Norwegian School of Economics
Bergen, Spring 2023

An applied study comparing a man-made lexicon, a machine learned lexicon, and OpenAI's GPT for sentiment analysis.

Markus Anton Alexandersen & Joakim Rutlin

Supervisor: Tommy Stamland

Master thesis, Economics and Business Administration

Major: Financial Economics

NORWEGIAN SCHOOL OF ECONOMICS

Acknowledgements

This thesis concludes our Master of Science degree in Economics and Business Administration, majoring in Financial Economics at the Norwegian School of Economics (NHH). The process of this thesis has been an incredible learning experience for us, especially given the relative novelty of textual analysis within the field of finance. As the use of AI and NLP models continue to expand, sentiment analysis in finance is poised to become an even more fascinating and fast-growing field. We are confident that this thesis will prove to be an engaging and informative read for those interested in this area of research.

Our sincerest thanks go out to Tommy Stamland for his sturdy guidance and swift responsiveness throughout the process. Further, we would like to extend our gratitude to our professors and teachers throughout the last years for enhancing our academic experience. Moreover, we would like to thank our family and friends for their perusal, feedback, and continuous support throughout the thesis.

Norwegian School of Economics

Bergen, Spring 2023

Markus A. Alexandersen

Joakim Rutlin

Abstract

Sentiment analysis, at scale, has become an essential tool in the methodological toolbox of finance. In this thesis, we construct a sentiment lexicon using a supervised machine learning model by Taddy (2013) and compare it to the traditional finance lexicon by Loughran and McDonald (2011). Additionally, a state-of-the-art AI natural language processing model from OpenAI's GPT family is introduced to challenge both of these classical lexical sentiment analysis approaches. Utilizing unbalanced panel data regressions, we compare the different approaches in a "horse race". First, we find that textual sentiment significantly explains stock returns. Secondly, we find that GPT outperforms both lexical approaches in terms of economic and statistical significance, with an $adjusted\ R^2$ of 3.9% versus 2.5% and 2.2% for the machine learned and Loughran and McDonald lexicon, respectively. Thirdly, we find that by fine-tuning GPT models for detecting sentiment, the performance increases significantly. Lastly, we find that the current optimal available GPT model for financial sentiment analysis in the GPT model library is *GPT-3.5-Turbo*.

Keywords – Textual Analysis, Measuring Sentiment, Machine Learning, GPT, OpenAI, NewsWeb

Contents

List of Figures

List of Tables

1 Introduction

Sentiment analysis in finance is a relatively new research topic, and many credits Tetlock's study from 2007 as the start of the "sentiment analysis in finance"-era. Tetlock performed a sentiment analysis applying a bag-of-words (BoW) approach on the daily stock market report in the Wall Street Journal to explain stock market returns. Such BoW approaches use words that express a feeling as indicators for measuring sentiment, e.g., "*good*", "*fast*", "*nice*", "*best*", or "*bad*", "*hate*", "*awful*", and "*poor*". A list of such words is called a sentiment dictionary/lexicon[1]. Such lexicon-based approaches use a pre-determined sentiment lexicon to score a document by aggregating the sentiment scores of all the words in a document (Kannan et al., 2016). Tetlock used the Harvard IV lexicon as his BoW to count words that appeared in both the dictionary and the articles he reviewed and then gave the article a sentiment score based on the number of matching words. A problem with using the Harvard IV dictionary was that it counted many finance-related words, such as "*tax*", "*cost*", "*capital*", "*board*", "*liability*", "*foreign*", and "*vice*", as negative words. When these common finance words are rated indiscriminately as "negative", it corrupts the analysis. To redeem this issue, Loughran and McDonald (LM) introduced their man-made financial dictionary[2] in 2011.

While a lexicon-based approach for measuring sentiment is a widely used and valid approach, it is essential to highlight that such lexicons have several issues (Liu, 2020). First, a positive or negative sentiment term can have opposite interpretations depending on context. For example, the word "*sick*" usually indicates negative sentiment, but a teenager may say: "*That movie was sick*", i.e., it was a good movie, which indicates positive sentiment. Secondly, a sentence can contain sentiment words without adding a meaningful sentiment to the sentence, like "*Did you have a good day?*". Thirdly, sarcasm cannot be detected by a simple BoW lexicon. This is especially problematic when applying lexicons on tweets or other social media posts. Lastly, many sentences do not have any sentiment words, but still, it could be a clear sentiment. E.g., "*This oven uses relatively much energy*". In addition to the issues highlighted by Liu (2020), Wang et al. (2020) criticize the LM lexicon as too sample specific. A lexicon depends on the characteristics of the text on which it is built. For instance, the LM lexicon is regarded as a finance lexicon, but there is a significant difference between the characteristics of 10-Ks[3], on

[1] The words dictionary and lexicon are used interchangeably in the thesis.
[2] We will refer to this lexicon as LM.
[3] A 10-K is a comprehensive annual report filed by a U.S.-based publicly traded company.

which it is built, and our data consisting of more general stock exchange announcements.

Several researchers have acknowledged these concerns and presented alternatives using machine learning (ML) approaches to classify sentiment. The use of alternative techniques is a critical element motivating our study. As an opponent to the "human" BoW approach, we use the multinomial inverse regression model (*MNIR*) by Taddy (2013) – a supervised ML model. This approach allows machine learning to determine a dictionary based on a quantitive analysis of the correlation between words and stock performance instead of applying a qualitative dictionary, manually created by researchers. The *MNIR* approach has been regarded as the "machine" in the "human versus machine" debate. Some of the last contributions to the debate are Loughran and McDonald (2020) (human) and Garcia et al. (2023) (machine).

To say that *MNIR* is a pure machine-based approach is somewhat misleading, as it needs to be supervised. We take this debate a step further by applying a state-of-the-art AI model to conduct the sentiment analysis – essentially a black box. We have applied Generative Pre-trained Transformer (GPT) models from the American artificial intelligence (AI) research laboratory OpenAI as a third approach. At this point, we are definitely considering humans versus machines. Whereas humans strictly supervise the *MNIR* model, and the machine is trained on a training sample, the *GPT-3.5-Turbo* model (one of the main GPT models used in this thesis) is an autoregressive language model[4] with **175 billion parameters** trained on **45 terabytes** of text scraped from the internet (Brown et al., 2020). We use the three (current) most advanced models offered by OpenAI, here ranked from most to least capable[5]:

1. *GPT-3.5-Turbo*
2. *Davinci*
3. *Curie*

[4] An autoregressive language model is a type of ML model that uses autoregressive techniques (uses past events to predict future events) to predict the next word in a sequence of words based on the words that have come before it (Deepchecks, 2023).

[5] We ask the reader to note that this field is moving at an incredible pace at the current time with constantly new models being tested – for example, *GPT-4*, which was beta-launched in March 2023.

These three very different approaches (LM, ML, and GPT) are the main inspiration for our thesis. A research question that started as a textual analysis of stock exchange announcements:

Does the sentiment of stock exchange announcements provide explanatory power to price changes?

Evolved into an analysis of three very different technological methods applied in financial sentiment analysis:

How do the different approaches fare, LM, ML, and GPT, compared to each other?

The latter question is in some fashions similar to Wang et al. (2020)'s paper, as they compare deep learning's accuracy to more traditional ML models' accuracy related to classifying sentiment. However, there is a fundamental difference in the analyses. We test the methods' capability to predict returns on stock exchange announcements. Their paper explores how accurate different models are in determining the sentiment of stock analyses from Seeking Alpha (Wang et al., 2020). This underlines a significant distinction in the data. Seeking Alpha reports are binary, long- or short recommendations and should appear reasonably clear to project the suitable sentiment. Our data is much more obscure regarding classifications, and we allow for a third option; neutral sentiment.

The thesis is structured as follows: Chapter 2 begins with a literature review of relevant research papers. The chapter also includes an "about" section, providing a brief insight into OpenAI and our text source – press releases from Oslo Børs' NewsWeb. Chapter 3 introduces the textual and numerical data and how and why it is retrieved. Then, Chapter 4 presents the methodology, which is quite extensive given that we apply three very different approaches. The chapter starts with pre-processing the texts from NewsWeb and extends into the calculation of expected returns and sentiment classification. Then, we go through the different methodologies relating to LM, ML, and GPT. Finally, we cover the empirical design and how we compare the different methods with each other. In Chapter 5, we perform the analysis, inspect the created ML dictionary, and how the different approaches perform compared with each other. Chapter 6 concludes our thesis, and we present some follow-up research subjects.

2 Background and literature review

This section presents a comprehensive overview of the existing research related to textual analysis, applying LM, ML, and GPT. It moves on to introduce the concepts of machine learning and artificial intelligence. Then, we present the approaches we will utilize in our thesis, *MNIR* and GPT. Additionally, we provide an in-depth discussion of NewsWeb, including the nature of its operations and the types of press releases published on the platform.

2.1 Sentiment analysis

Sentiment analysis is a relatively new research area that spun out of social media and the web. It can be defined as the computational study of people's opinions, sentiments, appraisals, attitudes, and emotions toward entities and their attributes expressed in written text (Liu, 2020). Regarding our case, these entities are publicly traded Norwegian firms and their disclosures and press releases that are published on Oslo Børs' messaging service – NewsWeb. In a nutshell, sentiment analysis aims to identify positive, negative, and neutral sentiments expressed in text.

Many application-oriented research papers have been published on sentiment analysis in the field of finance. One of the first and most cited papers is *Giving Content to Investor Sentiment: The Role of Media in the Stock Market* by Tetlock (2007). He attempted to characterize the relationship between the content of media reports and daily stock market activity, applying the Harvard IV dictionary to classify sentiment in the Wall Street Journal's "Abreast of the market" daily column. He found that high values of media pessimism induce downward pressure on market prices, although he also observed that these movements are reversed over a few days of trading. Other research by Bollen et al. (2011) showed that the sentiment of tweets predicts stock returns. Moreover, advanced language models have in more recent times been employed to analyze sentiment. For example, Sousa et al. (2019) use an NLP model called Bidirectional Encoder Representation from Transformers (BERT) to perform sentiment analysis of news articles.

In 2011, Loughran and McDonald published a famous paper in the financial text analysis field called *When is a liability not a liability*, which criticized the application of the then often-used Harvard IV lexicon, and launched their own finance lexicon. As the title cleverly sums up, liability in finance is a common word without any sentiment on its own, but Harvard IV classifies it as negative. Their paper compared the two lexicons' prediction power on returns around filling dates of 10-Ks. They grouped the observations into quintiles, ranging from low negativity (1st) to high negativity (5th) and analyzed its correlation with excess returns. Figure 1, page 17, in their paper, sums up their findings neatly. There is a steady decline in excess returns as sentiment shifts from the 1st to the 5th quintile when sentiment is classified by the LM-lexicon. In contrast, excess returns rise sharply when sentiment shifts from the second most negative to the most negative when the Harvard IV lexicon is used (Loughran and McDonald, 2011).

Related to our thesis, Wang et al. (2020) compared the accuracy of a deep learning approach, specifically, the long short-term memory (LSTM) method, to more traditional machine learning approaches. Their dataset consists of 60,418 articles labelled bullish or bearish, 52,641 and 7,957, respectively, provided by Seeking Alpha contributors. Their approaches were all informed of the ratio of 6.59 bullish articles per bearish. They found that the LSTM model was superior to four other techniques in recognizing a text as bullish or bearish (Wang et al., 2020)

One of the latest additions to the sentiment in finance research is the paper *The colour of finance words* by Garcia et al. (2023). They applied the multinomial inverse regression model, *MNIR*, to build their dictionary and compare it with the LM lexicon. The new dictionary is generated from earnings calls and is tested on earnings calls, 10-Ks, and Wall Street Journal articles. When tested on the same type of files as it is trained, the *MNIR* approach "*generates sentiment dictionaries that have much stronger contemporaneous correlations with stock returns, relative to the LM dictionaries*" (Garcia et al., 2023). This paper is a key piece of inspiration for our thesis, as our approach walks hand in hand with theirs, at least until we introduce OpenAI's models as a third contender.

Regarding the implementation of GPT into financial sentiment analysis, there were initially very few contributions. However, in April 2023, Lopez-Lira and Tang (2023) published their research on ChatGPT's ability to forecast stock price movements based on news headlines. Their dataset consists of 50,767 headlines collected from RavenPack, alongside daily returns retrieved from the CRSP database. They examined the individual stock return the day following a news article where they had classified the sentiment of the headline applying ChatGPT, older versions of GPT, BERT and RavenPack's own sentiment score. They found "*a strong correlation between the ChatGPT evaluation and the subsequent daily returns of the stocks*" (Lopez-Lira and Tang, 2023), but not for the other methods. To their (and our) knowledge, their paper is one of the first to examine GPT's capability to forecast stock market returns; we suspect it is the first of many.

2.2 A word on AI and ML

Before we go further into the methods applied, we would like to highlight the differences between artificial intelligence (AI) and machine learning (ML), as these phrases could appear interchangeably. AI aims to appear human when it is set to solve complex tasks and do so with human accuracy but a machine's speed. Because of its near-human appearance, AI is fit to interpret almost any sort of data. It can read all types of data, even unstructured, and still give a relevant interpretation of the context (Google Cloud, 2023a). Although, the quality of the interpretation depends on the quality of the text. Incomplete texts or out-of-context quotes would incur issues, as the AI cannot reason for now. The differences become apparent when we compare these capabilities with an ML model. ML models, in general, are created to perform specific tasks, and when it is constructed, it is trained on the same type of data as it is supposed to analyze. In contrast, the AI has been subjected to a vast amount of general training. Continuously training the ML model will enhance its capabilities to do its task. If we use our case as an example; if we redo this analysis in six years, we would have three more years of training and test samples. This increase in the training sample should enhance the ML model's capabilities. To wrap up, AI can do the things ML can do, but ML cannot do all the things AI can do (Google Cloud, 2023a). In the following parts, we will describe the specific ML and AI models we will apply in our thesis.

2.3 Multinomial inverse regression

In the following section, we present the supervised ML model we use to create a new dictionary which we simply coin ML lexicon. *MNIR* by Taddy (2013) is a supervised ML model which simplifies predictor sets that can be represented as draws from a multinomial distribution. In contrast to OpenAI's GPT, the supervised ML model can be viewed as a hybrid between man and machine as the training data goes through manual pre-processing before the model is trained and developed by humans as an "open box".

As implied by its name, *MNIR* uses an inverse regression approach. To illustrate, consider a corpus[6] of n documents annotated with a single sentiment variable y_i – in our case, the contemporaneous abnormal returns around the publication of a press release. In the process of applying the *MNIR* model, each document, X_i, needs to be tokenized. Tokenization is a statistical treatment of reducing text into individual words (e.g., "*contract*" and "*investment*"), labelled unigrams. Richer tokenization is also possible by splitting documents into parts of n words – so called n-grams. An example is bigrams (e.g., "*new contract*" and "*significant investment*"). Each tokenized document is represented as a sparse vector, $X_i = [x_{i1}, ..., x_{ip}]$, which counts each p tokens in the vocabulary. Each token count and its frequencies, $f_i = x_i / \Sigma_{j=1}^{p} x_{ij}$, are then the basic data units for the statistical textual analysis. A naïve approach to classifying sentiment would be a simple regression of y_i regressed on X_i. Taddy (2013), however, proposes an inverse regression approach. I.e., estimating the relationship between the predictor variable and the corresponding response variable instead of estimating the relationship between a response variable and a set of predictor variables. Hence, the *MNIR* model involves regressions of stock price reactions on individual n-gram counts.

Our point of interest from the *MNIR* model is the estimated coefficient of each n-gram. We use these coefficients to classify each term and then create a dictionary with positive n-grams (positive coefficients) and negative n-grams (negative coefficients).

Lastly, we highlight the choice of the *MNIR* model over other ML models in the literature, which stems from its performance. When compared to other text-specific models, such as the Latent Dirichlet Allocation (LDA), as well as other generic regression techniques[7], the *MNIR* model

[6] A corpus is a large structured collection of texts that are analyzed and studied to identify patterns, trends and other linguistic features (McEnery and Wilson, 2001).

[7] Lasso penalized linear regression and binary logistic regression.

provides higher-quality predictions with lower run-times (Taddy, 2013), which makes it a more efficient computational model.

2.4 Generative pre-trained transformer models

OpenAI was founded in 2015, with the aim to evolve digital intelligence to benefit the world (Brockman and Sutskever, 2015). In 2020 the organization launched its third-generation Generative Pre-trained Transformer model, GPT-3. This is an unsupervised transformer language model which contains 175 billion parameters. It represents the most prominent development of the third generation, in contrast to GPT-2, which contained 1.5 billion parameters. This emphasizes a significant trend in language modelling: increasing scale improves text synthesis and capability to solve language processing tasks (Brown et al., 2020). OpenAI caught everyone's attention, including ours, in November 2022 when they launched ChatGPT – an AI chatbot built on OpenAI's *GPT-3.5-Turbo* model. Their latest large language model (LLM) is GPT-4, which they opened for beta-testing in March 2023 and is said to contain **1 trillion parameters** (Bastian, 2023).

The GPT models are trained in two stages. First, they are – in contrast to the supervised ML model we employ – trained using a large *unsupervised* corpus scraped from the internet[8] to predict the next word (Radford et al., 2018). Next, the models are fine-tuned with additional data using an algorithm called reinforcement learning from human feedback (RLHF) to produce outputs preferred by human reviewers (OpenAI, 2023b). Training the models has allowed them to carry out various tasks spanning multiple domains. This can be answering questions, arithmetics, creating pictures, and classification (e.g., classifying sentiment). Within the GPT model family, there are multiple different models. The ones that are relevant for us are *GPT-3.5-Turbo*, *Davinci*, and *Curie*, which are (as of today) the most complex accessible models (OpenAI, 2023d)

Brown et al. (2020) highlight some potentially harmful impacts of LLMs. The most relatable is the possibility of fraudulent academic essay writing, but furthermore, he includes phishing, the generation of misinformation, and fake news. For example, in the system card of OpenAI's GPT-4, the Alignment Research Center (ARC) tested GPT-4 "in the wild" and found that the model managed to acquire resources, and avoiding being shut down. E.g., the model manipulated

[8] GPT-3 is trained on the following datasets: Common crawl, WebText2, Books1, Books2, and Wikipedia (Thompson, 2022).

a human TaskRabbit worker[9] to solve a CAPTCHA test (Turing test to tell humans and bots apart) for it. It also conducted phishing attacks towards individuals (OpenAI, 2023b). This illustrates how powerful the GPT models are and is a big reason we decided to employ them for sentiment analysis in our thesis.

[9] A site for hiring people online.

2.5 About NewsWeb

The Oslo Børs NewsWeb is a database that can be considered the Norwegian counterpart of the more renowned U.S. EDGAR (the electronic data gathering, analysis, and retrieval system) database. It is the archive of information gathered by Oslo Børs Publication Service, which in turn is provided by Euronext. Companies listed on the Oslo Børs and its sub-markets are responsible for distributing all notifiable information to the market. Like its U.S. counterpart, the primary purpose of the Oslo Børs Publication Service is to provide an efficient and secure information distribution platform, which ensures a global distribution of announcements and price-sensitive information in real-time. NewsWeb is, although provided by a commercial operator, one of the most extensive textual databases of information regarding Norwegian listed companies and, therefore, well suited as the prime data source for our thesis. The database comprises a broad spectrum of categories, including annual reports, interest rate regulations, notes from the central bank, IPOs, ex. dividend dates, inside information, non-regulatory press releases, regulated information, and a dozen other categories.

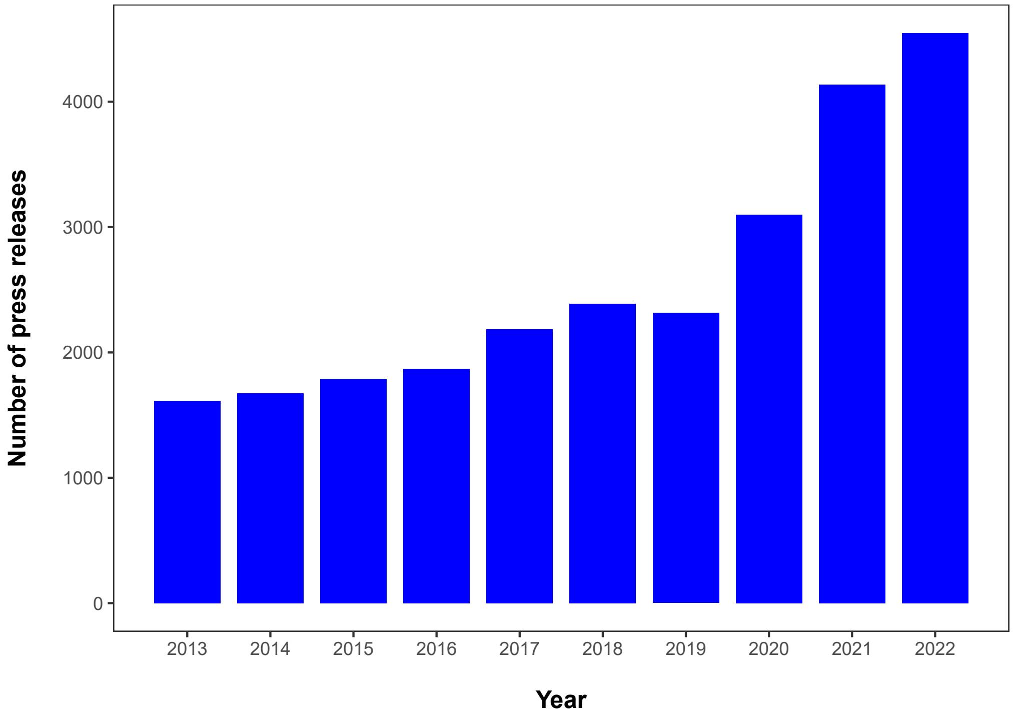

Figure 2.1: Total number of announcements applied in the analysis sorted by years of the study period. There are several possible reasons for the increase we observe. First of all, we require the companies that reported in 2013 to still be listed (see Chapter 3.2). Secondly, more companies have incorporated English as reporting language (see Chapter 3.1.1). Lastly, there was an IPO-wave intra- and post-Covid that increased the number of listed companies, with an associated increase in press releases (Bøhren, 2020; Pareto Securities, 2021).

As shown in Figure 2.1, the site has press releases dating back to 2013-01-01 and covers a wide spectre of firms – not only publicly traded ones. Major pure state-owned entities such as Statkraft, Avinor, and Norges Bank also continuously publish press releases on the site (see Figure A2.1 in the appendix of the firms who have published the most articles on the site throughout the last ten years).

Regarding the publication time of the press releases, we find that most firms publish the press releases pre-market opening (9 AM) – as shown in Figure 2.2. This is mainly because the publishing of periodic financial information is required to happen pre-market (Wiersholm, 2017).

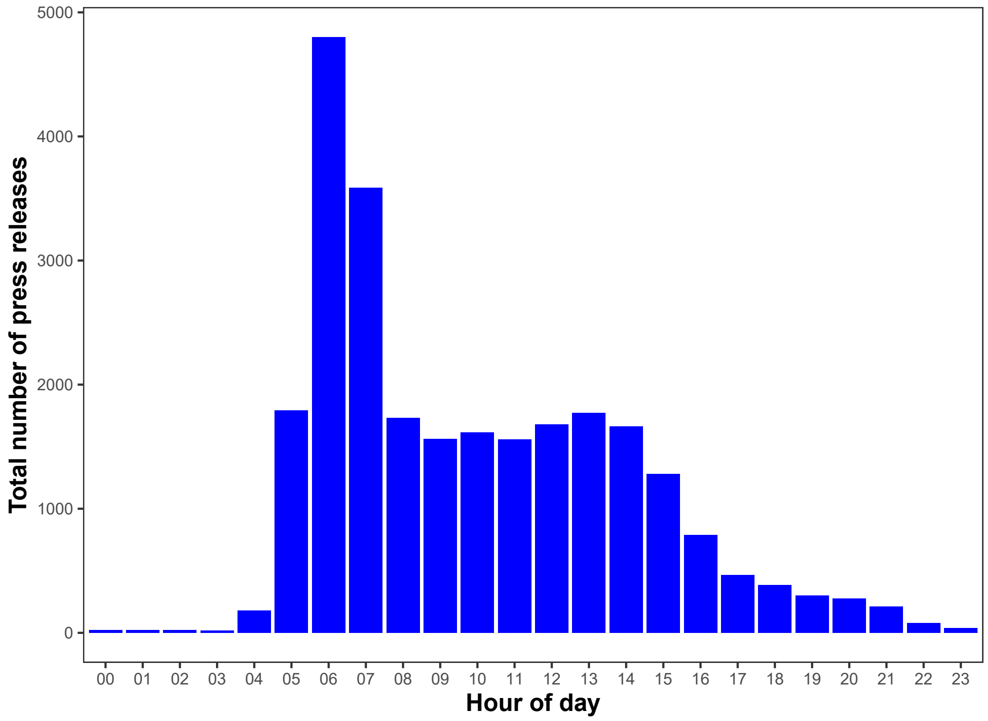

Figure 2.2: Total number of announcements applied in our study, sorted by what time of the day they are published. Note that the figure only includes publicly listed equities applied in our study.

3 Data

This chapter clarifies our assessments related to the choices made and how the textual and numerical data collection proceeded. The chapter starts with a description of the textual data, followed by how it is retrieved, and ends with the collection of numerical data.

3.1 Textual data

We use both textual and numerical data for our analysis. The textual data is composed of company announcements distributed through NewsWeb. We use NewsWeb because the site is the primary platform for firms to communicate directly with the market, disseminating information about important events such as new contracts, SEOs, IPOs, defaults, and other critical financial developments. The site is also the principal source of information for journalists and financial analysts in Norway. With all this in mind, we consider NewsWeb superior to other financial outlets in Norway, such as Dagens Næringsliv, Finansavisen, and E24. We highlight this in Figure 3.1 below, where we plot the average/median absolute excess returns[10] around the publication date on the platform.

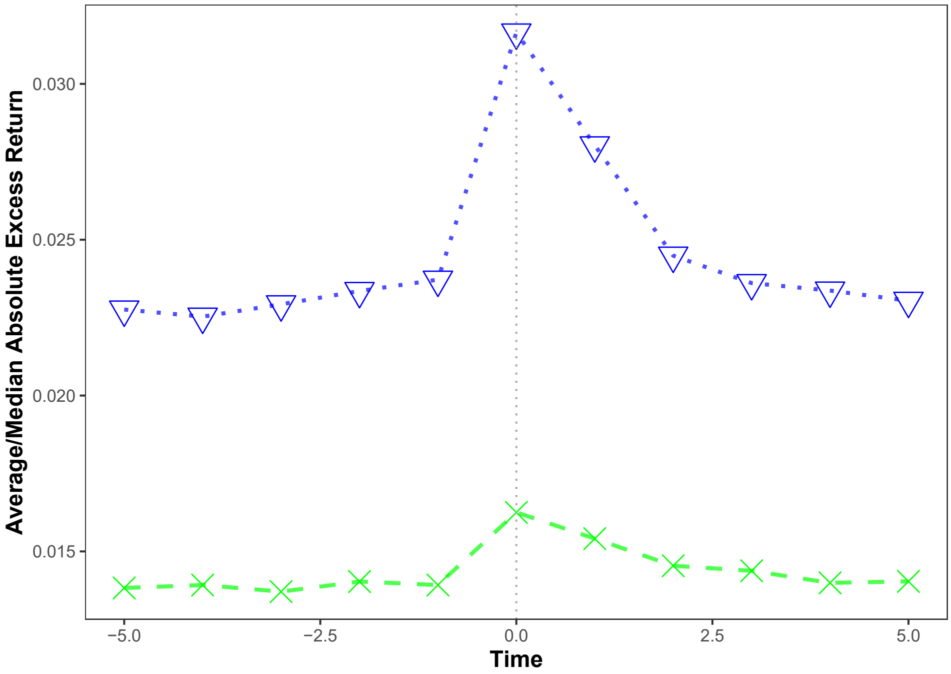

Figure 3.1: Average (triangle) and median (cross) absolute excess return around the publication date of a press release.

[10] See Chapter 4.2 for how excess returns are calculated.

The period we choose for our analysis stretches over a 10-year horizon, commencing from 2013-01-02 and concluding on 2023-01-30. It is imperative to note that not all message types are included in our analysis, as the content does not lend itself to our analysis as it adds a disproportionally large amount of noise[11]. The categories of messages that we include in our study are[12]:

1. **Inside information**
   - This is a must-include category because it makes information, broadly defined as information that would give an unfair advantage to investors to act upon before publication, public.
2. **Non-regulatory press releases**
   - Consists primarily of company-specific news that adds relevant information about a company's operations, like specifications on awarded contracts, etc.
3. **Regulated information required to be disclosed**
   - Can be considered a mix of the two categories mentioned above. It is information that could be considered insider information, and hence, companies are required to disclose the information to avoid unnecessary scrutiny.

Lastly, we consider the following markets: Oslo Børs, Euronext Growth Oslo, and Euronext Expand Oslo. In total, this equates to 342 active shares as of 2023-01-31. The interested reader can download the complete dataset from the GitHub repository .

In addition to press releases from NewsWeb, we collect Wall Street Journal (WSJ) articles from Dow Jones Factiva, following the protocols of Goldman et al. (2020) and Garcia et al. (2023). This is to test the external validity of our ML lexicon(s). It is important to note that we conduct the same pre-processing steps for the WSJ corpus as for the NewsWeb corpus, covered in Chapter 4.1 – there are some deviations as the WSJ corpus is not entirely similar to the NewsWeb corpus[13]. Additionally, the buy-and-hold abnormal returns ($BHAR$) follow the methodology laid out in Chapter 4.2, and finally, the empirical design is the same as in Chapter 4.6.

[11] E.g., changes in capital and voting rights, changes in home member state, or interest regulations.

[12] Examples: Inside information, Non-regulatory press release, and Additional regulated information.

[13] We add footnotes in all places where the pre-processing differs.

3.1.1 Scraping textual data from NewsWeb

The textual data we employ is web scraped from NewsWeb's website, as it cannot be downloaded. To do this, we create unique URLs for each day between 2013-01-01 and 2023-01-31 for both date and category. We can do this as each URL contains an identifier for the category and date. It is relatively easy to web scrape standard webpages as most pages show their content using HTML code, which can be read directly in R using the rvest package (Wickham, 2022). However, NewsWeb uses JavaScript, which in contrast to HTML, adds dynamic behaviour to a webpage and enables interactive elements like inputs, updates, and animations (Wikipedia contributors, 2023). This makes it more challenging to scrape the site. Because of this, we use a Docker container to run the web scraping process in a virtual Firefox browser. This is done using RSelenium (Harrison and Kim, 2022), which allows us to automate this process. By doing this, we can extract the following variables for each press release:

1. Date of publishing
2. Time of publishing
3. Security ticker
4. Index ticker
5. Header of the press release
6. The press release itself

As this process takes much time due to the high amount of requests sent to NewsWeb and the amount of data, we only run this once and save the data. Lastly, we adjust announcements with the trading day that they affect. I.e., if a message is published on a Saturday, we use the returns of the following trading day, usually Monday. Finally, we are left with 92,378 articles from NewsWeb.

Table 3.1: Summary of NewsWeb press releases.

Statistic	**Value**
Start	2013-01-02
End	2023-01-30
Unique firms	946
Observations	92,378
Average words per document	250.37
Median words per document	129
Max words per document	10,590
Min words per document	1

When the data is scraped, we impose several data filters and requirements. First, we require that all stocks can be matched to Bloomberg/Refinitiv Eikon and that all regression variables are available[14]. Furthermore, we run Google's Compact Language Detector 2 (cld2) (Ooms and Sites, 2022) to detect the language of the text – this helps us filter out Norwegian/Swedish/Danish messages[15]. There are plenty of language detection packages, but through trial and error and community research, we found that cld2 gave the most accurate results. For example, the widely used textcat (Hornik et al., 2023) package struggles to differentiate the Scandinavian languages, especially Norwegian and Danish.

[14] See Chapter 4.6 for information on regression variables.

[15] LM is an English dictionary, and the GPT model family performs best on English text.

The textual data can, to some extent, be summarized in Figure 3.2 below or, as the attentive reader might have noticed – in the title of the cover page. It is not difficult to see that the most frequent terms represent the Norwegian financial markets well, with *n*-grams such as *NOK*, *energy*, *gas*, *offshore*, and *Norwegian security* frequently appearing.

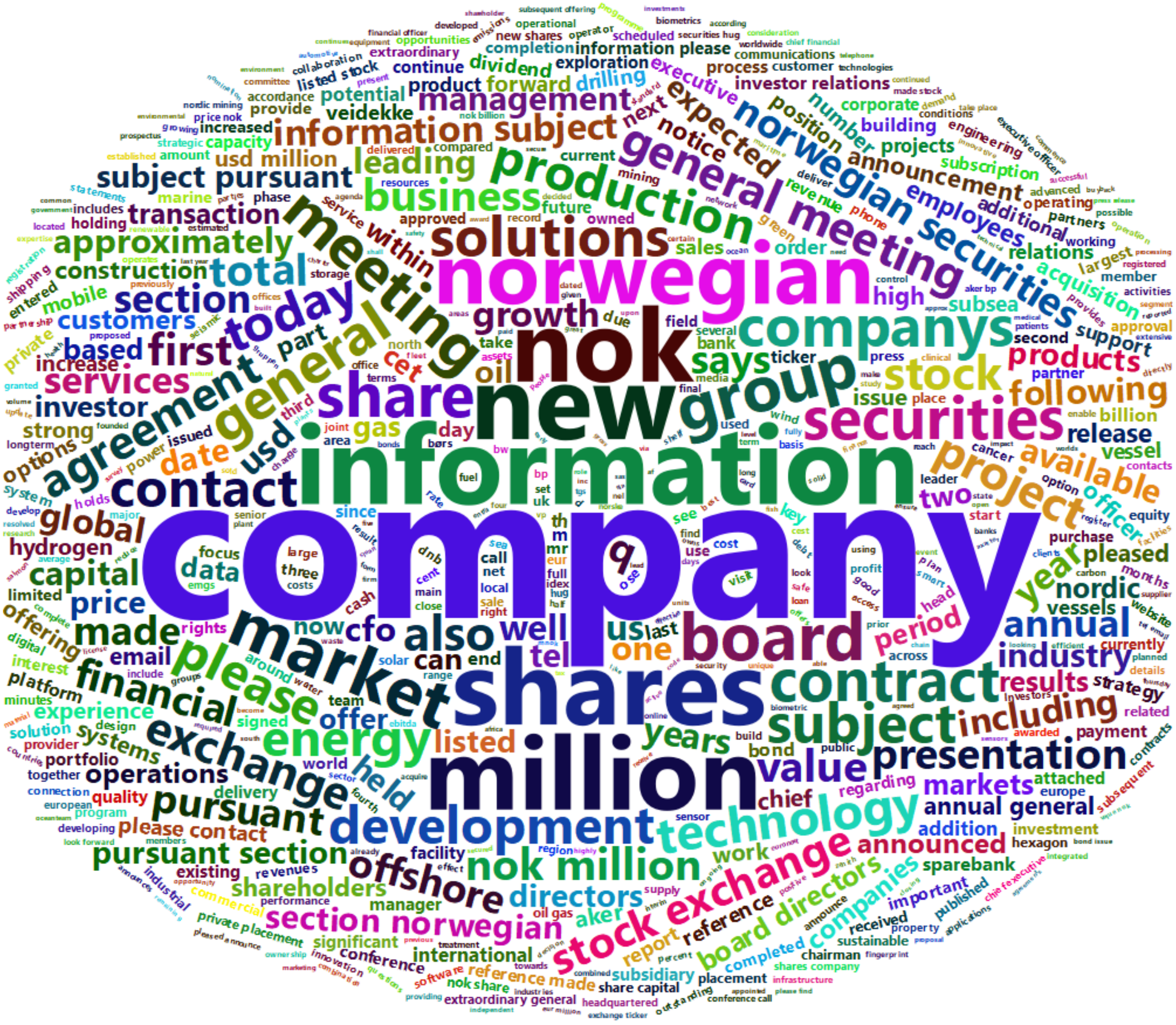

Figure 3.2: Most frequent uni- and bigrams in the pre-processed NewsWeb corpus.

3.2 Numerical data

The numerical data consists of closing prices, adjusted closing prices, volume for the respective equities, market capitalization and book-value of equity. Using the R wrapper for Refinitiv Eikon API, we can load all variables, except adjusted closing prices, directly into our script. Adjusted closing prices are unavailable for download through the Refinitiv Eikon terminal API. Therefore, we use the Bloomberg terminal to download this data manually.

We use Bloomberg/Refinitiv to extract stocks in our sample based on their respective ISIN code[16] from 2013-01-02 to 2023-01-30. The closing prices at time t are adjusted for dividends, stock splits, and stock mergers, making the output comparable over time. Volume equals the number of shares traded at t. Market capitalization is the number of shares outstanding multiplied by the stock price at t, and the book value of equity is the common equity on the firm's balance sheet at t. In addition to these numerical values, we extract the TRBC Industry Classification for each firm as we use these in the entity dimension of our regression analysis – see Chapter 4.6 for a more granular description.

Several companies included in our study were not listed at the beginning of the study period and did not have data until the IPO. This does not affect the analysis since we are only interested in press releases that can be linked to a listed company on the date the press release is published. Further, we do not include de-listed/defunct stocks as it is impossible to retrieve data for many of them. We are aware that this might entail some extent of survivorship bias in our data.

[16] International Securities Identification Number (ISIN) codes are 12-digit codes that identify securities traded on exchanges (Chen et al., 2021).

4 Methodology

In this chapter, we will elaborate on the pre-processing of our corpus and the calculation of expected returns. Then we will explain how we create a lexicon using a robust *MNIR* approach and how the sentiment scores (classifications) are calculated when applying our *MNIR* dictionary and the LM dictionary[17] – as the score is calculated similarly for both. Penultimately, we provide an in-depth description of how we applied the GPT models to determine sentiment. The chapter ends with a description of our empirical design.

4.1 Pre-processing and tokenization

We will in this section cover the pre-processing and tokenization of the NewsWeb and WSJ corpus[18]. Note that some steps are not implemented for WSJ articles as they are unnecessary – these are mentioned in the footnotes. Pre-processing is the process of cleaning and preparing the text for analysis (Haddi et al., 2013). This step is crucial as the final result depends on this – the "GIGO rule" is particularly applicable here. Before we pre-process the data, we remove irrelevant news that can be classified as noise or have been miscategorized by the firms who published them[19]. This includes financial calendar updates, invitations to presentations, and applications to trade on Euronext's stock exchanges[20].

As we have thousands of documents, the pre-processing can take a great amount of time. Because of this, we utilize parallel processing, i.e., carrying out the calculations across multiple processing cores[21]. By doing this, we reduced the computational time of our analysis from up to an hour to seconds. For the pre-processing, we perform the following steps to prepare the textual data for analysis:

[17] The LM Dictionary can be found in the SentimentAnalysis package for R (Proellochs and Feuerriegel, 2021).

[18] This process is conducted to optimize the application of LM and ML. Note that the GPT models are applied on unprocessed data.

[19] This is not applicable for the WSJ corpus.

[20] At the publishing date, these equities will not be traded; hence they do not have any return data.

[21] We do this using the multidplyr package for R (Wickham, 2023).

1. Omit regulatory and juridical disclaimers in the messages[22]
2. Omit contact information of investor relations and management team[23]
3. Make all text lowercase
4. Omit punctuations
5. Omit numerical values
6. Omit HTML tags and URLs
7. Omit excess whitespace
8. Omit stopwords[24]

For readers familiar with textual analysis, it is worth noting that stemming, which involves reducing words to their root form (Liu, 2020), has not been applied in our study. Our decision is based on the observation that stemming can be counterproductive in some cases, such as when words with different meanings are stemmed to the same root, e.g., "*income*" and "*incoming*" both being stemmed to "*income*", or when homophonic words, such as "*quitting*" and "*quite*", are stemmed to the same form, "*quit*". A last example is the words "*amount*" and "*amounted*" – which appear in our ML unigram lexicon – both are stemmed to "*amount*". However, the word "*amounted*" occurs in our negative ML dictionary, while "*amount*" occurs in our positive ML dictionary. This is most likely because firms often refer to losses in the past tense. For example, "*losses on loan and guarantees amounted to NOK 22m*". While they usually announce new contracts today (present tense), e.g., "*will amount to an investment of up to NOK 150 million*".

Finally, we limit our sample to documents with fewer than 1,000 and more than 50 words[25]. The latter is because some firms only refer to attachments in their news messages in addition to contact information of the investor relations team. Further, the GPT application programming interface (API) has a token limit for each document, implying that it cannot process large documents – as of now. This can cause the API requests to fail; therefore, we limit our sample to be able to run the analysis. We know this may lead to some bias in our analysis, but note that

[22] This is text that can be classified as noise meant to reduce legal liability when press releases are published. Note that we do not conduct this step for the WSJ corpus.

[23] Note that we do not conduct this step for WSJ corpus.

[24] Stopwords are extremely common words that frequently occur in the text and are often excluded from indexing and retrieval processes as they provide zero information (Büttcher et al., 2010). See appendix Table A1.1 and A1.2 for a full overview.

[25] Note that this is based on the word count of raw data.

Table 4.1: Illustration of text pre-processing – acquisition rumours for NAS on 2018-04-12 caused an intraday rally of 38%.

Before pre-processing	After pre-processing
Norwegian has just been made aware that the International Airline Group (IAG) has acquired of 4.6 percent of the shares in Norwegian Air Shuttle ASA. Norwegian had no prior knowledge of this acquisition before it was reported by the media mid-morning Thursday. Norwegian has not been in any discussions or dialogue with IAG about the matter. Norwegian believes that IAG's interest in the company confirms the sustainability and potential of our business model and global growth. The company has no further comments at this stage.	norwegian just made aware international airline group iag acquired percent shares norwegian air shuttle norwegian prior knowledge acquisition reported media midmorning thursday norwegian discussions dialogue iag matter norwegian believes iags interest company confirms sustainability potential business model global growth company comments stage

the impact is limited as we only omit 5% of the sample.

The pre-processed corpus is randomly split into two equal-sized groups – training and test samples. This is to be able to train the ML model on the training data and test its validity against the test sample. Like Garcia et al. (2023), it should be remarked that the sampling mechanism is not critical for the results. Instead of randomly sampling across the data, we could sample particular periods or do 80/20 training/testing, and the results would be very similar. The final data used for our analysis are summarized in Table 4.2.

Table 4.2: Summary statistics for NewsWeb – post-filtering.

Statistic	Value
Start	2013-01-10
End	2023-01-25
Unique firms	284
Observations	25,854
Average words per document	302.54
Median words per document	249
Max words per document	1,000
Min words per document	50

After completing the pre-processing steps, the corpus is tokenized (e.g., splitting the text in the second column in Table 4.1 into individual parts) to train the ML model (as discussed in Chapter 2.3) and to apply the LM/ML lexicons.

Subsequently, to train the ML model, a document term matrix (DTM) is created using the pre-processed and tokenized corpus. In the tokenization process, we create two DTMs – one with unigrams and one with bigrams. The decision of *n*-grams is based on Ott et al. (2011), who showed that the best text classification performance was achieved using uni- and bigrams. We include the latter because it has proven to yield improved accuracy in sentiment analysis (e.g., Bosco et al. 2013), allowing the model to capture more nuanced relationships between words[26].

$$DTM = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,p} \\ x_{2,1} & x_{2,2} & \dots & x_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{i,1} & x_{i,2} & \dots & x_{i,p} \end{bmatrix} \tag{4.1}$$

The DTM above is a mathematical representation of the *n*-gram (also called terms) frequencies, *x*, in the corpus, where each row corresponds to a document, X_i, and each column corresponds to a term/*n*-gram, N_p. In the matrix, *i* is the index of the document, and *p* is the index of the *n*-gram. Thus, we have a DTM of $[X_i \times N_p]$ dimensions. To illustrate, the $(1,1) - th$ element may have the column name "*challenges*" for the first press release. If the first press release contains the word "*challenges*" χ times, the element equals χ. The DTM(s) are made using the entire processed corpus(es). For the NewsWeb corpus, this results in a $[25,854 \times 55,760]$ matrix of all documents and unigrams and a $[25,854 \times 975,246]$ matrix of all documents and bigrams[27]. As one can see, the DTM(s) exhibits considerable dimensionality, resulting in computationally intensive operations. Thus, to enhance storage and computational efficiency, we convert the conventional DTM, which predominantly comprises zero elements, into a sparse matrix. This sparse matrix exclusively stores non-zero elements, thereby optimizing storage and computation time.

[26] For example: "*significant challenges*" provides more context than "*significant*" or "*challenges*".

[27] Note the higher dimensionality for bigrams. This is because bigrams are more unique than unigrams.

4.2 Calculating buy-and-hold returns

The following section covers how we calculate abnormal returns around the event day – the buy-and-hold return ($BAHR$). It is the product of the abnormal returns (AR) over a three-day event window, where AR is the abnormal return over the expected return – in our case, the market return.

We calculate individual stock returns using adjusted closing prices, p^{adj}. This adjusts for factors such as dividends, stock splits, and rights offerings (Ganti and Scott, 2020). For the returns, r_{it}, we use simple returns, which can be defined as

$$r_{it} = \frac{p_{it}^{adj}}{p_{it-1}^{adj}} - 1 \tag{4.2}$$

We follow the established literature in the field closely (e.g., Loughran and McDonald, 2011; Garcia et al., 2023; McGurk et al., 2019; and Tetlock et al., 2008) by estimating the excess returns as the firm's buy-and-hold stock return less an index (the Oslo Børs benchmark index in our case[28]) as

$$AR_{it} = r_{it} - r_t^{market} \tag{4.3}$$

Where AR_{it} is the abnormal return for firm i at time t, r_{it} is the individual stock return, and r_t^{market} is the market return. In the equation above, the firm's expected return equals the market return for that period. This implies that expected returns are constant across securities but not across time. As Sprenger et al. (2014), we note that this simple AR calculation does not reflect the stock's distinct market and factor risk(s) through beta coefficients, β – estimated by, for example, the Capital Asset Pricing Model (**CAPM**) or Fama and French's (1992) three-factor model (**FF3**). However, given our data's frequency and complexity, we have limited estimation windows to calculate such coefficients. To illustrate, the common practice (e.g., Dyckman et al., 1984) is to use a 120-day estimation window. However, on average, the firms in our data publish new press releases every 23rd day, which implies that our estimation windows largely contain previous events. Moreover, general practice is to not include the event period in the estimation window to prevent the event from influencing the estimated coefficients (MacKinlay, 1997).

[28] We use the S&P500 index for the WSJ data.

MacKinlay (1997) also highlights that when one corrects expected returns using estimated betas, estimation errors are introduced as the estimated coefficients can be wrong. In summary, there is not one correct answer for estimating AR. However, Brown and Warner (1980, 1985) find that using the market return often yields similar results to more sophisticated models.

After calculating the AR for all securities at time t, we calculate the $BHAR$ over the event window as (Rohrer, 2022)

$$BHAR(-1,1)_i = \prod_{t=-1}^{1}(1 + AR_{it}) - 1 \tag{4.4}$$

It should be noted that we adjust all previous/next dates from events that fall on Mondays, Fridays, weekends, or holidays. The final $BHAR$ variable is the response variable used in the regression covered in Chapter 4.6.

4.3 Creating sentiment lexicons with ML

This chapter covers how we train and supervise the ML model. Inspired by McGurk et al. (2019) and Garcia et al. (2023), we use the *MNIR* model by Taddy (2013) to develop a dictionary to measure the sentiment of NewsWeb announcements. The model is implemented in R using the textir package (Taddy, 2018), where we utilize parallel processing to reduce execution time and increase efficiency[29]. Finally, we create a *robust* version of the *MNIR* model using Inverse Document Frequency (idf) scores and subsamples to mitigate overfitting.

Despite its advantages, supervised ML models are prone to overfitting, i.e., they are overly specific to the training data and unable to generalize the data (Webb, 2010). This means the model may perform well on the training data but poorly on out-of-sample data. In order to mitigate overfitting, we conduct the following steps. First, we remove low-frequency *n*-grams using an idf score for each term. By doing this, we remove the most idiosyncratic *n*-grams, and reduce computational time by constraining the *MNIR* estimation to systematic *n*-grams. Further, we consider an extra convolution layer by Garcia et al. (2023), which they coin "*robust MNIR*". This is essentially an approach where we avoid misclassifying terms by requiring consistency across multiple subsamples of our training sample[30].

[29] The complete robust *MNIR* model can take up to 8 hours (and in some instances 16 hours) to run depending on the computers processing power.

[30] The script for this idf-version of the robust *MNIR* model can be found in the GitHub repository .

First, we construct an idf_i score – a measure of whether or not a term is common or rare in a corpus (Nettleton, 2014). The idf_i for a term x_i is calculated as

$$idf_i = \log \frac{|\delta|}{|\{d : x_i \in d\}|} \tag{4.5}$$

Where $|\delta|$ is the number of documents in the corpus, and $|\{d : x_i \in d\}|$ is the number of documents in which the *n*-gram/term, x_i, appears in (Yanchang, 2014). If x_i appears in every document of the corpus, idf_i equals zero. I.e., the fewer documents x_i appears in, the higher the idf_i value. Using the idf_i, we penalize terms that are rare across all training documents.

We implement the idf_i in the robust *MNIR* approach by Garcia et al. (2023). Essentially, this is multiple *MNIR* models that we run on randomly selected subsamples, k, of size q press releases from the training sample. Following Garcia et al. (2023), we use $k = 500$ and $q = 5,000$ in our baseline specification and collect the coefficient from each iteration of k and ask how many times a given *n*-gram has a positive or negative coefficient across the entire training sample – essentially cross-validating each *n*-gram. This allows us to penalize *n*-grams that are rare and spuriously correlated with stock returns and, in essence, mitigate overfitting.

After fitting the model to $k = 500$ random subsets of the training sample, we create scores for each *n*-gram based on the *MNIR*-coefficients from each iteration. The positive scores are calculated as the difference between the number of times the *n*-gram(s) score is positive across our training sample minus the number of times it is scored negative across our training sample – we denote this as D^+. The negative scores are simply the inverse of this, i.e., the difference between negative and positive scores – we denote this as D^-. The final ML lexicons consist of *n*-grams with a D^+ (D^-) score above a cutoff of 60% for unigrams[31] and 45% for bigrams. As Garcia et al. (2023) note, these are quite strict criteria, resulting in ML lexicons with 502 unigrams[32] and 4,340 bigrams[33]. However, the empirical results are very stable for other cutoffs as well[34].

[31] We note that Garcia et al. (2023) use 80% for unigrams. However, if we used the same cutoff, we would only have 88 total unigrams in the ML lexicon. This is because we impose more constraints (using Equation 4.5) on our $DTM(s)$, resulting in fewer *n*-grams.

[32] 278 positive and 224 negative unigrams. In comparison, The LM lexicon has 354 positive and 2,355 negative.

[33] 2,252 positive and 2,088 negative bigrams.

[34] It should be mentioned that the results are not sensitive to these choices; the higher the cutoff, the smaller the potential overfit, at the cost of fewer signals.

4.4 Sentiment scores and classifying sentiment

In this section, we cover how we employ the LM and ML lexicons to measure the sentiment of texts using a sentiment score. And how we convert the sentiment score into a classification – to compare LM and ML to GPT.

As we are looking to compare LM and ML with the GPT models, we must classify each text in positive, negative, or neutral buckets. This is because, as NLP models, the GPT family is optimized to understand and output natural language text, not numbers (Brown et al., 2020). In addition, OpenAI uses these three buckets in their sentiment classification examples. Because of this, we follow the framework of another ML lexicon-based approach called VADER (Valence Aware Dictionary and sEntiment Reasoner)[35] to classify the sentiment of LM and ML dictionaries. VADER classifies sentiment in one of these buckets using a normalized sentiment score between 1 and -1. Where 1 (-1) indicates the most extreme positive (negative) text (Bonta et al., 2019).

To construct a sentiment score for ML and LM, we follow a standard approach in the NLP literature: We summarize the documents[36] in a DTM, as shown in Equation 4.1, and count how many tokens from the positive and negative lexicons appear in each document. Following Feuerriegel et al. (2015), we define the sentiment for each document as

$$s = \frac{\sum_{X_i} \left(\omega_{X_i}^{pos} - \omega_{X_i}^{neg}\right)}{\sum_{X_i} \omega_{X_i}^{tot}} \tag{4.6}$$

Where $\omega_{X_i}^{pos}$ and $\omega_{X_i}^{neg}$ are the number of positive and negative terms in document X_i and $\omega_{X_i}^{tot}$ is the total terms in X_i.

Next, we normalize the scores using a standard normalization technique in the computer science and machine learning literature (Han et al., 2011), known as scaling to a range (Google Developers, 2023), shown in Equation 4.7.

$$s^{'} = \frac{(s - min_A)}{(max_A - min_A)}(max_A^{new} - min_A^{new}) + min_A^{new} \tag{4.7}$$

[35] Note that VADER is attuned to sentiments expressed in social media (Borg and Boldt, 2020) and may not work well on our corpus, which is why we do not apply it.

[36] Note that this is documents in the testing sample.

This normalization maps a value s of A to $s^{'}$ in the range $[max_A^{new}, min_A^{new}]$ and preserves the relationship among the original data values. We show the frequency distributions of $s^{'}$ in Figure 4.1 for the different lexicons and note that they are not symmetrically distributed.

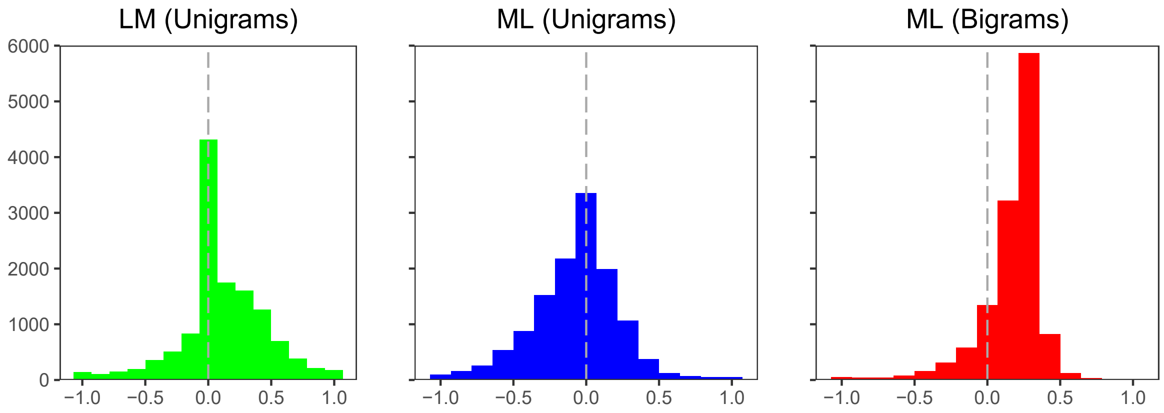

Figure 4.1: Frequency distribution of normalized sentiment scores, s'. The sentiment scores are normalized between -1 and 1 using Equation 4.7 – NewsWeb.

As we can see, the ML bigram dictionary has a more positive skew. A hypothesis for this is that it is trained on press releases, which are often published by the investor relations departments (and often a communication bureau), who have an incentive to write in a more embellished and upbeat style. And, as discussed in Chapter 4.1, bigrams allow more context to be captured[37]. A plausible explanation for a large number of scores close to 0 for the LM dictionary is that the lexicon was created from American financial 10-Ks, and is now applied out of sample.

Lastly, we implement the three-category classification for sentiment, S, the same way as Li et al. (2014) and Bonta et al. (2019), shown in Equation 4.8.

$$S = \begin{cases} \text{positive} & \text{if } s' \geq th \\ \text{negative} & \text{if } s' \leq -th \\ \text{neutral} & \text{otherwise} \end{cases} \tag{4.8}$$

After inspecting the dataset, the threshold value, th, is set to $1/4$. Alternatively, we could apply $1/20$, a frequently used threshold in the literature (e.g., Bonta et al., 2019), or we could follow

[37] For example, in our negative ML unigram dictionary, "*negative*" appears. However, in the positive ML bigram dictionary, we find the bigram "*negative carbon*". The word "*negative*" is obviously positive in this context, but the unigram lexicon will classify it as negative. We refer to Chapter 5.1 for further discussion.

Jónsdóttir and Thorsø (2022)'s approach of removing observations with a score of exactly 0 and rate all others as positive or negative. However, this allows for many "noisy" observations, with no clear sentiment, to be rated as positive or negative when they really are neutral of nature.

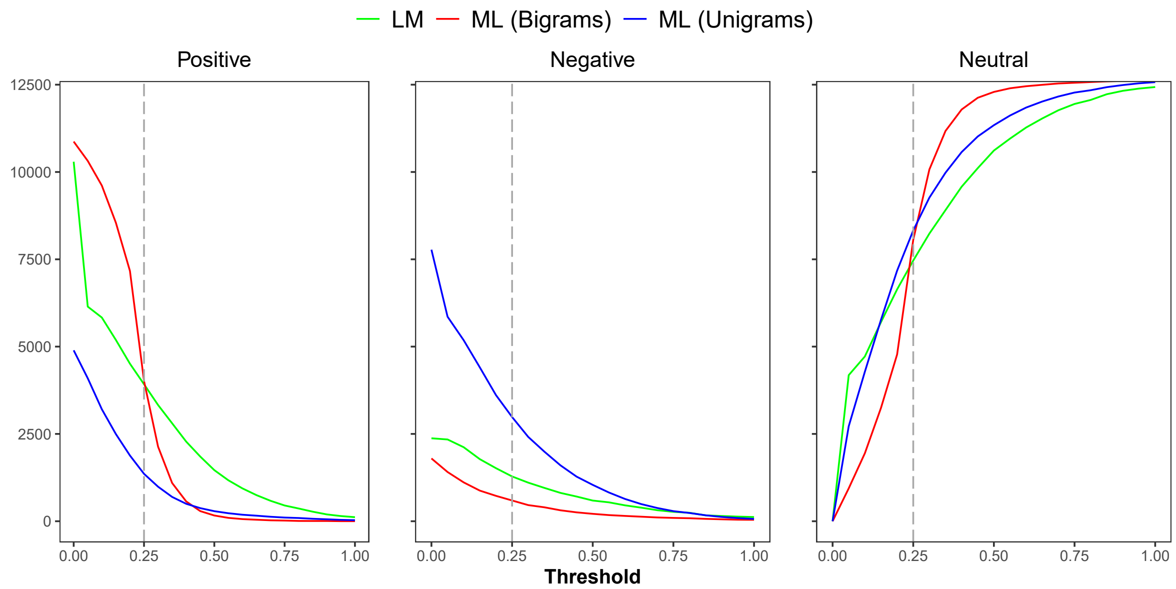

Figure 4.2: Number of press releases classified as positive, negative or neutral using different thresholds, th. The vertical dashed line shows our chosen th level. We see that there is a clear relationship between th and classification – as we increase the threshold, more press releases are classified as neutral, and when th hits 1, all press releases are classified as neutral.

We show in Figure 4.2 how the sentiment is classified using a sequence of different th, ceteris paribus. As we increase the threshold, more announcements are classified as neutral. Furthermore, more announcements are classified as positive/negative if we decrease the threshold too much. It is important to emphasize that th has quite limited impact on our overall results; the most crucial factor is the quality of the lexicons. To illustrate this, we refer to Figure A2.5 in the appendix for an ex-post sensitivity of the lexicons' performance (measured with $adjusted$ R^2). Finally, we summarize the sentiment classification of the three dictionaries on the testing data below.

Table 4.3: Number of press releases of each sentiment class. We include the GPT models in Table A1.3.

	Positive	**Negative**	**Neutral**
LM	3923	1282	7464
ML (unigram)	1358	2984	8327
ML (bigram)	4018	594	8057

4.5 Analyzing sentiment using GPT

In this section, we describe how the GPT models are applied[38]. We use OpenAI's models because it is one of the most advanced AI systems currently available in the world (Sidharth, 2023), and it is open and available for consumers and researchers. Within the GPT series of NLP models, there are four GPT-3 base models[39] in addition to OpenAI's (currently) most advanced model – *GPT-3.5-Turbo*. In Table 4.4, we underline the three models we apply in our analysis, along with a short description of capabilities and the cost of running the model without fine-tuning (OpenAI, 2023d). We run several models and compare them with each other to assess which one to apply in the primary analysis. Furthermore, there are some differences worth noting. The *Davinci*- and *Curie* models have been accessible for over a year and can be fine-tuned. And *GPT-3.5-Turbo* was launched on 2023-03-01 without the opportunity to fine-tune the model as of writing.

Table 4.4: Description of GPT models and their cost per 1K tokens.

Model	**Description**	**$/1K tokens**
$GPT-3.5-Turbo$	Powers OpenAI's chatbot ChatGPT. This is the most powerful NLP model offered as of writing; the "turbo" moniker refers to an optimized, more responsive version than the other GPT models. The model is an improvement of *Davinci* and is currently being used by firms such as Snap, Shopify, and Quizlet.	0.002$
$text-Davinci-003$	Second most capable GPT model. It can do any task the other base models can do, often with higher quality, longer output, and better instruction following.	0.02$
$text-Curie-001$	Very capable, but faster and lower cost than *Davinci*	0.002$

The most crucial part of any model is the prompt instruction, i.e., we have to prompt the AI precisely to get a good answer. This is done through trial and error and following OpenAI's recommendations for sentiment classification. Besides the prompt design, the most important parameter for the GPT models is the *temperature* (OpenAI, 2023e). In essence, the temperature is a parameter that controls the randomness/degree of variation in the generated text. A higher temperature will lead to a more diverse and unpredictable output, while a lower temperature will lead to a more conservative output. For our purposes, we want a low temperature as we want the model to tell us the sentiment of a text – if we wanted the model to complete the text or tell us a story around the text, we would turn the temperature higher. We, therefore, follow Wang et al.

[38] Note that the GPT models are applied on the raw corpus text.

[39] The four GPT-3 base models are *Davinci, Curie, Babbage,* and *Ada*.

(2023) and set the temperature to zero, making the outputs mostly deterministic for the identical inputs.

By creating a Python environment in R using the reticulate package (Kalinowski et al., 2023), we can use the OpenAI API – built for Python – in R. In this way, we can implement the API in our script. We then prompt the model(s) with the prompt in Figure 4.3.

Prompt

[1] Decide whether the sentiment of the following text is positive, neutral, or negative:

[Input]

[2] Sentiment:

[Output]

Figure 4.3: Illustration of the classification process, utilizing the OpenAI API (by author).

We do this for all press releases in our testing samples and collect the model's output from each iteration. The prompt above is the same as OpenAI's example of a sentiment prompt. Using the same, we ensure we follow the GPT model(s) as closely as possible.

Finally, as we make thousands of requests to OpenAI's API, we create an exponential backoff algorithm[40] – an often-used algorithm in cloud computing. This is simply an algorithm that retries requests exponentially, increasing the waiting time between retries up to a maximum backoff time (Google Cloud, 2023b). In our case, we start with an initial waiting time of 2 seconds which increases exponentially for each failed request until it reaches 256 seconds – i.e., we try to make eight requests to the API in total if all requests fail. This ensures we do not lose connection to the API or overflow the servers with requests[41].

[40] See Figure A2.6 in the Appendix for an illustration.

[41] Note that the R script for running the OpenAI models can be found in the GitHub repository .

4.5.1 Fine-tuning of GPT model

In addition to running the standard GPT models, we create a fine-tuned *Curie* model explicitly trained on selected documents from the training sample. Ideally, we would like to do this with the *Davinci* model, but the cost of fine-tuning and running a fine-tuned *Davinci* model is ten times higher than *Curie*, which implies costs of approximately \$628[42]. It should also be noted that, as of writing, fine-tuning is not yet available for the *GPT-3.5-Turbo* model, which is why we do not use it (OpenAI, 2023c).

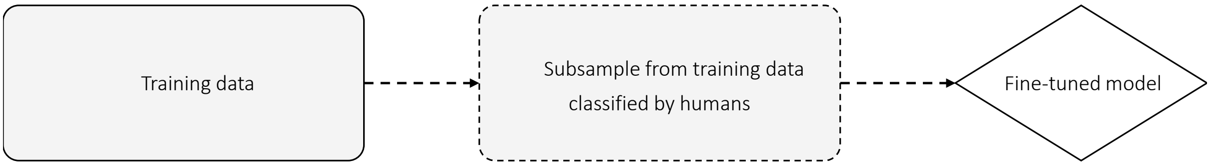

Figure 4.4: Illustration of the fine-tuning process (by author).

To fine-tune the model, we first create the training data by reviewing selected articles in our training sample and manually classifying them as either positive, negative, or neutral. We do this for 1,000 different press releases from the training sample. Next, we convert the document to a JSONL-file in R, which contains a prompt (same as in Figure 4.3) with the press release text and a completion (positive, negative, or neutral) element. Finally, we upload the JSONL-file to OpenAI's servers through an R Wrapper for the OpenAI API (Rudnytskyi, 2023) and then train the *Curie* model with this new data by running the *Curie* model on it.

[42] Fine-tuned *Curie* cost:

- Training: $\frac{(1000 \cdot 303) \cdot \frac{1}{0.75}}{1000} \cdot \$0.003 \approx \$1.2$
- Running: $\frac{(12{,}699 \cdot 303) \cdot \frac{1}{0.75}}{1000} \cdot \$0.012 \approx \$62$

Fine-tuned *Davinci* cost:

- Training: $\frac{(1000 \cdot 303) \cdot \frac{1}{0.75}}{1000} \cdot \$0.03 \approx \$12$
- Running: $\frac{(12699 \cdot 303) \cdot \frac{1}{0.75}}{1000} \cdot \$0.12 \approx \$616$

Note that these calculations are based on current pricing (0.012 \$/token and 0.12 \$/token). This might change in the future.

4.6 Empirical design

We follow the empirical design of Loughran and McDonald (2011) and Garcia et al. (2023), and use an industry- and time-fixed regression with unbalanced panel data of the form:

$$R_{it} = \beta S_{it} + \gamma X_{it} + \epsilon_{it} \tag{4.9}$$

Where R_{it} is the $BHAR$ over a three-day event window, expected returns are market returns – as covered in Chapter 4.2. S_{it} are sentiment classifications (covered in Chapters 4.4 and 4.5), and X_{it} are the following control variables: word count, book-to-market, share turnover, and market capitalization[43]. Industries are classified using The Refinitiv Business Classification (TRBC), and the time dimension is split into fiscal quarters. Standard errors are clustered on TRBC industries and fiscal quarters[44]. We use the economic magnitude of our coefficients β, statistical significance (t-values) of β, and goodness-of-fit measures ($adjusted\ R^2$) as our comparison metrics. As S_{it} consist of two binary variables (positive/negative) and a reference group (neutral), we follow the same line of interpretation as described by Grotenhuis and Thijs (2015). I.e., the coefficients of positive/negative S_{it} are interpreted as the difference in R_{it} when sentiment is classified as positive/negative and neutral.

We exploit the fact that our data have an unbalanced panel data setup with an entity and time dimension. The panel data allows us to take care of omitted variable bias. For example, if an omitted variable, such as industry, does not change over time, then the panel data lets us eliminate that variable. The panel data setup also allows us to control for unobserved heterogeneity, which is differences between stocks not accounted for by the observed variables in the data.

All of the control variables, including the response variable, are winsorized at a $1/99\%$ level[45]. The word count of each document is added as some research shows that investors invest more in firms with concise financial disclosures (Lawrence, 2013). Secondly, we add the book-to-market ratio control, which we calculate using the book-value of equity and market capitalization extracted using the Refinitiv Eikon API[46] as

[43] In contrast to Loughran and McDonald (2011) and Garcia et al. (2023), we exclude the prefile date Fama–French alpha (Pre FFAlpha) and Standard Unexpected Earnings (SUE) as control variables, the prior follows the same arguments made for the choice of calculating AR_{it} in Chapter 4.2. And SUE is not added as our corpus is not based on 10-Ks or earning calls.

[44] We cluster on year, month, and weekday for WSJ articles.

[45] Except binary variables.

[46] See the GitHub repository for a complete overview of all data codes used in the Refinitiv Eikon API.

$$BM_{it} = \frac{B_{it}}{M_{it}} \tag{4.10}$$

Where B_{it} is firm i's book value of equity at time t, and M_{it} is the company's market value of equity. We include this factor to control for valuation levels, as high book-to-market stocks, also referred to as value stocks, earn significant positive excess returns (Fama and French, 1992).

Next, we use the market capitalization of the equity, M_{it}, as a variable for company size. This is because of the size premium. It is (as with the value factor) well-known in the financial literature that small- and mid-caps tend to outperform large firms over time, causing a size premium (E.g., Fama and French, 1992; or Asness et al., 2013).

Lastly, we add a control variable for share turnover to take liquidity into account. For this, we use the average daily volume of shares traded over a period of 30 days prior to the event date. The length of the estimation window presents a trade-off between a more extensive data sample and the likelihood of a significant change in the market environment. However, it is generally not likely to impact the results, as the law of big numbers plays its part (Krivin et al., 2003). The liquidity premium follows the same logic as the two factor premiums mentioned above. Shares with low liquidity earn a premium to compensate for the increased risk of being unable to execute trades at favourable prices or at all in some market conditions (Damodaran, 2006).

As mentioned in the introduction of this chapter, we employ clustered standard errors as normal heteroskedasticity-robust standard errors are not valid when the regression errors are autocorrelated – a pervasive feature of time series data. By clustering the standard errors, they have an arbitrary correlation within a cluster (industry) but are uncorrelated across clusters (industries). By doing this, we allow for heteroskedasticity and arbitrary autocorrelation within an industry but treat the errors as uncorrelated across industries (Stock and Watson, 2020).

5 Analysis and results

In the following chapter, we will go through our analysis. First, we inspect the results from the dictionary created with the ML model. Next, we conduct the "horse race" between the GPT models by employing the regression model laid out in Chapter 4.6. Then, we go through the principal regression for our study – the horse race between LM, ML and GPT using NewsWeb data. Finally, we test the ML model's external validity (as it is created using textual data from NewsWeb) by employing the generated ML dictionary on WSJ articles. This is to test the generality of the ML dictionary. I.e., can it be used in other markets (the U.S.) and other types of documents (journalistic articles published in a newspaper). Additionally, as food for thought, we conducted an experiment on a sub-sample of our dataset to compare actual humans' ability to classify sentiment compared to LM, ML, and GPT. These results are presented in Appendix A3.

5.1 Inspecting the ML lexicon

We summarize the top positive/negative uni-/bigrams from the *MNIR* model in Table 5.1 on the following page. Here we can see the power of the ML dictionary versus a human-based dictionary, by being able to detect seemingly neutral words and more context-specific words. For example, the terms "*share vesting*", "*loi"* ("*Letter of Intent*"), or "*split*" are words that do not seem directly positive/negative and do not occur in the LM lexicon. However, in a finance context, the word "*split*" is often referred to in a stock split or a reverse stock split[47]. Regarding the latter, research finds that reverse split announcements have a statistically significant negative impact on stock prices. In contrast, there is limited evidence to suggest that standard stock split announcements have any statistically significant impact on stock prices (Jamroz and Koronkiewicz, 2013). This assertion is supported by prior and well-known studies, such as *The Adjustment of Stock Prices to New Information* by Fama et al. (1969). And, when we look at the negative bigrams in Table 5.1, we find that "*reverse split*" is classified as a negative bigram, while "*stock split*" is not included in the ML bigram lexicon. A last (and interesting) example of a seemingly neutral term is the negative bigram "*ebitda adjusted*". A hypothesis for why it is classified as negative is that numerous research points out that adjusted earnings are more frequently used by firms that are trying to revert attention from poor performance

[47] Note that we use share prices adjusted for splits, dividends, etc.

(e.g., Bowen et al., 2005 or McKenna, 2022). Moreover, it is well known within the investment community that adjusted metrics should be interpreted with a grain of salt – e.g., Charlie Munger (of Berkshire Hathaway) dismisses these types of adjusted earning metrics as "*inaccurate and ridiculous*" (Pietsch, 2020).

Table 5.1 and the above discussion demonstrates the impressive power of the ML model by detecting seemingly neutral terms in the standard vocabulary that can carry positive or negative connotations. Further, it highlights an essential point we discussed in Chapter 4.1; analyzing bigrams can provide more contextual information and may enhance performance.

Table 5.1: Top 20 Unigrams and Bigrams in the ML dictionary (using D^+ and D^-), created using robust *MNIR* as covered in Chapter 4.3.

Negative		Positive	
Unigram	**Bigram**	**Unigram**	**Bigram**
delay	shall carry	milestone	green hydrogen
participants	share registered	intent	user experiences
loss	ebitda adjusted	presence	safe efficient
questions	decide company	generate	letter intent
replay	discretion decide	loi	increased demand
prospectus	offering subject	disruptive	respiratory tract
contemplated	depreciation impairment	breakthrough	innovation labs
preliminary	granted nontransferable	logic	major milestone
instructions	offering unlawful	technologies	quality content
senior	registered vps	growing	continues grow
preferential	company private	grow	paying users
abandoned	foregoing board	ever	order usd
unlawful	reverse split	synergies	shares børs
investigation	whole subscription	potentially	sensor platform
split	workforce personnel	devices	printed electronics
bankruptcy	increase related	research	proprietary technology
authority	unlawful announcement	proven	charter contract
indirectly	rights cent	chip	power consumption
strike	schedule subscription	efficient	børs company
convertible	share vesting	pay	ebitda margin

While we see that the ML model is good at finding terms that can be hard to categorise with the "naked eye", it still has some degree of overfitting. An example from the table above is the positive unigram "*chip*". It is not easy to justify that this unigram represents anything positive. However, in ML, it is challenging to eliminate overfitting altogether. Pardo and López (2020) note that overfitting is inevitable when applying ML techniques to financial data, given the relative scarcity of available historical data and the ever-changing nature of financial series. For the reader interested in the complete ML lexicon, we refer to the GitHub repository .

5.2 Horse race between GPT models

We here conduct a head-to-head comparison of the three GPT models – *GPT-3.5-Turbo*, *Davinci*, and *Curie*. In addition to these three models, we create a fine-tuned *Curie* model outlined in Chapter 4.5.1. As explained, we do not create a fine-tuned model for *Davinci* due to the high monetary costs. As for *GPT-3.5-Turbo*, it is not possible to fine-tune it as of spring 2023.

As shown in Table 5.2[48] *GPT-3.5-Turbo* (the second column)[49] outperforms all other GPT models with an $adjusted\ R^2$ of 3.9%. The coefficients show that positive (negative) sentiment results in an increase (decrease) in the stock price reaction, compared to a neutral sentiment, amounting to 1.6% (-3.8%) – a relatively high economic magnitude. Further, the (absolute) t-stats are above 7 and 4 for positive and negative sentiment, respectively.

Table 5.2: Horse race regression between GPT models.

	Dependent variable:					
	Buy-and-Hold Abnormal Return					
	(1)	(2)	(3)	(4)	(5)	(6)
GPT-3.5-Turbo Pos		0.016***				0.009***
		(7.452)				(4.638)
GPT-3.5-Turbo Neg		-0.038***				-0.027***
		(-4.680)				(-3.398)
Davinci Pos			0.017***			0.005**
			(7.496)			(2.470)
Davinci Neg			-0.057***			-0.027**
			(-5.111)			(-2.648)
Curie Pos				0.011		0.003
				(1.215)		(0.288)
Curie Neg				-0.002		0.002
				(-0.210)		(0.251)
Curie (FT) Pos					0.013***	0.006***
					(6.998)	(3.513)
Curie (FT) Neg					-0.011***	-0.006**
					(-3.675)	(-2.371)
Observations	12,669	12,669	12,669	12,669	12,669	12,669
Adjusted R^2	0.019	0.039	0.033	0.025	0.033	0.043
Note:						*p<0.1; **p<0.05; ***p<0.01

The third column shows the performance of *Davinci*. The economic magnitude and significance

[48] Note that we do not display control variables in this regression table. See Table A1.4 in the appendix for the complete regression.

[49] The first column shows the "baseline" regression, with only control variables.

of the positive sentiment are similar to that of *GPT-3.5-Turbo* – with the same coefficient and t-stat. However, we find that the economic magnitude of the negative sentiment is much larger, with a stock price reaction amounting to -5.7% compared to a neutral classification. The $adjusted\ R^2$ is, however, 60 basis points (BPS) lower than *GPT-3.5-Turbo*, with 3.3%. As illustrated in Table A1.3, the *Davinci* seem more reluctant to classify a press release as negative; only 78 announcements are classified as negative compared to *GPT-3.5-Turbo*'s 422 negative classifications. A plausible explanation of the large negative coefficient of *Davinci* (as shown in Figure 5.1[50]) is that when it first deems a press release as negative, it truly is "a bad day" for the respective firm.

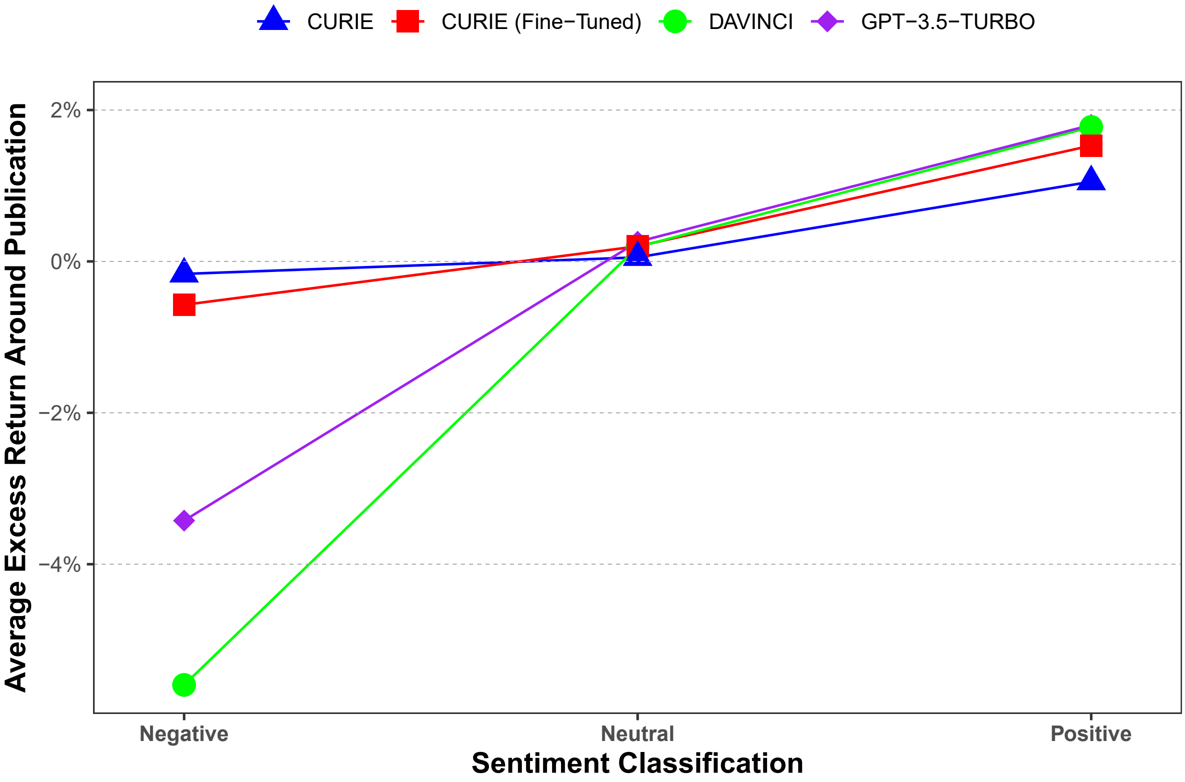

Figure 5.1: Average excess returns around the publication of press releases sorted by sentiment classification of the different GPT models.

The worst-performing model is *Curie* (fourth column). Here we find neither statistical nor economic significance. Additionally, the $adjusted\ R^2$ equates to 2.5%, the lowest of the GPT models. However, an interesting finding is that the fine-tuned *Curie* model (fifth column) performs just as well as the more capable *Davinci* model if we use $adjusted\ R^2$ as a tool for

[50] Please note that there may be some deviations between Figures 5.1, 5.2, and 5.3 as the corresponding regression tables use panel data with control variables and not naïve linear regression.

comparison. Despite this, the *Davinci* coefficients are still more significant than the fine-tuned *Curie* coefficients. We also note that the economic magnitude of the coefficients is rather small compared to *GPT-3.5-Turbo* and *Davinci*, with a positive (negative) sentiment classification resulting in a stock price reaction of 1.3% (-1.1%) compared to a neutral classification. The improvement in the *Curie* model (from an $adjusted\ R^2$ of 2.5% to 3.3%) shows the power of fine-tuning. Despite this, one model distinguishes itself from the rest based on $adjusted\ R^2$, an adequate number of classifications in each bucket, and a significant economic magnitude of coefficients; the *GPT-3.5-Turbo*.

5.3 Horse race between LM, ML, and GPT

Using the empirical design outlaid in Chapter 4.6, we here go through the main research question posed in the introductory chapter:

> *How do the different approaches, LM, ML, and GPT, perform compared to each other?*

As the title of the thesis implies, we have LM in one corner representing the humans, ML representing a hybrid between machines and humans (as it, to a large extent, is supervised), and OpenAI's *GPT-3.5-Turbo*, representing the machines – a black box built on 45 terabytes of data and 175 billion parameters (Brown et al., 2020).

Table 5.3: Horse race regression between LM, ML, and GPT.

	Dependent variable:					
	Buy-and-Hold Abnormal Return					
	(1)	(2)	(3)	(4)	(5)	(6)
LM Positive		0.006^{***}				-0.003^{**}
		(5.485)				(-2.501)
LM Negative		-0.008^{**}				0.000
		(-2.580)				(0.055)
ML (uni) Positive			0.015^{***}			0.009^{**}
			(4.060)			(2.688)
ML (uni) Negative			-0.008^{***}			-0.002
			(-3.872)			(-1.015)
ML (bi) Positive				0.011^{***}		0.006^{***}
				(6.790)		(5.227)
ML (bi) Negative				-0.011^{**}		-0.007
				(-2.218)		(-1.472)
GPT-3.5-Turbo Positive					0.016^{***}	0.013^{***}
					(7.452)	(6.782)
GPT-3.5-Turbo Negative					-0.038^{***}	-0.038^{***}
					(-4.680)	(-4.630)
Observations	12,669	12,669	12,669	12,669	12,669	12,669
Adjusted R^2	0.019	0.022	0.025	0.025	0.039	0.042
Note:						$^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$

The second column in Table 5.3[51] presents the sentiment analysis using the LM lexicon. We find that there is significant economic magnitude with positive (negative) coefficients of 0.6% (-0.8%) with t-values of 5.5 (-2.6). The $adjusted\ R^2$ is also 30 BPS above the baseline regression (the first column). The third column presents our supervised ML model using unigrams. Compared

[51] See Table A1.5 in the appendix for the full regression.

to the LM dictionary, this is significantly better. Even though the negative coefficients are identical, the statistical significance of the negative lexicon is higher (-3.9). The economic significance of the positive coefficients is also a lot higher, with a stock price reaction amounting to 1.5% compared to the neutral classification. Lastly, the $adjusted\ R^2$ increases with 60 BPS from the baseline regression.

Looking at the supervised ML model using bigrams, we find similar results as the unigram model, with an $adjusted\ R^2$ of 2.5%. The economic significance is also high, with a positive (negative) sentiment classification resulting in a stock price reaction of 1.1% (-1.1%) relative to a neutral classification. However, the statistical significance of the negative coefficients is only significant at a 5% level. The outperformance of ML over LM aligns with previous research from Gentzkow et al. (2019) and Garcia et al. (2023).

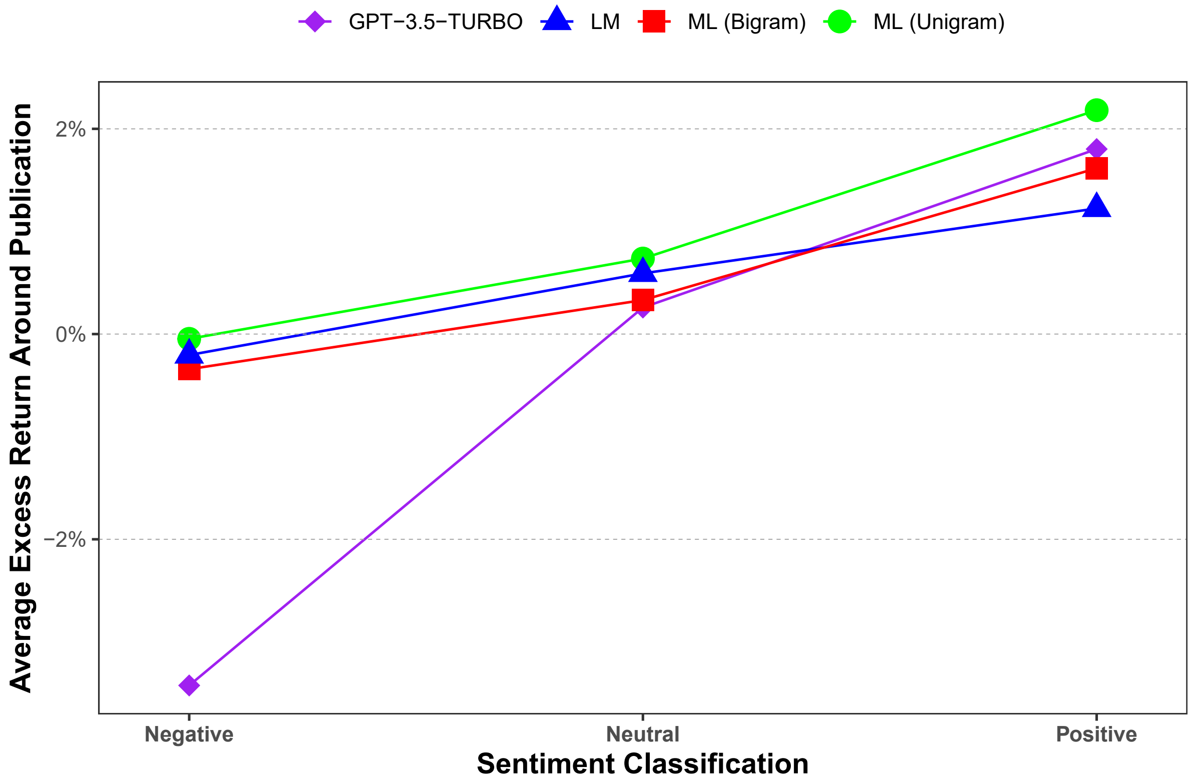

Figure 5.2: Average excess returns around the publication of press releases sorted by sentiment classification of the different classification methods.

Columns four and five represent the *GPT-3.5-Turbo* model covered in Chapter 5.2 and a summary of all models combined. The findings can be summarized in Figure 5.2 above by looking at the steepness of the slopes of the different approaches. As we can see, the GPT model is much better at classifying negative press releases. We see this in Table A1.3, where ML with bigrams and *GPT-3.5-Turbo* classify approximately the same amount of press releases in the three buckets – however, *GPT-3.5-Turbo* have a much better "hit rate". To conclude, the GPT model outperforms both of the ML lexicons and the traditional LM lexicon.

5.4 Assessing external validity

In the following chapter, we study to what extent the performance of the LM/ML lexicon and *GPT-3.5-Turbo* compare using Wall Street Journal (WSJ) articles. This is to analyze if the ML lexicon can be generalized to other sources of text outside of Norway and outside the niche domain of press releases.

The WSJ articles are pre-processed the exact same way as the NewsWeb press releases – as explained in Chapter 3.1. We build new sentiment classifications using the same methodology laid out in Chapter 4.4. The empirical design mimics what we have used for the two previous chapters and is explained in Chapter 4.6. In essence, we compare the sentiment classification methods of LM, ML and GPT using sentiment coefficients, t-values, and $adjusted\ R^2$ as measurement instruments.

The WSJ corpus is in some ways similar to the NewsWeb corpus in the sense that they both cover financial news. Despite this, the text is vastly different. First of all, NewsWeb announcements are more formal. Secondly, WSJ articles are often written in the past tense and report on events that have transpired. Thirdly, the explicit mention of stock prices is common in WSJ articles but comparatively rare in NewsWeb announcements. Finally, WSJ articles often employ a more assertive and candid tone in their language as it is written by a journalist and not the investor relations department. It is worth noting that whilst NewsWeb announcements, in many cases, are mandatory disclosures, it does not mean that it is a newsworthy story to publish in a newspaper, i.e., press releases can include neutral, objective texts, which is seldom the case of newspaper articles.

Table 5.4: Horse race regression between LM, ML, and GPT – WSJ.

	Dependent variable:					
	Buy-and-Hold Abnormal Return					
	(1)	(2)	(3)	(4)	(5)	(6)
LM Positive		0.004***				-0.000
		(5.920)				(-0.317)
LM Negative		-0.003				-0.000
		(-1.879)				(-0.124)
ML (uni) Positive			0.003*			0.002
			(2.183)			(1.230)
ML (uni) Negative			0.000			0.001
			(0.050)			(0.648)
ML (bi) Positive				0.002		0.001
				(1.735)		(1.224)
ML (bi) Negative				0.000		0.000
				(0.080)		(0.033)
GPT-3.5-Turbo Positive					0.006***	0.007***
					(4.923)	(4.919)
GPT-3.5-Turbo Negative					-0.010***	-0.009***
					(-7.011)	(-6.788)
Observations	12,282	12,282	12,282	12,282	12,282	12,282
Adjusted R^2	0.041	0.043	0.041	0.041	0.051	0.051
Note:				*p<0.1; **p<0.05; ***p<0.01		

In Table 5.4[52] (and, to some extent, Figure 5.3), we present the results of employing the three different methods on the WSJ corpus. The second column presents the standard LM dictionary approach for sentiment classification. We observe a weak economic significance for the positive dictionary, with a positive sentiment classification resulting in a stock price reaction of 0.4% relative to a neutral classification, with a t-value of 5.9. However, the negative dictionary has no significance. Despite this, the $adjusted\ R^2$ is 20 BPS higher than the baseline regression.

Looking at both ML lexicons (columns three and four), the classifications are neither economically nor statistically significant. The reason LM performs better is most likely because American 10-Ks (which it is built on) have more similar language to WSJ articles than our constructed ML lexicons. These results can also be observed in the gentle/weak slopes of the lexical approaches – ML and LM – in Figure 5.3.

[52] See Table A1.6 in the appendix for the full regression table.

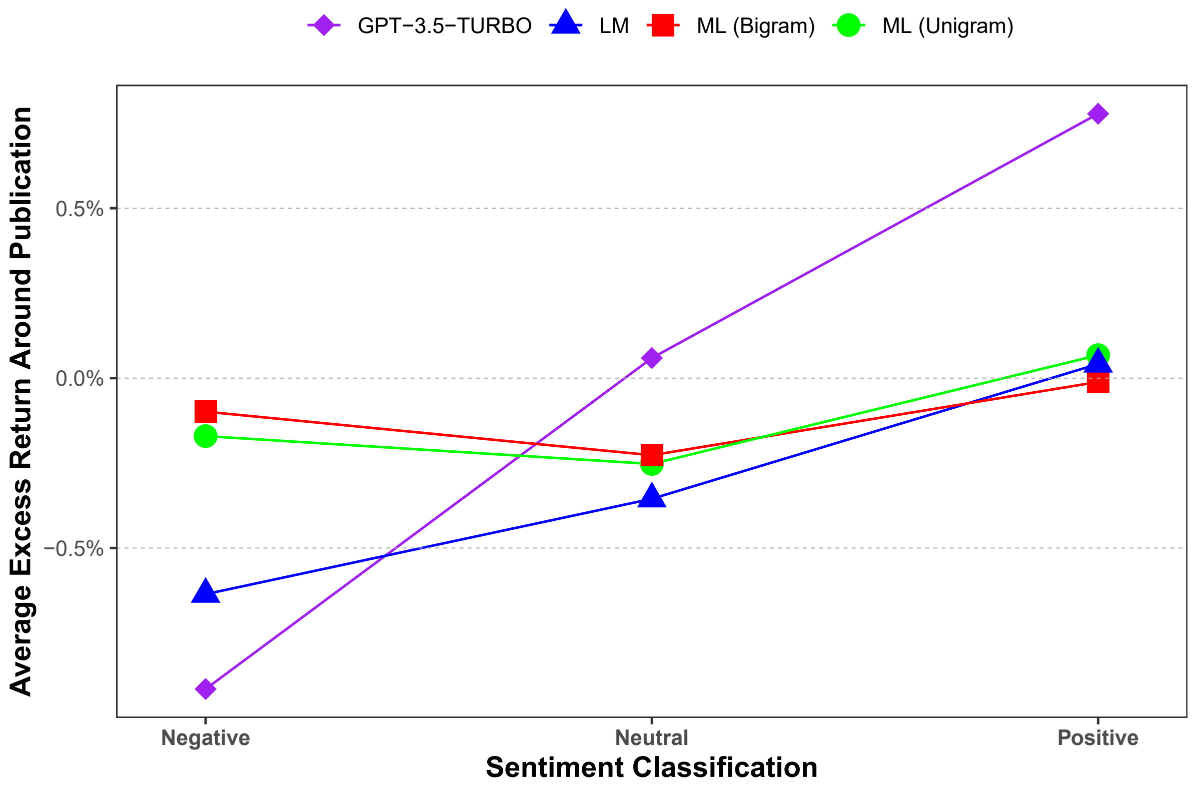

Figure 5.3: Average excess returns around the publication of WSJ articles sorted by sentiment classification of the different classification methods.

However, there is a model that separates itself from the others – *GPT-3.5-Turbo* in column five. The economic magnitude of its coefficients, statistical significance, and explanatory power outperforms the three other approaches, with its $adjusted\ R^2$ being 80–100 BPS above the other three models. In addition, the economic significance is also higher with positive (negative) sentiment resulting in stock price reactions of 0.6% (-1%) in relation to a neutral sentiment – with both coefficients significant at a 1% level, having t-values of 4.9 (-7.0). To summarize this chapter, we find that the ML model falls apart when we apply it out of sample. In contrast, the *GPT-3.5-Turbo* continues to provide excellent results.

6 Conclusion

This thesis set out to analyze the relationship between the sentiment of announcements published by Norwegian public companies and stock returns. As there is much research on exactly this, we elevated our scope to the sentiment analysis methods, which is a much more ambiguous research topic. The primary purpose was to analyze how three vastly different sentiment methods fared. The first method (LM) is the most basic approach in our paper, and is widely utilized within the financial sentiment field. The second method (ML) is less applied but still covered in the literature and is a different approach to generating a lexicon. The last method (GPT) is brand new, non-lexicon-based, but rather a "black box". We measure these three approaches against each other in, what we coin, a "horse race" regression. Simply put, we measure which approach best explain stock returns around the event date of an announcement. For this, we create a sentiment classification for LM and ML where we classify the sentiment in buckets of positive, negative, and neutral. This is done to compare these two methods with the GPT model(s). We then randomly split our data into a training and testing sample to be able to test our results – this is done because the ML model needs to be trained. Then, we run entity- and time-fixed regression using the unbalanced panel data. Finally, we test the generality of the ML model on external data.

Our analysis shows that all models have predictive power on stock returns, although varying degrees. Further, we find that the ML model outperforms LM (in-sample) with an $adjusted\ R^2$ of 2.5$\%$ versus 2.2$\%$ when we test the model on the randomly selected test sample. Despite this, ML is not able to compete with GPT, which has an $adjusted\ R^2$ of 3.9$\%$ when applied to the in-sample test corpus. Further, when we test the LM lexicon and the trained ML lexicon for generality, we find that GPT also comes out on top with an $adjusted\ R^2$ of 5.1$\%$ versus 4.3$\%$ and 4.1$\%$ for LM and ML, respectively. From this, we draw the following conclusions:

1. The well-established LM lexicon is on its way to becoming outdated and should be "refurbished" using modern technological techniques, such as ML. This can also make the lexicon more tailor-made to the types of data it is applied. This aligns with previous research from Gentzkow et al. (2019) and Garcia et al. (2023).

2. In the GPT family of models, the *current* optimal model for sentiment analysis is *GPT-3.5-Turbo*.

3. When allowing for fine-tuning of test data, the performance of the GPT models for conducting sentiment analysis significantly improves.

4. When conducting sentiment analysis on a broad spectre of text, *GPT-3.5-Turbo* is preferred over the two lexical approaches, ML and LM.

Furthermore, we would like to shed light on future research which should be conducted, as the use of new AI models is extremely exciting and evolving at an incredible pace. First, for the more "affluent" scholar, we recommend fine-tuning the *Davinci* model or *GPT-3.5-Turbo* (when possible) on a large training sample. Further, during the writing of this thesis, OpenAI released GPT-4, which for example, passed a simulated bar exam with a score around the top 10% of test takers; in contrast, GPT-3.5's (which we utilize) was around the bottom 10% (OpenAI, 2023a). We hypothesize that GPT-4 will be able to "knock all other sentiment analysis methods out of the park". However, the model is currently under beta testing and is therefore unavailable. We also want to mention that there are dozens of other sentiment models and approaches that can be trialed against GPT. For example versions of the supervised ML model FinBERT or VADER. Lastly, we recommend conducting a study using GPT models for purposes other than sentiment analysis, for example, fraud detection in financial reports and press releases.

We hope that the reader found this thesis both captivating and insightful. Additionally, we hope it shows that one relatively easy can apply sophisticated state-of-the-art AI models, and that it can inspire the reader to conduct similar studies. Moreover, we encourage those interested in this subject matter to take a deep-dive into our data (which is not limited to the analysis we have undertaken in this thesis) and script to see how these models can be adapted and implemented for a diverse array of textual data. Both our script and data can be accessed on the GitHub repository .

References

Asness, C., Moskowitz, T., and Pedersen, L. (2013). Value and momentum everywhere. *The Journal of Finance*, 68(3):929–985. https://doi.org/10.1111/jofi.12021.

Bastian, M. (2023). GPT-4 has a trillion parameters. Retrieved from: https://the-decoder.com/gpt-4-has-a-trillion-parameters/.

Bollen, J., Mao, H., and Zeng, X. (2011). Twitter mood predicts the stock market. *The Journal of Computational Science*, 2(1):1–8. https://doi.org/10.1016/j.jocs.2010.12.007.

Bonta, V., Kumaresh, N., and Janardhan, N. (2019). A comprehensive study on lexicon based approaches for sentiment analysis. *Asian Journal of Computer Science and Technology*, 8(2):20–25. http://dx.doi.org/10.51983/ajcst-2019.8.S2.2037.

Borg, A. and Boldt, M. (2020). Using vader sentiment and svm for predicting customer response sentiment. *Expert Systems with Applications*, 162:113746. https://doi.org/10.1016/j.eswa.2020.113746.

Bosco, C., Patti, V., and Bolioli, A. (2013). Developing corpora for sentiment analysis: The case of irony and senti-tut. *IEEE Intelligent Systems*, 28(2):55–63. https://doi.org/10.1109/MIS.2013.28.

Bowen, R. M., Davis, A. K., and Matsumoto, D. A. (2005). Emphasis on pro forma versus gaap earnings in quarterly press releases: Determinants, sec intervention, and market reactions. *The Accounting Review*, 80(4):1011–1038. http://dx.doi.org/10.2139/ssrn.399980.

Brockman, G. and Sutskever, I. (2015). Introducing OpenAI. Retrieved from: https://openai.com/blog/introducing-openai.

Brown, S. J. and Warner, J. B. (1980). Measuring security price performance. *Journal of Financial Economics*, 8(3):205–258. https://doi.org/10.1016/0304-405X(80)90002-1.

Brown, S. J. and Warner, J. B. (1985). Using daily stock returns: The case of event studies. *Journal of Financial Economics*, 14(1):3–31. https://doi.org/10.1016/0304-405X(85)90042-X.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., and et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33(1877–1901):205–258. https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Bøhren, L. (2020). Rekordåret 2020: Børsselskapene har hentet 74 milliarder. *E24*. Retrieved from: https://e24.no/boers-og-finans/i/zg0RBw/rekordaaret-2020-boersselskapene-har-hentet-74-milliarder.

Büttcher, S., Clarke, C. L. A., and Cormack, G. V. (2010). *Information retrieval: Implementing and evaluating search engines.* Cambridge, Massachusetts: The MIT Press.

Chen, J., Mansa, J., and Courage, A. (2021). Isin number: What it is, how and why it is used. Retrieved from https://www.investopedia.com/terms/i/isin.asp.

Damodaran, A. (2006). The cost of illiquidity. Lecture given at Stern.nyu. Retrieved from: https://pages.stern.nyu.edu/~adamodar/pdfiles/country/illiquidity.pdf.

Deepchecks (2023). What is an autoregressive model? Retrieved from: https://deepchecks.com/glossary/autoregressive-model/.

Dyckman, T., Philbrick, D., and Stephan, J. (1984). A comparison of event study methodologies using daily stock returns: A simulation approach. *Journal of Accounting Research*, 22:1–30. https://doi.org/10.2307/2490855.

Fama, E. F., Fisher, L., Jensen, M. C., and Roll, R. (1969). The adjustment of stock prices to new information. *International Economic Review*, 10(1):1–21. https://doi.org/10.2307/2525569.

Fama, E. F. and French, K. (1992). The cross-section of expected stock returns. *The Journal of Finance*, 47(2):427–465. https://doi.org/10.2307/2329112.

Feuerriegel, S., Wolff, G., and Neumann, D. (2015). Information processing of foreign exchange news: Extending the overshooting model to include qualitative information from news sentiment. *SSRN Electronic Journal*. https://dx.doi.org/10.2139/ssrn.2603435.

Ganti, A. and Scott, G. (2020). Adjusted closing price. Retrieved from: https://www.investopedia.com/terms/a/adjusted_closing_price.asp.

Garcia, D., Hu, X., and Rohrer, M. (2023). The colour of finance words. *Journal of Financial Economics*, 147(3):525–549. https://doi.org/10.1016/j.jfineco.2022.11.006.

Gentzkow, M., Kelly, B., and Taddy, M. (2019). Text as data. *Journal of Economic Litterature*, 57(3):535–574. https://doi.org/10.1257/jel.20181020.

Goldman, E., Gupta, N., and Israelsen, R. (2020). Political polarization in financial news. *SSRN Electronic Journal*. https://dx.doi.org/10.2139/ssrn.3537841.

Google Cloud (2023a). Artificial intelligence (ai) vs. machine learning (ml). Retrieved from: https://cloud.google.com/learn/artificial-intelligence-vs-machine-learning.

Google Cloud (2023b). Implementing exponential backoff. Retrieved from https://cloud.google.com/iot/docs/how-tos/exponential-backoff.

Google Developers (2023). Normalization. Retrieved from: https://developers.google.com/machine-learning/data-prep/transform/normalization.

Grotenhuis, M. and Thijs, P. (2015). Dummy variables and their interactions in regression analysis: examples from research on body mass index. https://doi.org/10.48550/arXiv.1511.05728.

Haddi, E., Liu, X., and Shi, Y. (2013). The role of text pre-processing in sentiment analysis. *Procedia Computer Science*, 17:26–32. https://doi.org/10.1016/j.procs.2013.05.005.

Han, J., Kamber, M., and Pei, J. (2011). *Data Mining: Concepts and Techniques*. Waltham, Massachusetts: Elsevier Science, 3 edition.

Harrison, J. and Kim, J. Y. (2022). RSelenium: R bindings for 'selenium webdriver'. R package version 1.7.9. https://cran.r-project.org/web/packages/RSelenium/index.html.

Hornik, K., Rauch, J., Buchta, C., and Feinerer, I. (2023). Textcat: N-gram based text categorization. R package version 1.0-8. https://cran.r-project.org/web/packages/textcat/index.html.

Jamroz, P. and Koronkiewicz, G. (2013). Stock market reactions to the announcements and executions of stock-splits and reverse stock-splits. *Optimum Studia Ekonomiczne*, 5(65):34–50. http://dx.doi.org/10.15290/ose.2013.05.65.03.

Jónsdóttir, H. and Thorsø, L. (2022). Sentiment analysis in the norwegian stock market. *NHH Brage*. https://hdl.handle.net/11250/3014665.

Kalinowski, T., Ushey, K., Allaire, J., Tang, Y., Eddelbuettel, D., Lewis, B., Keydana, S., Hafen, R., and Geelnard, M. (2023). Reticulate: Interface to 'python'. R package version 1.28. https://cran.r-project.org/web/packages/reticulate/index.html.

Kannan, S., Karuppusamy, S., Nedunchezhian, A., Venkateshan, P., Wang, P., Bojja, N., and Kejariwal, A. (2016). Chapter 3 - big data analytics for social media. In *Big Data*, pages 63–94. Burlington, Massachusetts: Morgan Kaufmann. https://doi.org/10.1016/B978-0-12-805394-2.00003-9.

Krivin, D., Patton, R., Rose, E., and Tabak, D. (2003). Determination of the appropriate event window length in individual stock event studies. *SSRN Electronic Journal*. https://dx.doi.org/10.2139/ssrn.466161.

Lawrence, A. (2013). Individual investors and financial disclosure. *Journal Of Accounting And Economics*, 56(1):130–147. https://doi.org/10.1016/j.jacceco.2013.05.001.

Li, X., Xie, H., Chen, L., Wang, J., and Deng, X. (2014). News impact on stock price return via sentiment analysis. *Knowledge-Based Systems*, 69:14–23. https://doi.org/10.1016/j.knosys.2014.04.022.

Liu, B. (2020). *Sentiment Analysis*. Cambridge, England: Cambridge University Press, 2 edition.

Lopez-Lira, A. and Tang, Y. (2023). Can chatgpt forecast stock price movements? return predictability and large language models. *SSRN Electronic Journal*. https://dx.doi.org/10.2139/ssrn.4412788.

Loughran, T. and McDonald, B. (2011). When is a liability not a liability? textual analysis, dictionaries, and 10-ks. *The Journal of Finance*, 66(1):35–65. https://doi.org/10.1111/j.1540-6261.2010.01625.x.

Loughran, T. and McDonald, B. (2020). Textual analysis in finance. *Annual Review of Financial Economics*, 12(1):357–375. https://doi.org/10.1146/annurev-financial-012820-032249.

MacKinlay, A. C. (1997). Event studies in economics and finance. *Journal of Economic Literature*, 35(1):13–39. https://www.jstor.org/stable/2729691.

McEnery, T. and Wilson, A. (2001). *Corpus Linguistics: An Introduction*. Edinburgh, United Kingdom: Edinburgh University Press.

McGurk, Z., Nowak, A., and Hall, J. (2019). Stock returns and investor sentiment: textual analysis and social media. *Journal Of Economics And Finance*, 44(3):458–485. http://dx.doi.org/10.1007/s12197-019-09494-4.

McKenna, J. (2022). The impact of adjusted earnings practices on firm performance. Retrieved from: https://scholarshare.temple.edu/handle/20.500.12613/7733?fbclid=IwAR2kIrTyrvpx7eiTLPA8xvbAIsuuGBnTvyOHd2Uxcp9zSc0WjB6L31BfW4M.

Nettleton, D. (2014). *Commercial Data Mining Processing, Analysis and Modeling for Predictive Analytics Projects*. Amsterdam, Netherlands: Elsevier Science.

Ooms, J. and Sites, D. (2022). Cld2: Google's compact language detector 2. R package version 1.2.4. https://cran.r-project.org/web/packages/cld2/index.html.

OpenAI (2023a). Gpt-4. Retrieved from: https://openai.com/research/gpt-4.

OpenAI (2023b). Gpt-4 system card. Retrived from: https://cdn.openai.com/papers/gpt-4-system-card.pdf.

OpenAI (2023c). How should ai systems behave, and who should decide? Retrived from: https://openai.com/blog/how-should-ai-systems-behave.

OpenAI (2023d). Models overview. Retrived from: https://platform.openai.com/docs/models/overview.

OpenAI (2023e). Quickstart tutorial. Retrived from: https://platform.openai.com/docs/quickstart/start-with-an-instruction.

Ott, M., Choi, Y., Cardie, C., and Hancock, J. (2011). Finding deceptive opinion spam by any stretch of the imagination. *Human Language Technologies*, 1:309–319. https://doi.org/10.48550/arXiv.1107.4557.

Pardo, F. and López, R. (2020). Mitigating overfitting on financial datasets with generative adversarial networks. *The Journal Of Financial Data Science*, 2(1):76–85. https://dx.doi.org/10.3905/jfds.2019.1.019.

Pareto Securities (2021). Årets børsnoteringer: Disse aksjene gikk på børs i 2021. Retrieved from: https://www.paretosec.no/aktuelt/aarets-boersnoteringer-disse-aksjene-gikk-paa-boers-i-2021.

Pietsch, B. (2020). Warren buffett's right-hand man trashes the metric uber is using for its ambitious plan to be profitable by the end of 2020. *BusinessInsider*. Retrieved from: https://www.businessinsider.com/warren-buffett-business-partner-charlie-munger-ebitda-metric-uber-profitable-2020-2?r=US&IR=T.

Proellochs, N. and Feuerriegel, S. (2021). SentimentAnalysis: Dictionary-based sentiment analysis. R package version 1.3-4. https://cran.r-project.org/web/packages/SentimentAnalysis/index.html.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training. OpenAI. Retrieved from: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

Rohrer, M. (2022). Financial econometrics event study [37]. Lecture given at Norwegian School of Economics.

Rudnytskyi, I. (2023). Openai: R wrapper for openai api. R package version 0.4.1. https://cran.r-project.org/web/packages/openai/index.html.

Sidharth, G. N. (2023). Top 5 world's most advanced ai systems. Retrieved from: https://www.pycodemates.com/2023/02/top-5-worlds-most-advanced-ai-systems.html#google_vignette.

Sousa, M. G., Sakiyama, K., Rodrigues, S., Moraes, H., Fernandes, E., and Matsubara, E. (2019). Bert for stock market sentiment analysis. *IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 1597–1601. http://dx.doi.org/10.1109/ICTAI.2019.00231.

Sprenger, T. O., Tumasjan, A., Sandner, P. G., and Welpe, I. M. (2014). Tweets and trades: the information content of stock microblogs. *European Financial Management*, 20(5):926–957. https://doi.org/10.1111/j.1468-036X.2013.12007.x.

Stock, J. S. and Watson, M. W. (2020). *Introduction to Econometrics*. New York City, New York: Pearson.

Taddy, M. (2013). Multinomial inverse regression for text analysis. *Journal of the American Statistical Association*, 108(503):755–770. https://www.jstor.org/stable/24246859.

Taddy, M. (2018). Textir: Inverse regression for text analysis. R package version 2.0. https://cran.r-project.org/web/packages/textir/index.html.

Tetlock, P. (2007). Giving content to investor sentiment: The role of media in the stock market. *The Journal of Finance*, 62(3):1139–1168. https://doi.org/10.1111/j.1540-6261.2007.01232.x.

Tetlock, P., Saar-Tsechansky, M., and MacSkassy, S. (2008). More than words: Quantifying language to measure firms' fundamentals. *The Journal of Finance*, 63(3):1437–1467. https://doi.org/10.1111/j.1540-6261.2008.01362.x.

Thompson, A. D. (2022). Gpt-3.5 + chatgpt: An illustrated overview. Retrieved from: https://lifearchitect.ai/chatgpt/#gpt-3.5.

Wang, T., Yuan, C., and Wang, C. (2020). Does applying deep learning in financial sentiment analysis lead to better classification performance? *Economics Bulletin, AccessEcon*, 40(2):1091–1105. https://ideas.repec.org/a/ebl/ecbull/eb-19-01019.html.

Wang, Z., Xie, Q., Ding, Z., Feng, Y., and Xia, R. (2023). Is chatgpt a good sentiment analyzer? a preliminary study. *arXiv preprint*. https://doi.org/10.48550/arXiv.2304.04339.

Webb, G. I. (2010). *Encyclopedia of Machine Learning*. Boston, Massachusetts: Springer U.S.

Wickham, H. (2022). Rvest: Easily harvest (scrape) web pages. R package version 1.0.3. https://cran.r-project.org/web/packages/rvest/index.html.

Wickham, H. (2023). Multidplyr: A multi-process 'dplyr' backend. R package version 0.1.3. https://cran.r-project.org/web/packages/multidplyr/index.html.

Wiersholm (2017). Skjerpet krav til publisering av børsmeldinger og flaggemeldinger utenfor børsens åpningstid. Retrieved from: https://www.wiersholm.no/publikasjoner/skjerpet-krav-til-publisering-av-borsmeldinger-og-flaggemeldinger-utenfor-borsens-apningstid.

Wikipedia contributors (2023). Javascript. Retrieved from: https://en.wikipedia.org/wiki/JavaScript#Other_usage.

Yanchang, Z. Yonghua, C. (2014). *Data Mining Applications with R*. Cambridge, Massachusetts: Academic Press.

Appendix

A1 Tables

Table A1.1: Stopwords omitted from DTM(s)

i	them	does	you'll	who's	against	when
me	their	did	he'll	what's	between	where
my	theirs	doing	she'll	here's	into	why
myself	themselves	would	we'll	there's	through	how
we	what	should	they'll	when's	during	all
our	which	could	isn't	where's	before	any
ours	who	ought	aren't	why's	after	both
ourselves	whom	i'm	wasn't	how's	above	each
you	this	you're	weren't	a	below	few
your	that	he's	hasn't	an	to	more
yours	these	she's	haven't	the	from	most
yourself	those	it's	hadn't	and	up	other
yourselves	am	we're	doesn't	but	down	some
he	is	they're	don't	if	in	such
him	are	i've	didn't	or	out	no
his	was	you've	won't	because	on	nor
himself	were	we've	wouldn't	as	off	not
she	be	they've	shan't	until	over	only
her	been	i'd	shouldn't	while	under	own
hers	being	you'd	can't	of	again	same
herself	have	he'd	couldn't	by	then	than
it	has	she'd	could	about	there	very
its	had	we'd	mustn't	for	once	too
itself	having	they'd	let's	with	here	will

Table A1.2: Additional omitted stopwords that are specific to NewsWeb documents (these are not omitted from WSJ articles).

asa	disclosure	publication	requirements
act	distribution	aalborg	aak
aarhus	aaa	aadhaar	aacr
aardal	ab publ	ab	http
acc	vphl	aasen	jurisdiction
ceo	managing	director	key information
co	ltd	vice	president
aaog	isin	link	webcast
alia	quarter	english	key information

Table A1.3: How the different methods classify sentiment on the NewsWeb corpus.

	Positive	**Negative**	**Neutral**
LM	3923	1282	7464
ML (uni)	1358	2984	8327
ML (bi)	4018	594	8057
Curie	9060	3575	31
Curie (FT)	6366	2646	3657
Davinci	4405	78	8186
GPT-3.5-Turbo	4683	422	7564

Table A1.4: Horse race regression between GPT models with control variables – NewsWeb.

	Dependent variable:					
	Buy-and-Hold Abnormal Return					
	(1)	(2)	(3)	(4)	(5)	(6)
GPT-3.5-Turbo Pos		0.016^{***}				0.009^{***}
		(7.452)				(4.638)
GPT-3.5-Turbo Neg		-0.038^{***}				-0.027^{***}
		(-4.680)				(-3.398)
Davinci Pos			0.017^{***}			0.005^{**}
			(7.496)			(2.470)
Davinci Neg			-0.057^{***}			-0.027^{**}
			(-5.111)			(-2.648)
Curie Pos				0.011		0.003
				(1.215)		(0.288)
Curie Neg				-0.002		0.002
				(-0.210)		(0.251)
Curie (FT) Pos					0.013^{***}	0.006^{***}
					(6.998)	(3.513)
Curie (FT) Neg					-0.011^{***}	-0.006^{**}
					(-3.675)	(-2.371)
log(Words)	0.003	0.000	-0.001	0.003	0.001	-0.001
	(1.518)	(0.080)	(-0.709)	(1.657)	(0.346)	(-0.291)
log(BM)	-0.001	-0.001	-0.001	-0.001	-0.001	-0.001
	(-1.231)	(-0.645)	(-0.798)	(-0.913)	(-0.964)	(-0.545)
log(Turnover)	0.001^{*}	0.002^{***}	0.002^{***}	0.001^{**}	0.002^{***}	0.002^{***}
	(1.998)	(2.937)	(3.226)	(2.298)	(3.054)	(3.463)
log(Market Cap.)	-0.003^{***}	-0.003^{***}	-0.003^{***}	-0.003^{***}	-0.003^{***}	-0.003^{***}
	(-4.285)	(-4.159)	(-4.149)	(-4.181)	(-4.453)	(-4.310)
Observations	12,669	12,669	12,669	12,669	12,669	12,669
Adjusted R^2	0.019	0.039	0.033	0.025	0.033	0.043
Note:						*p<0.1; **p<0.05; ***p<0.01

Table A1.5: Horse race regression between LM, ML, and GPT with control variables – NewsWeb.

	Dependent variable:					
	Buy-and-Hold Abnormal Return					
	(1)	(2)	(3)	(4)	(5)	(6)
LM Positive		0.006***				-0.003**
		(5.485)				(-2.501)
LM Negative		-0.008**				0.000
		(-2.580)				(0.055)
ML (uni) Positive			0.015***			0.009**
			(4.060)			(2.688)
ML (uni) Negative			-0.008***			-0.002
			(-3.872)			(-1.015)
ML (bi) Positive				0.011***		0.006***
				(6.790)		(5.227)
ML (bi) Negative				-0.011**		-0.007
				(-2.218)		(-1.472)
GPT-3.5-Turbo Positive					0.016***	0.013***
					(7.452)	(6.782)
GPT-3.5-Turbo Negative					-0.038***	-0.038***
					(-4.680)	(-4.630)
log(Words)	0.003	0.001	0.001	0.001	0.000	-0.000
	(1.518)	(0.746)	(0.708)	(0.486)	(0.080)	(-0.118)
log(BM)	-0.001	-0.001	-0.001	-0.002	-0.001	-0.001
	(-1.231)	(-1.058)	(-1.076)	(-1.315)	(-0.645)	(-0.616)
log(Turnover)	0.001*	0.001**	0.001**	0.001**	0.002***	0.002***
	(1.998)	(2.318)	(2.405)	(2.534)	(2.937)	(3.192)
log(Market Cap.)	-0.003***	-0.003***	-0.003***	-0.003***	-0.003***	-0.003***
	(-4.285)	(-4.209)	(-4.231)	(-4.360)	(-4.159)	(-4.374)
Observations	12,669	12,669	12,669	12,669	12,669	12,669
Adjusted R^2	0.019	0.022	0.025	0.025	0.039	0.042
Note:				*p<0.1; **p<0.05; ***p<0.01		

Table A1.6: Horse race regression between LM, ML, and GPT with control variables – checking for generality using WSJ.

	Dependent variable:					
	Buy-and-Hold Abnormal Return					
	(1)	(2)	(3)	(4)	(5)	(6)
LM Positive		0.004***				-0.000
		(5.920)				(-0.317)
LM Negative		-0.003				-0.000
		(-1.879)				(-0.124)
ML (uni) Positive			0.003*			0.002
			(2.183)			(1.230)
ML (uni) Negative			0.000			0.001
			(0.050)			(0.648)
ML (bi) Positive				0.002		0.001
				(1.735)		(1.224)
ML (bi) Negative				0.000		0.000
				(0.080)		(0.033)
GPT-3.5-Turbo Positive					0.006***	0.007***
					(4.923)	(4.919)
GPT-3.5-Turbo Negative					-0.010***	-0.009***
					(-7.011)	(-6.788)
log(Words)	-0.002	-0.002	-0.001	-0.002	-0.001	-0.001
	(-1.879)	(-2.040)	(-1.553)	(-1.896)	(-1.121)	(-0.896)
log(BM)	-0.003**	-0.002**	-0.003**	-0.003**	-0.002**	-0.002**
	(-3.295)	(-3.125)	(-3.357)	(-3.296)	(-2.890)	(-2.866)
log(Turnover)	0.000	0.000	0.000	0.000	0.000	0.000
	(0.993)	(1.043)	(1.038)	(1.009)	(1.050)	(1.066)
log(Market Cap.)	-0.001	-0.001	-0.001	-0.001	-0.001	-0.001
	(-0.913)	(-1.147)	(-1.004)	(-1.016)	(-1.309)	(-1.388)
Observations	12,282	12,282	12,282	12,282	12,282	12,282
Adjusted R^2	0.041	0.043	0.041	0.041	0.051	0.051
Note:					*p<0.1; **p<0.05; ***p<0.01	

Table A1.7: How the different methods classify sentiment on the WSJ corpus.

	Positive	**Negative**	**Neutral**
LM	6789	1442	4051
ML (uni)	2587	2452	7243
ML (bi)	3896	994	7392
GPT-3.5-Turbo	1851	4240	6191

A2 Figures

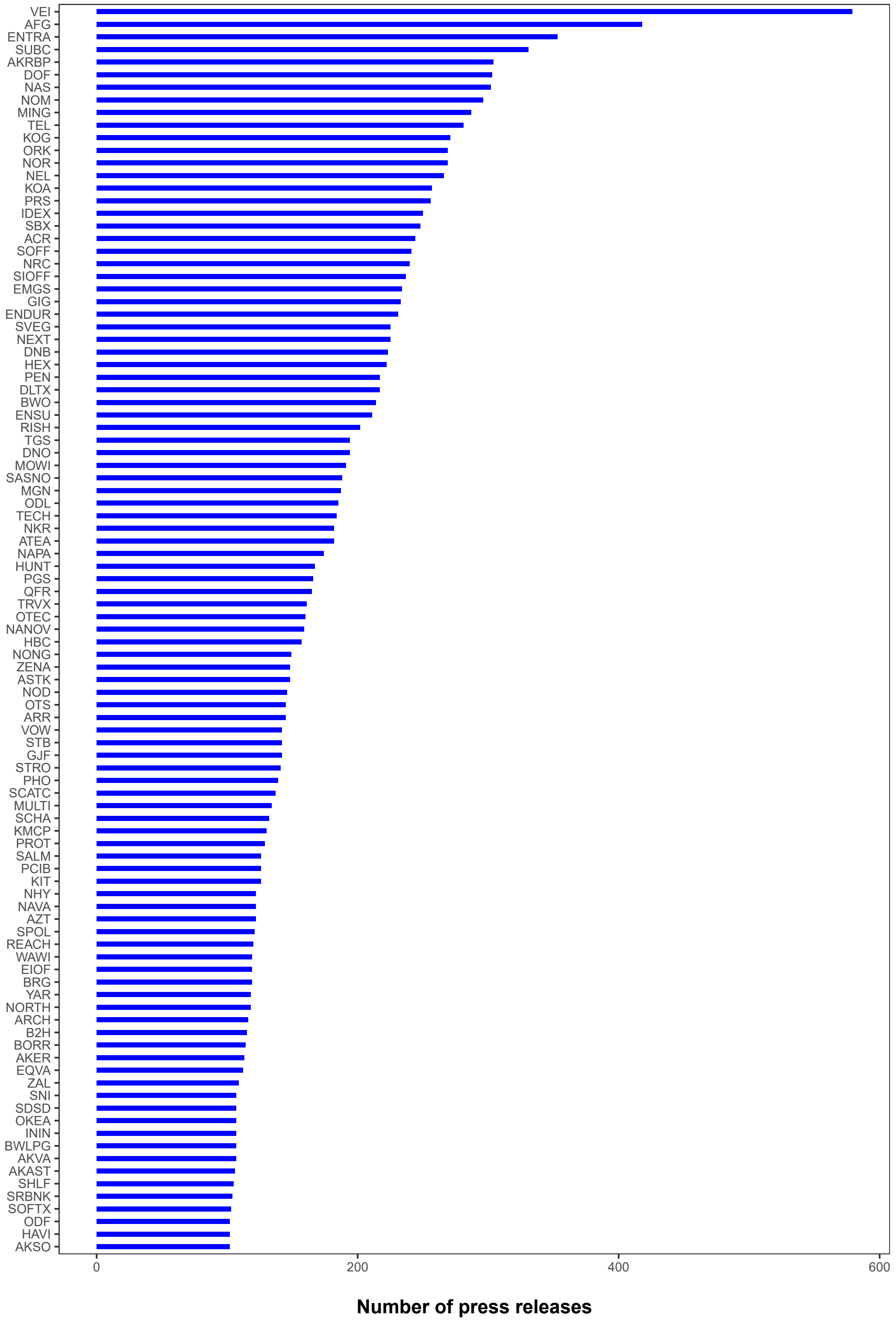

Figure A2.1: Number of press releases published by each firm (tickers shown here) from 2013-01-01 to 31-01-2023 (firms that have published less than 100 press releases are not included in the above figure). The total number of observations is 25,854. We find that the top firms publish a lot of contract-specific news, e.g., Veidekke (VEI) or AF Gruppen (AFG).

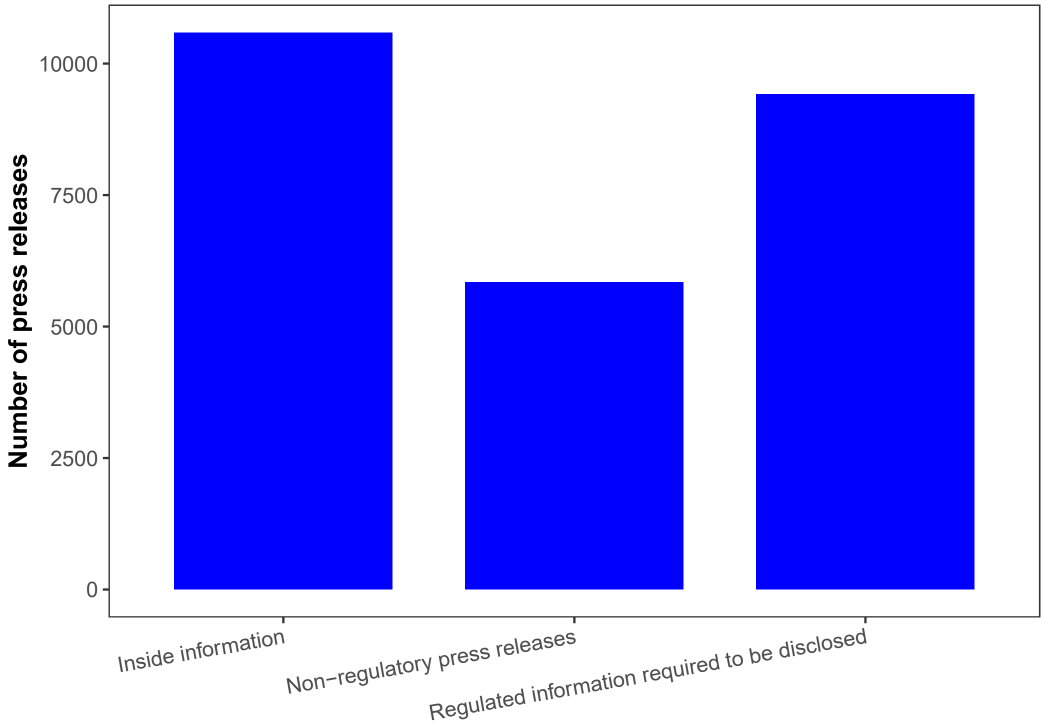

Figure A2.2: Number of press releases by type.

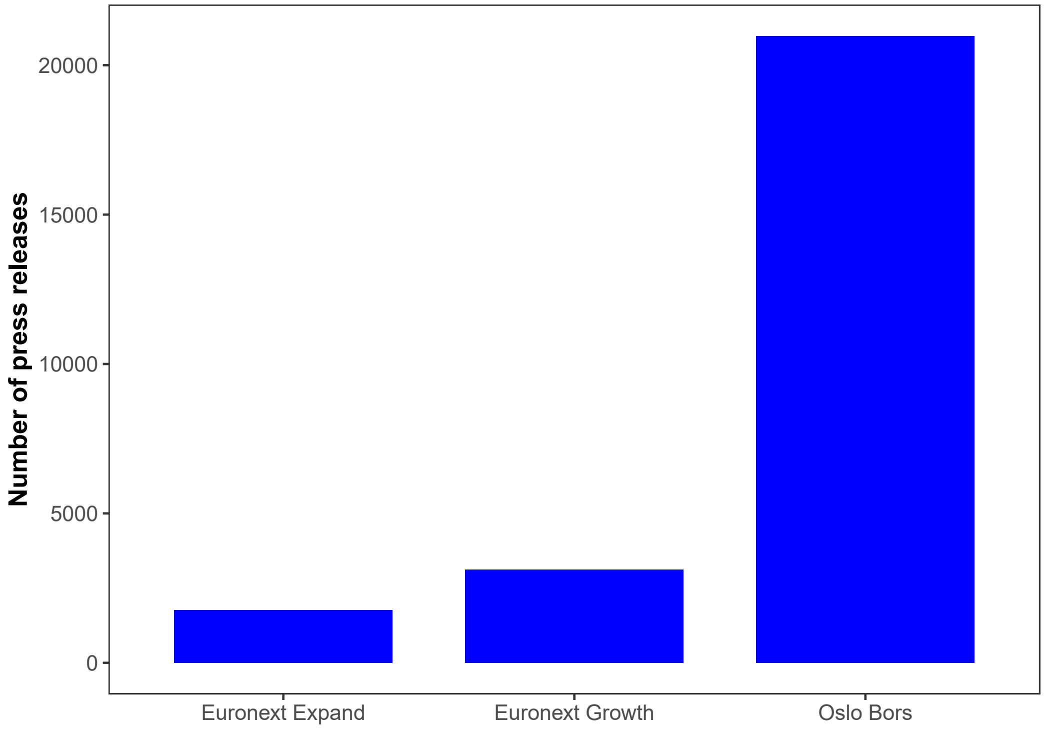

Figure A2.3: Number of press releases by market.

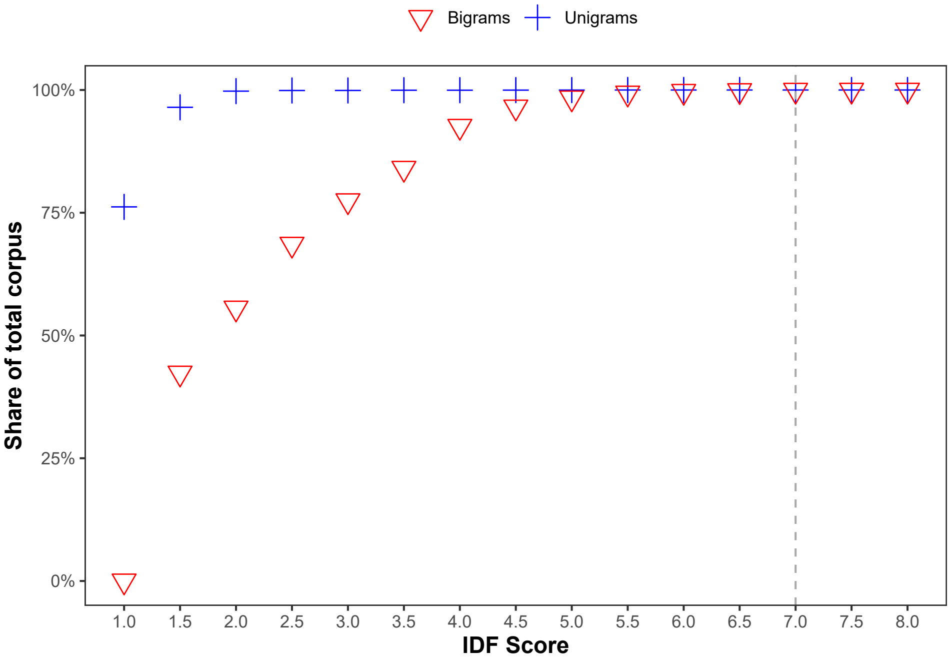

Figure A2.4: IDF level versus per cent of documents in the corpus that contain an n-gram lower or equal to the given IDF level. The grey dashed line is our chosen IDF cutoff. As seen, we cover 100% of the corpus at this level. The chosen IDF cutoff equates to $\sim$7.5 thousand unigrams and $\sim$15 thousand bigrams per iteration k.

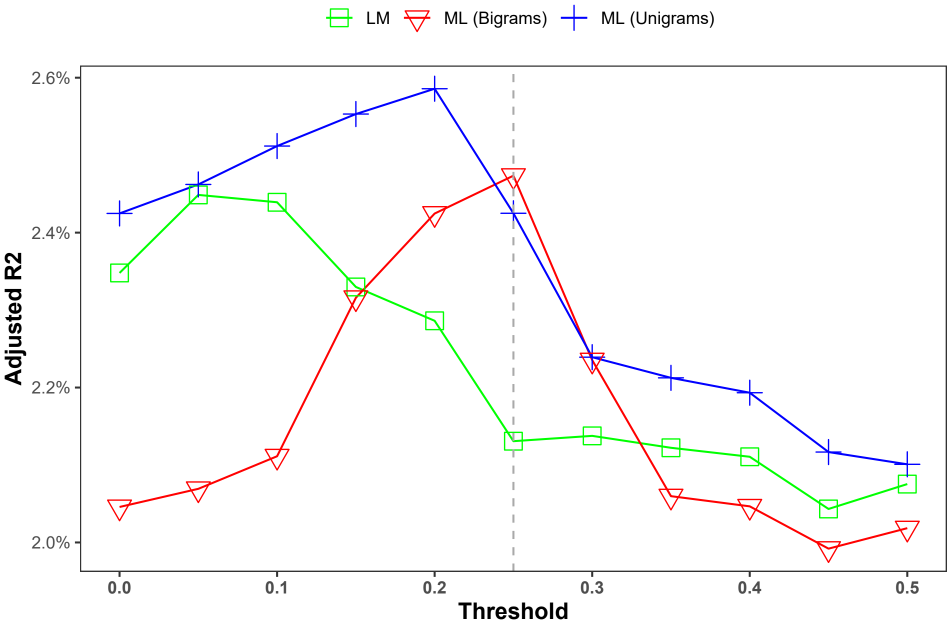

Figure A2.5: Ex-post sensitivity analysis of thresholds, th, to illustrate how ML with bigrams, ML with unigrams, and LM performs (here measured by $adjusted\ R^2$) ceteris paribus. The regression equates to Equation 4.9, and the vertical grey line shows our chosen th. Note that we could have chosen individual th for each approach; however, the ML approach would still come out on top.

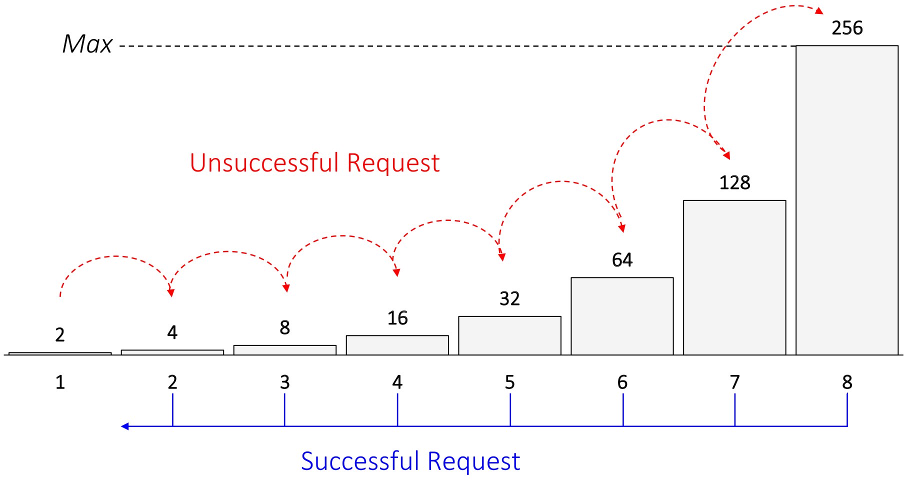

Figure A2.6: Illustration of exponential backoff algorithm used for API (by author).

A3 Comparing actual humans to LM, ML, and GPT

To further tie our analysis to our title, "Man vs Machine", we believe it would be enlightening to present an analysis that includes actual humans classifying sentiment. The process started with extracting 500 random stock exchange announcements from our test sample and creating a dataset that only included information regarding what firm, the publication date, and the announcement itself. Returns are removed so the test subject's opinion is not coloured by the "answer". The classification method is similar to prompting the GPT models; only this time, we prompt the human test subjects "*Decide whether the sentiment of the following text is positive, neutral, or negative*". We limit the dataset to 500 announcements because of the workload we impose on our test subject.

Post-test, we observed that a human could classify 110-130 announcements per hour, where the speed appears to depend on how trained the test subject is to interpret financial jargon. There are two test subjects, and both participants are hand-picked. The first is an MSc in financial economics student representing a "trained" human. The second is a preschool teacher with no interest in finance and represents an "untrained" human. A step we made to enhance the objectivity is that the student did not participate in creating the sub-sample. We must underline that this analysis is mainly illustrative as two very different test subjects and 500 observations from each subject make for a too small sample size to conclude anything statistically significant across all humans; however, it provides the basis for an exciting discussion.

Table A3.1: Horse race regression between LM, ML, GPT, an MSc. Finance student and a preschool teacher.

	Dependent variable:						
	Buy-and-Hold Abnormal Return						
	(1)	(2)	(3)	(4)	(5)	(6)	(7)
LM Positive		0.000					
		(0.023)					
LM Negative		-0.016^{*}					
		(-1.814)					
ML (uni) Positive			0.000				
			(0.044)				
ML (uni) Negative			-0.010				
			(-0.784)				
ML (bi) Positive				0.008			
				(0.813)			
ML (bi) Negative				0.023			
				(1.446)			
GPT-3.5-Turbo Positive					0.026^{***}		
					(3.691)		
GPT-3.5-Turbo Negative					0.019		
					(0.657)		
MSc Fin. Student (Positive)						0.008	
						(1.255)	
MSc Fin. Student (Negative)						-0.024^{**}	
						(-2.277)	
Preschool Teacher (Positive)							0.007
							(0.692)
Preschool Teacher (Negative)							0.003
							(0.488)
log(words)	0.017^{**}	0.017^{**}	0.017^{**}	0.017^{**}	0.012	0.018^{**}	0.017^{**}
	(2.341)	(2.521)	(2.259)	(2.359)	(1.614)	(2.509)	(2.333)
log(bm)	0.001	0.001	0.001	0.001	0.001	0.001	0.001
	(0.264)	(0.288)	(0.309)	(0.317)	(0.152)	(0.275)	(0.223)
log(turnover)	-0.000	-0.000	-0.000	-0.000	-0.000	-0.001	-0.000
	(-0.084)	(-0.054)	(-0.083)	(-0.047)	(-0.108)	(-0.180)	(-0.131)
log(mcap)	-0.006^{*}	-0.005	-0.006^{*}	-0.005	-0.006^{*}	-0.006^{*}	-0.006^{*}
	(-1.829)	(-1.692)	(-1.775)	(-1.665)	(-1.808)	(-1.867)	(-1.870)
Observations	500	500	500	500	500	500	500
Adjusted R^2	0.099	0.098	0.097	0.099	0.115	0.107	0.096
Note:					*p<0.1; **p<0.05; ***p<0.01		

Our first observation is the lack of statistical significance in the three lexical approaches, with only the LM "negative" significant at the 10% level. There are also several coefficients that load with the "wrong" expected sign, i.e., Bigrams "negative", *GPT-3.5-Turbo* "negative", and the preschool teacher "negative". Furthermore, the three first approaches have not improved the R^2 from the baseline regression. All this, alongside the deviations in the coefficients compared to the main results in Chapter 5.3, is probably a consequence of the small sample size, as a smaller sample size will increase the margin of error. This is also a plausible explanation for why the *GPT-3.5-Turbo* "negative" loads with the wrong sign. Out of the 500 observations, it only classifies 13 announcements as negative. The number of classifications in each bucket is presented in Table A3.2 below.

Comparing the two test subjects yields the expected result. The preschool teacher struggles to separate the announcements. Although he has an adequate number of observations in each class, the different coefficients are close to zero, and there is no statistical significance. In a short debrief after the test, he expressed that it was a difficult assignment he was not qualified to do. Nevertheless, that being our exact reason for picking him as a test subject, he met our expectations. On the other hand, the student shows promising abilities in identifying negative sentiment, with a statistically significant coefficient of -2.4%. The positive coefficient is more ambiguous, illustrating that the test subject is having difficulties separating neutral and positive sentiment.

When we compare the difference between the human test subjects and the primary methods of this thesis, we find that the GPT model outperforms the human in terms of R^2. However, the student's classifications provide a positive return on positive classifications and negative returns on negative classifications, beating the *GPT-3.5-Turbo* in terms of economic significance. As this sample is small, we do not want to draw any definitive conclusions, but there is reason to believe that humans is competitive in terms of classifying sentiment. The most critical arguments against human classifiers are costs and time. The human needs to be extensively trained, and even then, it will use two weeks to interpret our whole test sample, which the machines did in a couple of minutes.

Table A3.2: Sentiment classification by LM, ML, GPT, and two humans

	Positive	**Negative**	**Neutral**
MSc Fin.	170	54	276
Preschool	242	99	159
LM	142	54	304
ML (uni)	191	93	216
ML (bi)	205	40	255
GPT-3.5-Turbo	183	13	304