Tilburg University

MEASURING DOMAIN-SPECIFIC SENTIMENT TO PREDICT STOCK PRICES

THE WALLSTREETBETS 'MEME-STOCK' SAGA

STEFAN WINTER

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
AT THE SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
OF TILBURG UNIVERSITY

STUDENT NUMBER

2067606

COMMITTEE

Dr. Peter Hendrix
Dr. Raquel Garrido Alhama

LOCATION

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science &
Artificial Intelligence
Tilburg, The Netherlands

DATE

December 28, 2021

ACKNOWLEDGMENTS

I want to thank Dr. Peter Hendrix for his invaluable guidance during this thesis. Additionally, I want to thank Sveva Cadura and Anna Carolina van Melis for their support during the writing process.

MEASURING DOMAIN-SPECIFIC SENTIMENT TO PREDICT STOCK PRICES

THE WALLSTREETBETS 'MEME-STOCK' SAGA

STEFAN WINTER

Abstract

Until the GameStop short-squeeze in early 2021, the impact of changes in sentiment of the Reddit discussion board *WallStreetBets* on the financial market was vastly underappreciated. Due to the novelty of the WallStreetBets phenomenon, there is also almost no research available on that topic. This thesis will explore methodologies on how to measure sentiment of the discussion board and use the sentiment to predict changes of the GameStop stock. One of the challenges when measuring the sentiment of WallStreetBets is the usage of novel domain-specific words and terminologies, which are shown to have a big impact on the results of sentiment analysis. Hence, this thesis proposes a method to create a dataset that covers the sentiment of text data which includes the terminology of a given domain. It will be shown that sentiment analysis machine learning models that use the domain-specific text corpus as input outperform general purpose lexicons, which are currently commonly used by both academia and industry to measure the sentiment of WallStreetBets.

1 INTRODUCTION

Modern society has been able to access vast amounts of information, communicate ideas, and become part of communities with the advent of the internet. Online discussion boards are playing a critical role by providing a platform where people can do so. Those discussion boards are also used by a variety of people to talk about the stock market and discuss trading strategies. Recently, the Reddit forum WallStreetBets has become one of the most well-known and influential investing online-forums.

Even though the Reddit subforum was created in 2012 already, it received the majority of its media exposure in 2021 as a result of a short-squeeze of the GameStop (GME) stock, which drove the stock price up hundreds of percentage points (Diangson & Jung, 2021). Over the ensuing months, however, the stock price experienced extraordinary volatility. Prices fluctuated by double-digit percentage points which not only lead to gains, but also to large losses for market participants. Research shows that discussion board activity can be one cause of increased volatility (Das & Chen, 2007). Interestingly, finance scholars did not consider Reddit as a platform capable of having such a significant impact on the financial markets. As a result, the site has been neglected in their research (Long, Lucey, & Yarovaya, 2021).

However, it was neither the volatility nor the rapid price appreciation in the beginning of the short-squeeze of the Gamestop stock that astounded market observers. Instead, it was the unprecedented decentralized and coordinated buying of Gamestop shares by members of the WallStreetBets community that attracted attention (Anand & Pathak, 2021). Organizing the mass-coordinated buying of stock, however, requires that enough participants share the same sentiment. Furthermore, social media sentiment has a particularly strong impact on uninformed traders, of which WallStreetBets has plenty (Danbolt, Siganos, & Vagenas-Nanos, 2015). It is argued, that coordinated investments will also occur in the future, mainly due to the influence of social media and other online platforms on our society today (Semenova & Winkler, 2021). Hence, it is of the utmost importance to study and understand the impact of WallStreetBets. This thesis attempts to answer one of the numerous questions that have arisen with the growing popularity of Wallstreetbets, by answering the following research question:

> *Can sentiment analysis of the WallStreetBets Reddit-forum be used to predict daily changes in the stock price of Gamestop?*

To begin, it must be determined how the discussions about the Gamestop stock on WallStreetBets should be handled to serve as good input features for sentiment analysis. One of the challenges is the heavy use of peculiar terminology and domain-specific phrases on the WallStreetBets forum, as well as many novel words (Anand & Pathak, 2021). According to recent research, sentiment lexicons and text-corpora with a focus on a certain domain produce superior sentiment analysis results compared to a general-purpose sentiment lexicon or text-corpus (Park, Lee, & Moon, 2015). Furthermore, the text data needs to be cleaned and pre-processed in order to be accurately processed by a machine learning algorithm (Jemai, Hayouni, & Baccar, 2021). As a result, the following sub-research question was formed:

RQ1 *How can the domain-specific language of the Reddit forum WallStreetBets be incorporated into sentiment analysis?*

Subsequently, machine learning models can be trained to perform sentiment analysis. However, each machine learning algorithm has its own idiosyncrasies and assumptions, and no single classifier works optimally in all possible scenarios. Hence, it is a good idea to evaluate the results and performance of different machine learning algorithms. As a result, the best model with a given set of hyperparameters can be selected to solve a particular problem (Raschka & Mirjalili, 2019, p. 53).

This thesis will explore traditional machine learning methods such as Naive Bayes (NB) and Support Vector Machines (SVMs), as well as deep learning methods like Long Short Term Memory (LSTM) and Bidirectional Encoder Representations from Transformers (BERT). Due to the high dimensionality of textual data, deep learning methods have shown to outperform traditional machine learning techniques. That can be explained by the ability of deep learning methods to automatically learn the most important features, whereas traditional methods may suffer from the curse of dimensionality (Xianghua, Jingying, Jainqiang, Min, & Huihui, 2018). As was mentioned earlier, however, no classifier works best in all scenarios which is why the next sub-research question needs to be answered:

RQ2 *How do different sentiment analysis approaches perform based on the predefined evaluation metrics Accuracy, Precision, Recall and F_1-Score?*

An accurate measure of sentiment can represent an infromative feature, which is why the impact of sentiment on changes in stock prices has gained attention by researchers in recent years. However, predicting stock prices is a complex undertaking due to the nonlinearity and nonstationarity in the time series data of stock prices (Nikou, Mansourfar, & Bagherzadeh, 2019). Generally, this means that certain attributes of the data change over time, making it hard to forecast stock prices (Shetty & Ismail, 2021). One established stock-price forecasting method is Auto Regressive Integrated Moving Average (ARIMA), which captures temporal structures in time-series data, has shown excellent predictive results (Caginalp & Constantine, 1995). However, it is not designed to include other features, such as sentiment. This is why this thesis will also compare newer models such as LSTMs and XGBoost, which have demonstrated strong predicitive capabilities with regards to time-series data (Chen, Zhang, Mehlawat, & Jia, 2021). This leads to the following sub-research question:

RQ3 *Do machine learning models show stronger predictive capabilities for changes in the stock price of Gamestop if sentiment obtained from WallStreetBets is included as a feature, based on the predefined evaluation metrics MAE, MSE and RMSE?*

By answering the three sub-research questions, a scientific and thought-out answer for the main research question can be found.

2 RELATED WORK

Gauging sentiment of online forums to predict movements in stock prices has been a research subject for many years now. Das and Chen (2007) conducted a study on the *Yahoo!* message board, which was amongst the first ones on the internet for investors to exchange ideas. Others did similar studies on platforms such as *Twitter, Reddit and StockTwits* and (Anand & Pathak, 2021; Gu & Kurov, 2020; Piñeiro-Chousa, Vizcaino-Gonzalez, & Perez-Pico, 2017). However, since the WallStreetBets *'meme-stock movement'* is a relatively recent phenomenon, there is very little research on that topic and none that accounts for the novel terminology used on the forum to create sentiment models which can be used to predict changes in stock prices. As a result, incorrect conclusions may be drawn.

2.1 *Domain Specific Terminology*

Even though the literature suggests many innovative ways to enhance model performance by a few percentage points, the biggest benefits seem to come from high quality input data in the form of a domain-specific knowledge base. It has been demonstrated that adapting data to a certain domain results in more accurate sentiment analysis results (Park et al., 2015). Furthermore, it is argued that there is no general-purpose sentiment lexicon that can be optimally applied on all domains. This is due to different meaning of terms, depending on the domain. A good example is the word *unpredictable*, which would be associated with negative sentiment for a car but can be a positive label for movies (Pang & Lee, 2008).

One proposed approach is to adapt sentiment lexicons to a specific domain (Yue, Malu, Umeshwar, & ChengXiang, 2011). This adapted lexicon can then be searched to find and score the sentiment of a specific word (Muhammad, 2014). While lexicon-based methods have found widespread adoption, mainly due to their simplicity, more advanced machine learning methods have shown strong performance with regards to domain specificity (Yanyan, Fulian, Jianbo, & Marco, 2020).

Machine learning methods can be used to automatically detect and identify domain-specific words in sentences. By doing so it is assumed that the algorithm can not only detect whether domain-specific words are used, but also identify the position of the term in the sentence. Hence, it is possible to detect new meanings of words in an already existing text corpus. In addition, this approach also allows to classify novel words, that do not yet exist in a lexicon (Zhengqi, Zhewei, & Yang, 2019). This can be achieved by having models that formulate domain-specific word detection as a sequence-labelling task. Furthermore, novel domain-specific words can be learned by understanding the contextual structure of a sentence (Zhengqi et al., 2019). For example, out-of-vocabulary tokens can be learned in the hidden layers of LSTMs (Hochreiter & Schmidhuber, 1997).

To train machine learning models, however, a labeled corpus is needed. Obtaining one is not without its challenges. For example, working with multiple human annotators can lead to discrepancies in the annotation results (Jin-Dong, Tomoko, & Junichi, 2008; Salah & Gayar, 2019). Additionally, it is hard to estimate the total annotation cost which can depend of various factors. One example would be whether the annotator is capable of fluently understanding the language for the given task (Arora, Nyberg, & Rose, 2009). Additionally, labelling an entire dataset incurs extremely high costs, which can be avoided. With the support of an Active Learner, a complete domain-specific corpus with its respective labels can be created using only partial annotations (Park et al., 2015).

One of the key concepts of Active Learners is that if a machine learning algorithm is allowed to choose the data from which it learns, it will achieve higher accuracy with less training data. If a considerable amount of the data is unlabeled, this is especially desirable. As a result, the total cost of annotation can be reduced drastically. Research shows that the total number of manual annotations can be reduced by 80% when using an Active Learner instead of randomly selecting data to label (Baldridge & Osborne, 2004).

In comparison, if data are manually annotated at random (passive learning), the annotator will invest a lot of time into labeling irrelevant instances. That is especially true if the class distribution of the data is imbalanced or if there are many very similar documents. For example, if a specific feature set appears on only 1% of instances, the annotator would have to label 1000 documents to cover the feature set on 10 relevant documents. When it comes to document similarity, large clusters of very similar documents might be identifiable. Because features may be barely distinctable, the annotator might spend a lot of effort labeling uninformative instances when selecting them randomly. An Active Learner, on the other hand, suggests which instances the annotator should label. Those instances can be determined on various quantitative metrics (Miller, Linder, & Mebane, 2020).

Based on the reviewed literature, an Active Learner seems to result in a highly accurate labeled dataset that contains domain-specific terminology which can then be learned by machine learning models.

2.2 *Sentiment Analysis*

Once an annotated text corpus is obtained and pre-processed it can be used by machine learning models to perform sentiment analysis.

To further optimize performance, models can be improved, by applying a character-based convolutional neural network to encode the spelling of words (Zhengqi et al., 2019). Other research shows that the accuracy of an LSTM can be improved by introducing Word2Vec to the LSTM. As a result, the one-hot encoded input to the LSTM is converted into a low dimensional vector that covers the semantic similarity of the words in it. Due to the lower dimensionality, over-fitting can be prevented and the network may also need less parameters (Xiao, Wang, & Zuo, 2018). However, Gennaro, Buonanno, and Palmieri (2021) argue that there is almost no research on specific choices of hyperparameters for the Word2Vec model. Hence, this thesis will add to the literature by considering varying hyperparameters of the Word2Vec embedding.

Other research demonstrates the importance of large pre-trained models using transfer learning (Deng et al., 2009). Devlin, Chang, Lee, and Toutanova (2019) introduce BERT, a pre-trained model that uses the English Wikipedia and the BooksCorpus, which shows promising results. One of the advantages is that only one output layer needs to be added to the model to achieve state-of-the-art sentiment analysis performance. However, it is also shown that BERT lacks domain awareness. Hence, it cannot differentiate between properties of source and target domains. One proposed solution is to replace a random subset of tokens by a *MASK* token. The corresponding hidden values of the MASK token can be fed into the output layer. This adaption to a specific domain is shown to slightly enhance the performance of a vanilla implementation of BERT. However, it is also argued that a vanilla implementation can still outperform other machine learning models (Du, Sun, Wang, Qi, & Liao, 2020). Other research that compares BERT to a lexicon approach shows that on average BERT achieves better performance. However, the better performance cannot be observed on all analyzed test sets which keeps the authors "optimistic about the lexicon-based approach in general" (Kotelnikova, Paschenko, Bochenina, & Kotelnikov, 2021).

However, the implementation of deep-learning models is oftentimes much more complex than traditional machine learning methods. Furthermore, deep-learning models typically also require much more computing power to train the network. The Naive Bayes method, in contrast, is very easy to implement and fast to train. As a result, the classifier is oftentimes used as a baseline for text classification. Multinomial Naive Bayes (MNB), one type of the Naive Bayes classifier, has established itself as the de-facto standard for text classification (Abbas et al., 2019). MNB is a frequency based method that calculates the conditional probability of a word belonging to a specific class (Susanti, Djatna, & Kusuma, 2017). It is argued that MNB performs better than many rule-based lexicas, which are oftentimes used as baseline models (S. Wang & Manning, 2012). Furthermore, MNB seems to perform especially well if the words in the document are shown to be significant for the classification problem (Sharif, Hoque, & Hossain, 2019).

2.3 *Stock Price Prediction*

Using machine learning to predict stock prices has been a research topic for many years now. It was shown that high accuracy is attainable within a short prediction time span (Schöneburg, 1990). However, De Gooijer and Hyndman (2006) found that most published papers mainly focus on time-series forecasting while excluding other features that may also lead to an improvement in performance. It is shown that social media sentiment can have a direct effect of how market participants perceive a company, which can lead to changes in the stock price of companies. This is especially true for smaller firms with low analyst coverage (Feng & Johansson, 2019). Other researchers show that sentiment obtained from Twitter can be used to predict returns of a broader stock market index (Gu & Kurov, 2020). Additionally, Antweiler and Frank (2004) uncover that positive sentiment has a very significant relationship with returns. In other research the emotions of discussions on WallStreetBets are studied by performing sentiment analysis.

Long et al. (2021) tried to uncover the impact of specific emotions such as *"Angry, Fear, Happy, Sad and Surprise"* from the comments on WallStreetBets discussions on intraday changes of the stock price of the affected stock. While they conclude that the tone as well as the number of comments have an impact on the stock price, they show that the number of comments is not directly related to sentiment. Additionally, they argue it is the number of comments that is posted within an hour that has the biggest effect on 1-minute changes of the stock price. Furthermore, the paper shows that the emotions *Sad, Angry and Surprise* have a significant impact on the gamestop 1-minute stock price. The Happy sentiment does not show a significant impact on 1-minute price changes, however, a causality test showed a link between the Happy sentiment and intraday returns of the GME stock. Hence, the authors confirm that Reddit sentiment has an impact on the stock market. They also argue that any asset that is targeted by a large crowd from WallStreetBets can become a subject of excessive volatility, without being driven by any fundamental reasons. Lyócsa, Baumöhl, and Vyrost (2021) also showcased that as the discussion volume on WallStreetBets increased, the volatility of certain stocks got amplified. Additionally, the research of Zaghum, Mariya, Imran, and Shoaib (2021) also found that sentiment of investors on WallStreetBets affected the returns of the Gamestop stock. However, they also demonstrate that other features such as the put-call ratio and the short-sale volume had a strong impact on the stock price. As a result, this thesis will also include other features such as the trading volume as input to the machine learning models.

Most of the aforementioned research on WallStreetBets, however, uses general purpose lexicas to perform sentiment analysis. Hence, this thesis will add to the literature by first selecting the model that shows the most promising performance based on selected evaluation metrics and then implements the predicted sentiment into the stock price prediction model. However, not all algorithms and models can implement sentiment analysis into their stock price prediction.

One of those models is ARIMA, which is widely used in the financial industry. ARIMA works well on time-series data due to its ability to catch time-series specific features (Vuong, Dat, Mai, Uyen, & Bao, 2021). It does so by gauging the strength of one dependent variable based on other independent ones that may change. One of the limitations of ARIMA is that the time-series data needs to be stationary, meaning there are no trends in the data. To achieve stationarity, the data can be differentiated. By applying the *Augmented Dickey-Fuller* test, the condition for stationarity can be tested (Ivanovic, Bogdan, & Baresa, 2013).

Other models that can include sentiment are LSTMs. They have shown strong results for time series prediction (Rammurthy & Patil, 2021). That is because of its strengths in analyzing connections among time-series data by using LSTM's memory function. Other feed-forwards neural networks, as a comparison, cannot handle the complex time correlation between information. Even though the results seem promising when using an LSTM to predict the stock prices, the performance of the LSTM can be enhanced by accounting for sentiment as well (Jin, Yang, & Liu, 2020; H. Wang, Guo, & Chen, 2019).

Even though the literature extensively studies various time-series prediction methods, there is very little research on stock price prediction for stocks that are heavily discussed on online forums such as WallStreetBets and even less research that also incorporates the sentiment of those online forums. This thesis tries to close that gap in the literature based on the case of Gamestop.

3 METHODOLOGY

3.1 *General Description*

This subsection describes the methodology used to

- create a domain-specific annotated dataset,
- using the dataset for sentiment analysis and
- predicting the stock price of GameStop.

For better understanding of the methodologies used, this paragraph will provide a brief summary of the entire workflow before explaining each step in more detail. To begin with, stock prices and posts from the WallStreetBets subreddit are obtained for GameStop. Since the posts are completely unlabeled, ten percent of the data are manually labeled to perform some data exploration and gain some basic understanding. The annotated targets are either *bearish, neutral or bullish*. Generally, bearish is associated with negative sentiment where investors assume a decline in stock prices. Bullish, on the other hand, relates to positive sentiment where investors hope for rising stock prices. Subsequently, an Active Learner is applied to label the rest of the data. As a result, *Ground Truth* data are created which can be used for the machine learning sentiment models. Subsequently, the model with the best evaluation metrics is used to predict the sentiment of the entire dataset. The predicted sentiment can then be applied as an additional input feature for the stock price prediction task.

3.1.1 *The Case for a Semi-Supervised Method over an Unsupervised Method to Label Data*

Since the data obtained from Reddit is unlabeled, it cannot be fed into supervised machine learning algorithms. That is because supervised sentiment analysis methods rely on labeled data (Sazzed & Jayarathna, 2021). One approach to label data is using unsupervised machine learning models. Unsupervised models are commonly applied in Natural Language Processing and text classification (Namcheol & Ghang, 2019). However, unsupervised models are a better choice for uncovering hidden patterns in a dataset, especially without any a priori knowledge of the structure of the data. As a result, unsupervised models excel at summarizing or exploring a large text corpus. For the case at hand, a t-Distributed Stochastic Neighbor embedding (t-sne) algorithm was applied on the data to extract similarity features and project them onto a lower dimension (Binu & Sony, 2020). As can be seen in Figure 1, admittedly at a low dimension, the majority of the

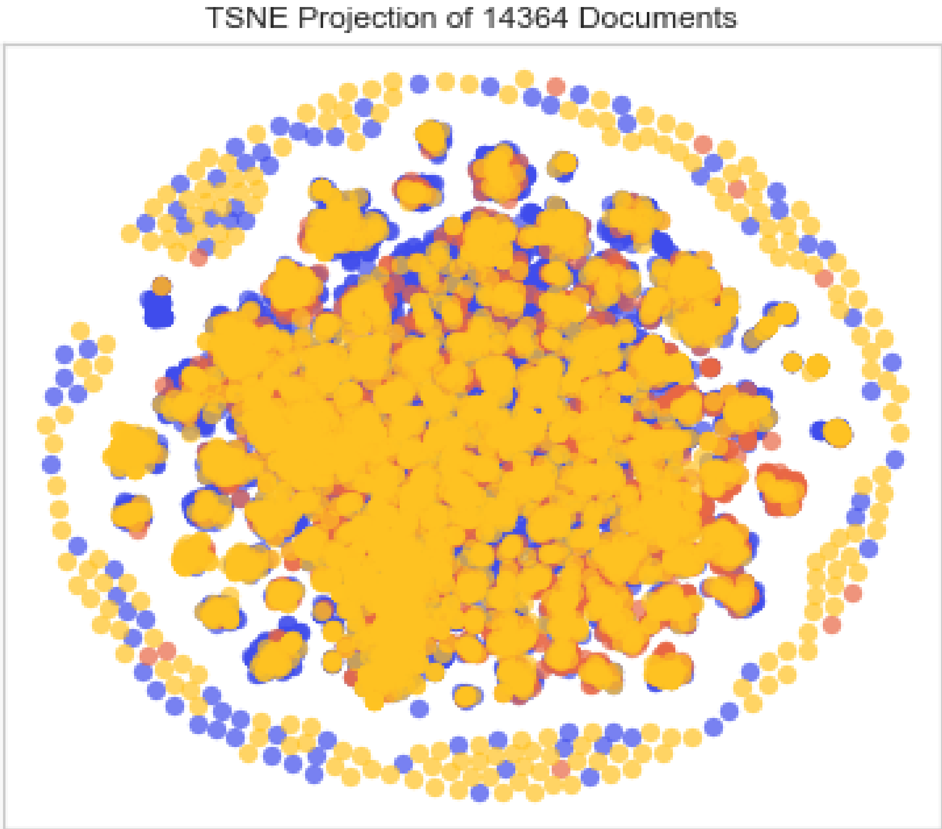

Figure 1: t-sne visualization of seed data

data do not belong to any particular cluster. The data used for the t-sne algorithm was the manually labeled text data that was transformed into a Term Frequence-Inverse Document Frequence (tf-idf) representation.

Even though there are some approaches to clustering high dimensional data, it generally is difficult to do so accurately. One of the explanations is the increased sparsity and the difficulty to distinguish between the distances of specific instances (Tomasev, Radovanovic, Mladenic, & Ivanovic, 2014). As a result, an unsupervised approach was not chosen for this thesis.

However, labelled data is still needed to train supervised machine learning models for sentiment analysis. While manually labelling all data might be the most accurate solution, it is associated with high costs (Miller et al., 2020). Hence, this thesis proposes the implementation of an Active Learner. With its support, a complete domain-specific corpus can be labeled while only relying on partial annotations (Park et al., 2015). As a result, a domain-specific labeled text corpus is created that can be used to compare the performance of different supervised machine learning algorithms.

3.1.2 *Active Learner Workflow*

The illustrated workflow in Figure 2 provides an overview of how an Active Learner works. To begin with, cleaned and pre-processed data needs to be available that can be used by the Active Learner. Furthermore, the Active Learner can also be trained with some initial training data, which is also referred to as the seed. By using clustering algorithms, the seed data can be selected and labeled methodologically, which allows the Active Learner to achieve higher accuracy faster when compared to randomly picking the initial seed data (Kang, Ryu, & Kwon, 2004). All the unlabeled instances will become the pool data, which need to be labeled. The seed data is fed into the Active Learner and trains an estimator, which needs to be defined when creating the Active Learner.

In addition, a query strategy needs to be defined, based on which the Active Learner queries new instances from the aforementioned pool. A query strategy evaluates the informativeness of unlabeled samples. Common strategies are *uncertainty sampling, query-by-committee, expected model change, expected error reduction and variance reduction*.

While each strategy has its own intricacies, all essentially try to find instances that are hard for the model to classify and hence might benefit from manual annotation. After the query function selected instances from the pool, an oracle needs to label those. An oracle normally is at least one human with knowledge on how to annotate the data at hand (Settles, 2009). Once the new instances are labeled, those instances need to be removed from the pool, since they are now part of the labeled data. The Active Learner then needs to be taught the new instances, which he can use to adjust the model. After each iteration, the results can be evaluated. A common performance measure for Active Learners is *accuracy*.

If a predefined stopping criterion is not yet met, the query strategy selects more instances from the pool and repeats the process. If the stopping criterion is met, the process ends (Lu, Henchion, & Namee, 2019).

3.1.3 *Sentiment Analysis Models*

The next section explores the machine learning models that will be used to perform sentiment analysis on the domain-specific corpus created by the Active Learner.

Naïve Bayes (NB)

NB is a probabilistic supervised machine learning model. By working probabilistically, the classifier assigns the probability of belonging to a given class based on certain features (Jemai et al., 2021). Because of the high dimensionality of textual data, which can be handled very well by

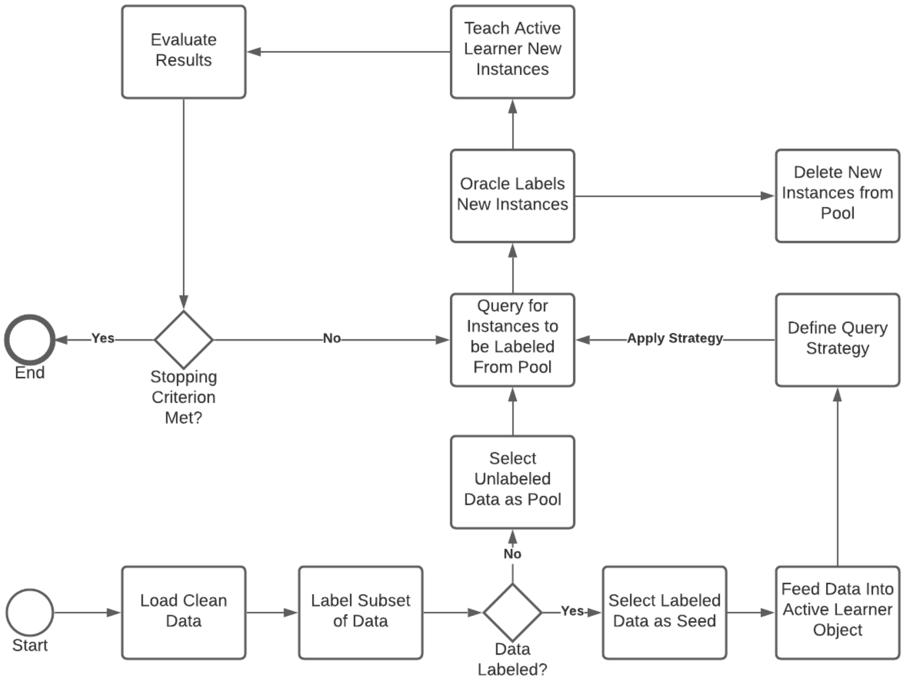

Figure 2: Visualized Workflow of an Active Learner. Created with lucid.app

NB, this algorithm has established itself as one of the standards for sentiment analysis. This thesis will use Multinomial Naïve Bayes to classify the sentiment of the text. This is due to the model's ability to handle larger vocabulary sizes. In addition, the algorithm is simple to implement, suitable for real-time applications, and highly scalable. However, the algorithm's prediction accuracy is frequently lower than that of other sentiment analysis techniques (Song, Kim, Lee, Kim, & Youn, 2017). Due to the easy implementation and fast training of the algorithm, Naïve Bayes will serve as the baseline classifier.

Long Short Term Memory (LSTM)

LSTMs are becoming increasingly popular for sentiment classification. LSTMs are built on a recurrent neural network architecture (RNN). In an RNN the neurons are connected to themselves through time. As a result, the input from a time instance t_i will also be used as an input for the next time instance t_{i+1}. That leads to the problem of vanishing gradients, which means that it is hard for the model to learn long-term dependencies. This occurs because in a long sequence, like a sentence, as the loss gradients are backpropagated through the RNN, they may shrink to zero (Ribeiro, Tiels, Aguirre, & Schön, 2020). LSTMs are designed to overcome that problem. The LSTM architecture does so via its four constituents: A memory cell which can remember a lot of information from previous states, an input gate which controls the inputs into the neurons, an output gate with an activation function and lastly a forget gate which resets the neuron (Priyantina & Sarno, 2019). When training an LSTM, it is shown that using a pretrained Word2Vec embedding can help solving the curse of dimensionalty that might occur when using a one-hot encoded input (Xiao et al., 2018). Hence, a Word2Vec embedding layer is used in the LSTM.

Bidirectional Encoder Representations from Transformers (BERT)
BERT is a relatively new machine learning algorithm developed by Google in 2018 and mainly designed for natural language processing. BERT is pretrained on the English Wikipedia and BooksCorpus. Because of the pretraining users won't need as much computing power to achieve good results, even if the dataset is relatively small (Devlin et al., 2019). The BERT github page even states that "Most NLP researchers will never need to pre-train their own model from scratch" (Google Research, 2020).

Valence Aware Dictionary for sEntiment Reasoning (VADER)
Due to wide spread usage of lexicons, the VADER sentiment lexicon was also included. The results of VADER are intended for illustrative purposes only, as a lexicon based approach is not within the research scope of this thesis. The VADER lexicon does not have any hyperparameters to tune. It is only possible to optimize the classification results by changing the thresholds to classify the associated sentiment. VADER is specifically designed to classify social media sentiment and also includes emoticons, acronyms and *slang* words (Hutto & Gilbert, 2015).

3.1.4 *Evaluation Metrics for Sentiment Classification*

Typically, *accuracy, precision, recall and the F-score* are used as evaluation metrics to assess the performance of a sentiment analysis model.
Accuracy is the percentage of correctly predicted observations over all observations. However, accuracy should only be used if the classes in the data are balanced.
Precision expresses the proportion of how many classes were classified as positive, that actually are positive.
Recall refers to the percentage of total relevant results that were correctly classified. Hence, it is a good metric to see if the model was able to find all relevant instances in a dataset.
F-score is a metric that combines precision and recall (Garcia, 2020).

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

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

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

$$F_\beta = \frac{(1 + \beta^2) * (Precision * Recall)}{\beta^2 * Precision + Recall}$$

3.1.5 *Stock Price Prediction Models*

The next section explores the machine learning models that will be used to predict the stock price of GameStop.

ARIMA

One of the standard methods for time series forecasting is ARIMA. Therefore, it is commonly used in forecasting stock prices. However, it has some limitations, especially with regards to modelling the non-linear relationships between variables. As a result, some prerequisites are required to create good results (Siami-Namini, Tavakoli, & Siami-Namin, 2018). First of all, ARIMA does not work well with seasonal data. Seasonality can be identified by plotting the stock prices and plotting the Autocorrelation and Partial Autocorrelation. Furthermore, ARIMA only works on stationary time-series. Even though the time series can also be differenced, and hence be transformed from a non-stationary to a stationary, via the hyperparameter d it is better to difference the time series before applying the ARIMA model. As a result, the applied dickey fuller test can be used to test for stationarity (Jain & Mallick, 2017). If the null hypothesis can be rejected, the time series is stationary at a significance level.

LSTM without Sentiment Features

As stock prices are generally volatile, non-stationary and can have changes in their statistical properties it is beneficial to use models that can learn those attributes. Since LSTMs are able to capture such contextual information, they are shown to outperfrom many other methods on fincancial time series datasets (Preeti, Bala, & Singh, 2019). Furthermore, LSTMs can solve the vanishing gradient problem that oftentimes arises in normal RNNs. This thesis implements a Vanilla LSTM, which is a model with only a single LSTM layer. This implementation was chosen, because a more complex implementation, such as a stacked-LSTM, does not necessarily outperform a Vanilla LSTM for stock price prediction tasks (Hai et al., 2020).

LSTM with Sentiment Features
One advantage of LSTMs is that they can also include other features into their time-series prediction. When including other input features besides the stock price, the time series is considered a multivariate time series (Liu & Songcan, 2019).

3.1.6 *Evaluation Metrics for Stock Price Prediction*

Standard evaluation metrics for time-series data include MAE, MSE and RMSE (Rezaei, Faaljou, & Mansourfar, 2021). Mean Absolute Error (MAE) computes the absolute difference between the actual and predicted price. As a result, the MAE is on the same unit scale as the output value. Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) are commonly used to evaluate stock price prediction tasks. MSE represents the squared distance betwen the actual and predicted prices. Root Mean Squared Error (RMSE) simply takes the square-root of MSE. Hence, the output value is the same unit as the required output and can still represent the dispersion of results. In the formulas y_i represents the actual value and $\hat{y}_i$ the predicted value (Chen et al., 2021). Those metrics are defined as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

3.2 *Experimental Setup*

Explain Section a little bit.

3.2.1 *Data*

While Reddit does offer an official API, the API is most useful for streaming data. There are some strict limitations on accessing large amounts of historical data. As a result, the official API is not the best choice for this thesis. However, *pushshift.io* provides a solution for the strict limits. The FAQ on the pushshift subreddit states, that pushshift data is best used to:

- 'Analyze large quantities of Reddit data'
- 'Grab data for a specific date range in the past'
- 'Search for comments'
- 'Aggregate data'

Pushshift copies data from Reddit at the time it is posted. Since Pushshift uses the document-based database Elastic, it is extremely fast to query data (Brasetvik, 2015). However, currently Pushshift does not regularly update certain metadata, such as scores, edits to a submission's text or comments. Hence, there might be some minor inconsistencies of what is shown on Reddit and what is in the database. The scores, for example can easily be accessed via the official reddit API and, if needed, joined with the data obtained from Pushshift. Based on the data verification that was performed for this thesis, the number of comments only deviates by a marginally small amount. It is hypothesized that the small difference can be explained by forum moderators deleting spam. Those spam comments are assumed to not have a big impact the thesis anyways, which is why the small difference in the number of comments do not need to be addressed. To access Pushshift, this thesis uses an API wrapper called *PMAW*. Since requests are I/O-bound, PMAW is multithreaded. Hence, requests can be run asynchronously which allows the data to be loaded much faster (Podolak, 2021). When making the API request, the most important parameters are the following:

- subreddit: Name of the subreddit
- q: The search term based on which the subreddit is queried
- before: The starting date of the query
- after: The end date of the query

For this thesis all Gamestop (GME) related posts between January 1st, 2020 and October 26th, 2021 were requested for the subreddit WallStreetBets. The query returns 89 columns. Most of which, however, can be dropped since they either aren't useful for this thesis or contain no data. The most important columns are the number of comments, the title of the post and the content of the post. Emoticons are also included in the content text. In total 179,544 posts were obtained.
Of all obtained posts, 10% or 17,955 were manually labeled as either bearish, neutral or bullish. This was done by creating a graphical user interface that displayed the title and text of every tenth Reddit post. The relevant target sentiment was then assigned to each post by clicking a button by

either a mouse click or a keyboard shortcut. Of all manually labelled data 3479 (19.4%) are bearish, 5119 (28.5%) are neutral and 9357 (52.1%) are bullish.

3.2.2 *Active Learner Implementation*

To implement an Active Learner the *modAL* package was used. modAL was designed with modularity, flexibility and extensibility as high priorities (Danka & Horvath, 2018). The estimator defined in the Active Learner object is a Support Vector Machine (SVM). A SVM was chosen because of its strong generalization performance (Firmino, Baptista, Firmino, Oliveira, & Paiva, 2014). Additionally, SVMs can be used to solve both regression and classification problems. For the case at hand, the algorithm needs solve a classification problem, by optimally separating the data between bearish, neutral and bullish instances. Classification is done by finding a hyper-plane with the biggest margin, meaning it looks for the greatest distance to the nearest sample points (Jemai et al., 2021). SVMs use spatial transformations, commonly known as kernel functions, to fit the hyperplane. By doing so the data is projected into a higher dimensional space, which makes them easier to separate. Kernels can be linear, RBF or others. The radial basis function (RBF) kernel is best used for non-linear problems and is a general-purpose kernel that is often used in pattern recognition problems. The linear kernel, on the other hand, is typically used when there are only two classes present. A good example for that might be positive and negative sentiment (Firmino et al., 2014).

The initial seed data to train the SVM-estimator in the Active Learner was the data that was annotated manually. Since an SVM cannot handle text data, the data had to be preprocessed and converted to a tf-idf representation. Furthermore, the implementation of the Active Learner in this thesis deviates from the literature a little bit: The literature that was reviewed does not set aside a test set from the initial seed data and the accuracy of the Active Learner is evaluated on the entire seed data after every iteration. While the literature does not explain why this approach was taken, I hypothesize that is due to the cost associated with labeling the data. This thesis will not deviate from well established machine learning practices and therefore set aside 20% of the seed data as test data, which will be used to evaluate the performance of the Active Learner after every iteration (Raschka & Mirjalili, 2019, p. 196).

Uncertainty sampling was chosen as the query strategy because it has been demonstrated to be a strong baseline strategy. This query strategy assumes, that instances that are far from the decision boundary are adequately explainable and instances close to the decision boundary are uncertain. Naturally, this complements the SVM-estimator very well. As a result, the Active Learner queries the samples about which it is most uncertain about (Osborne & Baldridge, 2004). Two human oracles labelled an additional 10% of the data, which were chosen by the Active Learner. Those 10% were chosen by the Active Learner in ten iterations, meaning the Active Learner was retrained every time an additional 1% of the data was labelled. Even though modAL supports the retraining of the estimator based on the new instances only, the estimator was retrained on all data. This was decided because the training of the SVM estimator did not take too long and more training data usually leads to better results (Find citation).

3.2.3 *Data Preprocessing*

The research by Jemai et al. (2021) presents a system for structuring a sentiment analysis project, which was also used applied in this thesis. The data collection phase is the first step, where textual data is obtained from a source. The data is then cleaned in the second step, the data pre-processing phase. To do so, several actions need to be performed. One of them is tokenization. This is a natural language processing technique in which a large body of text is broken down into multiple sentences, each of which is then broken down into a list of words. Stop words such as is, the, a and other common words are also removed during the pre-processing phase. If stop words are included, they may play a negative role in sentiment classification and increase the overall vocabulary size while having little predictive power. (Zhao & Gui, 2017). In addition, special characters such as @ and urls should also be removed. It is also suggested that the text is converted to lowercase. As the final step, the research proposes lemmatization. By doing so, the structure of a word is analyzed and converted to its normalized form. The research conducted by Camacho-Collados and Pilehvar (2018) shows that lemmatization improves sentiment analysis results especially when using domain-specific datasets.

Since it is shown that having data with emoticons leads to more accurate results than data without emoticons, this thesis does not remove emoticons from the text corpus (Parveen & Pandey, 2016).

3.2.4 *Sentiment Analysis Models Implementation*

The next section explores how the machine learning models were implemented and how their optimal hyperparameters were determined. *Explain* for each model: Before training the models, 20% of the data were set aside as the test set. To account for class imbalances, stratification was applied. Also better explain why stratification, etc. See feedback prof.

Naïve Bayes (NB)
To train the classifier, the data was first converted to a tf-idf representation. By using tf-idf, a weight is assigned to each word. The tf is calculated by the number of times a word can be counted in a document. The idf adjusts the importance of the respective word (Guia, Silva, & Bernardino, 2019). The classifier uses five-fold gridsearch cross-validation to find the optimal parameters for *alpha* and *fit_prior*. Write better explanation for tf-idf.

Long Short Term Memory (LSTM)
Before training the LSTM, data it first is fed into a Word2Vec model to learn the word embeddings. As explained in the literature, this can enhance the performance of the model by learning the similarity between words. Selecting optimal parameters for the Word2Vec model is oftentimes neglected, even though that may lead to performance differences. The Word2Vec hyperparameters that were analyzed is the vector_size, min_count and window. The optimal respective hyperparameters are 50, 1 and 1. Those were determined based on a comparative intrinsic evaluation (Schnabel, Labutov, Mimno, & Joachims, 2015). The input and output dimensions of the Embedding layer of the LSTM, as well as the weights are taken from the word2vec model. The output dimensions of the embedding layer are also used as the units for the subsequent LSTM layer. Furthermore, the model adds a Dropout layer to improve generalization. To find the optimal dropout parameter and optimizer for the model, a loop runs through a set of parameters when fitting the model. The optimal model is determined by evaluating the performance on the validation set, which is 20% of the training data. The final Dense output layer uses softmax as its activation function. Furthermore, the model uses categorical crossentropy as its cost function and accuracy as its metric. The optimizer is also chosen based on the hyperparameters provided.

Bidirectional Encoder Representations from Transformers (BERT)
To train BERT, the token *[CLS]* and *[SEP]* are added to the beginning and end of the input sequence. Subsequently, the tokens are converted to IDs using the tokenization module from the bert library. Even though a maximum of 512 tokens can be used when training BERT, this implementation only uses a maximum of 100 tokens due to computational reasons. Furthermore, this implementation uses an Adam optimizer and a constant learning rate of 0.00001. Future work should include an optimization of those parameters. The model is trained with a batch size of two over three epochs. Write about hyperparameter tuning.

Valence Aware Dictionary for sEntiment Reasoning (VADER)
Vader can easily be implemented, by looking up the associated score of words in the lexicon. Typical threshold values for the obtained compound score are as follows: If the score is greater than or equal to 0.05, it is associated with positive sentiment. If the score is smaller than or equal to -0.05 it is associated with negative sentiment. For scores in between, neutral sentiment is assigned (Hutto & Gilbert, 2015). To obtain optimal evaluation metrics, however, the thresholds 0.03, 0.05, 0.07 and 0.10 and their negative counterparts were also analyzed.

3.2.5 *Stock Price Prediction Implementation*

When splitting time series data, it needs to be split into windows. The training set consists of the first 80% of the timeseries data and the test set of the remainding 20% of the data. To select the optimal hyperparameters, a rolling forecast is implemented. By doing so, the model sequentially adds the most recent observation and rebuilds the model (Siami-Namini et al., 2018).

ARIMA
For the case at hand, the p-value is almost 0, which indicates statistical significane at the 1% level. The stock price data does not include a seasonality pattern and the applied dickey-fuller test shows stationarity after differencing the time series data once. The hyperparameters that can be optimized are the number of lag observations included in the modle (p), the number of times the data needs to be differenced (d) and the size of the moving average window (q) (Siami-Namini et al., 2018). As a result the following hyperparameters were used: (5,0,0), (1,0,1), (2,0,2), (0,0,5), (1,0,0), (0,0,1), (0,0,0).

LSTM without Sentiment Features
text

LSTM with Sentiment Features
text

3.3 *Data, Code and Ethics Statements*

To query the data from pushshift, the explanation of pmaw API wrapper from Github was used: `https://github.com/mattpodolak/pmaw`
The data was manually verified, by comparing specific, randomly-sampled,

instances with the actual posts on reddit.

To label the initial train set, I created a graphical user interface using tkinter: `https://docs.python.org/3/library/tkinter.html`

To create the t-sne visualization, I relied on the documentation provided by Yellowbrick: `https://www.scikit-yb.org/en/latest/api/text/tsne.html`

To implement the Active Learner, I used modAL: `https://github.com/modAL-python/modAL`

All scikitlearn packages and classes, such as train_test_split, TfidfVecotricer, LabelEncoder, GridSearchCV, Pipeline, SVM and NB were implemented by utilizing material provided during the Machine Learning course at Tilburg University, taught by Dr. Güven and Dr. Önal.

The LSTM was implemented by using material provided during the Deep Learning course at Tilburg University, taught by Dr. Vanmassenhove and Dr. Saygili.

To implement BERT the following tutorial was used: `https://www.kaggle.com/xhlulu/disaster-nlp-keras-bert-using-tfhub/notebook`

The code for this thesis is shared in the following github repository: `https://github.com/StefanWinterToo/Master-Thesis`

All graphics used in this thesis were created by myself.

Due to time constraints, this first-submission does not use the full set of all possible hyperparameters and trains the LSTM and BERT only on a subset of the data. Furthermore, for the Active Learner the VADER sentiment lexicon was used as an oracle.

To the best of my knowledge, the literature used was referenced appropriately.

4 RESULTS

Figure 3 shows the accuracy of the Active Learner. As can be seen in the graph, the performance of the Active Learner slightly decreases over time. This is due to the oracle not doing a good job at labeling queried instances.

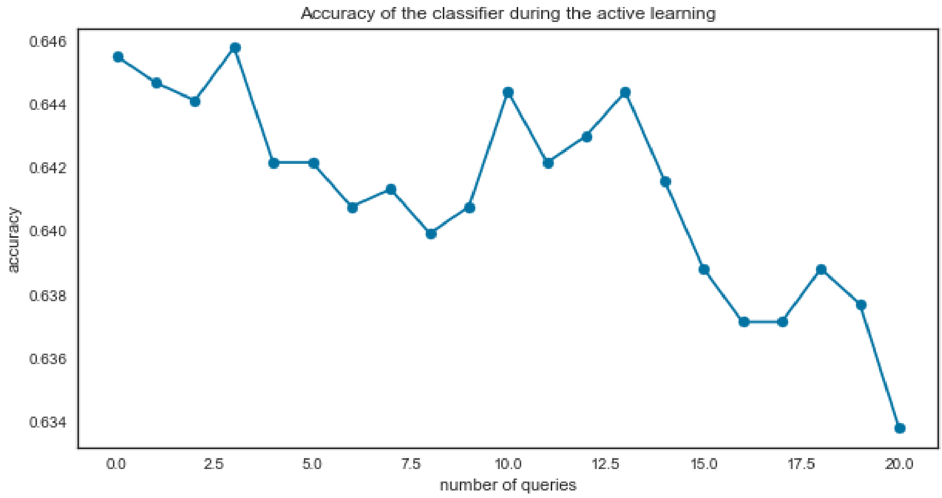

Figure 3: Accuracy of Active Learner over 20 iterations

Table 1: Test scores for NB, LSTM and BERT

	Evaluation Metrics			
Model	Accuracy	Precision	Recall	F_1-Score
NB	**0.79**	**0.74**	**0.82**	**0.76**
LSTM	0.50	0.53	0.53	0.53
BERT	0.52	0.17	0.33	0.23
VADER	0.39	0.38	0.39	0.38

To determine which model delivered the best comparative performance, they were evaluated based on metrics outlined in Table 1. Since lexicons are widely used in the literature, the VADER sentiment lexicon was also included in the results. Based on the evaluation metrics in Table 1, the NB baseline model outperformed other more complex models. Interestingly, BERT even underperformed the lexicon approach on most metrics. However, it can be seen that a general purpose lexicon vastly underperforms other sentiment models.

The accuracy of the LSTM on the validation set is highest after 10 epochs, while the loss starts to increase by a lot. As a result, the weights and other parameters seem to be optimal at epoch 10, which is why they will be used for the final model.

Since BERT is already pre-trained, it does not need as many epochs to optimize a model. As can be seen in figure 6, there is almost no change to the accuracy, meaning the model does not benefit from training over more epochs.

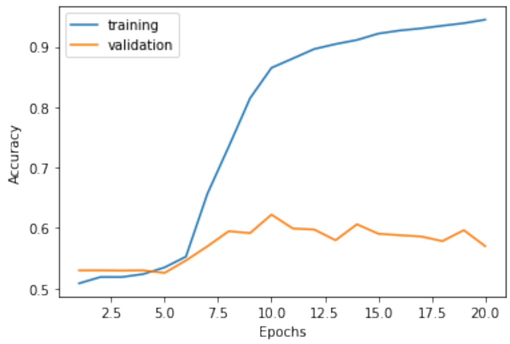

Figure 4: Accuracy of LSTM over epochs

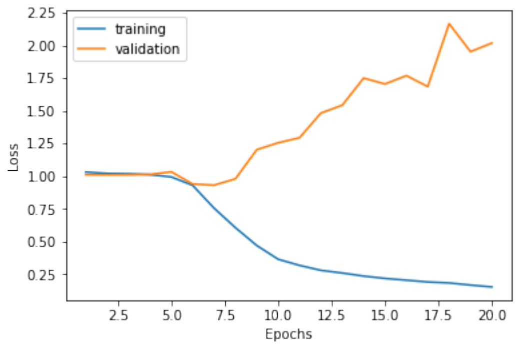

Figure 5: Loss of LSTM over epochs

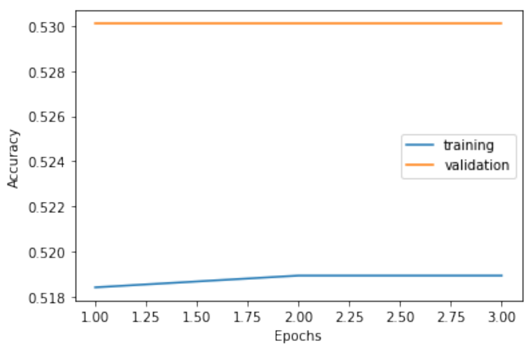

Figure 6: Accuracy of BERT over epochs

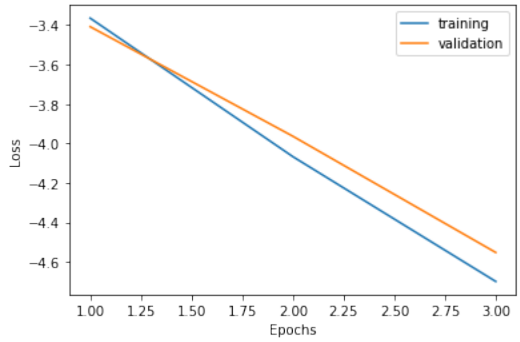

Figure 7: Loss of BERT over epochs

5 DISCUSSION

The 'to the moon' WallStreetBets movement had a tremendous impact on the lives of individuals, both to the positive and negative. Besides that, however, many investment funds have also been negatively impacted by the recent short-squeezes. While it might seem noble to root for individuals who try to force large funds out of their positions at big losses, it is easy to forget that many of those funds manage money for charitable endowments, pensions and others. Furthermore, such disruptions to the financial markets can harm its stability, thus causing spillover effects which can also negatively impact the lives of many people (Lyócsa et al., 2021). By being able to accurately measure and monitor the sentiment on WallStreetBets, market participants and regulators are able to preemptively take measures.

However, since the wallstreetbets subreddit has become very popular just recently, there is little academic research about the impact of the community on financial markets so far. Even though there is some research about sentiment analysis on wallstreetbets, that research does not use state of the art algorithms to perform sentiment analysis. This thesis shows that the wide-spread use of lexicons is not the best way to monitor sentiment and the adaption of better algorithms is urgently needed.

Not only did this thesis compare the performance of different models, but also proposed a highly efficient and reliable way to create a domain-specific annotated corpus, which can be used as the input to aforementioned models. To my knowledge, this thesis is the first research that creates a domain-specific corpus for the WallStreetBets forum. Researchers, such as Talamás (2021), specifically propose future work on "inclusion of features derived from alternative manipulation of the data like sentiment analysis could lead to new insights". I strongly believe that the methods proposed in my thesis lead to better sentiment classifiers, which can then be used in other scientific or industrial applications.

6 CONCLUSION

This thesis proposes the use of an Active Learner to drastically reduce the total cost of annotation. As a result, it becomes more feasible to create a fully labeled domain-specific dataset. Once a fully labeled dataset is obtained, it can be used in supervised learning algorithms. The results show that using state of the art models underperform the simple baseline NB classifier.

REFERENCES

Abbas, M., Ali, K., Memon, S., Jamali, A., Memon, S., & Ahmed, A. (2019, 03). *Multinomial naive bayes classification model for sentiment analysis.* doi: 10.13140/RG.2.2.30021.40169

Anand, A., & Pathak, J. (2021). Wallstreetbets against wall street: The role of reddit in the gamestop short squeeze. *Indian Institute of Management Bangalore Research Paper Series*.

Antweiler, W., & Frank, M. (2004, 02). Is all that talk just noise? the information content of internet stock message boards. *Journal of Finance*, *59*, 1259-1294. doi: 10.2139/ssrn.282320

Arora, S., Nyberg, E., & Rose, C. (2009, 01). Estimating annotation cost for active learning in a multi-annotator environment. *HLT-NAACL*. doi: 10.3115/1564131.1564136

Baldridge, J., & Osborne, M. (2004, jul). Active learning and the total cost of annotation. In *Proceedings of the 2004 conference on empirical methods in natural language processing* (p. 9-16). Barcelona, Spain: Association for Computational Linguistics. Retrieved from https://aclanthology.org/W04-3202

Binu, M., & Sony, G. (2020). Dimensionality reduction and visualisation of hyperspectral ink data using t-sne. *Forensic Science International*, *311*, 110194. Retrieved from https://www.sciencedirect.com/science/article/pii/S0379073820300566 doi: https://doi.org/10.1016/j.forsciint.2020.110194

Brasetvik, A. (2015, February 15). *Uses of elasticsearch, and things to learn.* Web. Retrieved from https://www.elastic.co/blog/found-uses-of-elasticsearch

Caginalp, G., & Constantine, G. M. (1995). Statistical inference and modelling of momentum in stock prices. *Applied Mathematical Finance* *2*, 225-242.

Camacho-Collados, J., & Pilehvar, M. T. (2018). *On the role of text preprocessing in neural network architectures: An evaluation study on text categorization and sentiment analysis.*

Chen, W., Zhang, H., Mehlawat, M. K., & Jia, L. (2021). Mean-variance portfolio optimization using machine learning-based stock price prediction. *Applied Soft Computing*, *100*, 106943. Retrieved from https://www.sciencedirect.com/science/article/pii/S1568494620308814 doi: https://doi.org/10.1016/j.asoc.2020.106943

Danbolt, J., Siganos, A., & Vagenas-Nanos, E. (2015). Investor sentiment and bidder announcement abnormal returns. *Journal of Corporate Finance*, 164-179.

Danka, T., & Horvath, P. (2018). *modal: A modular active learning framework for python.*

Das, S. R., & Chen, M. Y. (2007). Yahoo! for amazon: Sentiment extraction from small talk on the web. *Management Science*, 1375-1388.

De Gooijer, J. G., & Hyndman, R. (2006). 25 years of time series forecasting. *International Journal of Forecasting*, *22*, 443-473. doi: 10.1016/j.ijforecast.2006.01.001

Deng, J., Dong, W., Socher, R., Li, L.-J., Kai, L., & Li, F.-F. (2009). Imagenet: A large-scale hierarchical image database. In *2009 ieee conference on computer vision and pattern recognition* (p. 248-255). doi: 10.1109/CVPR.2009.5206848

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *Bert: Pre-training of deep bidirectional transformers for language understanding.*

Diangson, B., & Jung, N. (2021). *Bet it on reddit: The effects of reddit chatter on highly shorted stocks.*

Du, C., Sun, H., Wang, J., Qi, Q., & Liao, J. (2020). Adversarial and domain-aware bert for cross-domain sentiment analysis. In *Acl.*

Feng, X., & Johansson, A. C. (2019). Top executives on social media and information in the capital market: Evidence from china. *Journal of Corporate Finance*, *58*, 824-857. Retrieved from https://www.sciencedirect.com/science/article/pii/S0929119918303225 doi: https://doi.org/10.1016/j.jcorpfin.2019.04.009

Firmino, A., Baptista, C., Firmino, A., Oliveira, M., & Paiva, A. (2014). A comparison of svm versus naive-bayes techniques for sentiment analysis in tweets: A case study with the 2013 fifa confederations cup. In *Proceedings of the 20th brazilian symposium on multimedia and the web* (p. 123–130). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi-org.tilburguniversity.idm.oclc.org/10.1145/2664551.2664561 doi: 10.1145/2664551.2664561

Gennaro, G., Buonanno, A., & Palmieri, F. (2021, 11). Considerations about learning word2vec. *The Journal of Supercomputing*, *77*. doi: 10.1007/s11227-021-03743-2

Google Research. (2020, March 11). *bert.* Retrieved from https://github.com/google-research/bert

Gu, C., & Kurov, A. (2020). Informational role of social media: Evidence from twitter sentiment. *Journal of Banking and Finance*, *121*, 105969. Retrieved from https://www.sciencedirect.com/science/article/pii/S0378426620302314 doi: https://doi.org/10.1016/j.jbankfin.2020.105969

Guia, M., Silva, R. R., & Bernardino, J. (2019). Comparison of naive bayes, support vector machine, decision trees and random forest on sentiment analysis. In *Kdir.*

Hai, P. N., Tien, N. M., Hieu, H. T., Chung, P. Q., Son, N. T., Ha, P. N., & Son, N. T. (2020). An empirical research on the effectiveness of different lstm architectures on vietnamese stock market. In *2020 international conference on control, robotics and intelligent system* (p. 144-149). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3437802.3437827 doi: 10.1145/3437802.3437827

Hochreiter, S., & Schmidhuber, J. (1997, 12). Long short-term memory. *Neural computation*, *9*, 1735-80. doi: 10.1162/neco.1997.9.8.1735

Hutto, C., & Gilbert, E. (2015, 01). Vader: A parsimonious rule-based model for sentiment analysis of social media text..

Ivanovic, Z., Bogdan, S., & Baresa, S. (2013, 06). Forecasting croatian stock market index: Crobex. *UTMS Journal of Economics*, *4*(2), 79-91. Retrieved from https://www.proquest.com/scholarly-journals/forecasting-croatian-stock-market-index-crobex/docview/1399281625/se-2

Jain, G., & Mallick, B. (2017). A study of time series models arima and ets.

Jemai, F., Hayouni, M., & Baccar, S. (2021). Sentiment analysis using machine learning algorithms. *International Wireless Communications and Mobile Computing*, 775-779.

Jin, Z., Yang, Y., & Liu, Y. (2020). Stock closing price prediction based on sentiment analysis and lstm. *Neural Computing and Applications*, *32*. doi: 10.1007/s00521-019-04504-2

Jin-Dong, K., Tomoko, O., & Junichi, T. (2008, 02). Corpus annotation for mining biomedical events from lterature. *BMC bioinformatics*, *9*, 10. doi: 10.1186/1471-2105-9-10

Kang, J., Ryu, K. R., & Kwon, H. (2004). Using cluster-based sampling to select initial training set for active learning in text classification. In *Pakdd.*

Kotelnikova, A., Paschenko, D., Bochenina, K., & Kotelnikov, E. (2021). *Lexicon-based methods vs. bert for text sentiment analysis.*

Liu, J., & Songcan, C. (2019). Non-stationary multivariate time series prediction with selective recurrent neural networks. In A. C. Nayak & A. Sharma (Eds.), *Pricai 2019: Trends in artificial intelligence* (p. 636-649). Cham: Springer International Publishing.

Long, C., Lucey, B. M., & Yarovaya, L. (2021). 'i just like the stock' versus 'fear and loathing on main street': The role of reddit sentiment in the gamestop short squeeze. *SSRN Electronic Journal*.

Lu, J., Henchion, M., & Namee, B. M. (2019). *Investigating the effectiveness of representations based on word-embeddings in active learning for labelling text datasets.*

Lyócsa, S., Baumöhl, E., & Vyrost, T. (2021). Yolo trading: Riding with the herd during the gamestop episode. *Finance Research Letters*.

Miller, B., Linder, F., & Mebane, W. R. (2020). Active learning approaches for labeling text: Review and assessment of the performance of active learning approaches. *Political Analysis*, *28*(4), 532–551. doi: 10.1017/pan.2020.4

Muhammad, A. (2014, 05). Detection and scoring of internet slangs for sentiment analysis using sentiwordnet. *Life Science Journal*, *11*, 66-72. doi: 10.6084/M9.FIGSHARE.1609621

Namcheol, J., & Ghang, L. (2019, 04). Automated classification of building information modeling (bim) case studies by bim use based on natural language processing (nlp) and unsupervised learning. *Advanced Engineering Informatics*, *41*. doi: 10.1016/j.aei.2019.04.007

Nikou, M., Mansourfar, G., & Bagherzadeh, J. (2019, 12). Stock price prediction using deep learning algorithm and its comparison with machine learning algorithms. *Intelligent Systems in Accounting, Finance and Management*, *26*. doi: 10.1002/isaf.1459

Osborne, M., & Baldridge, J. (2004, 01). Ensemblebased active learning for parse selection. In (p. 89-96).

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, *2*, 1-135. Retrieved from https://doi.org/10.1561/1500000011 doi: 10.1561/1500000011

Park, S., Lee, W., & Moon, I.-C. (2015). Efficient extraction of domain specific sentiment lexicon with active learning. *Pattern Recognition Letters*, *56*, 38-44.

Parveen, H., & Pandey, S. (2016, 01). Sentiment analysis on twitter dataset using naive bayes algorithm. In (p. 416-419). doi: 10.1109/ICATCCT.2016.7912034

Piñeiro-Chousa, J., Vizcaino-Gonzalez, M., & Perez-Pico, A. M. (2017). Influence of social media over the stock market. *Psychology and*

Marketing, *34*, 101-108. doi: 10.1002/mar.20976
Podolak, M. (2021, October 1). *Pmaw: Pushshift multithread api wrapper.* Web. Retrieved from https://github.com/mattpodolak/pmaw
Preeti, Bala, R., & Singh, R. P. (2019). Financial and non-stationary time series forecasting using lstm recurrent neural network for short and long horizon. In *2019 10th international conference on computing, communication and networking technologies (icccnt)* (p. 1-7). doi: 10.1109/ICCCNT45670.2019.8944624
Priyantina, R., & Sarno, R. (2019, 06). Sentiment analysis of hotel reviews using latent dirichlet allocation, semantic similarity and lstm. *International Journal of Intelligent Engineering and Systems*, *12*, 142-155. doi: 10.22266/ijies2019.0831.14
Rammurthy, S. K., & Patil, S. B. (2021). An lstm-based approach to predict stock price movement for it sector companies. *International Journal of Cognitive Informatics and Natural Intelligence*, *15*, 1-12.
Raschka, S., & Mirjalili, V. (2019). *Python machine learning*. Packt Publishing.
Rezaei, H., Faaljou, H., & Mansourfar, G. (2021). Stock price prediction using deep learning and frequency decomposition. *Expert Systems with Applications*, *169*, 114332. Retrieved from https://www.sciencedirect.com/science/article/pii/S0957417420310228 doi: https://doi.org/10.1016/j.eswa.2020.114332
Ribeiro, A. H., Tiels, K., Aguirre, L. A., & Schön, T. (2020, 26–28 Aug). Beyond exploding and vanishing gradients: analysing rnn training using attractors and smoothness. In S. Chiappa & R. Calandra (Eds.), *Proceedings of the twenty third international conference on artificial intelligence and statistics* (Vol. 108, pp. 2370–2380). PMLR. Retrieved from https://proceedings.mlr.press/v108/ribeiro20a.html
Salah, R., & Gayar, N. E. (2019). *Sentiment analysis using unlabeled email data.* EasyChair Preprint no. 2080.
Sazzed, S., & Jayarathna, S. (2021). Ssentia: A self-supervised sentiment analyzer for classification from unlabeled data. *Machine Learning with Applications*, *4*, 100026. Retrieved from https://www.sciencedirect.com/science/article/pii/S2666827021000074 doi: https://doi.org/10.1016/j.mlwa.2021.100026
Schnabel, T., Labutov, I., Mimno, D., & Joachims, T. (2015, 01). Evaluation methods for unsupervised word embeddings. In (p. 298-307). doi: 10.18653/v1/D15-1036
Schöneburg, E. (1990). Stock price prediction using neural networks: A project report. *Neurocomputing*, *2*(1), 17-27. Retrieved from https://www.sciencedirect.com/science/article/pii/092523129090013H doi: https://doi.org/10.1016/0925-2312(90)

90013-H

Semenova, V., & Winkler, J. (2021). *Reddit's self-organised bull runs: Social contagion and asset prices.*

Settles, B. (2009). Active learning literature survey..

Sharif, O., Hoque, M. M., & Hossain, E. (2019). Sentiment analysis of bengali texts on online restaurant reviews using multinomial naïve bayes. In *2019 1st international conference on advances in science, engineering and robotics technology (icasert)* (p. 1-6). doi: 10.1109/ICASERT.2019.8934655

Shetty, D. K., & Ismail, B. (2021). Forecasting stock prices using hybrid non-stationary time series model with ernn. *Communications in Statistics - Simulation and Computation*, *0*(0), 1-13. Retrieved from https://doi.org/10.1080/03610918.2021.1872631 doi: 10.1080/03610918.2021.1872631

Siami-Namini, S., Tavakoli, N., & Siami-Namin, A. (2018). A comparison of arima and lstm in forecasting time series. In *2018 17th ieee international conference on machine learning and applications (icmla)* (p. 1394-1401). doi: 10.1109/ICMLA.2018.00227

Song, J., Kim, K., Lee, B., Kim, S., & Youn, H. Y. (2017). A novel classification approach based on naïve bayes for twitter sentiment analysis. *KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS*, 2996-3011.

Susanti, A., Djatna, T., & Kusuma, W. (2017, 09). Twitter's sentiment analysis on gsm services using multinomial naïve bayes. *Telkomnika (Telecommunication Computing Electronics and Control)*, *15*, 1354-1361. doi: 10.12928/TELKOMNIKA.v15i3.4284

Talamás, J. (2021, 05). *Social media effects on the market: Reddit data analysis on stocks.* doi: 10.13140/RG.2.2.24180.88960

Tomasev, N., Radovanovic, M., Mladenic, D., & Ivanovic, M. (2014). The role of hubness in clustering high-dimensional data. *IEEE Transactions on Knowledge and Data Engineering*, *26*(3), 739-751. doi: 10.1109/TKDE.2013.25

Vuong, P. H., Dat, T. T., Mai, T. K., Uyen, P. H., & Bao, P. T. (2021). Stock-price forecasting based on xgboost and lstm. *Computer Systems Science and Engineering*, *40*, 237-246. Retrieved from http://www.techscience.com/csse/v40n1/44219 doi: 10.32604/csse.2022.017685

Wang, H., Guo, Z., & Chen, L. (2019). Financial forecasting based on lstm and text emotional features. In *2019 ieee 8th joint international information technology and artificial intelligence conference (itaic)* (p. 1427-1430). doi: 10.1109/ITAIC.2019.8785505

Wang, S., & Manning, C. (2012, 07). Baselines and bigrams: Simple, good sentiment and topic classification. In (p. 90-94).

Xianghua, F., Jingying, Y., Jainqiang, L., Min, F., & Huihui, W. (2018). Lexicon enhanced lstm with attention for general sentiment analysis. *IEEE Access*, 71884-71891.

Xiao, L., Wang, G., & Zuo, Y. (2018). Research on patent text classification based on word2vec and lstm. In *2018 11th international symposium on computational intelligence and design (iscid)* (Vol. 01, p. 71-74). doi: 10.1109/ISCID.2018.00023

Yanyan, W., Fulian, Y., Jianbo, L., & Marco, T. (2020, 08). Automatic construction of domain sentiment lexicon for semantic disambiguation. *Multimedia Tools and Applications*, *79*. doi: 10.1007/s11042-020-09030-1

Yue, L., Malu, C., Umeshwar, D., & ChengXiang, Z. (2011). Automatic construction of a context-aware sentiment lexicon: An optimization approach. In *Proceedings of the 20th international conference on world wide web* (p. 347-356). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi-org.tilburguniversity.idm.oclc.org/10.1145/1963405.1963456 doi: 10.1145/1963405.1963456

Zaghum, U., Mariya, G., Imran, Y., & Shoaib, A. (2021). A tale of company fundamentals vs sentiment driven pricing: The case of gamestop. *Journal of Behavioral and Experimental Finance*.

Zhao, J., & Gui, X. (2017, 02). Comparison research on text pre-processing methods on twitter sentiment analysis. *IEEE Access*, *PP*, 1-1. doi: 10.1109/ACCESS.2017.2672677

Zhengqi, P., Zhewei, S., & Yang, X. (2019, November). Slang detection and identification. In *Proceedings of the 23rd conference on computational natural language learning (conll)* (pp. 881–889). Hong Kong, China: Association for Computational Linguistics. Retrieved from https://aclanthology.org/K19-1082 doi: 10.18653/v1/K19-1082