



# MEASURING SENTIMENT

THIS IS THE SUBTITLE

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 1, 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 SENTIMENT

THIS IS THE SUBTITLE

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 this phenomenon, there is also almost no research available on that topic. This thesis will explore methodologies on how to best measure sentiment of the discussion board. One of the challenges when measuring 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). However, it was not the rapid price appreciation that amazed market participants. 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. According to studies, social media sentiment has a particularly strong impact on uninformed traders (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).

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).

Hence, this thesis will try to answer the following Research Question:

*How can sentiment analysis best be performed on the WallStreetBets Reddit-forum?*

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 best 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 *Which sentiment analysis approach performs best on predefined key performance indicators?*

## 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) did a study on the Yahoo! message board, which was amongst the first ones on the internet for investors to exchange ideas. Lyócsa, Baumöhl, and Vydrost (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.

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.

However, since the WallStreetBets ‘*meme-stock movement*’ is a relatively recent phenomenon, there is very little research on the impact of WallStreet-

Bets on individual stocks, especially with regards to sentiment analysis. Additionally, of all the published research none account for the domain-specific language used on the forum. Because of the frequent usage of terminology that is specific to WallStreetBets, this can lead to incorrect conclusions.

Of course, this also applies to research in other fields, which usually also use a general-purpose sentiment lexicon, because of the cost associated with building a domain-specific one. However, it has been demonstrated that using a domain-specific knowledge base results in more accurate sentiment analysis results (Park et al., 2015). It is argued that there is no general-purpose sentiment lexicon that can be optimally applied on all domains. In different domains, some terms can have completely different meanings. A good example is the word “unpredictable”, which would have negative sentiment for electronics but can be a positive label for movies. It has been demonstrated that by adapting sentiment lexicons to a certain domain performance for sentiment classification can be enhanced (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 also shown stronger performance (Yanyan, Fulian, Jianbo, & Marco, 2020).

For this reason some research deviates from lexicon-based approaches. Instead, they examine how deep 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 (sentence-level detection), but also identify the exact position of the term in the sentence (token-level identification). 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. 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, Zhewei, & Yang, 2019). Those out-of-vocabulary tokens can be learned in the hidden layers of LSTMs (Hochreiter & Schmidhuber, 1997). 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). 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 domain-specific labeled data.

Creating a domain-specific annotated corpus to train machine learning models, however, is not without its own 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).

If data are manually annotated at random, the annotator will invest a lot of time into labeling irrelevant instances. This may incur costs which could be avoided with an Active Learner. It is argued that Passive Learning, or randomly selecting instances to be labeled by an annotator, is especially costly 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).

### 3 METHOD

#### 3.1 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 WallStreet-Bets. 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 bearish, neutral or bullish.



### 3.2 Data Preprocessing

The research by Jemai et al. (2021) presents a system for structuring a sentiment analysis project. 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. Data tokenization is one of the actions. 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 not only may play a negative role in sentiment classification but also increase the overall vocabulary size (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 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). Eventually, term frequency-inverse document frequency (tf-idf), was applied on the text corpus. Using this representation allows the extraction of the most descriptive terms in a document and easy computations. However, it fails to capture semantics and word embeddings. For computational reasons, only words that occur at least five times were included in the tf-idf representation.

### 3.3 The Case for a Semi-Supervised Method over an Unsupervised Method

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 exploration 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

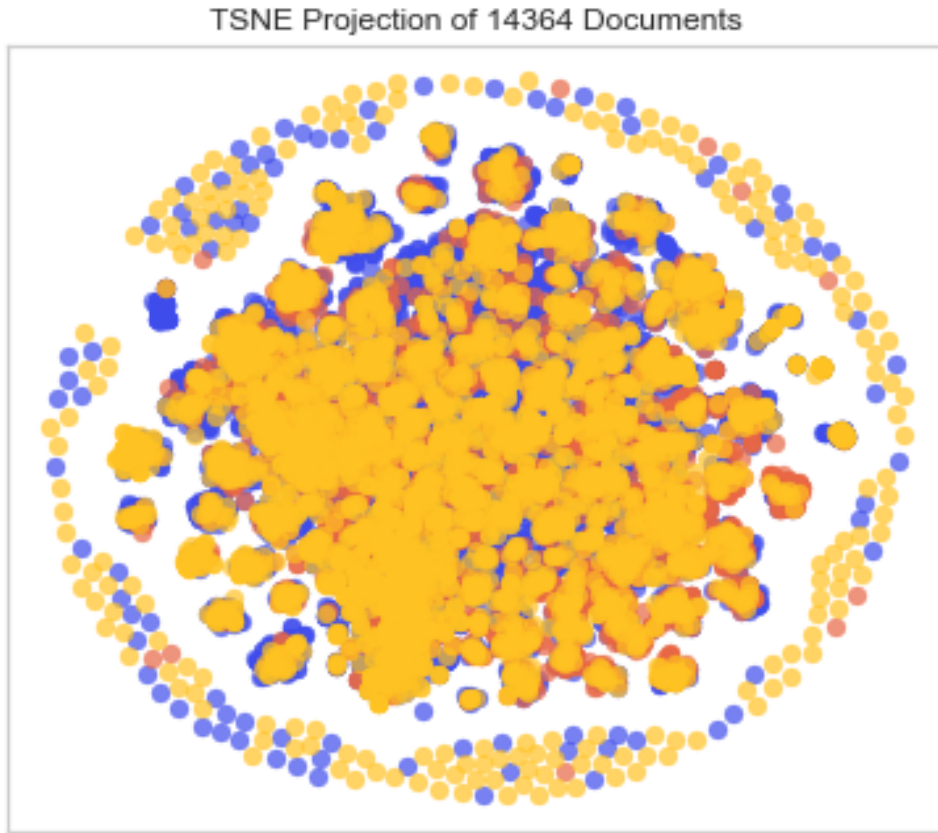


Figure 1: t-sne visualization of seed data

(Binu & Sony, 2020). As can be seen in the visualization below, admittedly at a low dimension, the majority of the data do not belong to any particular cluster.

Even though there are some approaches to clustering high dimensional data, it generally is difficult to do so. One of the explanations for that is the increased sparsity and the difficulty to distinguish between the distances of specific instances (Tomasev, Radovanovic, Mladenec, & Ivanovic, 2014).

Once labeled instances are obtained, supervised learning methods can be applied. One of the major disadvantages of supervised models, however, is the cost associated with manually labelling the data (Miller et al., 2020). This thesis proposes a method to create a labeled dataset for the fraction of the total annotation cost. As a result, a domain-specific labeled text corpus is created, which can be used to compare the performance of different supervised machine learning algorithms. The proposed methodology is an Active Learner. With its support, a complete domain-specific corpus can be labeled while only relying on partial annotations (Park et al., 2015).

### 3.4 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 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).

### 3.5 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 to solve a classification problem, by optimally separating the data between

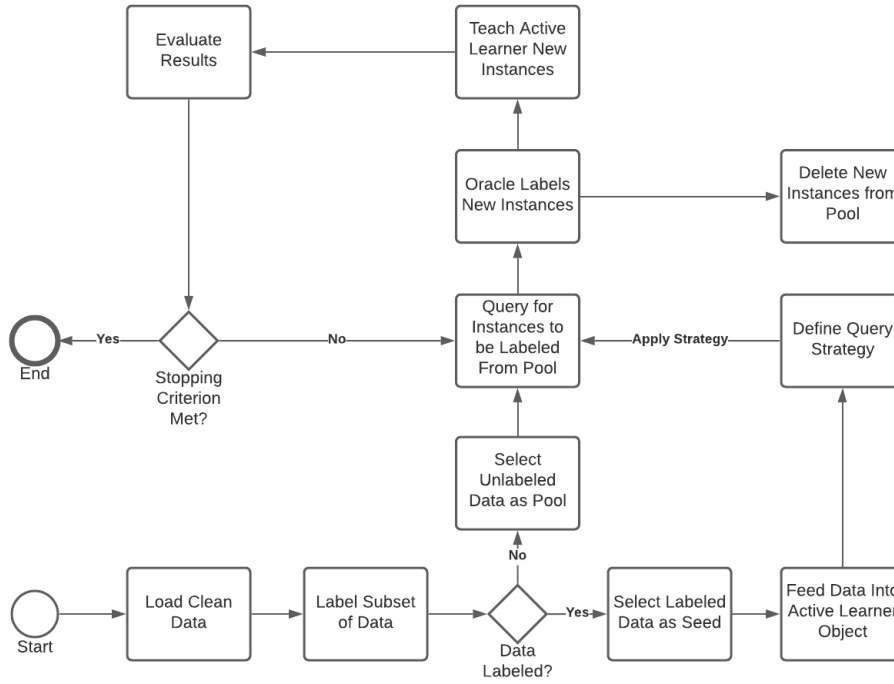


Figure 2: Visualized Workflow of an Active Learner. Created with lucid.app

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. 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 had to be labeled manually. To do so, every tenth instance was annotated. 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. *Insert citation from my ML book*

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 & Baldrige, 2004).

### 3.6 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.

Because a SVM was used in the Active Learner to label the ground truth data, it will not be applied in the sentiment classification task. Otherwise, the results might lack robustness and be biased.

Before training the models, 20% of the data were set aside as the test set. To account for class imbalances, stratification was applied.

#### 3.6.1 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 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.

#### 3.6.2 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. 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).

### Implementation

To feed the data into the LSTM, it first is converted from a tf-idf representation into a one-hot encoded array with three dimensions. However, one-hot encoding fails to incorporate similarity between words, which is why an Embedding layer was added to the LSTM. The layer takes the number of words in the text corpus as input. The output dimensions which will be used in the subsequent LSTM layer are a hyperparameter and chosen by evaluating the performance on the validation set, which is 20

#### 3.6.3 *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, Chang, Lee, & Toutanova, 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)

#### 3.7 *Hyperparameter Tuning*

In order to find the best performing models, some hyperparameter tuning steps were taken. For the implementation of the Naïve model, five-fold grid search cross-validation was used to find the best parameters in a pre-defined parameter grid. For the deep learning models, LSTM and BERT, a loop was created that iterates over a set of parameters. Within each iteration, the model is fit on the training data while setting aside 20% of the data for validation. The best performing model and its respective parameters will then be used to classify the test data.

#### 3.8 *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, TfidfVecotrizer, 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://skimai.com/fine-tuning-bert-for-sentiment-analysis/>

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

Note: Currently the code mainly consists of notebooks. However, before the final submission, I will properly structure the code.

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. 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.



Table 1: Best scoring models classifying bots, on Twitter and Facebook respectively.  $F_1$  scores report positive (bot) class. Outline text left (l) and numbers right (r).

PCA	Models	$F_1$ score	
		Twitter	Facebook
300	Linear SVM ( $C = 0.1$ )	0.51	<b>0.91</b>
	Random Forest ( $S = 5, F = 5$ )	0.71	0.85
	Naive Bayes	0.61	0.73
500	Linear SVM ( $C = 0.1$ )	0.55	0.84
	Random Forest ( $S = 5, F = 5$ )	<b>0.76</b>	0.71
	Naive Bayes	0.41	0.64
	Majority	0.50	0.60

#### 4 RESULTS

I didn't add my results yet. I will make a table as below and also add some graphics.

You have results and want to show them — probably in a table of some kind as you can see in Table 1. Highlight important scores with `\textbf{}`, use booktabs commands for structure: `\toprule \midrule \bottomrule`. APA does not allow vertical lines.

#### 5 CONCLUSION

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 not only



tries to shine some light on those new and influential market participants, but also tries to put forward some methods that work best to perform sentiment analysis on the forum.

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 can lead to better sentiment classifiers, which can then be used in other scientific or industrial applications.

#### REFERENCES

- 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*.
- 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
- Baldrige, 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>
- Camacho-Collados, J., & Pilehvar, M. T. (2018). *On the role of text pre-processing in neural network architectures: An evaluation study on text categorization and sentiment analysis*.
- 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.
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Naacl*.
- Diangson, B., & Jung, N. (2021). *Bet it on reddit: The effects of reddit chatter on highly shorted stocks.*
- 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
- Google Research. (2020, March 11). *bert*. Retrieved from <https://github.com/google-research/bert>
- Hochreiter, S., & Schmidhuber, J. (1997, 12). Long short-term memory. *Neural computation*, 9, 1735-80. doi: 10.1162/neco.1997.9.8.1735
- Jemai, F., Hayouni, M., & Baccar, S. (2021). Sentiment analysis using machine learning algorithms. *International Wireless Communications and Mobile Computing*, 775-779.
- Jin-Dong, K., Tomoko, O., & Junichi, T. (2008, 02). Corpus annotation for mining biomedical events from literature. *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*.
- 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., Henschon, 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
- Osborne, M., & Baldridge, J. (2004, 01). Ensemblebased active learning for parse selection. In (p. 89-96).
- 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
- Podolak, M. (2021, October 1). *Pmaw: Pushshift multithread api wrapper*. Web. Retrieved from <https://github.com/mattpodolak/pmaw>
- 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
- Raschka, S., & Mirjalili, V. (2019). *Python machine learning*. Packt Publishing.
- 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>
- Semenova, V., & Winkler, J. (2021). *Reddit's self-organised bull runs: Social contagion and asset prices*.
- Settles, B. (2009). Active learning literature survey..
- 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.
- Tomasev, N., Radovanovic, M., Mladenec, 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
- Xianghua, F., Jingying, Y., Jainqiang, L., Min, F., & Huihui, W. (2018). Lexicon enhanced lstm with attention for general sentiment analysis. *IEEE Access*, 71884-71891.

- 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

#### APPENDIX A

If you have nothing to append: remove this. You can do a page referral for these, like: Appendix A (page 18).

#### APPENDIX B

And this!