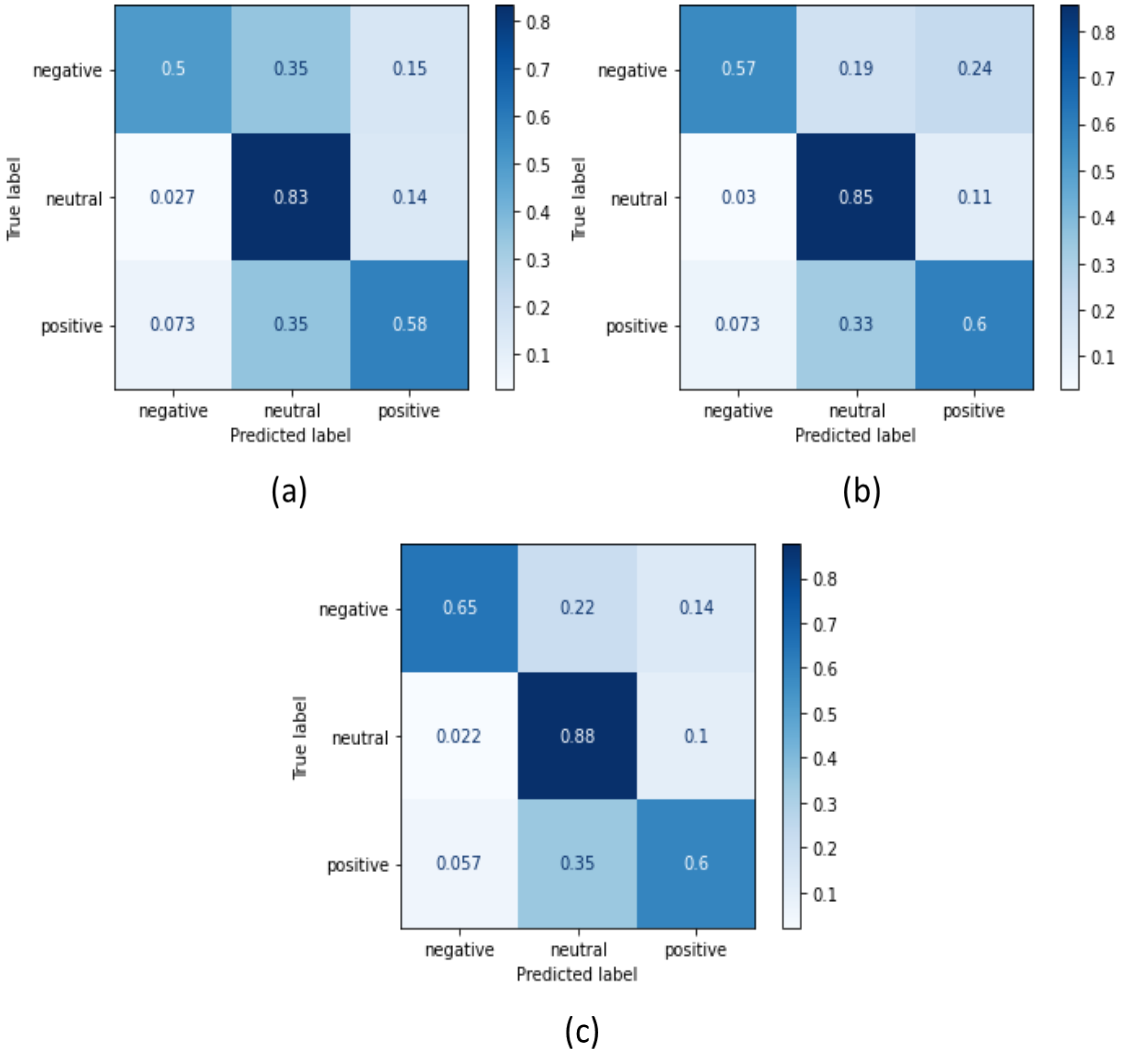

Figure 4: Confusion matrix for (a) BOW, (b) USE and (c) BERT models

with an accuracy of 58 percent and 50 percent, respectively. In figure 4(b), the USE model predicts neutral, positive, and negative sentiments with an accuracy of 85 percent, 60 percent, and 57 percent, respectively. In figure 4(c), the BERT model predicts neutral, positive, and negative sentiments with 88 percent, 60 percent, and 65 percent accuracy, respectively. The results reveal

that all three models predict neutral sentiment notably better than positive and negative sentiments, with negative sentiments having the lowest prediction accuracy of the three. The significant difference in neutral prediction accuracy with respect to negative and positive sentiments in all models is due to the imbalance in the dataset as seen in Figure 3. Which leads to overtraining of the neutral sentiments. The poor prediction accuracy for negative sentiments is not a favorable situation for an investor. Because any misclassification of negative sentiments could result in massive losses. For example, misclassifying a negative sentiment as a positive sentiment could lead an investor to purchase stock in a company whose stock price is expected to fall as a result of the negative sentiment. Similarly, misclassifying negative sentiment as neutral may result in an investor holding stock in a firm where he needs to sell it to avoid loss.

The accuracy of the BOW and USE models in predicting negative emotion is notably poor, with accuracy of 50 percent and 57 percent, respectively. That means the model has a one-in-two chance of misclassifying the negative sentiment as neutral or positive. However, the BERT model predicts negative sentiments with a 67 percent accuracy, which is much higher than the BOW and USE models. That equates to one in every three chances of misclassifying negative sentiment as neutral or positive. However, the BERT model has a 60% accuracy rate in predicting positive sentiment and a 35% possibility of misclassifying positive sentiment as neutral, which could potentially result in lower gains.

When compared to SVM classifiers using BOW and USE, it is clear that the SVM classifier using BERT has the least overfitting, the highest performance measures, and the highest accuracy scores for various sentiments in the confusion matrix. BERT's supremacy was expected given that it has been pre-trained on an absurd amount of data and can capture the context of a sentence far better than BOW and USE. As a result, in reality, the BERT model can be utilized for SA of financial news. By doing so, the project's aim has been achieved. However, this should not be regarded as a stand-alone trading strategy due to the model's shortcomings, which have been mentioned. To address the identified issues, more study is required. Given that BERT models are pre-trained using deep neural networks, using with Neural Networks for SA could result in higher performance than SVM and perhaps overcome all or some of the shortcomings listed.

Chapter 5

5. Sentiment Analysis using BERT & Neural Networks

5.1. Introduction

The preceding chapter demonstrated that a linear SVM classifier utilizing BERT text representation could effectively categorize financial news with micro-averaged Precision, Recall, and F-Score values of 0.742, 0.706, and 0.722, respectively. The model, however, has limits, since there was evidence of overfitting and the model performing poorly on negative sentiments. Recent studies have revealed that Neural Networks outperform SVM machine learning approaches in terms of performance. Behdenna et al. [21] highlighted in her assessment of document-level SA that, while SVM machine learning approaches have been prominent in this field, deep learning methods have attracted the interest of researchers in recent years due to their significant outperformance of traditional methods. Moraes et al. [68] conducted an empirical comparison of SVM and Artificial Neural Networks (ANN) in the context of SA. The research was conducted using four distinct datasets: benchmark movie review datasets and a collection of reviews gathered from amazon.com in four various product domains: GPS, Books, and Cameras. The final results demonstrated that ANN yielded results that were better or equivalent to SVM's. Gupta et al. [69] used Decision Trees, Logistic Regression, SVM, and Neural Networks for sentiment analysis of Twitter postings in another study. The train dataset contained around 1.6 million labelled tweets, while the test dataset contained approximately 500 labelled tweets. The results revealed that Neural Networks outperformed all methods, with an accuracy of 80%.

However, there have been studies in which SVM performs better than Neural Networks and also where both SVM and Neural Network methods perform equally poorly or produce unacceptable results. For example, a study conducted by Porshnev et al. [70] for predicting stock market indicators based on historical data and data from Twitter SA revealed that SVM performed better than Neural Networks, however the difference was not significant. Chiruzzo et al. [71] carried out a SA study in which they compared SVM, Convolution Neural Networks (CNN),

and Long Short Term Memory (LSTM) networks. In this study, the following approaches were employed for each method: SVM with word embeddings, centroids, and manually constructed features, CNN with word embeddings as input, and LSTM with word embeddings. The results showed that there was no clear winner among the classifiers, since different methods performed better on different datasets.

Interestingly, Araci's [72] study, which used the same dataset as this project, was able to get f-scores of 0.64 and 0.7 using LSTM classifiers with GLoVE embeddings and LSTM classifiers with ELMo embeddings, respectively. When these results are compared to the results of this project, it is clear that an SVM classifier utilizing BERT outperformed Neural Networks using traditional text representations. Therefore, given that this project has established that BERT outperforms traditional text representations, and recent studies indicating that Neural Networks outperform SVM in most cases, the aim of this chapter is to use Neural Networks using BERT to try and achieve better performance and address the shortcomings that the SVM classifier using BERT demonstrated.

5.2. Methodology

The same methodology and BERT model detailed previously was utilized to preprocess and encode the dataset into train and test datasets. Therefore, the train dataset (bert_fntrain) comprised 3,872 rows and 768 columns, while the test dataset (bert_fntest) had 970 rows and 768 columns and was ready for SA using Neural Networks. There are different types of Neural Networks such as Multi-layer Perceptron (MLP), Autoencoder and Denoising Autoencoder, CNN, RNN, LSTM, Attention Mechanism with RNN, Memory Network and Recursive Neural Network [7]. Given the complexity of Neural Networks, SA using Neural Networks can form a project on its own. Therefore, a basic MLP neural network was used. This is one of the most commonly used Neural Networks which are nothing more than nonlinear regression and discriminant models. For more details on MLP, refer to [73].

The MLPclassifier was then loaded from the Scikit-Library, along with GridSearchCV, for hyperparameter tuning. This MLPclassifier implements a MLP algorithm that trains using backpropagation and supports multi-class classification by applying SoftMax as the output

function. For the MLPclassifier, all the parameters were left as default except for the number of iterations, neurons and the hidden layers. Therefore, the combination of neurons and hidden layers used were as follows: (100,), (200,), (100, 50,), (100, 100,), (100, 50, 25,). Each element in the tuple represents the number of neurons in that hidden layer. For example, (100,) has one hidden layer with 100 neurons. The different values of epochs used were 150, 200 and 250. A five-fold cross validation to prevent overfitting, a random state of seven to control randomness and macro-averaging f-score to evaluate the performance of the models was used again. The Grid-Search approach was now employed with the above mentioned parameters, resulting in a total of 15 models. The best performing model determined by its validation score was saved and used for evaluating the test dataset.

5.3. Results and Discussion

Table 4 shows the macro-averaged validation and train f-scores of 15 MLP classifier models trained on BERT text representation for different number of neurons, hidden layers, and iterations. The models are displayed in descending order of the validation score, which was used to evaluate the models' performance. The model's performance varies from 0.685 to 0.626. When the train and validation scores in Table 4 are compared, there is evidence of overfitting, as demonstrated in Table 2 for SVM classifier models employing BERT. Following on, models with two hidden layers outperform models with one and three hidden layers. However, overfitting can be seen in a model with 2 hidden layers with 100 and 50 neurons at 150 iterations. Besides that, the number of iterations used does not have any effect on the model performance as the difference in f-scores are insignificant for the same number of neurons and hidden layers. Therefore, the best performing model with the parameters of two hidden layers with each having 100 neurons and 250 iterations were used to evaluate the test dataset. Unless otherwise stated, the following discussion will be limited to this model and the test dataset.

Table 5 shows the macro-averaged Precision, Recall and F-Score of SVM and MLP classifiers using BERT. The performance measures of the SVM classifier using BERT is the same as that in Table 3, it was entered into Table 5 for the ease of comparison with MLP classifier performance measures. The MLP classifier has a macro-averaged Precision, Recall and F-Score of

0.703, 0.694, and 0.692, respectively. These performance score are relatively lower than that of scores exhibited by SVM classifier. Furthermore, the performance scores of MLP classifier using BERT are very similar to that of SVM classifier using USE as seen in Table 3.

Table 4: Macro-averaged f-scores for MLP classifier using BERT

Validation Score	Train Score	Hidden Layers Sizes	Iterations
0.685	0.799	(100, 100,)	250
0.680	0.775	(100, 100,)	150
0.680	0.788	(100, 100,)	200
0.676	0.765	(100, 50,)	200
0.676	0.765	(100, 50,)	250
0.675	0.760	(100,)	150
0.670	0.759	(100,)	200
0.668	0.764	(100,)	250
0.664	0.744	(100, 50, 25,)	150
0.664	0.747	(100, 50,)	150
0.656	0.732	(100, 50, 25,)	250
0.656	0.732	(100, 50, 25,)	200
0.626	0.691	(200,)	250
0.626	0.691	(200,)	200
0.626	0.691	(200,)	150

Table 5:Macro-averaged Precision, Recall and F-Score SVM and MLP models using BERT

Method	Precision	Recall	F-Score
SVM	0.742	0.706	0.722
MLP	0.703	0.694	0.692

Figure 5 depicts the MLP classifier's confusion matrix while employing BERT. The model correctly classifies negative, neutral, and positive feelings with accuracy rates of 55%, 79%, and 74%, respectively. As seen in Figure 4(c), the model performs much worse than an SVM classifier

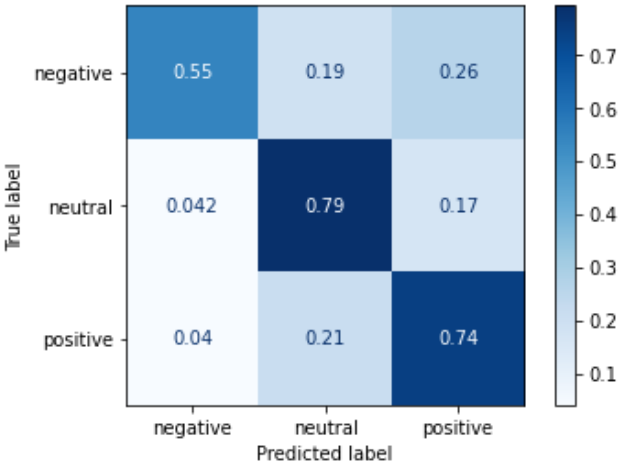

Figure 5: Confusion matrix of MLP classifier using BERT

employing BERT at predicting negative and neutral attitudes. However, when it comes to predicting positive feelings, the models outperform the SVM classifier with BERT. Nonetheless, an analysis of these results leads to the conclusion that SVM classifiers using BERT outperform Neural Networks using BERT. As a result, the aim of this chapter to try to improve the performance and overcome the flaws that the SVM classifier utilizing BERT presented is unsuccessful.

Chapter 6

6. Conclusion and Future Work

The aim of this project was to use deep neural networks to analyze the sentiments of financial news. In doing so, the project utilized pre-trained deep neural network language models such as USE and BERT to represent text in a way that is suitable for use with machine learning methods. These models' performance were compared to that of a baseline BOW model. SVM classifiers utilizing BERT outperformed both the USE and BOW models, according to the results. BERT's dominance was due to its ability to capture the context of a sentence significantly better than BOW and USE. The macro-averaged Precision, Recall, and F-Score values for the SVM classifier using BERT were 0.742, 0.706, and 0.722, respectively. Despite the strong performance of this BERT model, evidence of overfitting was discovered during hyperparameter tuning, as well as overtraining on neutral sentiments. Following that, Neural Networks (MLP Classifier) were employed to try to improve on the results obtained by the SVM classifier using BERT. The results showed that Neural Networks could not outperform SVM. Nonetheless, Neural Networks outperformed SVM classifiers in predicting positive sentiments. The only two drawbacks of this study were signs of overfitting and the classifiers' inability to detect negative sentiments with better accuracy. This may be overcome by adding additional negative sentiments to the dataset, which would resolve the huge imbalance in the dataset shown in Figure 3, resulting in a better fit of the model and more accurate predictions of the negative sentiment.

In conclusion, combining BERT with machine learning methods outperforms standard text representation methods significantly, and SVM exhibits the high performance in terms of SA. In reality, the work done for this project can be applied to real-world SA. The limitations addressed in the project, however, must be considered. Future work should focus on overcoming the study's limitations, and now that it has been established that BERT is the best language model for SA, BERT should be fine-tuned on diverse datasets such as financial news headlines, tweets, and so on, where BERT itself acts as a classifier instead of other machine learning methods.

References

[1] A. M. Turing, “COMPUTING MACHINERY AND INTELLIGENCE,” *Comput. Mach. Intell. Mind*, vol. 49, pp. 433–460, 1950.

[2] “Natural Language Processing Market | 2021 - 26 | Industry Share, Size, Growth - Mordor Intelligence.” https://www.mordorintelligence.com/industry-reports/natural-language-processing-market (accessed Jan. 02, 2022).

[3] G. G. Chowdhury, “Natural language Processing,” *Annu. Rev. Inf. Sci. Technol.*, vol. 37, no. 1, pp. 51–89, 2005, doi: https://doi.org/10.1002/aris.1440370103.

[4] E. D. Liddy, “Natural Language Processing,” *Cent. Nat. Lang. Process.*, 2001, Accessed: Dec. 13, 2021. [Online]. Available: https://surface.syr.edu/cnlp.

[5] R. Feldman, “Techniques and applications for sentiment analysis,” *Commun. ACM*, vol. 56, no. 4, pp. 82–89, 2013, doi: 10.1145/2436256.2436274.

[6] E. Cambria, D. Das, S. Bandyopadhyay, and A. Feraco, *A Practical Guide to Sentiment Analysis*. Socio-Affective Computing, 2017.

[7] L. Zhang, S. Wang, B. Liu, and C. Bing Liu, “Deep learning for sentiment analysis: A survey,” 2018, doi: 10.1002/widm.1253.

[8] C.-J. Wang, M.-F. Tsai, T. Liu, and C.-T. Chang, “Financial Sentiment Analysis for Risk Prediction,” pp. 14–18, 2013, Accessed: Jul. 13, 2021. [Online]. Available: http://www.nd.edu/.

[9] B. Liu, “Sentiment Analysis and Subjectivity,” *Handb. Nat. Lang. Process.*, pp. 627–666, 2010.

[10] L. C. Yu, J. L. Wu, P. C. Chang, and H. S. Chu, “Using a contextual entropy model to expand emotion words and their intensity for the sentiment classification of stock market news,” *Knowledge-Based Syst.*, vol. 41, pp. 89–97, Mar. 2013, doi: 10.1016/J.KNOSYS.2013.01.001.

[11] “The rise of retail investing,” Sep. 08, 2021. https://unitedfintech.com/blog/the-rise-of-retail-investing/ (accessed Jan. 02, 2022).

[12] H. Lee, M. Surdeanu, B. Maccartney, and D. Jurafsky, “On the Importance of Text Analysis for Stock Price Prediction,” Accessed: Jul. 11, 2021. [Online]. Available: http://nlp.

[13] Y. Yu, W. Duan, and Q. Cao, “The impact of social and conventional media on firm equity value: A sentiment analysis approach,” *Decis. Support Syst.*, vol. 55, no. 4, pp. 919–926, Nov. 2013, doi: 10.1016/J.DSS.2012.12.028.

[14] X. Li, H. Xie, L. Chen, J. Wang, and X. Deng, “News impact on stock price return via sentiment analysis,” *Knowledge-Based Syst.*, vol. 69, no. 1, pp. 14–23, Oct. 2014, doi: 10.1016/J.KNOSYS.2014.04.022.

[15] S. W. K. Chan and M. W. C. Chong, “Sentiment analysis in financial texts,” *Decis. Support*

Syst., vol. 94, pp. 53–64, Feb. 2017, doi: 10.1016/J.DSS.2016.10.006.

[16] S. G. Chowdhury, S. Routh, and S. Chakrabarti, "News Analytics and Sentiment Analysis to Predict Stock Price Trends," Accessed: Dec. 27, 2021. [Online]. Available: www.ijcsit.com.

[17] J. Bollen, H. Mao, and X.-J. Zeng, "Twitter mood predicts the stock market," Accessed: Dec. 27, 2021. [Online]. Available: http://www.sca.isr.umich.edu/.

[18] B. Li, K. C. C. Chan, and C. Ou, "Public Sentiment Analysis in Twitter Data for Prediction of A Company's Stock Price Movements," 2014, doi: 10.1109/ICEBE.2014.47.

[19] V. S. Pagolu, K. N. Reddy, G. Panda, and B. Majhi, "Sentiment analysis of Twitter data for predicting stock market movements," *Int. Conf. Signal Process. Commun. Power Embed. Syst. SCOPES 2016 - Proc.*, pp. 1345–1350, Jun. 2017, doi: 10.1109/SCOPES.2016.7955659.

[20] K. Ravi and V. Ravi, "A survey on opinion mining and sentiment analysis: Tasks, approaches and applications," *Knowledge-Based Syst.*, vol. 89, pp. 14–46, Nov. 2015, doi: 10.1016/J.KNOSYS.2015.06.015.

[21] S. Behdenna, F. Barigou, and G. Belalem, "Document Level Sentiment Analysis: A survey," *EAI Endorsed Trans. Context. Syst. Appl.*, vol. 4, no. 13, p. 154339, Mar. 2018, doi: 10.4108/EAI.14-3-2018.154339.

[22] A. M. Rahat, A. Kahir, and A. K. M. Masum, "Comparison of Naive Bayes and SVM Algorithm based on Sentiment Analysis Using Review Dataset," *Proc. 2019 8th Int. Conf. Syst. Model. Adv. Res. Trends, SMART 2019*, pp. 266–270, Feb. 2020, doi: 10.1109/SMART46866.2019.9117512.

[23] A. L. F. Alves, A. A. Firmino, M. G. De Oliveira, and A. C. De Paiva, "A Comparison of SVM Versus Naive-Bayes Techniques for Sentiment Analysis in Tweets: A Case Study with the 2013 FIFA Confederations Cup," 2014, doi: 10.1145/2664551.2664561.

[24] S. Tan and J. Zhang, "An empirical study of sentiment analysis for chinese documents," *Expert Syst. Appl.*, vol. 34, no. 4, pp. 2622–2629, May 2008, doi: 10.1016/J.ESWA.2007.05.028.

[25] Y. Al Amrani, M. Lazaar, and K. E. El Kadirp, "Random Forest and Support Vector Machine based Hybrid Approach to Sentiment Analysis," *Procedia Comput. Sci.*, vol. 127, pp. 511–520, Jan. 2018, doi: 10.1016/J.PROCS.2018.01.150.

[26] K. Lu and J. Wu, "Sentiment Analysis of Film Review Texts Based on Sentiment Dictionary and SVM," doi: 10.1145/3319921.3319966.

[27] M. Hagenau, M. Liebmann, and D. Neumann, "Automated news reading: Stock price prediction based on financial news using context-capturing features," *Decis. Support Syst.*, vol. 55, no. 3, pp. 685–697, Jun. 2013, doi: 10.1016/J.DSS.2013.02.006.

[28] A. K. Uysal and S. Gunal, "The impact of preprocessing on text classification," *Inf. Process. Manag.*, vol. 50, no. 1, pp. 104–112, Jan. 2014, doi: 10.1016/J.IPM.2013.08.006.

[29] D. Cer *et al.*, “Universal Sentence Encoder.” [Online]. Available: https://tfhub.dev/google/.

[30] J. Devlin, M.-W. Chang, K. Lee, K. T. Google, and A. I. Language, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” [Online]. Available: https://github.com/tensorflow/tensor2tensor.

[31] J. Eisenstein, *Introduction to Natural Language Processing*. 2018.

[32] D. M. E. D. M. Hussein, “A survey on sentiment analysis challenges,” *J. King Saud Univ. - Eng. Sci.*, vol. 30, no. 4, pp. 330–338, Oct. 2018, doi: 10.1016/J.JKSUES.2016.04.002.

[33] A. Kumar and T. M. Sebastian, “Sentiment Analysis: A perspective on its Past, Present and Future,” *Intell. Syst. Appl.*, vol. 10, pp. 1–14, 2012, doi: 10.5815/ijisa.2012.10.01.

[34] L. Yue, W. Chen, X. Li, W. Zuo, and M. Yin, “A survey of sentiment analysis in social media,” *Knowl. Inf. Syst.*, vol. 60, pp. 617–663, 2019, doi: 10.1007/s10115-018-1236-4.

[35] F. Hemmatian, · Mohammad, and K. Sohrabi, “A survey on classification techniques for opinion mining and sentiment analysis,” 2017, doi: 10.1007/s10462-017-9599-6.

[36] H. Kaur, V. Mangat, and Nidhi, “A survey of sentiment analysis techniques,” *Proc. Int. Conf. IoT Soc. Mobile, Anal. Cloud, I-SMAC 2017*, pp. 921–925, Oct. 2017, doi: 10.1109/I-SMAC.2017.8058315.

[37] A. D. ’ Andrea, F. Ferri, and P. Grifoni, “Approaches, Tools and Applications for Sentiment Analysis Implementation,” *Int. J. Comput. Appl.*, vol. 125, no. 3, pp. 975–8887, 2015, Accessed: Dec. 21, 2021. [Online]. Available: http://messenger.yahoo.com/features/emoticons.

[38] W. Medhat, A. Hassan, and H. Korashy, “Sentiment analysis algorithms and applications: A survey,” *Ain Shams Eng. J.*, vol. 5, no. 4, pp. 1093–1113, Dec. 2014, doi: 10.1016/J.ASEJ.2014.04.011.

[39] Z. Zhang, Q. Ye, Z. Zhang, and Y. Li, “Sentiment classification of Internet restaurant reviews written in Cantonese,” *Expert Syst. Appl.*, vol. 38, no. 6, pp. 7674–7682, Jun. 2011, doi: 10.1016/J.ESWA.2010.12.147.

[40] VAPNIK and V. N., *The Nature of Statistical Learning*, 2nd ed. 1995.

[41] P. Routray, C. Kumar Swain, and S. Prava Mishra, “A Survey on Sentiment Analysis,” *Int. J. Comput. Appl.*, vol. 76, no. 10, pp. 975–8887, 2013.

[42] S. Aftab, I. Ali, and M. Ahmad, “Sentiment Analysis of Tweets using SVM,” *Artic. Int. J. Comput. Appl.*, vol. 177, no. 5, pp. 975–8887, 2017, doi: 10.5120/ijca2017915758.

[43] Jason Brownlee, “A Gentle Introduction to the Bag-of-Words Model,” *Deep Learning for Natural Language Processing*, Oct. 09, 2017. https://machinelearningmastery.com/gentle-introduction-bag-words-model/ (accessed Jan. 02, 2022).

[44] Jacob Devlin and Ming-Wei Chang, “Google AI Blog: Open Sourcing BERT: State-of-the-Art Pre-training for Natural Language Processing,” Nov. 02, 2018. https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html (accessed Jan. 03, 2022).

[45] “Universal Sentence Encoder.” https://tfhub.dev/google/collections/universal-sentence-encoder/1 (accessed Dec. 20, 2021).

[46] “Classifying Tweets With LightGBM and the Universal Sentence Encoder | by Grid Search | Medium,” Sep. 10, 2020. https://medium.com/@invest_gs/classifying-tweets-with-lightgbm-and-the-universal-sentence-encoder-2a0208de0424 (accessed Jan. 07, 2022).

[47] Saad Arshad, “Using USE (Universal Sentence Encoder) to Detect Fake News | by Saad Arshad | Towards Data Science,” Nov. 19, 2019. https://towardsdatascience.com/using-use-universal-sentence-encoder-to-detect-fake-news-dfc02dc32ae9 (accessed Jan. 07, 2022).

[48] Rani Horev, “BERT Explained: State of the art language model for NLP | by Rani Horev | Towards Data Science,” Nov. 10, 2018. https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270 (accessed Jan. 01, 2022).

[49] Anirban Sen, “Text Classification — From Bag-of-Words to BERT — Part 6(BERT) | by anirban sen | Analytics Vidhya | Medium,” Jan. 12, 2021. https://medium.com/analytics-vidhya/text-classification-from-bag-of-words-to-bert-part-6-bert-2c3a5821ed16 (accessed Jan. 02, 2022).

[50] A. Vaswani *et al.*, “Attention Is All You Need.”

[51] “Classify text with BERT | Text | TensorFlow.” https://www.tensorflow.org/text/tutorials/classify_text_with_bert (accessed Jan. 02, 2022).

[52] “BERT Overview.” https://tfhub.dev/google/collections/bert/1 (accessed Jan. 02, 2022).

[53] M. Gomes De Sousa *et al.*, “BERT for Stock Market Sentiment Analysis,” doi: 10.1109/ICTAI.2019.00231.

[54] A. Chiorrini, C. Diamantini, A. Mircoli, and D. Potena, “Emotion and sentiment analysis of tweets using BERT,” 2021, Accessed: Jan. 07, 2022. [Online]. Available: https://code.google.com/archive/p/word2vec/.

[55] “Project Jupyter | Home.” https://jupyter.org/ (accessed Dec. 18, 2021).

[56] “Welcome to Python.org.” https://www.python.org/ (accessed Dec. 18, 2021).

[57] “NumPy.” https://numpy.org/ (accessed Dec. 18, 2021).

[58] “pandas - Python Data Analysis Library.” https://pandas.pydata.org/ (accessed Dec. 18, 2021).

[59] “An introduction to seaborn — seaborn 0.11.2 documentation.”

https://seaborn.pydata.org/introduction.html (accessed Dec. 18, 2021).

[60] “scikit-learn: machine learning in Python — scikit-learn 1.0.1 documentation.” https://scikit-learn.org/stable/ (accessed Dec. 18, 2021).

[61] “TensorFlow.” https://www.tensorflow.org/ (accessed Dec. 18, 2021).

[62] P. Malo, A. Sinha, P. Takala, P. Korhonen, and J. Wallenius, “Good Debt or Bad Debt: Detecting Semantic Orientations in Economic Texts,” 2013.

[63] “sklearn.feature_extraction.text.CountVectorizer — scikit-learn 1.0.1 documentation.” https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html (accessed Dec. 19, 2021).

[64] “Universal Sentence Encoder Large.” https://tfhub.dev/google/universal-sentence-encoder-large/5 (accessed Dec. 20, 2021).

[65] “BERT English Uncased - L12H768A12.” https://tfhub.dev/tensorflow/bert_en_uncased_L-12_H-768_A-12/4 (accessed Dec. 20, 2021).

[66] “BERT English Uncased Preprocess.” https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3 (accessed Dec. 20, 2021).

[67] R. Prabowo and M. Thelwall, “Sentiment analysis: A combined approach,” *J. Informetr.*, vol. 3, no. 2, pp. 143–157, Apr. 2009, doi: 10.1016/J.JOI.2009.01.003.

[68] R. Moraes, J. F. Valiati, and W. P. Gavião Neto, “Document-level sentiment classification: An empirical comparison between SVM and ANN,” *Expert Syst. Appl.*, vol. 40, no. 2, pp. 621–633, Feb. 2013, doi: 10.1016/J.ESWA.2012.07.059.

[69] A. Gupta, A. Singh, I. Pandita, and H. Parashar, *Sentiment Analysis of Twitter Posts using Machine Learning Algorithms*. .

[70] A. Porshnev, I. Redkin, and A. Shevchenko, “Machine learning in prediction of stock market indicators based on historical data and data from Twitter sentiment analysis,” 2013, doi: 10.1109/ICDMW.2013.111.

[71] L. Chiruzzo and A. Rosá, “RETUYT-InCo at TASS 2018: Sentiment Analysis in Spanish Variants using Neural Networks and SVM,” 2018, Accessed: Jan. 06, 2022. [Online]. Available: https://www.cs.york.ac.uk/semeval-2013/.

[72] D. T. Araci, “FinBERT: Financial Sentiment Analysis with Pre-trained Language Models,” 2019.

[73] “Neural network models (supervised) — Multi-layer Perceptron.” https://scikit-learn.org/stable/modules/neural_networks_supervised.html (accessed Jan. 07, 2022).