Transformers and Tradition: Using Generative AI and Deep Learning for Financial Markets Prediction

London School of Economics and Political Sciences

by

Toby J. Wade

A thesis submitted to the Department of Statistics of the London School of Economics for the degree of Doctor of Philosophy

London, June 2024

Declaration

I certify that the thesis I have presented for examination for the MPhil/PhD degree of the London School of Economics and Political Science is solely my own work other than where I have clearly indicated that it is the work of others (in which case the extent of any work carried out jointly by me and any other person is clearly identified in it). The copyright of this thesis rests with the author. Quotation from it is permitted, provided that full acknowledgement is made.

This thesis may not be reproduced without my prior written consent. I warrant that this authorisation does not, to the best of my belief, infringe the rights of any third party.

Abstract

Artificial intelligence has revolutionized numerous industries, and financial markets are no exception. With the ability to process vast amounts of data quickly and accurately, AI algorithms have been increasingly used in finance to predict stock prices, detect fraud, and optimize investment strategies. However, the full potential of AI in finance still needs to be explored, and researchers continue to explore new ways to apply machine learning techniques to financial challenges. This thesis investigates whether advanced Generative AI and Deep Learning techniques are more effective in extracting information for predicting financial markets than conventional natural language processing methods.

The first part of this thesis analyzes quarterly SEC 10-Q filings for S&P 500 companies from January 2000 to December 2019 to show how artificial intelligence techniques can provide reasoning about changes in corporate disclosures indicative of future company performance. This thesis finds that by leveraging the reasoning capabilities of the Claude2 large language model on the Management Discussion & Analysis section of a 10-Q, negative excess returns of -5.5% over 180 days (-11% annualized) can be avoided. The paper introduces two novel approaches: A) Concatenating Deep Learning architectures comparing quarterly filings, and B) Summarization methods using Claude2 to extract sentiment signals related to significant business risks, profitability, legal, and market pressures. Together, these techniques demonstrate new ways of expanding beyond rudimentary natural language processing approaches that many investment firms have historically used, such as lexicons and cosine similarity, to answer fundamental questions related to firm performance.

The second part of the thesis takes a step further, developing an enhanced sentiment model and utilizing Bitcoin subreddit data from December 2010 to January 2022 to predict the price of Bitcoin 60 days in advance. The Reddit text data is known for its high noise level, with non-relevant price information such as advertisements or technical advice. This noise can significantly impact the accuracy of the predictions. To address this, the research proposes a novel approach that combines a Few-Shot RoBERTa topic classification model with sample augmentation on training data powered by ChatGPT. This approach effectively reduces the noise, creating a more robust sentiment signal. The enhanced sentiment signal is then integrated with other Bitcoin on-chain features in a nonlinear multivariate LightGBM model. The results clearly demonstrate the impact of noise reduction, with the F1 score for predicting the sign of Bitcoin 60 days in advance increasing from 0.26 to 0.63 on the test set.

Acknowledgements

First, I would like to thank Professor Clifford Lam for his support, guidance and insights in regards to all parts of my thesis. This research would not have been possible without his advice and hints. With his support I was able to go deep and develop novel research ideas within the space financial markets applications using Natural Language Processing. Further, I would like to thank my secondary supervisor Professor Kostas Kalogeropoulos for his suggestions, advice and guidance throughout the PhD research process.

Additionally, I would like to thank colleagues and friends over the years where I have cultivated a passion for applying Natural Language Processing techniques to financial markets. Specifically, Michael Meninder, German Vera Concha, John Menzies and Ben Morton.

I owe my deepest gratitude to my wife Nicole DeSantis for encouraging me to apply in the first place and to pursue this dream of achieving a PhD. Without her continued support of this thesis, it most certainly would not have been possible. Further, I hope that one day my two sons will appreciate the love of learning to pursue a PhD and be inspired themselves to do something similar.

Contents

Appendix

List of Figures

List of Tables

Chapter 1

Introduction

The landscape of investment strategies, including action by fundamental and systematic investors, has traditionally relied on established financial metrics, such as price-to-earnings ratios or factors like momentum and value for stock selection. However, the emergence of alternative data sources, including unstructured data like text, images, and audio, has augmented these traditional signals and expanded the horizon of analytical tools. Over the last decade, natural language processing (NLP) has been pivotal in extracting valuable insights from textual sources like social media (Twitter, Reddit forums), news, earnings calls, SEC filings, and central bank statements for financial market prediction.

Converting unstructured text into numerical representations, NLP enables systematic analysis to identify signals related to sentiment, semantics, and topics. Conducting comprehensive analysis of textual data on a large scale presents significant challenges, as manually processing thousands of documents across various companies becomes impractical. Traditional methods employed by fundamental investors involve listening to earnings calls and meticulously reading financial disclosures to enhance their analytical processes. The introduction of Natural Language Processing (NLP) has revolutionized this landscape, offering an innovative solution by facilitating large-scale automated analysis to extract trading signals from textual data. However, a notable trade-off exists in this approach. While fundamental investors using manual methods can derive richer insights, their capacity is limited to a specific number of firms.

On the other hand, systematic analysis, although providing breadth, needs more in-depth, detailed insights. The arrival of Deep Learning and Large Language Models has the potential to reconcile this trade-off, allowing for a synthesis of both depth and breadth in analysis. This research aims to compare the effectiveness of advanced generative A.I. and deep learning methods with conventional natural language processing techniques in extracting relevant information for predicting financial market dynamics.

Traditional natural language processing (NLP) methods, such as sentiment lexicons and cosine similarity metrics, must be updated to retain their efficacy in corporate disclosures in finance. Public companies modify their disclosure language to evade detection using these conventional NLP techniques. Cao et al. (2022) observed that companies with higher-than-expected machine downloads of their filings tend to eliminate negative words that sentiment lexicons would flag strategically. The machine downloads suggest a self-reinforcing cycle whereby firms modify their language to manipulate sentiment analysis. As a result, there is a growing need for more sophisticated natural language processing (NLP) techniques that transcend traditional counting-based approaches and can leverage the power of Large Language Models and Deep Learning algorithms to provide more nuanced insights into company disclosures.

Natural Language Processing (NLP) is a pervasive technology, propelled by the advent of the internet age, with applications spanning chatbots, language translation, web search, spam filtering, artificial intelligence, etc. Natural Language Processing is the process of converting unstructured text data to structure. Traditional NLP techniques involve putting structure with applications, such as Named Entity Recognition, that can allocate words to a pre-defined category (such as person name, location, or company) to Lemmatization, which reduces a word to its root form.

Within the domain of NLP, there are two pivotal subcomponents known as Natural Language Understanding (NLU) and Natural Language Generation (NLG). NLU focuses on sentiment analysis, topic clustering, and semantic textual. NLG encompasses Large Language Models, such as ChatGPT, introduced by OpenAI, and many other competing models. ChatGPT has achieved unprecedented popularity, even surpassing the Turing test in some instances. Confirming its rise in popularity, UBS, an investment bank, has confirmed that ChatGPT is the fastest-growing consumer application in history, reaching 100 million users just two months after its launch.

In tandem, the proliferation of text data within finance has enabled some financial investors to process the data to augment their trading strategies systematically. A common approach is using sentiment lexicons like the Loughran-McDonald financial dictionary that Loughran and McDonald (2011a) introduced, which provides lists of positive and negative words tailored to the finance domain, with additional dictionaries measuring uncertainty and legal risk. Further, Loughran-McDonald introduced readability metrics like the Fog Index that measure sentence complexity. However, the evolving nature of disclosure language and deception tactics demands more sophisticated methods.

Prior work by Cohen et al. (2020) showed that tracking textual changes in disclosures using cosine similarity can signal future performance. Their "Lazy Prices" paper found firms with more active changes tended to underperform. This paper extends that work beyond similarity metrics to leverage modern NLP methods for deeper reasoning. Specifically, this research within Part 1 (which includes chapters 2 and 3) applies large language models (LLMs) that can analyze full document context and meaning. The reasoning capacity of LLMs allows for identifying nuanced changes in business conditions, risks, and profitability, thereby finding nontrivial clues about future firm performance. Additionally, this research utilizes concatenated deep learning architectures to process 10-Q filings, leading to nontrivial excess returns.

Chapter 2 provides an overview of the SEC filings dataset and key preprocessing steps, utilizing ChatGPT to identify the most and least similar topics between two 10-Q filings and comparing the results with traditional approaches such as cosine similarity. Chapter 3 delves into concatenated Deep Learning architectures spanning Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and semantic text matching. These architectures are employed to classify if the 90-day excess return will be positive or negative by comparing text data from the Management Discussion and Analysis section.

In chapter 3, this research introduces a novel methodological approach called Summarize and Sentimentize, utilizing the Claude2 large language model to summarize fundamental changes between filings related to risks, profitability, and other business conditions. Compressing the texts into insightful summaries, this research extracts sentiment signals predictive of performance. Two primary approaches for developing Summarize and Sentimentize are compared: Zero-Shot and In-context training. The Summarize and Sentimentize approach enables the prediction of negative excess returns of -11% on an annual basis by assigning sentiment ratings to the summarized textual changes.

In conclusion, this research contributes novel methodologies within modern artificial intelligence to uncover subtle signals in corporate disclosures for improved investment insights. The Summarize and Sentimize methodology leveraged Large Language Models such as Claude2 to identify language changes related to risks and profitability and provide a nuanced analysis of 10-Q filings. We are applying concatenated deep learning architectures, including CNN, RNN, and Max Embeddings models, on time series quarterly 10-Q filings to classify future returns by learning nuanced textual patterns. These advanced methods move beyond conventional NLP techniques to extract nontrivial predictive power from qualitative narratives. By applying the latest advances in NLP, this research can realize the rich insights in unstructured disclosures that remain underutilized by traditional techniques.

Part II of the thesis (which includes both chapter 4 and 5) shifts the focus on constructing a social sentiment signal with reduced noise using Bitcoin Reddit data, and in Chapter 4, the focus shifts to blockchain technology, providing an in-depth background on Reddit data related to the subreddit Bitcoin. This chapter introduces the on-chain features of Bitcoin. It meticulously breaks down noisy categories such as “advertisement” and “technical advice”, which do not contribute to the price-related information of Bitcoin. Definitions are provided for the primary on-chain indicators, including the on-chain feature Transfers Volume Exchanges Net, which refers to cryptocurrency quantities transferred in and off exchanges. Typically, when users transfer to exchanges, there is an intent to sell[1]. In contrast, if Bitcoin is moved off the exchange, it indicates someone just bought it.

In Chapter 5, the methodology for noise reduction in Reddit posts unrelated to the price of Bitcoin is introduced to find a stronger predictive signal for the price of Bitcoin 60 days out. The first part of a three-step process aims to predict Bitcoin's price by combining Few-Shot Topic classification with sample augmentation using ChatGPT. The second step involves estimating overall Reddit-based sentiment related to Bitcoin. In contrast, the third and final step integrates the aggregated sentiment signal, after having noise removed and smoothed over a daily time frame through an Exponentially Weighted Moving Average, into a nonlinear multivariate LightGBM model for forecasting the direction of Bitcoin's price direction 60 days in the future. Notably, the Reddit category and sentiment classification benefit from augmentation using an approach called AugGPT using ChatGPT. RoBERTA classification models were also employed, with a Naive Bayes Classifier as a benchmark model.

The outcome of this methodology shows a significant increase in the F1 score for predicting Bitcoin prices 60 days ahead, soaring from 0.26 to 0.63 on the test set after noise reduction. The gain in F1 score underscores the efficacy of the proposed method in mitigating noise within social media data, specifically Reddit, and enhancing the overall predictive capability by honing in on Reddit posts

[1] Note Bitcoin Miners normally sell when rewarded with Bitcoin for verifying transactions, so they would have to transfer the proceeds to an exchange to do this

related to the price of Bitcoin. This research demonstrates that leveraging generative A.I. techniques, such as RoBERTA and ChatGPT, outperforms traditional natural language processing techniques, such as Naive Bayes Classifier.

In conclusion, this thesis leverages two distinct data sets for the S&P 500 stocks and Bitcoin. While U.S. stocks and Bitcoin share some commonalities as investment assets, attracting global investors, they also exhibit substantial differences. As a decentralized peer-to-peer currency and volatile digital asset, Bitcoin represents a unique technology and inflation hedge, similar to digital gold. In contrast, individual S&P 500 stocks are part of the legacy financial system, representing a nuanced perspective across 500 companies' performances. For this thesis, the SEC 10-Q data set for each individual stock represents an ideal data set for developing innovative methodologies to measure textual differences utilizing Deep Learning and generative A.I. Simultaneously, the Reddit dataset for Bitcoin proves well suited for noise removal and sentiment analysis. In the end, the research demonstrates that the advanced generative A.I. and Deep Learning techniques outperform traditional natural language processing techniques across two distinct datasets.

Part I

Semantic Textual Analysis on 10-Qs

Chapter 2

Literature Review, Data Collection and Exploration

This chapter covers a literature review, data collection, and exploration of SEC 10-Q filings for S&P 500 stocks.

2.1 Background and Literature Review

2.1.1 Literature Review on SEC Filings

Among the most cited authors in applying text processing techniques to SEC financial filings are Loughran and McDonald (2011a). Their seminal work examines filings' readability, sentiment, and uncertainty, relying on financial dictionaries. An earlier contribution in the field comes from Li (2008), who introduced one of the first readability metrics for financial filings—the Fog index. This metric gauges the average length of a sentence and the percentage of complex words. Li found that companies with harder-to-read financial filings tended to have lower reported earnings. However, Loughran and McDonald demonstrated that the Fog index is a suboptimal metric for business documents. They proposed an alternative proxy for readability in the context of a 10-Q document, suggesting the natural log of the overall file size. Their findings indicate that firms with larger file sizes were associated with greater stock return volatility, analyst dispersion, and absolute earning surprises.

Historically, a prevalent approach among researchers has been to construct sentiment indicators using dictionaries to capture the document's tone. The lexicon approach involves tabulating the number of positive, negative, and uncertain words within the text. Early attempts, such as those by Hanley and G. (2010), utilized the Harvard General Inquiry dictionary to measure the tone of initial prospectuses (Form S1) from IPOs, which can be linked to first-day returns. Initially designed for social sciences, this dictionary posed limitations when applied to financial documents, as noted by Loughran and McDonald (2011b). As a result, this motivated the authors to create a custom financial dictionary that has since become the standard for measuring tone in financial documents. A significant disadvantage of the Harvard General Inquiry dictionary lies in its classification of words such as *tax*, *liability*, and *depreciation* as negative, which, in a financial context, does not accurately reflect their nature. These words, frequently found in 10-Q documents, are not inherently negative in the financial domain. While these dictionaries have been the standard in quantitative investment research, the evolving landscape of Natural Language Processing techniques, such as Transformers, is poised to exert pressure on their continued usage.

The research most relevant for the 10-Q section of this thesis centers around document similarity analysis. The earliest research comparing document similarity between financial filings originates with Brown and Tucker (2011), who found that stock price performance positively correlates with language changes when using a cosine similarity score on 10-Ks. The authors also found that the usefulness of changes in 10-Ks has deteriorated over time in terms of their impact on stock prices. In contrast, a popular paper by Cohen et al. (2020) presents an opposing perspective. Their study finds stock price responses through textual similarity methods applied to 10-Qs when constructing portfolios. In addition to the cosine similarity score, their textual difference scores incorporated various metrics such as Jaccard similarity, minimum edit distance, and simple similarity between quarter-on-quarter measures in 10-Qs.

Another sub-branch of research focuses on trend analysis of topic changes, exemplified by Dyer et al. (2017), by applying Latent Dirichlet Allocation (LDA) on SEC filings. Specifically, the authors suggest that the length of SEC filings has increased due to recent FASB and SEC mandates. Their findings indicate that three of the 150 topics—fair value, internal controls, and risk factor disclosures—significantly contributed to the overall extension. Building on this, Brown et al. (2017) delved into meaningful topics for predicting financial misreporting using samples from the SEC enforcement actions dataset. Their research revealed that an increased focus on these topics heightened the likelihood of financial misreporting.

While these traditional semantic textual difference approaches have provided valuable insights, they have inherent context and language structure limitations. There remains a notable gap in research utilizing the latest advancements in Deep Learning and Large Language Model technologies for analyzing financial SEC filings, aiming to extract insights into why companies modify their disclosures. In Chapter 3, we present our novel research findings using concatenated deep learning architectures and Large Language Model methodologies. However, before we delve into these findings, the rest of this chapter introduces the 10-Q data and conducts an exploratory analysis.

2.2 Data and Exploration

2.2.1 Data Collection

The research focused on the meticulous data collection for each stock within the S&P 500 universe from January 2000 to December 2019. Given the dynamic nature of the S&P 500 index, its constituents change over time, so we intentionally captured the historical composition. This approach was crucial to avoid survivor bias and ensure the inclusivity of all relevant data. Compiling this comprehensive dataset is the first step before we get to modeling, aligning with one of the main motivations of this thesis—to detect significant company failures through the analysis of substantial language changes from one 10-Q document to the next.

Figure 2.1 illustrates the temporal evolution of the number of stocks resulting from our data collection efforts. Our approach involves pulling a comprehensive stock list encompassing approximately 800 companies in the S&P 500 universe over the past twenty years.

The SEC filings data was sourced from Wharton Research Data Services (WRDS), where we extracted raw text data, SEC Central Index Code, and file size. The dataset also encompasses various date columns, including filing publication date,

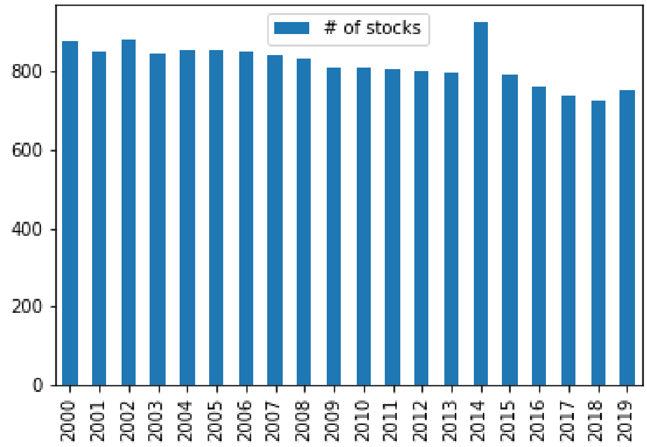

FIGURE 2.1: Number of stocks over time

initial filing submission, and filing review date. For this research, we specifically consider the filing publication date, as it marks the moment when the filing goes public and could potentially impact a company's stock price.

In the early stages of this research, we initially utilized web scraping on the SEC EDGAR website to extract all the filings. This approach involved systematically processing each filing, handling different data formats (i.e., HTML or text-based files), and removing tables from the documents. However, further investigation revealed that the same SEC filing data could be effectively retrieved from WRDS. As a result, this was more advantageous, as the WRDS SEC filing data is cleaner, contains reliable date-time stamps for realistic modeling, and is already structured in a Python DataFrame format. Additionally, although too late in the stage for this thesis, one can now retrieve SEC filing data directly from the SEC via a new API that facilitates the systematic collection of SEC filing documents while accessing specific sections.

Furthermore, WRDS was utilized for pulling reliable price data for each of the S&P 500 constituents to calculate the financial returns of stocks. Initially, the

WRDS price data did not incorporate stock split adjustments, requiring us to account for this by re-indexing the data. Since this thesis focuses on predicting idiosyncratic returns instead of overall market performance, each S&P 500 stock return was subtracted by the S&P 500 Index return to generate excess returns. This is standard practice in finance to eliminate the overall market effect, allowing for an assessment of whether a stock is underperforming or outperforming the overall S&P 500 Index. As a final step, the data was transformed into a classification problem by generating a binary variable: 1 if the excess return is positive and 0 if the excess return is negative. The subsequent discussion, the Management's Discussion and Analysis section, details how we extract the most relevant section of the 10-Q filing.

2.2.2 Management's Discussion and Analysis Section

Within the various sections of the SEC filings, boilerplate language is often prevalent, including topics such as Controls and Procedures or Minor Safety Disclosures. However, for this thesis, we focus on the "Management's Discussion and Analysis of Financial Condition and Results of Operations" section. This section allows company management to articulate their narrative in their own words. The significance of this section lies in its potential to contain language that reflects significant changes in a company's results. In particular, management details accounting changes, material shifts in a company's results, critical assumptions, and adjustments from previous years or quarters.

It is important to note that most of the literature omits discussing the practical challenge of obtaining critical sections of the SEC filings. The difficulty arises due to the inconsistencies in how the financial filings have been reported over time.

Loughran and McDonald (2011a) discuss these challenges at length, attributed to several factors: 1) Some companies may not include various sections, 2) Others may be inconsistent in where certain sections are reported, and 3) There can be variations in the naming of sections. Despite these challenges, achieving a close to 70 percent hit rate is considered favorable, as discussed with McDonald, who employed regex to extract this information. For the 10-Q dataset, we achieved a 69 percent hit rate, providing a sufficiently robust sample size. The ultimate goal for this dataset is to serve as a foundation for training Deep Learning models and systematically employing Large Language Models to predict whether a company's stock price will rise or fall over the next 180 days.

An example of Apple's Dec 2021 10-Q filing can be found on the SEC website here:

https://www.sec.gov/Archives/edgar/data/320193/000032019322000007/aapl-20211225.htm

To illustrate further, see below the first part of Apple's Management Discussion and Analysis in the Dec 2021 10-Q filing (Note: we only report the first few pages for brevity, but please see the URL for further details).

Quarterly Highlights

Business Seasonality and Product Introductions

The Company has historically experienced higher net sales in its first quarter compared to other quarters in its fiscal year due in part to seasonal holiday demand. Additionally, new product and service introductions can significantly impact net sales, cost of sales, and operating expenses. The timing of product introductions can also impact the Company's net sales to its indirect distribution channels as *these channels are filled with new inventory following a product launch, and channel inventory of an older product often declines as the launch of a newer product approaches. Net sales can also be affected when consumers and distributors anticipate a product introduction.*

COVID-19 Update

The COVID-19 pandemic has had, and continues to have, a significant impact around the world, prompting governments and businesses to take unprecedented measures, such as restrictions on travel and business operations, temporary closures of businesses, and quarantine and shelter-in-place orders. The COVID-19 pandemic has at times significantly curtailed global economic activity and caused significant volatility and disruption in global financial markets. The COVID-19 pandemic and the measures taken by many countries in response have affected and could in the future materially impact the Company's business, results of operations, and financial condition, as well as the price of the Company's stock.

During the first quarter of 2022, aspects of the Company's business continued to be affected by the COVID-19 pandemic, with a significant number of the Company's employees working remotely and certain of the Company's retail stores operating at limited capacity or temporarily closing at various times. The Company has reopened substantially all of its other facilities, subject to operating restrictions to protect public health and the health and safety of employees, and it continues to work on safely reopening the remainder of its facilities, subject to local rules and regulations. At times, some of the Company's component suppliers and logistical service providers have experienced disruptions, resulting in supply shortages that affected sales worldwide. Similar impacts or other disruptions could occur in the future.

The extent of the continuing impact of the COVID-19 pandemic on the Company's operational and financial performance is uncertain and will depend on many factors outside the Company's control, including the timing, extent, trajectory, and duration of the pandemic, the emergence of new variants, the development, availability, distribution, and effectiveness of vaccines and treatments, the imposition of protective public safety measures, and the impact of the pandemic on the global economy and demand for consumer products. Refer to Part I, Item 1A of the 2021 Form 10-K under the heading "Risk Factors" for more information.

2.2.3 Data Preparation

The dataset spans from January 1995 to December 2019, containing nearly 28,669 10-Q documents after a series of data pre-processing steps. To ensure data accuracy, the 10-Qs files containing a zero file size are removed to prevent any erroneous records. In terms of text pre-processing for the Deep Learning section, all digits and punctuation such as "$", ",", and ":" are excluded. For Large Language Models, we keep the original text untouched as punctuation context can be helpful to the model.

We implement additional corrective measures to address potential cases where the Management Analysis and Discussion section may have captured incorrect text segments. Specifically, 10-Qs with word counts below a minimum threshold and greater than a maximum threshold are excluded. Additionally, we evaluate the total word count from one document to the next. The respective row is removed if the absolute difference surpasses the 95th percentile. This approach has proven effective in filtering out any potentially noisy data. Following these processing steps, the dataset is refined to 22,002, with 17,863 utilized for training and validation and 4,139 reserved for the test dataset.

2.2.4 Frequency Analysis

A typical representation in text processing involves examining the rank order of the top occurring words in the dataset, aligning with Zipf's Law. Figure 2.2 illustrates this power law distribution, emphasizing that the most prevalent words are typically within the top 10. It's important to note that this chart is conducted after the removal of stopwords (e.g., “the”, “an”, “a”, etc.). If stopwords were retained, the percentages of the top words could increase to 20-25 percent. Removing stopwords reduces noise for non-sequence-based concatenated deep learning models introduced in the next section (such as Max Embeddings and CNN, as opposed to LSTM).

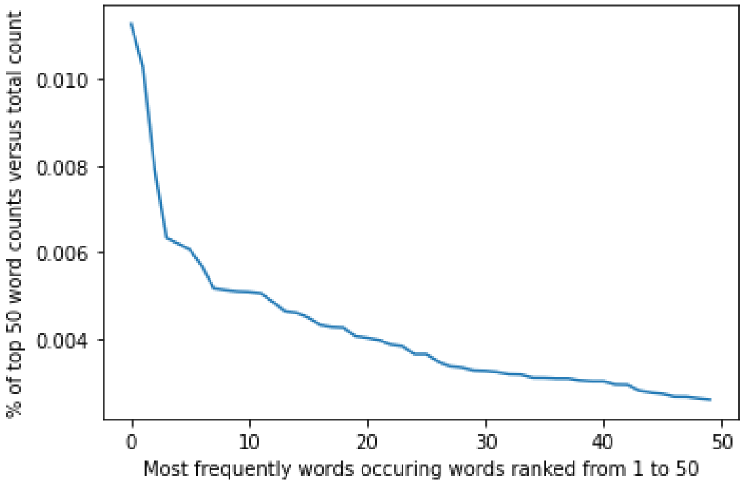

FIGURE 2.2: Zipf's law demonstrating the power law distribution

To identify the most common words within the dataset, Table 2.1 illustrates the top 40 words present within our corpus. This word list aligns with expectations, given that the Management Discussion and Analysis sections typically discuss operating, income, sales, etc. Companies use this section to disclose context around their reported financial numbers and any other risks. If one were to drill down per each sector or industry group within the S&P 500, such as the energy or technology sectors, the word list would contain more idiosyncratic and sector-specific terms.

Table 2.1: List of top 40 most common words

Rank	Words	Frequency	%	Rank	Words	Frequency	%
1	net	1946713	1.1%	21	rate	697729	0.4%
2	income	1774833	1.0%	22	expense	688486	0.4%
3	cash	1369661	0.7%	23	business	671592	0.4%
4	operating	1095557	0.6%	24	expenses	664642	0.4%
5	sales	1072329	0.6%	25	debt	633324	0.3%
6	interest	1048410	0.6%	26	revenues	632332	0.3%
7	total	983239	0.6%	27	capital	602726	0.3%
8	due	895283	0.5%	28	revenue	585439	0.3%
9	tax	887846	0.5%	29	based	580550	0.3%
10	related	882546	0.5%	30	per	568532	0.3%
11	costs	880563	0.5%	31	company's	566791	0.3%
12	assets	873113	0.5%	32	loss	562195	0.3%
13	statements	841184	0.5%	33	notes	554038	0.3%
14	consolidated	804444	0.5%	34	securities	552671	0.3%
15	increase	797882	0.5%	35	market	539584	0.3%
16	value	779398	0.5%	36	certain	539154	0.3%
17	stock	749335	0.4%	37	fair	537886	0.3%
18	compared	741254	0.4%	38	term	536664	0.3%
19	increased	738632	0.4%	39	including	528685	0.3%
20	credit	704386	0.4%	40	approximately	525493	0.3%

Inspecting the word distribution per each 10-Q document across the entire corpus, Figure 2.3 illustrates a positively skewed distribution. This distribution was used as a guide to determine the thresholds for data cleaning per document in cases where the Python regex failed.

Figure 2.4 provides insights into the average word count over time per document. The graph illustrates an initial increase, stabilizing from 2004 onward.

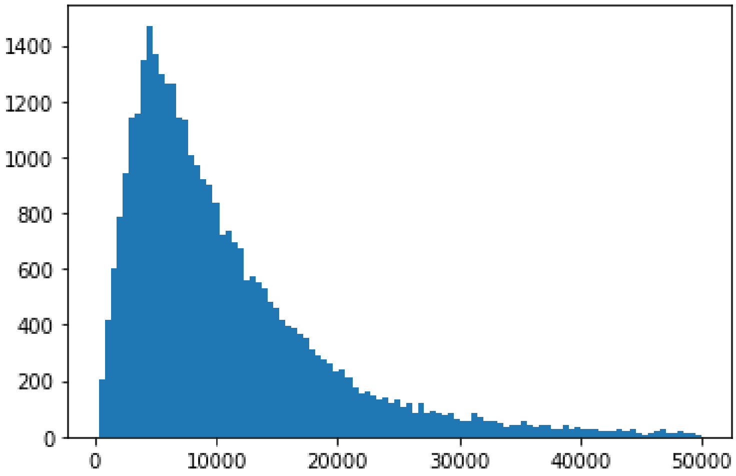

Figure 2.3: Histogram of total word frequency

This observation suggests that, in general, companies tended to maintain consistent boilerplate language in their 10-Q filings unless compelled to disclose material changes.

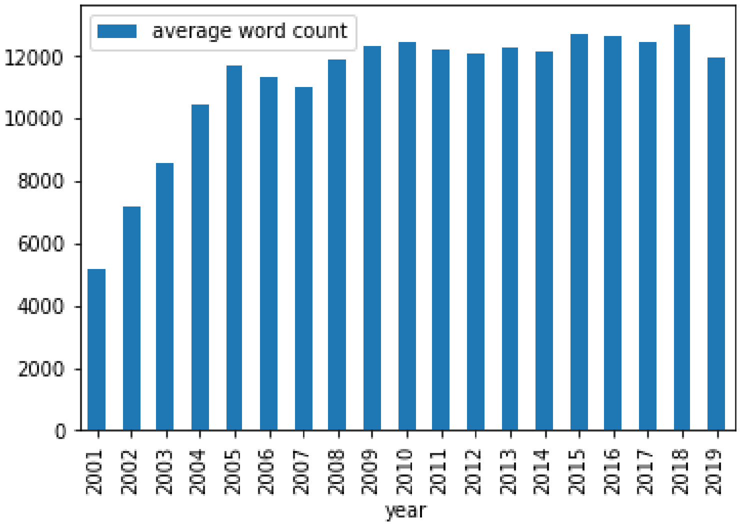

Figure 2.4: Average word count by year

2.2.5 Background on Cosine Similarity

Cosine similarity is a standard benchmark metric in NLP for estimating the similarity between two documents. Given its widespread use in existing research papers within the SEC literature review section, this metric is adopted as our baseline for evaluating any new models introduced. Mathematically, it represents the cosine of the angle between two vectors in a multidimensional space. Given two vectors A and B, the cosine similarity can be expressed using the following equation.

$$\text{Cosine similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{2.1}$$

Where A_i and B_i represent the vectors A and B, respectively, with values of 1 indicating the documents are identical and -1 if they are opposite.

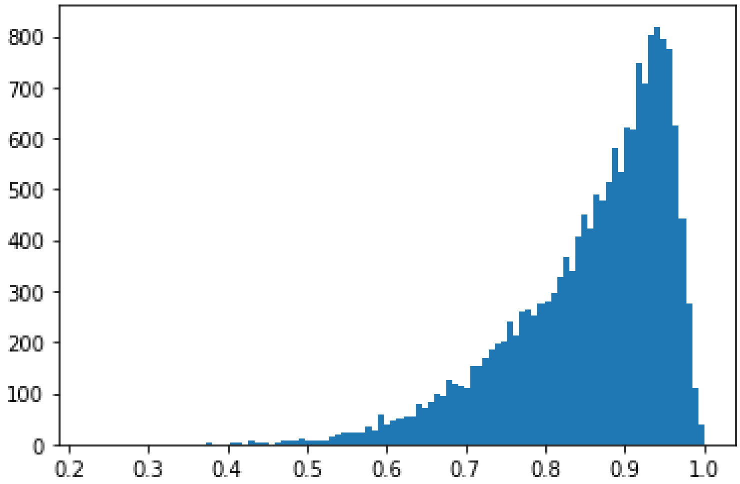

FIGURE 2.5: Histogram of Cosine Similarities

Cosine similarity is computed across our text corpus, and Figure 2.5 illustrates the overall distribution of this metric. Most of the data appears to be broadly classified as similar language, given that the mean is 0.9, with 1 representing a perfect match. This observation aligns with the notion that most companies tend not to update their 10-Q language unless something material compels disclosure.

2.3 Summary

This chapter has provided an in-depth exploratory analysis of the textual data within the Management's Discussion and Analysis section across 10-Q filings for S&P 500 constituents. A comprehensive dataset was compiled, consisting of 28,669 filings spanning from 1995 to 2019. Initial frequency analysis revealed common phenomena in textual data, such as the power law distribution according to Zipf's law, and established data cleaning thresholds.

Traditional metrics, such as cosine similarity, were introduced as benchmarks for subsequent sections. The cosine similarity calculation for 10-Q filings indicated a high average similarity score 0.9 between consecutive filings. This finding supports the notion that companies maintain boilerplate language from quarter to quarter unless there is a material disclosure. An investigation over time revealed the average word count per filing has stabilized since 2004, further supporting the notion of minimal change in disclosure content.

In summary, this comprehensive exploratory analysis provides a strong foundation for the subsequent section, where concatenated deep learning and large language models are employed to predict whether excess returns on S&P 500 stocks will rise or fall over the next 180 days. Specifically, for concatenated deep learning models, we consider max embeddings, convolutional neural nets, and Long Short Term Memory models. Claude2 is utilized for its extended context windows to process large 10-Q sections for the Large Language Model, exploring both zero-shot and in-context methodologies.

Chapter 3

Semantic 10-Qs Matching with Deep Learning and LLMs

This chapter provides background on traditional Deep Learning approaches covering Convolutional Neural Networks, Recurrent Neural Networks, and Semantic Text Matching. The following sections introduce how concatenating these Deep Learning architectures can be used to identify duplicate questions, inspired by Quora's engineering blog[1]. We then extend this concatenating Deep Learning framework for financial markets prediction, where instead of a duplicate question, we substitute quarterly filings per the current quarter comparing the previous quarter where the ultimate classification task is predicting if the excess return of a stock will be positive or negative. This approach is novel from the time series perspective as it allows the various deep learning architectures to learn semantic differences between 10-Q filings for predicting the sign of the stock's excess return. A significant contribution of this thesis in this section is adapting techniques that are effective for duplicate question detection in Quora to the new problem context

[1] https://quoraengineering.quora.com/Semantic-Question-Matching-with-Deep-Learning

of evaluating the similarity between financial disclosures. Later in the thesis, the results demonstrate significant predictive accuracy over the benchmark, indicating why concatenating deep learning frameworks can detect subtle but meaningful language changes between two 10-Q filings.

With the recent advances in Large Language Models (LLM), the chapter's next part uses Anthropic's Claude2. Claude2 at the time is advantageous over OpenAI's ChatGPT or GPT4 given its 100k long context windows, which is ideal for 10-Q filings where the typical length of just the Management Discussion and Analysis section averages around 50k words (equates to about 70k tokens). This research introduces a new methodology called Summarize and Sentimentize that harnesses large language models to process lengthy financial disclosures into condensed, insightful summaries, evaluating if linguistic changes in these generated texts can predict excess returns for S&P 500 stocks. Specifically, the Claude2 model summarizes fundamental shifts across filings related to risks, profitability, and business conditions. By compressing filings into summaries and assigning a sentiment score on these summaries, we find this methodology can predict excess returns for S&P 500 stocks. We examine two training approaches for constructing this approach: 1) Zero-Shot prompting with natural instructions and 2) In-context learning using previous filing summary examples with excess returns.

In this chapter, the thesis makes a novel contribution by introducing new ways to concatenated Deep Learning Models on financial disclosure data that can process semantic textual differences between two 10-Q filings for stock market prediction. Further, a new Summarize and Sentimentize approach is developed using summarization methods employing Claude2, where the summaries are related to significant risks, profitability, legal considerations, and market pressures. The final step generates a sentiment score from these summaries that is either “positive”, “neutral” or “negative”. Collectively, these methodologies go beyond conventional natural language processing techniques like lexicons and cosine similarity, which are typically used for financial disclosures and represent contributions for this thesis.

3.1 Introduction to NLP with Deep Learning

Some of the primary NLP tasks that are typical applications include entity name recognition, parts of speech tagging, sentiment analysis, semantic textual similarity, text classification, and relation extraction, to name a few. As Goldberg (2016) points out in his Primer on Neural Network Models for NLP, traditionally, NLP techniques usually involve training linear models (i.e., Support Vector Machines or Logistic regression) with high dimensional sparse features. In contrast, with the advantages in both computational power and neural network architectures, Deep Learning for NLP has begun to significantly outperform on the majority of the benchmarks as tracked by one of the better GitHub folders maintained by Sebastian Ruder[2].

In the following sections, we introduce three common neural network architectures: Convolution Neural Networks, Long-Short-Term Memory, and Semantic Text matching. These architectures are ultimately used within a concatenating architecture, allowing the model to detect language changes at different levels and abstractions.

[2] see website here https://github.com/sebastianruder/NLP-progress

3.1.1 Convolutional Neural Network

Convolutional Neural Networks (CNN) are a popular neural network used heavily within computer vision. It is inspired by the visual cortex of animal brains, where receptive fields process multiple filters of input. CNN maps multiple convolution layers, each performing a convolution operation that applies some function to process inputs that can ultimately be filtered for prediction.

An illustration of a CNN can be seen in a classic paper Sermanet and LeCun (2011) that recognizes traffic signs for autonomous driving. Figure 3.1 shows the overall architecture that first processes the initial input image and is fed forward through two stages of convolutions and subsampling (max pooling most common approach), where ultimately it uses a linear classifier for the final operation. To account for non-linearities, the author uses *tanh*, where it was convention to use sigmoid functions at the time.

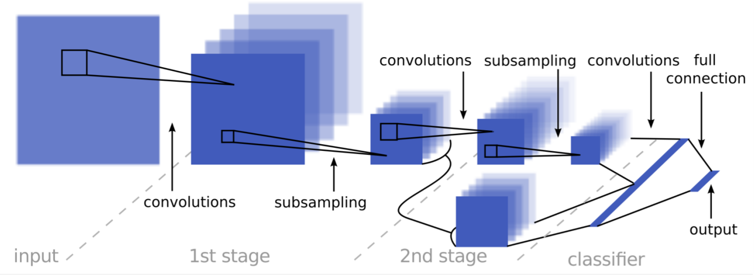

Figure 3.1: Convolutional Neural Network Traffic Signs Example

In Natural Language Processing, CNNs are just as competitive with sequenced-based models when on NLP tasks such as sentiment analysis. Goldberg (2016) and Zhang et al. (2018b) highlight one of the main contributing reasons why CNNs are helpful is their ability to extract information from features locally to find clues about class membership. For instance, it may boil down to a few keywords or phrases that might be the most important for predicting class outcomes, no matter where they appear in the document. CNNs can help isolate these local features, regardless of whether they appear in the first or later part of the document. In the context of the 10-Q dataset, CNNs may have an advantage over LSTM models discussed in the next section, given how long the average document length is (averages 5000 words) and how LSTM models suffer from vanishing or exploding gradients for extremely long sequences.

3.1.2 Recurrent neural networks

Recurrent neural networks (RNNs) are a branch of neural networks designed for processing sequential data. They are popular in speech recognition and natural language processing. Given the sequential nature of language, even if sequences are extremely long, we will be using RNNs as one of the competing models for the thesis.

A RNN model as specified in Graves (2011), is an input sequence $x = (x_1, ..., x_T)$ that can compute a hidden vector sequence $h = (h_1, ..., h_T)$ for an output sequence $y = (y_1, ..., y_T)$ through the below equations when going from $t = 1$ to T.

$$h_t = \gamma(W_{xh}x_t + W_h h_{t-1} + b_h) \tag{3.1}$$

$$y_t = W_{hy}h_t + b_y \tag{3.2}$$

where b are the bias vectors, W represents the weight matrices and γ is some nonlinear function.

A type of RNN model called Long Short-Term Memory (LSTM) will be utilized as one of the deep learning models. The LSTM has an advantage over traditional

RNNs in that it has the capacity to store long-term information through memory cells. The LSTM framework is advantageous for exploiting long-term information contained within financial data. As specified by Graves (2011), γ above can also be represented as a composite function with the following equations:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{3.3}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \tag{3.4}$$

$$c_t = f_t c_{t-1} + i_t \tanh\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right) \tag{3.5}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_{t-1} + b_o) \tag{3.6}$$

$$h_t = o_t \tanh\left(c_t\right) \tag{3.7}$$

σ is the logistic sigmoid function, and i, f, c, and o are the input, forget, cell, and output gates for activation vectors. These gates enhance the ability of the model to forget or keep relevant information by updating the weights. The traditional RNN utilizes information from the previous time step so that an LSTM can incorporate immediate and more extended time scales.

Typically, LSTM or sequenced-based models are popular Deep Learning models within NLP, given that language is a sequenced data set. However, our SEC filings data has an average of 5000 words per document, and LSTM models typically struggle with long sequences past 500 words.

Removing stopwords within NLP is traditionally a famous text-cleaning stage where words such as “the”, “is”, and “and” are removed to focus on more high-impact words. The standard NLTK python package stopwords removal is utilized for the CNN models and Sum Embeddings. However, the stopwords are kept for LSTM models. Our decision to keep stopwords in our LSTM model is primarily because researchers such as Jeremy Howard have noted in his FASTAI NLP course, especially for RNN-based models, that stopwords can contain meaningful information about the context of a sentence. Refer to the Transformer section B.1 in the Appendix that illustrates Deep Learning architectures like Transformers can still have an autoregressive nature but can do the model training in a parallel processing manner versus LSTM models that train sequentially.

3.1.3 Semantic Text Matching

Semantic text matching is another active area of research that investigates the similarity of sentences and questions for various tasks. One recent practical task has come from the company Quora, where they have recently tried to identify duplicate questions to minimize redundancy on their platform as described in their 2017 engineering blog[3]. They estimate three Deep Learning approaches with similar accuracy rates, including LSTM with concatenation, LSTM with distance and angle, and decomposable attention. Additionally, another well-known labeled sentence pair data set curated by Bowman et al. (2015) has aided with natural language inference by understanding contradictions and hypotheses.

For this thesis, we can draw parallels with Semantic Question Matching by comparing textual differences between different quarters in the 10-Q documents. This can help identify if any key language has changed significantly that can be linked to under or overperformance in excess returns.

[3] https://quoraengineering.quora.com/Semantic-Question-Matching-with-Deep-Learning

One of the most common architectures for comparing sentence similarity was specified by Merity (2016), which computes a simple summation of two Glove Word Embeddings representing each document pair. One prominent feature of this model architecture can be adjusted so that the max operator on the word embeddings can be used per the Quora adaptation instead of the sum. Note that summation across the dimension of word embeddings is a simple but effective way of aggregating meanings across individual words. In contrast, for each dimension in the word embeddings, the max takes the most significant value, which can be helpful to emphasize the most critical aspects of each question. As a straightforward illustrative example, consider a simplified two-dimensional example using "cash" and "debt".

- Word "cash": [0.8, 0.4]
- Word "debt": [0.9, 0.5]

For instance, dimension 1 (i.e., column 1) could represent the financial health of a company, whereas dimension 2 (i.e., column 2) might represent the presence of debt or financial obligations.

Figure 3.2 shows the modified model adaptation by Quora where once the two embeddings are combined, it gets fed into four layers with every 200 neurons, ReLU activation functions, sigmoid activation for the final layer with binary cross entropy for loss function and Adam for optimization.

As per the Deep Learning diagram in figure 3.2, other architectures can be substituted per each question. This thesis also substitutes in both CNNs and Bidirectional LSTM before being concatenated into the neural network architecture.

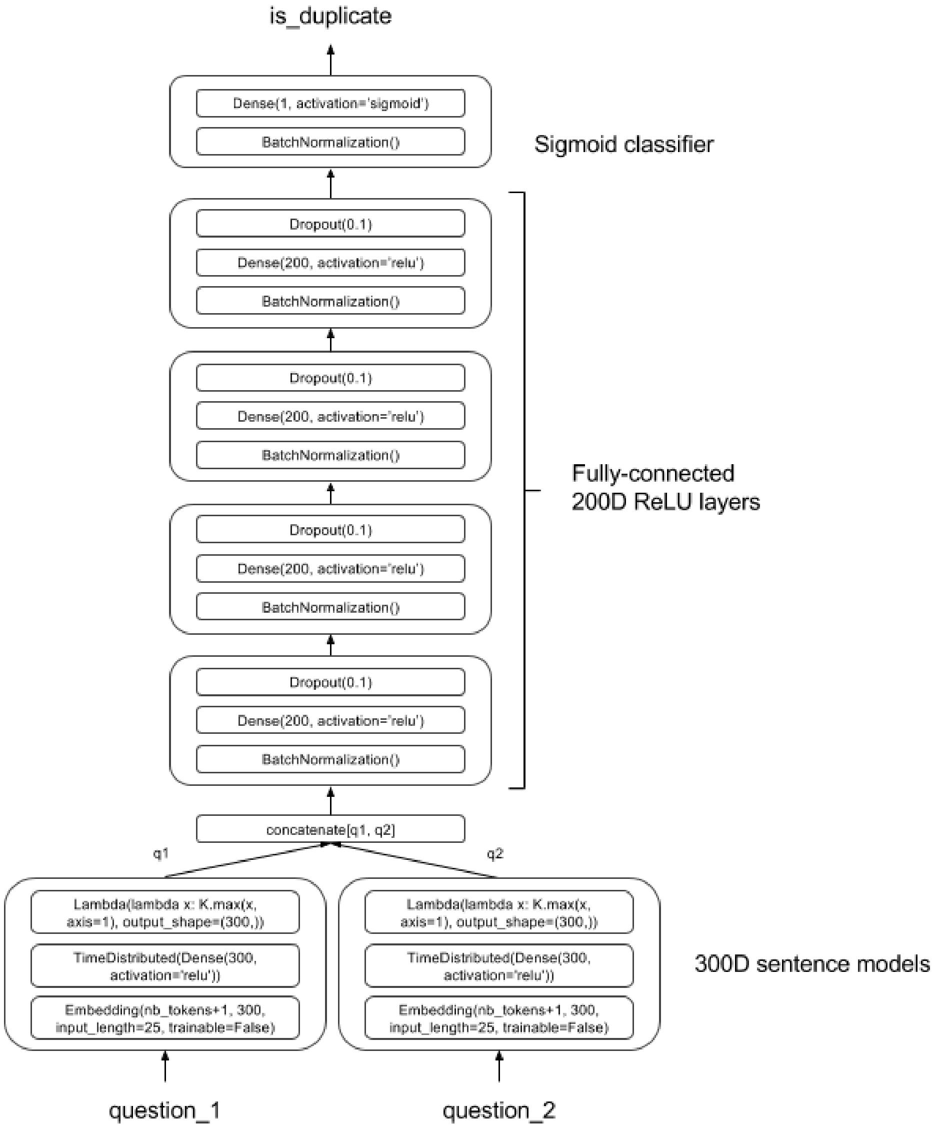
is_duplicate
Dense(1, activation='sigmoid')
BatchNormalization()
Sigmoid classifier
Dropout(0.1)
Dense(200, activation='relu')
BatchNormalization()
Dropout(0.1)
Dense(200, activation='relu')
BatchNormalization()
Dropout(0.1)
Dense(200, activation='relu')
BatchNormalization()
Dropout(0.1)
Dense(200, activation='relu')
BatchNormalization()
Fully-connected 200D ReLU layers
concatenate[q1, q2]
q1
q2
Lambda(lambda x: K.max(x, axis=1), output_shape=(300,))
TimeDistributed(Dense(300, activation='relu'))
Embedding(nb_tokens+1, 300, input_length=25, trainable=False)
Lambda(lambda x: K.max(x, axis=1), output_shape=(300,))
TimeDistributed(Dense(300, activation='relu'))
Embedding(nb_tokens+1, 300, input_length=25, trainable=False)
300D sentence models
question_1
question_2

Figure 3.2: Quora question matching architecture

Separating the current quarter and previous quarter 10-Q into its neural network architecture, allows the overall model to learn the appropriate model weights to understand if semantic language differences can predict the sign of excess stock returns 180 days out. Each of the 10-Q documents represents a different point in time, so the deep learning architecture has a time series component that is inherently built into the architecture.

3.2 Methodology Overview

The major goal for compiling the 10-Q documents is to determine if language between the current versus the previous quarter can classify if the excess return of a stock is positive or negative over the next 180 days. The overall objective at hand can be formulated as follows:

$$f(doc\ 10Q_t, doc\ 10Q_{t-1}) - > 0\ or\ 1 \tag{3.8}$$

0 represents a stock with a negative excess return, and 1 is a positive excess return.

One central hypothesis for this thesis is that Deep Learning models can provide additional forecasting power over and above simplistic text similarity measures such as Cosine Similarity. We use Cosine similarity within a logistic classification model as the primary benchmark model. For the overall methodology, we test three different deep learning concatenating architecture models versus our benchmark defined as follows:

- Cosine Similarity Logistic Regression (as benchmark model)
- Max of Embeddings Concatenation
- CNN Concatenation
- Bidirectional LSTM Concatenation

The following sections will go into more depth about each of these architectures.

3.2.1 Cosine Similarity Logistic Regression (as benchmark model)

As per the previous section 2.2.5 where Cosine Similarity is defined, this is fed into a traditional logistic regression model. We use logistic regression as part of the Sklearn Python package, where L2 regularization is applied by default. Logistic regression is used as the basis for a benchmark model for the rest of the concatenated models defined below.

3.2.2 Max of Embeddings

As introduced in the Semantic Text Matching section 3.1.3, this approach takes what was described in that section. Still, it implements it using 10-Q filings using the current and previous quarters predicting the sign of excess stock returns. As a Deep Learning benchmark, the model hyperparameters and architecture are kept the same as the Quora example for consistency. Specifically, we use 200 neurons across four layers, Relu for activation functions and the final layer using the sigmoid activation function.

3.2.3 Concatenated CNN

Before reaching the concatenation layer, the CNN model architecture needs to be defined. The first processing step is to create a numerical representation of each token by feeding and training an embedding matrix using Glove Embeddings with 300 as the dimension layer (see 3.2.4.1 below) and the input length set to 5000.

Note that CNN models can handle more considerable input lengths than LSTM models due to the vanishing gradient problem they suffer.

The model hyperparameters are defined as follows. First, we feed the embeddings layer into a Conv1D[4] with filters set to 100, kernel size set to 4, and the activation layer set to Relu. Dropout is then applied using a 50% threshold using Maxpooling1D[5] with pool size set to 2. Finally, the architecture is fed into a flattening layer. The CNN architecture is repeated twice per the 10-Q quarterly filings before it gets concatenated.

Figure 3.3 illustrates the entire concatenated CNN structure where ultimately the CNN concatenated layer gets passed into two more fully connected layers where the first has 30 neurons and the second containing 15 neurons with each using a 50% dropout with Relu as the activation function. At the last step, it gets passed into a Sigmoid activation function predicting two classes. The optimizer used was Adam with a binary cross entropy loss function.

3.2.4 Bidirectional LSTM Concatenation

Figure 3.4 illustrates the concatenated BiDirectional LSTM architecture. Note that this architecture contains an additional layer with the total amount of neurons going from 500 to 100. Note that the experimentation process improved accuracy when the number of neurons decreased as they passed through each fully connected layer. Further, accuracy improved using a BiDirectional LSTM instead of a vanilla LSTM, allowing learning from right to left and left to right.

[4] Conv1D layer is a one-dimensional convolutional operation, which is ideal for text processing applications where it helpful to extract features from sequential data.

[5] Maxpooling1D is used, representing a one-dimensional that reduces the feature extraction dimension even further.

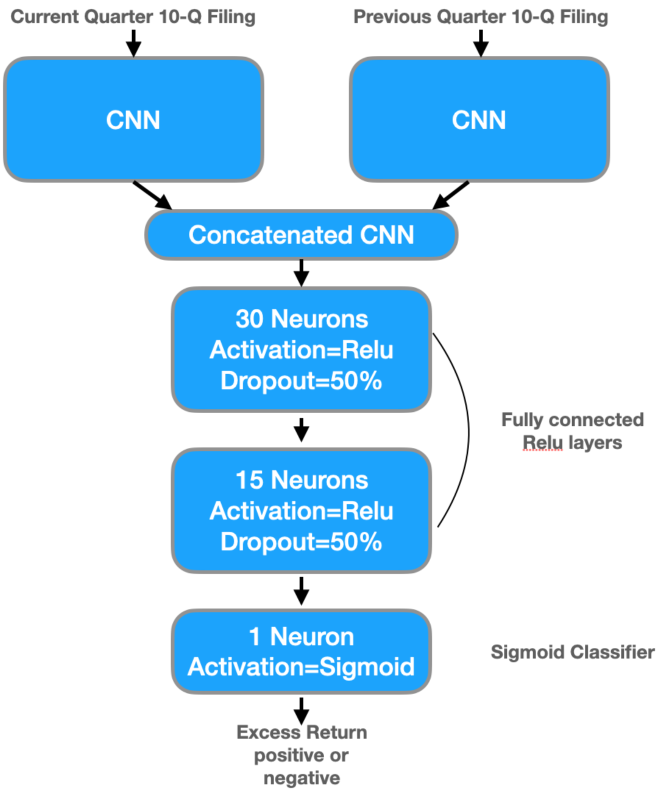

Figure 3.3: Concatenated Convolutional Neural Network architecture

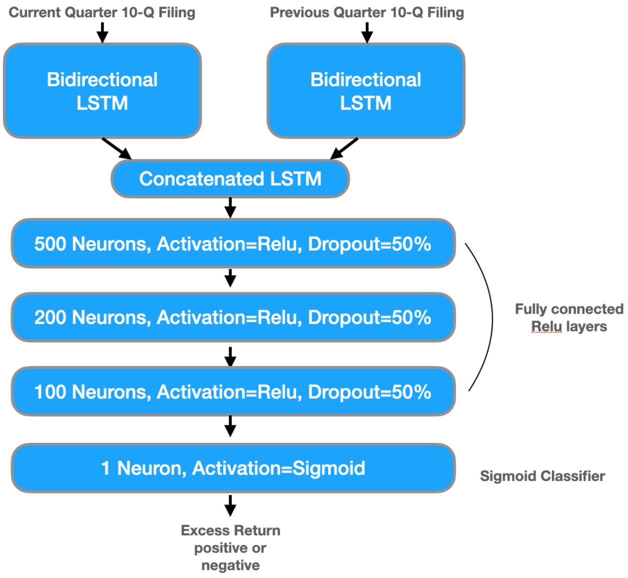
Current Quarter 10-Q Filing
Previous Quarter 10-Q Filing
Bidirectional
LSTM
Bidirectional
LSTM
Concatenated LSTM
500 Neurons, Activation=Relu, Dropout=50%
200 Neurons, Activation=Relu, Dropout=50%
100 Neurons, Activation=Relu, Dropout=50%
Fully connected
Relu layers
1 Neuron, Activation=Sigmoid
Sigmoid Classifier
Excess Return
positive or
negative

Figure 3.4: Concatenated BiDirectional LSTM architecture

3.2.4.1 Transfer Learning with Glove Embeddings

We utilize the pre-trained Glove embedding introduced by Pennington et al. (2014), which acts as a Transfer Learning mechanism for all the Deep Learning models listed above. Glove embeddings are trained on five different corpora from either Wikipedia (averaging 1.5 billion tokens) to a Common Web Crawl (42 billion tokens), where the goal is to find the best word-to-word analogies using a global statistics approach.

The last step sets "Trainable=True" within the Keras python package, which fine-tunes the embeddings matrix on the 10-Q filing financial corpus. This step makes the matrix domain specific to financial text data while leveraging the power of transfer learning from billions of tokens trained on Wikipedia and the Common Crawl.

3.2.4.2 Other hyper-parameter assumptions

Another critical assumption is how long the sequence length should be for padding and if it is pre or post-padding. First, padding length is essential to standardize the data across multiple documents. For the latter point, typically, most NLP use pre as they are focused on predicting the next word in a sequence, but in our case, we use post since the first part of the sequence is the most important, and the ultimate target variable is a classification model. For the concatenated Max Embeddings and concatenated CNN, the max sequence length taken was 5000 as this was the average document length after only examining the Management Discussion and Analysis section. However, for the concatenated BiDirectional LSTM, the padding was set to 500 due to the vanishing gradient problem.

We utilize the callback features for all models, where the best model is saved as a function of the validation accuracy. Generally, Deep Learning models tend to begin to overfit, and this ensures the best model is picked before this occurs, even with high dropout rates. Additionally, all models use a 50 percent dropout rate, binary cross entropy for the loss function, and Adam optimization.

3.3 Beyond traditional NLP using LLMs

We are still in the early stages of understanding what Large Language Models can and cannot do and to what extent they will challenge existing Natural Language Processing approaches, ranging from topic discovery, sentiment, summarization, name entity recognition, and question and answer. The performance of all these NLP tasks can vary significantly depending on the domain application, specifics on the model implementation, and exploring fine-tuning. In general, one of the research findings for this thesis is that LLMs are better at broad-based tasks such as sentiment analysis. However, LLMs need help to classify and determine more nuanced topics.

This section introduces a Summarize and Sentimentize approach using Claude2 as the base Large Language Model. Note that we used Claude2 specifically over ChatGPT at the time, given its ability to process large context windows, which is suitable for large 10-Q documents. Further, this thesis introduces a Zero-Shot and an In-Context learning approach for evaluating the performance of the Summarize and Sentimentize approach. First, we introduce some nuances about what Claude2 is in the following section.

3.3.1 Introduction to Claude2

Claude2 is based on the decoder-only transformer architecture (see the Introduction to Transformers section B.1 in the Appendix for further details). As detailed by Bai et al. (2022), the original Claude model parameter counts can range from 10 million to 52 billion with a fixed context window of 8192 (Note that the context window is how many tokens can be passed as an input to generate a response).

The next evolution within the Claude models is Claude2, which has a total parameter count of 132 billion with a context window of up to 200k tokens, but for the API, they have restricted it to only 100k max size. Claude2 significantly outperforms other Large Language Models in handling longer context windows, compared with OpenAI's ChatGPT context window of about 8k. The context window capability is critical when dealing with SEC filings, as the median word length size is around 50k. Note for this thesis, the Management and Discussion Analysis section from figure 2.3 the median word size is 10k with the max up to 50k. Hence, for this section, there was an emphasis on using Claude2 over ChatGPT for the base Large Language Model to develop our Summarize and Sentimentize methodology. Note for the sample augmentation in the Sentiment modeling with noise removal in Chapter 5; the thesis used ChatGPT as the token lengths are relatively small.

Claude2 is also augmented using the Reinforcement Learning from Human Feedback approach as detailed by Ouyang et al. (2022) to fine-tune a base-level language model using a reward signal derived from human evaluators. The authors then collect data from over 40 human labelers on how they rate what they would prefer to see using reinforcement learning via proximal policy optimization (PPO) as detailed by Schulman et al. (2017).

3.3.2 Zero-shot LLM Summarize & Sentimentize

3.3.2.1 Methodology

See algorithm 1 designed for the Summarize and Sentimentize, where it first summarizes the 10-Q language changes due to business risks, profitability, legal and market pressures with a final step of assigning a sentiment score based on the tone of that summary. In contrast to traditional NLP methods, harnessing the power of large language models is a distinct advantage of utilizing the power of reason over simplistic cosine textual similarity measures. This type of LLM processing can allow one to focus on the parts of the text that matter the most.

The primary prompt instructions are defined below. These instructions are passed in per each 10-Q, compared with the previous quarter's 10-Q, and asked of Claude2 for sentiment extraction based on how the language is changing due to profitability, business risks, etc.

Prompt instructions are defined as:

"Please only respond with one word representing a continuous score between zero and one indicating if the 10-Q language changed due to major business risks, profitability, legal, market pressures, etc., and zero indicating maximum change and 1 indicating no change. Then please summarize with short paragraph with 3-4 sentences of any significant changes that might have an impact on the companies underlying business profitability and if no major changes then simple say 'no changes'. Then based on the summary, indicate if sentiment will be 'positive' or 'neutral' or 'negative' for the company and make this only one word only. And then separating the score and the summary and sentiment word by a @ symbol."

Algorithm 1 Zero-shot LLM Summarize & Sentimentize

```
Input: Pass in Prompt Instructions
for each data item $d_i$ in the dataset do
    Please see the current 10-Q filings $d_i$ and the previous quarters $d_{i-1}$
    Pass Prompt Instructions
    Send the prompt to Claude2 and receive a response $r_i$
    Process $r_i$ to extract "positive", "neutral" or "negative"
end for
```

Note that we first pass in the data from the current and previous 10-Q quarters as background context data before passing the prompt instructions through the prompt definition at the end. This data processing is done deliberately as research from Liu et al. (2024) found that accuracy improves if the most critical information within the context passed in as part of the prompt resides either in the beginning or the end. Specifically, the authors find that accuracy tends to be worse when the most critical information resides in the middle of the context that is passed (Hence the title of their paper "Lost in the Middle"). As a result, we consider the main prompt instructions as part of the algorithm 1 for how to Summarize and Sentimentize the 10-Q documents, the most critical piece of information that is crucial for the large language model to understand. Further research would explore additional sensitivities involving the location of this relevant information, but for now, we rely on their research to justify this approach.

Lastly, we use the default hyper-parameters within Claude2 except for setting the temperature parameter to zero. Temperature controls the amount of randomness per response. If the temperature equals one, the LLM is more creative, while if it is set to zero, the purpose is for more analytical tasks. In our case, where the objective is to determine what language has changed from a profit and loss and other legal risk considerations, choosing a temperature equal to zero is the appropriate setting.

3.3.2.2 Advance Micro Devices Case Study

To better understand the Zero-shot Summarize and Sentimentize approach, this section examines the models generated summaries and associated sentiment assignments for Advanced Micro Devices (AMD), a popular stock within the S&P 500 universe. Presenting results for AMD is just an illustrative deep dive on one stock, but in the Results section, we report across the entire universe of S&P 500 stocks.

As per Wikipedia, AMD was founded in 1969 as an American semiconductor company that designs and manufactures computer processors, graphics processing units (GPUs), and related technologies. The examples below showcase two Summarize and Sentimentize examples for AMD. Specifically, the URLs are shared per each 10-Q document for both 2013-10-30 and 2014-05-01 and the subsequent period for both 2014-05-01 to 2014-07-31[6]. For the first period comparison from 2013-10-30 to 2014-05-01, the summary indicates a decline in profitability due to challenges in the PC market, and the model assigned a negative sentiment where the negative excess return over the next 180 days was down -39.5%. The following 10-Q reporting in 2014-07-31 indicated the reporting was slightly more consistent with the model assigning a neutral sentiment. The excess return over the next 180 period for this was down -25.5%, indicating some carryover from the previous period. From figure 3.5, AMD's stock price history between 2013 and 2016 shows the initial drop in stock price occurred around Sept 2014, lasting for a year or so.

10-Q filing for Advanced Micro Devices on 2013-10-30:

[6] Note that URLs are provided for the 10-Qs otherwise, the text would be too cumbersome to read in the thesis

https://ir.amd.com/sec-filings/filter/quarterly-filings/content/0001193125-13-418892/0001193125-13-418892.pdf

10-Q filing for Advanced Micro Devices on 2014-05-01:

https://ir.amd.com/sec-filings/filter/quarterly-filings/content/0001193125-14-177314/0001193125-14-177314.pdf

Summarize and Sentimentize Generated Response comparing 10-Q documents from 2013-10-30 and 2014-05-01 *0.4 @ The 10-Q indicates some notable changes for AMD due to market pressures and profitability. AMD saw decreased revenue and margins in their computing solutions segment due to challenging PC market conditions and the increasing popularity of tablets. However, their graphics segment revenue increased from sales of semi-custom SOC products. Overall, there was a slight increase in revenue but a decline in profitability. @ negative*

10-Q filing for Advanced Micro Devices on 2014-05-01:

https://ir.amd.com/sec-filings/filter/quarterly-filings/content/0001193125-13-201635/0001193125-13-201635.pdf

10-Q filing for Advanced Micro Devices on 2014-07-31:

https://ir.amd.com/sec-filings/filter/quarterly-filings/content/0001193125-13-315281/0001193125-13-315281.pdf

Summarize and Sentimentize Generated Response comparing 10-Q documents from 2014-05-01 and 2014-07-31. *0.7@ There were some changes in AMD's business related to product launches, market conditions, and financial management, but no major shifts. AMD continued executing its transformation strategy, launching* *new products, and managing expenses. They repaid some debt and issued new debt. The overall business strategy seems consistent with prior quarters. @neutral*

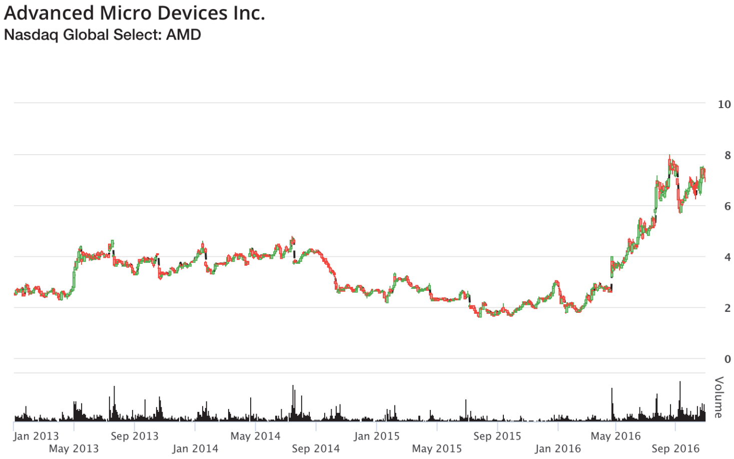

FIGURE 3.5: Stock Price for Advance Micro Systems from Jan 2013 Dec 2016

3.3.3 In Context Training LLM Summarize & Sentimentize

3.3.3.1 Methodology

Similar to the Summarize and Sentimentize methodology described above, we augment that approach by first passing in training data as part of the context to allow the LLM to learn at each iteration. Specifically, the training data comprises Summarize and Sentimentize summaries from 750 individual stocks randomly selected from the training data set, along with whether or not the excess return is negative, neutral, or positive. We then pass this training data along with the associated Summarize and Sentimentize prompt instructions per the stated algorithm 2. Note that the prompt instructions are broken into parts one and two so that the LLM can be followed and instructed easily.

Prompt instructions for part 1 are defined as:

"As context, please see the below summary of 750 companies comparing two 10-Q filings and the associated stock price return impact:"

$Xtrain$

Where $Xtrain$ is a concatenated list of 750 companies representing the training data that gets passed each call with each element representing the 10-Q summary as defined in the Summarize and Sentimentize approach in the previous section and whether or not the excess return was either "positive", "neutral" or "negative". Note that $Xtrain$ is passed in as an entire string as part of the prompt instructions for each call in the 2.

Prompt instructions part 2 are defined as:

"then based on this latest summary, indicate if the stock price impact will be 'positive' or 'neutral' or 'negative' for the company and make this only one word only."

Algorithm 2 In Context Training LLM Summarize & Sentimentize

```
for each data item $d_i$ in the dataset do
    Pass in Prompt Instructions part 1
    Pass in $d_i$ summary of the $Xtest_i$
    Pass in Prompt Instructions part 2
    Send both instructions to Claude2 and receive a response $r_i$
    Process $r_i$ to extract "positive", "neutral" or "negative" for price prediction
end for
```

Where $Xtest$ is also a summary representing the 10-Q summary as defined in 1 but does not include any labeled data on whether or not the excess return was "positive" or "negative".

In this approach, we extend past sentiment and ask the LLM to make financial market predictions based on the In-context training data set of the summaries and historically observed signs of the excess return.

Further, fine-tuning a model on price information has two significant downsides. The first is that a large language model like Claude2 itself has already been trained throughout the entire history of events up until 2023, so it could introduce forward-looking bias. The second concern is that training on price could potentially be a noisy process for a large language model as opposed to simply assigning a sentiment score. Assigning a sentiment score on a piece of text data is far easier than, say, predicting financial asset returns.

3.4 Results

In this section, we report the results across all the models discussed across the universe of S&P 500 stocks. We report accuracy metrics such as the F1 score in the Model Results section below, where we have a specific objective, such as training a classification model with either the Concatenating Deep Deep Learning or Cosine Similarity Logistic Regression. However, in the Excess Returns Results section, we report the Excess Returns for both the classification models as well as the LLM-based models. Note that we do not have explicit accuracy metrics for LLM, as no model was trained from scratch.

Additionally, within Finance, accuracy metrics only matter so much. Excess Return metrics are what end investors primarily evaluate portfolio managers or trading strategies on. Some trading strategies, like Momentum, can have a low accuracy rate, but when they are successful, they generate large excess returns when they do get it right. As a result, Excess Returns are what primarily matters. Further, in the realm of statistics, Excess Returns can be thought of as continuous versus binary or regression versus classification as capturing more depth.

3.4.1 Model Results

3.4.1.1 Concatenation Deep Learning Models

In this section, we compare all the Concatenated Deep Learning models along with the Cosine Similarity Logistic Regression benchmark using F1 scores per each positive and negative class. Most of the Concatenated Deep Learning models outperform the benchmark model regarding the negative F1 score while they under-perform the positive F1 score as seen in table 3.1. However, with the Cosine Similarity Logistic Regression model, there are signs that the model is not able to fit well, as upon analyzing the confusion matrix, it isn't very objective toward one-sided predictions. As a result, we will see in the Excess Returns Results section that the Cosine Similarity Logistic Regression generates Excess Returns in the opposite direction of the intended prediction. Overall, the Concatenated Deep Learning models outperform in terms of consistency with their predictions between the negative and positive classes, which plays into the models outperforming in predicting stock performance.

Model	F1 Score Negative	F1 Score Positive
Cosine Similarity Logistic	0.31	0.63
CNN concatenation	0.48	0.53
Max of Glove Embeddings	0.49	0.54
LSTM concatenation	0.48	0.53

Table 3.1: F1 Scores per the negative and positive test set

3.4.2 Excess Return Results

3.4.2.1 Concatenation Deep Learning Excess Returns

We find, as per table 3.2, that all the Concatenated Deep Learning architectures outperform in terms of predicting negative and positive excess returns versus the Cosine Similarity Logistic regression benchmark, which generates the opposite prediction. In particular, the CNN concatenation and the Max of Embeddings outperform the LSTM concatenation. This out-performance can most likely be attributed to the fact that LSTM models have limitations in passing in most of the text corpus since we restricted the padding length to 500 due to the vanishing gradient problem, in comparison with CNN and Max of Embeddings where we can process the majority of the text corpus with a 5000 padding length.

Model	Negative	Neutral	Positive
Cosine Similarity Logistic	0.03	0.01	-0.074
CNN concatenation	-0.053	0.012	0.003
Max of Glove Embeddings	-0.037	-0.004	0.047
LSTM concatenation	-0.03	-0.03	-0.002

Table 3.2: Average excess returns per prediction label for test set measured 180 days out

3.4.2.2 Zero-shot LLM Summarize & Sentimentize Results

Tables 3.3 and 3.4 contain the results when aggregating excess returns on the test set given the sentiment label. The tables show an event study from 30 to 180 days out to see how sustainable the signal is. Table 3.3 shows the average of excess returns while table 3.4 uses the median. In both the median and average case, it is clear there is a strong signal when sentiment is negative as the excess returns substantially negative, ranging from down -1.8% to -5.6% over the 30-day and 180-day periods, respectively. These results suggest that when companies change their 10-Qs due to profitability and other business concerns, there is a strong negative signal on the future stock price.

In contrast, there seems to be some marginal out-performance when the sentiment is labeled positive, ranging from up 0.1% to 1.7%. These results suggest that companies do not significantly change their 10-Q language as much when they see positive profitability and business improvements due to being lazy. Excess returns are defined as any financial returns when subtracting out the performance of the S&P 500 Index to take out the overall market movements. Compared with the cosine similarity benchmark without running it through a logistic regression model as per table 3.5, the Zero-shot LLM summary and sentiment analysis outperforms by a factor of two.

As a result, there is evidence that rudimentary natural language processing techniques, such as cosine similarity, underperform Large Language Models that can be synthesized and fine-tuned for nuanced reasoning. In addition, the Summarize and Sentimentize approach has another advantage: it is not forced to be in the 20th or 80th percentile as we did with the classification models, as the approach allows us to assign sentiment when observed.

Sentiment	30day	60day	90day	180day
Negative	-0.018	-0.022	-0.037	-0.056
Neutral	0.003	0.001	-0.000	-0.000
Positive	0.007	0.007	0.01	0.017

Table 3.3: Zero-shot LLM Summarize & Sentimentize average excess return on test period

Sentiment	30day	60day	90day	180day
Negative	-0.018	-0.02	-0.03	-0.041
Neutral	0.003	0.002	-0.002	0.001
Positive	0.001	0.004	0.007	0.023

Table 3.4: Zero-shot LLM Summarize & Sentimentize median excess return on test period

Buckets	30day	60day	90day	180day
Lower 20% percentile	-0.0015	-0.007	-0.016	-0.02
Between 20-80% percentile	0.001	-0.000	0.0006	0.003
Upper 80% percentile	0.006	0.008	0.007	0.011

Table 3.5: Average Excess returns for Cosine Similarity (no model) benchmark over test period (no model)

3.4.2.3 In-context training LLM Summarize & Sentimentize

Tables 3.6 and 3.7 contain similar results as the Zero-shot LLM section when aggregating excess returns over the test set with the exception that the prediction under-performance is not quite as pronounced. Further, the average excess returns for predicting positive performance are negative for the In-context training, indicating this approach struggled in this regard. Future research can explore this area further as, intuitively, it would seem more data would be better for transformers to learn. However, in contrast, this could contribute more noise for the large language model to ascertain the "positive", "neutral", or "negative" predicted signal.

Sentiment	30day	60day	90day	180day
Negative	-0.036	-0.064	-0.035	-0.063
Neutral	0.00	-0.008	-0.01	-0.016
Positive	0.003	0.009	0.006	-0.034

Table 3.6: In context training LLM Summarize & Sentimentize average excess return on test period

Sentiment	30day	60day	90day	180day
Negative	-0.037	-0.058	-0.000	-0.051
Neutral	0.003	0.000	0.004	0.007
Positive	0.018	0.007	0.012	0.001

TABLE 3.7: In context training LLM Summarize & Sentimentize median excess return on test period

3.4.2.4 Bringing it all together

Table 3.8 collates the annualized excess return from each methodology, including the cosine similarity benchmarks, concatenating deep learning models, and the summarize & sentimentize LLM methodologies. Across all the methods, it is clear that both summarize & sentimentize methodologies perform the best on the downside as these approaches can predict which stocks will under perform the most. However, on the upside, both Max of Glove Embeddings and the Zero-shot LLM Summarize & Sentimentize can outperform other methodologies as these approaches can better predict which stocks will outperform. Compared to the benchmark, the cosine similarity using logistic regression is the worst performer as the excess returns are the opposite of its prediction (i.e., stocks it predicts to outperform actually underperform and vice versa). Regarding the cosine similarity baseline benchmark that originated from Cohen et al. (2020), which sorts stocks based on the underlying cosine similarity metric, this is still relatively competitive but still falls short of the newer deep learning and LLM methodologies.

Model	Negative	Neutral	Positive
Cosine Similarity Logistic	0.03	0.01	-0.074
CNN concatenation	-0.053	0.012	0.003
Max of Glove Embeddings	-0.037	-0.004	0.047
LSTM concatenation	-0.03	-0.03	-0.002
Cosine Baseline (No model)	-0.04	0.006	0.02
Zero-shot LLM Summarize & Sentimentize	-0.11	0.0	0.032
In context training LLM Summarize & Sentimentize	-0.063	-0.016	-0.034

Table 3.8: Annualized Average excess returns across all methodologies

3.5 Summary

This chapter introduces new applications and models using deep learning and large language models to analyze financial disclosures for stock market prediction. Traditionally, NLP approaches for financial text data have relied on simple methods like cosine similarity between word vectors or lexicons to capture aspects like sentiment. While helpful, these techniques have limitations in fully capturing nuanced reasoning within lengthy and complex disclosure documents. As an alternative, we propose concatenating different deep learning architectures like convolutional neural networks and Max embeddings to compare two different 10-Q documents to model semantic shifts that may predict future stock returns. Testing three different concatenating models, we find they substantially outperform a cosine similarity benchmark in predicting excess stock returns. CNN and embedding concatenation work best because they can handle disclosures of a more significant length, unlike LSTMs.

Going beyond deep learning, we introduce a new Summarize and Sentimentize approach using the large language model Claude2. This method harnesses Claude2 first to summarize fundamental changes between two filings related to risks, profitability, legal issues, market pressures, and other critical information. It then assigns positive, neutral, or negative sentiment labels to these summaries and ultimately predicts excess stock returns 180 days out. With zero-shot prompting, we find negative sentiment strongly predicts poor future returns, while positive sentiment shows a weaker signal. Interestingly, adding training data through in-context learning introduces noise that degrades performance, but future research could be done to try to refine this process. Overall, the Summarize and Sentimentize methodology substantially outperforms standard NLP techniques like cosine similarity or lexicons, demonstrating the power of large language models to synthesize nuanced reasoning.

In summary, this chapter makes significant contributions by adapting techniques like semantic text matching with deep learning and introducing the new Summarize and Sentimentize approach to leverage financial disclosures for stock prediction in ways not done before. The results provide compelling evidence that modern Deep Learning and Large Language Model methods can better harness the wealth of insights within lengthy, complex disclosure documents compared to relying on simple statistics. We demonstrate that these AI techniques offer considerable promise for synthesizing meaningful information from financial text data.

Part II

Noise-Reduced Sentiment Analysis of Bitcoin on Reddit

Chapter 4

Exploratory Reddit Data Mining for Bitcoin

This chapter provides an overview of blockchain technology and introduces Reddit and on-chain data. The thesis's contribution in this chapter is the data analysis and exploratory analysis.

4.1 Literature review and Background

4.1.1 Introduction to Blockchain technology

Blockchain technology builds trust, delivers transparency, and creates value by having a peer-to-peer network on a decentralized, immutable ledger. Crypto was invented as an alternative to the traditional banking system that aims to democratize finance that can serve everyone worldwide, including the unbanked in developing countries.

Blockchain technology is a series of transactions attached and labeled as blocks. It starts with the original genesis block, where the next block builds on top of this one in a series of blocks, thereby protecting the system from any destructive attackers who try to alter or change the past for their benefit. Since Bitcoin began trading in early 2010, the system has withstood several attacks.

The primary application of Blockchain technology is cryptocurrencies, with Bitcoin being the first to launch in 2008. Bitcoin, the most popular cryptocurrency, utilizes Blockchain for transactions. Similar to PayPal but without a central authority, users can send money globally 24/7. Bitcoin serves as a means of payment and a store of value, often likened to digital gold. Additionally, hedge funds engage in speculative trading of Bitcoin as an inflation hedge. Chainanalysis, a leading blockchain analytics firm, reported the creation of nearly 480 million wallet addresses by December 2018[1].

Cryptocurrencies provide a significant pillar for Blockchain technology by introducing value creation. For instance, for Bitcoin, miners solve cryptographic hashing functions to verify transactions[2]. Another application is a cryptocurrency called Filecoin, which holds nearly 2% of the world's data on its Blockchain, and those who offer up storage get rewarded with Filecoin tokens. Typically, once payment is received for their services, either for computing or storage, miners will immediately transfer to an exchange to sell tokens back to USD to pay for their fixed costs, such as electricity.

[1] https://www.chainalysis.com/blog/bitcoin-addresses/

[2] Bitcoin mining is the process of validating the information in a blockchain block by generating a cryptographic solution that matches specific criteria. When a correct solution is reached, a reward in the form of bitcoin and fees for the work done is given to the miner(s) who reached the solution first

In addition to cryptocurrencies, Blockchain technology holds potential applications in various sectors such as banking, asset transfers, smart contracts, supply chain monitoring, insurance, gaming, and recording scientific discoveries and votes. Although the overall technological transformation with Blockchain is in its early stages, there are clear indications that traditional companies are venturing into the industry. In finance, for instance, Treasury securities are now traded on-chain, and Central Banks use Decentralized protocols for cross-border payments. According to a 2021 United Nations study[3], various industries are adopting this technology. The study cites examples such as Walmart partnering with IBM to enhance food safety in their supply chains and a blockchain platform called Everledger, which records the provenance of over 2 million diamonds.

4.1.2 In the beginning, there was Bitcoin

In a famous email in 2008 to a cryptography mailing list, Satoshi Nakamoto (i.e., pseudonymous name) sent his abstract and a PDF describing Bitcoin as "I've been working on a new electronic cash system that's fully peer-to-peer, with no trusted third party." Nakamoto (2009) in his white paper suggests a network of timestamp transactions by hashing them to form an ongoing chain to record transactions where a proof-of-work system must be utilized if a record is attempted to be changed. The longest chain provides proof of the transaction history observed.

Proof-of-work is the fundamental building block behind Bitcoin and many other cryptocurrencies. Satoshi's paper describes it as a distributed timestamp server that validates new transactions on a peer-to-peer basis. A more straightforward

[3] HARNESSING BLOCKCHAIN FOR SUSTAINABLE DEVELOPMENT: PROSPECTS AND CHALLENGES

definition of proof-of-work comes from Yaga et al. (2018), where a user gets the right to publish the next block if they can solve a cryptographic puzzle where the solution is the proof that they have performed the work. Further, the authors describe the puzzle's difficulty as set up to be around every 10 minutes; a new block is solved, rewarding a miner (computer node computationally dedicated resources) before block publication. This proof-of-work to verify transactions helps ensure security, maintains decentralization, incentivizes miners, and establishes a consensus. However, having every miner verify each block is considered by some to be an inefficient use of computer resources and not sustainable. As a result, this has led to the invention of proof-of-stake consensus mechanisms for some cryptocurrencies (i.e., Ethereum) based on how much each miner has of a particular token. As the computational complexity grows, more is needed for a single computer to solve these cryptographic puzzles. As a result, "pools" of computer resources are required to address this problem.

Satoshi also implemented a crucial element, the system of rewards, to motivate miners to allocate computer resources. Miners receive compensation either in the form of newly minted coins when they successfully create a block or through transaction fees when Bitcoin is utilized as payment. This reward system serves as a financial incentive, compelling miners to adhere to the established rules since generating new coins is more lucrative than undermining the wealth they accumulate through mining.

According to Narayanan et al. (2016), there have been hundreds of failed attempts to create cryptographic payment systems. In fact, the technology was available for Bitcoin to be made in 1994 on the back of three technologies: Merkle trees (1979), blockchain data structure (Haber and Stornetta, 1991) and proof-of-work (1993)[4]. In fact, Satoshi usually gets credited for creating the proof-of-work system. Still, Cynthia Dwork and Moni Naor invented this to circumvent denial-of-service attacks[5]. Hashing functions were invented in the 1980s with Merkel Trees by Ralph Merkle. It is hard to say why everything came together for Bitcoin in 2008-2010, leading to greater adoption. Still, it certainly was good timing, with some people losing trust in traditional finance due to the 2008 financial crisis, bank runs and collapses, and Cyprus seizing retail funds from banks.

4.1.3 Not just Bitcoin

According to CoinMarketcap, nearly 10,000 cryptocurrency projects exist as of the end of 2022. However, only a few hundred of these projects have significant market capitalization. Major applications extend beyond Bitcoin, including decentralized finance (DeFi), stablecoins, and Non-Fungible Tokens (NFTs).

They were introducing Ethereum, the second-largest cryptocurrency by market capitalization, which introduced the concept of smart contracts—executable code within a blockchain. Ethereum acts as a platform similar to an iPhone, enabling the development of various applications with use cases ranging from insurance and supply chain to decentralized finance.

DeFi encompasses applications such as lending, insurance, automated trading for spot and derivatives, and yield farming. According to DeFiLama[6], the total value

[4] CoinTelegraph article "Could Bitcoin have launched in the 1990s — Or was it waiting for Satoshi?"

[5] Denial-of-service is a cyber-attack cyber-attack where the attackers make a computer or network unavailable.

[6] https://defillama.com

locked in DeFi tokens reached nearly $47 billion as of Nov 29, 2023, across prominent tokens like Lido, Maker, JustLend, Aave, and Uniswap.

NFTs represent a novel form of digital assets, allowing the original creator to maintain rights and, in some cases, collect royalties. NFTs include digital art, music, and any asset represented on-chain. For instance, the digital ownership of automobiles or real estate will soon be recorded on the Blockchain, reducing the need for paper trails or title insurance.

Stablecoins maintain a one-to-one match with the USD, eliminating volatility for users entering the digital currency space. According to DeFiLama, the total value locked in stablecoins reached $128 billion as of Nov 28, 2023. Stablecoin cross-border payments are virtually free for wallet-to-wallet transfers, providing an efficient option for workers sending money internationally. However, some uncertainty still needs to be made regarding how issuers maintain liquidity and assets in their reserves.

4.1.4 Literature review

A substantial body of academic literature has explored utilizing diverse data sources to predict cryptocurrency markets. Data sources include online social data, text-based sentiment, and on-chain data (note we introduce what on-chain data is in the following sections).

Before we introduce critical research that has been done within cryptocurrency prediction, we want to lay some groundwork on how financial markets behave. Farmer (2002) provided a seminal explanation of financial market functions, drawing an analogy to biological ecosystems. He posited that the market comprises various trading strategies akin to competitive, predator-prey, or mutualistic interactions in biology, which continually evolve and adapt. Farmer's agent-based model presents a simplified version of financial markets, contrasting short-term momentum traders with long-term value traders. As a result, this model accounts for the formation of short-term trends through the release and interpretation of news.

Seeking quantitative signals in financial news, Cristescu et al. (2023) extensively studied the impact of media coverage on stock price trends using wavelet analysis. Their work highlighted a monumental shift in influence from traditional news outlets like Dow Jones and Reuters to social platforms such as Twitter and Reddit within the cryptocurrency sector. This paradigm change was attributed to the limited mainstream media coverage of the nascent crypto asset class at the time, making social platforms disproportionately impactful.

Perhaps the most comprehensive thesis on Bitcoin using social media data is a University College of London (UCL) PhD Thesis by Phillips (2019) that takes an in-depth look at various social media data and its potential predictive power on Bitcoin. The data the thesis considers spans from Reddit (derived features such as the number of posts per day, new authors, and subscriber growth), google trends and Wikipedia count-based metrics, Twitter count data, and bitcoin.org forum data. Note that the UCL thesis's primary emphasis is on count-based metrics as opposed to text data. The main exception to this is a small part of the thesis uses a simple topic-based Latent Dirichlet Allocation (LDA) model to figure out which topics Granger causes the price of Bitcoin. The UCL thesis provides valuable contributions by utilizing epidemic modeling to predict Bitcoin price bubbles and wavelet coherence analysis between online factors and Bitcoin. This thesis extends past count-based metrics and solely focuses on harnessing value from sentiment-based signals within Reddit.

In the early months of 2021, Reddit experienced a noteworthy event involving the "wallstreetbets" subreddit community. This group of retail investors collaborated to identify and collectively invest in a selection of heavily shorted stocks[7], such as Gamestop, Bed Bath and Beyond, and AMC, among others. The subreddit forum created a unique dynamic akin to a David and Goliath narrative, as hedge funds on the opposing side had predicted a decline in stock prices. Ultimately, this manifested in a rare moment in history when retail investors emerged victorious, causing the closure of the prominent hedge fund Melvin Capital. To test this relationship, Wang and Luo (2021) extracted Reddit text data on the wallstreetbets subreddit related to Gamestock and found that sentiment from the posts is able to predict daily price movements for Gamestock. Their sentiment relies on a simple lexicon called VADAR for social data that sums up positive and negative words to construct a sentiment model.

Further, they extend the vocabulary of this lexicon using word embeddings from Word2Vec and Bert. Chalkiadakis et al. (2022), utilize a similar approach by augmenting a lexicon with embeddings applied to crypto news to gauge retail adoption by using a multimodel causality between crypto sentiment and price. However, the authors need to address how to deal with the massive amount of noise contained within Reddit posts that can reduce the impactfulness of the signal derived from social media data. As a result, Part 2 of this thesis introduces a novel methodology for addressing the additional noise.

[7] The CFA institute defines Short selling as the practice of selling borrowed securities – such as stocks – hoping to be able to make a profit by repurchasing them at a price lower than the selling price. In other words, when you sell short a stock, you're looking to profit from a decline – rather than an increase – in price.

The most exciting aspect of how cryptocurrencies set themselves apart from other asset classes, like equities, foreign exchange, and the bond markets, is their transparency for measuring transaction activity on the Blockchain (also referred to as on-chain data). However, the sheer magnitude of the raw transaction-level blockchain data is incredibly difficult to work with. Blockchain data has provided an opportunity for on-chain analytics vendors, such as Cryptoquant and Glassnode, to aggregate hundreds of indicators and other representations of the data on an aggregated basis. Kim et al. (2022) collate over 254 on-chain variables utilizing Self Attention Multiple Long short term memory (SAM-LSTM) in order to predict the price of Bitcoin with promising results. The on-chain variables span from the number of BTC addresses with different balances, market capitalization, exchange flow indicators, and unrealized profit variables.

Additionally, Herremans and Lowa (2022) investigate using Cryptoquant data and whale alert tweets to predict Bitcoin's subsequent day volatility. They ultimately find their approach helps forecast extreme volatility periods. For this thesis, we will use on-chain indicators as additional features for Bitcoin prediction to augment the Reddit sentiment without noise signal.

4.2 Reddit Data and Exploration

4.2.1 Reddit Introduction

Reddit is a social news aggregation website comprised of distinct communities called subreddits dedicated to specific topics. Users can share and comment on posts, voting them up or down. As of July 2018, Reddit had over 330 million active users across over 138,000 active subreddits (out of 1.2 million total)[8].

The subreddit of focus for this thesis is “r/bitcoin”, which contains over 4.7 million members, ranking it 42nd in terms of the largest Reddit communities. This subreddit is dedicated to discussions around Bitcoin. The other potential subreddit considered was “r/cryptocurrency”; however, this would have posed additional challenges for topic modeling while having significantly fewer overall posts than r/bitcoin.

On Reddit, posts can be links, images, videos or text. Users can comment on submissions, reply to other comments, and vote positively or negatively within each subreddit. This crowdsourced curation model surfaces popular content. Moderators oversee each subreddit’s rules and norms. Reddit provides a window into current discussions and sentiment around niche interests through targeted subreddits like “r/bitcoin”. Analyzing associated textual data offers insights into Bitcoin discourse among early adopters and enthusiasts.

4.2.2 Data Collection

To extract Reddit text data, the Pushshift API was utilized, which is an open-source tool for retrieving Reddit content. The API can collect comments and submissions for a specified subreddit of interest. For this thesis, we focused solely on submission texts to analyze authors’ original content and intent. Just submissions allowed easier filtering of irrelevant posts like advertisements or technical questions unrelated to Bitcoin prices.

[8] Chicago Tribune article “Reddit to open Chicago office as part of advertising push”

After processing through the API, the resulting Reddit dataset spanned December 2010 to January 2022, comprising 786,739 observations. The data included each post's date, author, title, body text, and upvote ratio. A further filtering stage kept only submissions with text in both the title and body, yielding a refined dataset of 257,435 posts. Requiring body text signifies meaningful thought beyond a title-only post. The title and body text were merged into a combined text field to simplify feeding into models.

Figure 4.1 shows the annual submission count for the r/bitcoin subreddit. From 2010-2012, activity was low as Bitcoin and Reddit were still emerging. Data is partially missing in 2013. Given these limitations, price prediction analyses focused only on data from 2014 onward. Visible spikes in 2017 and 2021 correspond to Bitcoin price bubbles. 2022 only contains partial January data. As Bitcoin gained traction, submissions on r/bitcoin grew substantially, providing a rich source of discourse to analyze as the basis for this thesis.

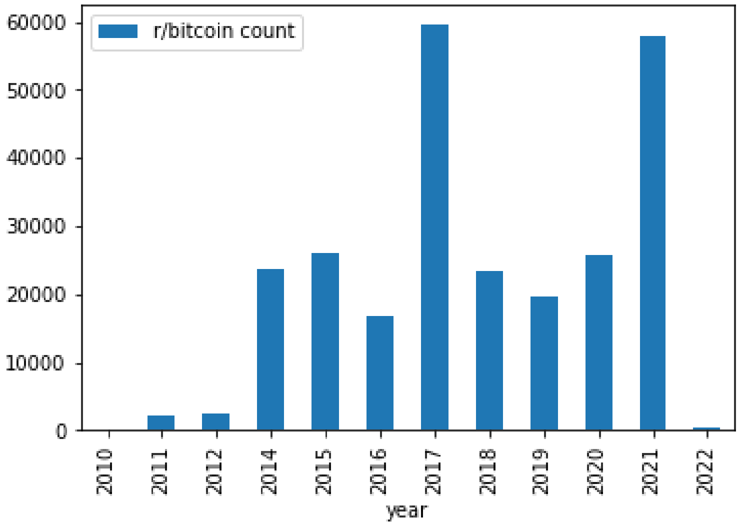

FIGURE 4.1: Number of Reddit "r/bitcoin" subreddit submissions by year

4.2.3 Exploratory Data Analysis

Similar to SEC filings and many other text datasets, Zipf's law is present in the Reddit data, as shown in Figure 4.2. A power law distribution is visible, with the top 50 words making up many overall counts. Most high-frequency words are stopwords like "the", "to", etc. For SEC filings, stopwords were removed since CNN models do not rely on sentence context. However, for Transformer models, keeping stopwords and punctuation can provide valuable signals about the linguistic context that improves sentiment classification. While not informative alone, these words help models learn relationships between terms that define sentiment. Unlike CNNs, Transformers can utilize the full range of text to understand sentiment cues from context.

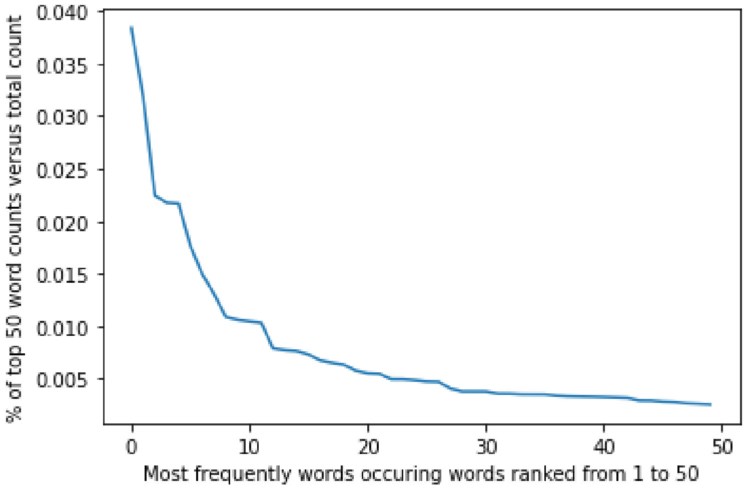

FIGURE 4.2: Zipf's Law demonstrating power distribution

Figure 4.3 shows the distribution of total words per Reddit submission. In contrast to more extended SEC filings, Reddit posts have fewer words per submission, following a left-skewed distribution. On average, Reddit submissions contain around 100 words, far less than the approximately 7,500 words in SEC 10-Q filings. The condensed nature of social media discourse creates a better fit for using models such as RoBERTa, which we leverage in chapter 5. The length of text passages impacts the modeling options available for NLP tasks. Short texts provide more flexibility compared to long passages.

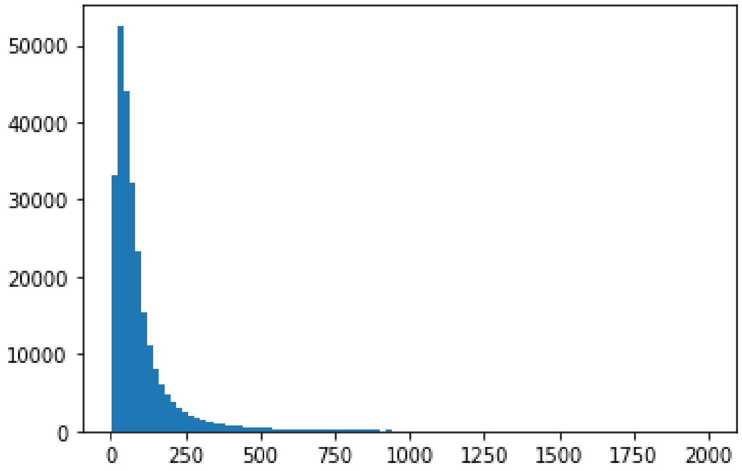

Figure 4.3: Histogram of Reddit "r/bitcoin" Word Frequency

4.2.4 Manual labeling for few-shot learning

After reviewing many submissions, it is clear that many posts are spam-oriented, involving either advertisements or generic technical advice questions. Both of these have no fundamental bearing on the price of Bitcoin. In order to jumpstart the removal of these noisy categories, over 1000 example posts from the "r/bitcoin" subreddit were manually labeled to allow few-shot learning. As shown in Table 4.1, of a sample of 1000, nearly 27% were users asking for technical advice such as "what is the best cold storage wallet?" or "what is the best exchange to trade on?". Additionally, 7% were attributable to advertisements. As a result, the categories of advice and advertisements are treated as noise in this thesis, as they have no meaningful impact on the potential price of Bitcoin.

Similar to labeling the noisy categories, table 4.2 shows the distribution of manual labeling for sentiment as well across the 1000 samples. We can see that 70.8% of

Categories	Count	Count %
other	329	33.1%
advice	330	33.0%
price	125	12.5%
musings	141	14.1%
advertisements	73	7.3%
total	1000	

TABLE 4.1: Reddit category classification within the "r/bitcoin" subreddit

the data have been classified as neutral, where it is slightly biased towards bias labels for being positive versus negative, i.e., 17.8% versus 11.3%.

Categories	Count	Count %
neutral	707	70.8%
positive	178	17.8%
negative	113	11.3%
total	1000	

TABLE 4.2: Reddit sentiment classification within the "r/bitcoin" subreddit

Labeling data is invaluable for fine-tuning models and evaluating classification performance. Specifically, 500 observations are used for training, with 250 for validation and 250 held out for testing. While a relatively small sample, we can benefit from transfer learning, as the underlying BERT models were trained on hundreds of millions of tokens. Howard and Ruder (2018) developed Universal Language Model Fine-tuning for Text Classification (ULMFIT), demonstrating comparable performance to training from scratch on 100x more data using just 100 labeled examples. As a result, manually labeling sentiment and topics for 1000 Reddit posts can leverage few-shot learning with Roberta models as described in chapter 5, removes noisy categories to obtain a cleansed sentiment signal with the ultimate goal of predicting Bitcoin's price 60 days out, despite limited labeling. Carefully curating a small labeled dataset enables the harnessing of pre-trained knowledge in transformer networks like Roberta.

4.3 On-chain Indicators Data and Exploration

In addition to social sentiment, a range of on-chain indicator features are included when forecasting Bitcoin's price direction using a multivariate nonlinear model. This section introduces on-chain data, collection methods, and exploratory analysis.

4.3.1 Introduction to On-chain Indicators

On-chain data refers to metrics and indicators derived from blockchain transaction activity. They summarize blockchain health, adoption, and usage patterns. Common examples include several addresses holding over $100 in bitcoin, mining fees, and new coins generated. This thesis utilizes data from Glassnode, a leading blockchain intelligence provider, offering thousands of on-chain indicators. Categories span addresses, distributions, fees, market signals, supply, transactions, and unspent/spent ratios.

A key advantage of cryptocurrencies is the wealth of immutable on-chain data. By recording every transaction, the Blockchain provides a rich basis for analysis over traditional finance. Glassnode leverages this transparency to construct hundreds of indicators measuring adoption, sentiment, fundamentals, and early warning signals.

4.3.2 Data Collection

Glassnode provides a wealth of on-chain indicators capturing various facets of blockchain activity, offering transparency unavailable in traditional markets. Compared to foreign exchange, cryptocurrencies like Bitcoin enable observing transaction flows, wallet balances, miner economics, and adoption metrics on-chain.

This thesis utilizes the following Glassnode indicators covering core aspects of Bitcoin's Blockchain:

1. Glassnode indicator summary

    (a) Net unrealized profit loss: the difference between Unrealized Profit and Unrealized Loss to determine whether the network as a whole is currently in a state of profit or loss

    (b) Puell multiple: is a ratio of daily coin issuance (in USD) and the 365 moving average of daily coin issuance providing an oscillator derived from miner profitability and income stress

    (c) SOPR (Spent Output Profit Ratio): indicator provides insight into macro market sentiment, profitability, and losses taken over a particular time-frame. It reflects the degree of realized profit for all coins moved on-chain.

    (d) SOPR Adjusted (Spent Output Profit Ratio): similar construction and interpretation to the standard SOPR metric; however, excludes all transaction volume for coins with a lifespan younger than 1hr.

    (e) Transfers Volume Exchanges Net: The difference in volume flowing into and out of exchanges, i.e., the net flow of coins into/out of exchanges.

(f) Reserve Risk: A cyclical indicator that tracks the risk-reward balance relative to the confidence and conviction of long-term holders.

(g) CDD (Coin Days Destroyed): is a measure of economic activity that gives more weight to coins that haven't been spent for a long time.

(h) Addresses 1k min: Number of wallet addresses with greater than $1k balances.

(i) Revenue from miner fees: The percentage of miner revenue derived from fees, i.e. fees divided by fees plus minted coins.

(j) Thermocap: is the aggregated amount of coins paid to miners and serves as a proxy to mining resources spent. It measures the true capital flow into Bitcoin and is computed as the aggregate coinbase transactions multiplied by the price in USD at the time they were mined.

(k) Marketcap: The market capitalization (or network value) is defined as the product of the current supply by the current USD price.

The Glassnode indicators encompass various blockchain attributes, from market capitalization and user adoption to exchange activity and miner revenue. This transparency into on-chain dynamics is unparalleled compared to traditional finance. These metrics will be utilized as control features when predicting Bitcoin's price direction 60 days out. The overarching goal is to evaluate whether filtering noisy topics from Reddit sentiment can enhance price forecasting accuracy. On-chain data is crucial as we will have this data for both the benchmark and the new proposed methodology. If the cleansed social media signal proves more predictive, even controlling for blockchain factors, it demonstrates the value of careful text filtering and classification.

4.3.3 Exploratory Data Analysis

Figure 4.4 shows the distribution of each Glassnode indicator. Though the LightGBM model does not require standardized data, as does this automatically within the Python library, a rolling 365-day z-score is computed to bring all metrics to a standard scale and enable comparisons with traditional models.

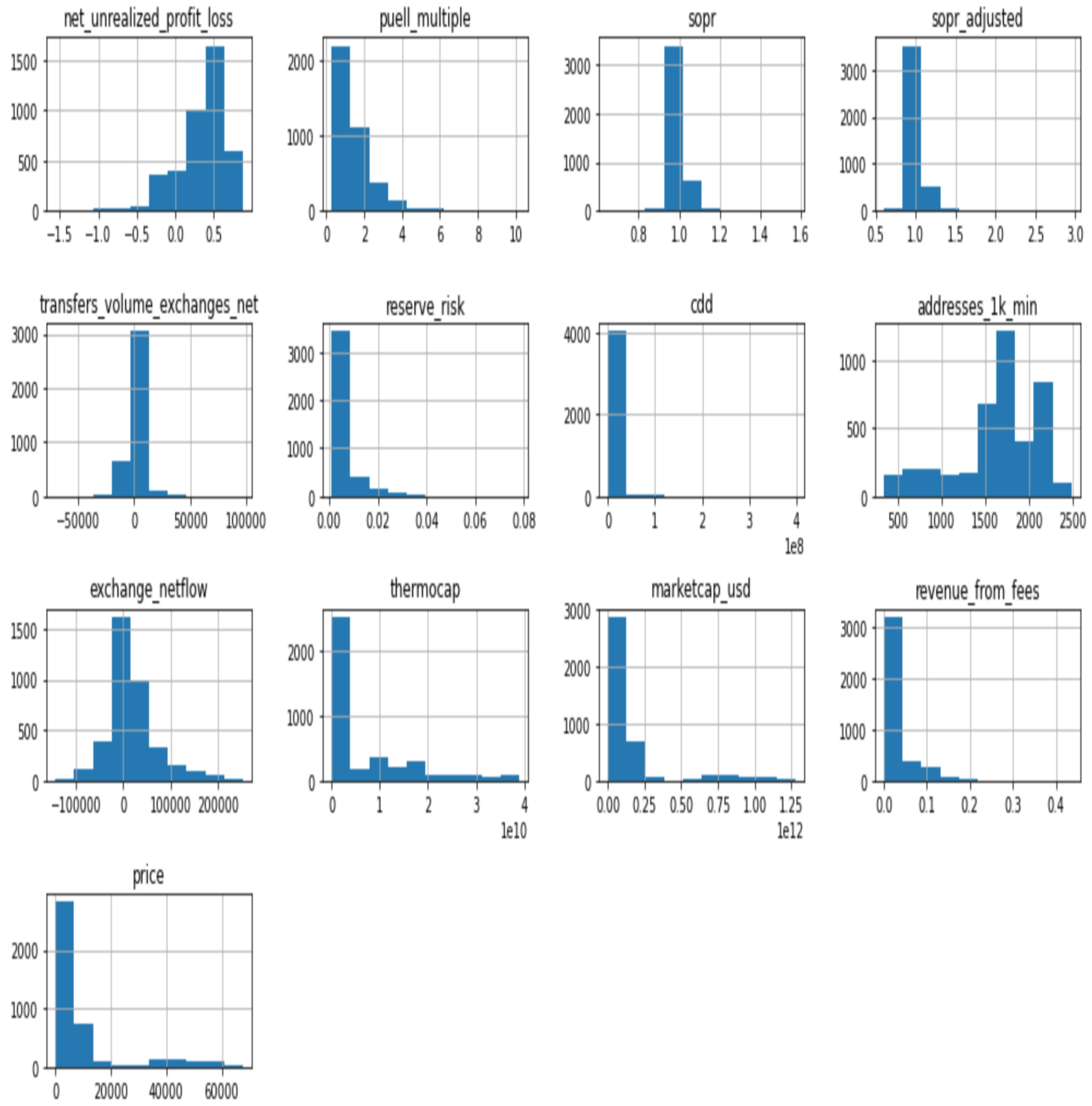

FIGURE 4.4: Histogram of each of the Glassnode features

Figure 4.5 evaluates multicollinearity between indicators through their correlation matrix. Most features exhibit low to moderate correlation, with a few exceptions.

The Puell Multiple and Reserve Risk are highly correlated, implying related information. Thermocap and Marketcap also show elevated correlations as both quantify global network value.

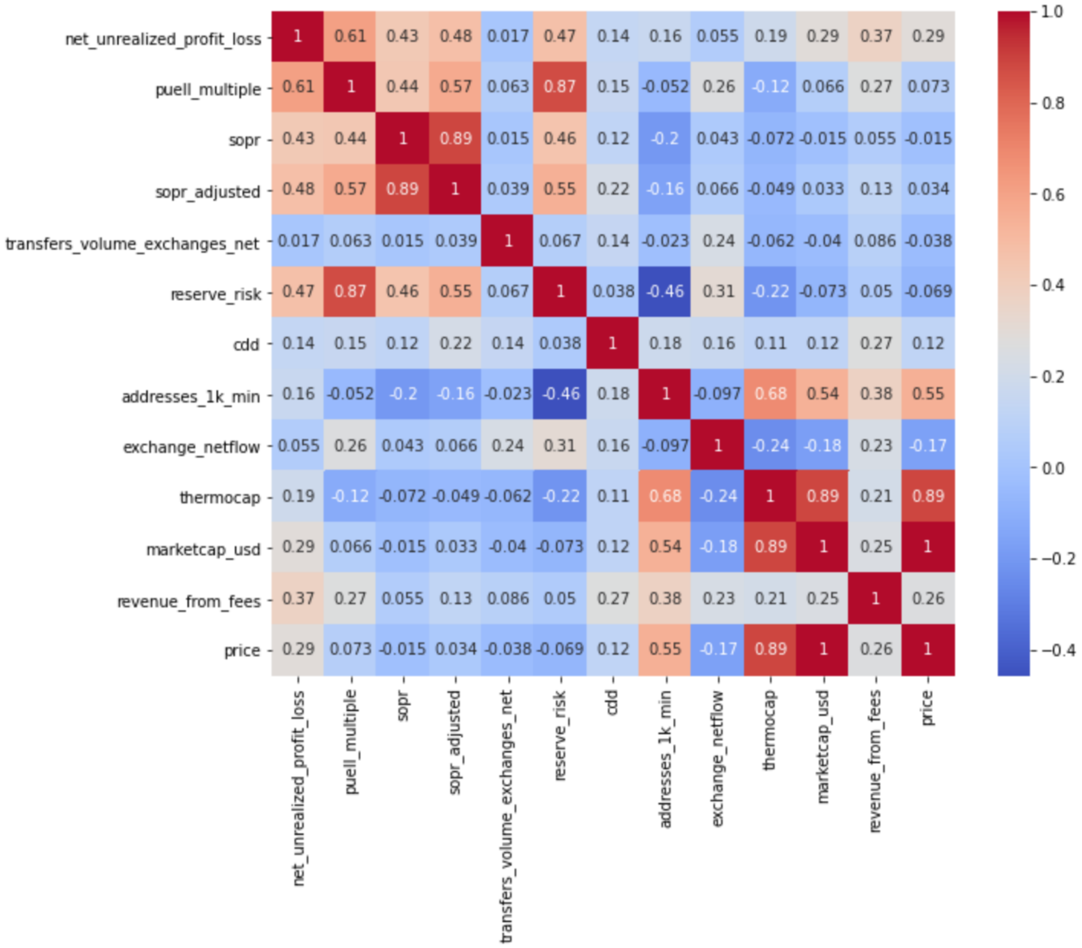

Figure 4.5: Glassnode feature correlation matrix

In summary, distributions show that most on-chain metrics have some skewness, which normalizing via a z-score standardization should help address some of this. Correlation analysis identifies a handful of highly related indicators, but overall, the features appear sufficiently independent. This exploratory analysis provides insights into the dimensionality and relationships between blockchain activity metrics from Glassnode.

4.4 Summary

This chapter comprehensively explores blockchain technology, focusing particularly on Bitcoin and its interactions within the Reddit community. It starts by explaining the fundamentals of blockchain technology, emphasizing its role in establishing trust, transparency, and value through a decentralized ledger. As the pioneering application of this technology, Bitcoin is highlighted for its global financial implications and its role in democratizing finance.

The chapter delves into the origin story of Bitcoin, tracing back to Satoshi Nakamoto's 2008 proposal. It discusses the critical elements of Bitcoin's infrastructure, such as the proof-of-work mechanism and the incentivization model for miners. The narrative then extends to the broader cryptocurrency landscape, noting the rise of thousands of crypto projects, with Ethereum and its smart contracts gaining significant attention.

Much of the chapter is devoted to a literature review, analyzing various data sources and methodologies for predicting cryptocurrency markets. This includes a deep dive into the role of social media, mainly Reddit, in influencing Bitcoin's market dynamics. The chapter examines Reddit as a data source, focusing on the "r/bitcoin" subreddit. It details the methods of data collection and the challenges involved, such as identifying noisy categories like advertisements or irrelevant content.

The exploratory data analysis reveals insights into the nature of Reddit discussions about Bitcoin, highlighting the prevalence of Zipf's law in the dataset and discussing the implications of post lengths for model selection. Manual labeling for topic and sentiment is employed, laying the foundation for developing Few-Shot learning classification models that will be described in chapter 5.

The chapter also introduces on-chain indicators as a novel aspect of cryptocurrency analysis. It explains the significance of on-chain data in providing a transparent and detailed view of blockchain activity, which is unavailable in traditional financial markets. Glassnode's on-chain indicators are explored, showing their utility in understanding market dynamics and predicting price movements.

In conclusion, this chapter sets the stage for predictive analytics in the cryptocurrency domain, blending social media sentiment analysis with on-chain data insights. It positions the Reddit platform, particularly the "r/bitcoin" subreddit, as a valuable source of sentiment and opinion, which, when combined with on-chain metrics, offers a rich dataset for understanding and forecasting Bitcoin's market behavior.

Chapter 5

Few Shot Sentiment Analysis with reduced noise

This chapter presents a new methodology for enhancing sentiment analysis. It systematically applies encoders such as BERT and RoBERTa to reduce noise from Bitcoin SubReddit data. The sentiment approach also integrates Large Language Models (LLMs) to augment training samples. This section presents empirical evidence substantiating that integrating state-of-the-art generative AI and transformer technologies outperforms text classification models compared to traditional Naive Bayes classification approaches.

An essential contribution of this thesis is the development of a novel noise reduction methodology for sentiment on social media. This development involves a RoBERTa-based topic classification model, refined through training sample augmentation using ChatGPT. The chapter introduces a cohesive methodology that merges three distinct models: RoBERTa for topic classification, RoBERTa for sentiment classification, and the LightGBM multivariate model for Bitcoin price classification. This three-part framework is designed to predict the sign of Bitcoin prices 60 days in advance.

There is a noticeable gap in research explicitly targeting augmented sentiment analysis, especially in filtering out noisy topic categories from social media datasets. This thesis distinguishes itself by combining Few-Shot learning with LLM-based training sample augmentation, offering enhanced topic assignment compared to unsupervised approaches like Latent Dirichlet Allocation (LDA). This method represents a significant advancement in sentiment analysis, particularly for social media data.

The chapter starts with an introduction to Sentiment Analysis, laying a foundation for understanding the concept. It then introduces various models: BERT, RoBERTa, and the Naive Bayes Classifier, outlining our methodology for isolating noisy topics in Sentiment Analysis. We also discuss the results of using the BERT model for topic and sentiment classification.

Subsequently, we introduce our sample augmentation methodology using ChatGPT and analyze the results of combining RoBERTa with this method. There is also a section benchmarking the augmented RoBERTa model against ChatGPT alone. In conclusion, the chapter synthesizes our findings into a reduced noise sentiment signal integrated into a multivariate LightGBM model as a practical application.

5.1 Sentiment Analysis Introduction

Sentiment analysis is a widely utilized natural language processing technique for extracting attitudes, perceptions, and feelings from text passages, with applications spanning finance, customer insights, and political analysis. The term "Sentiment Analysis" first appeared in a paper by Zhang et al. (2018a), where they employed a Naive Bayes classifier to construct sentiment for online reviews of restaurants, movies, and products. The seminal work by Zhang et al. (2018b) extensively covers Sentiment Analysis and Opinion Mining, noting that the field is known by various names, including opinion mining, opinion extraction, sentiment mining, subjectivity analysis, affect analysis, emotion analysis, and review mining. These terms are now generally encompassed within the term "Sentiment Analysis."

Furthermore, Zhang et al. (2018b) categorizes sentiment analysis into three main areas: document, sentence, and entity and aspect. Document-level analysis pertains to the sentiment of an entire text passage or review. The sentence-level analysis focuses on classifying sentences as subjective or objective. However, it may fall short in cases where objective sentences contain opinions, such as *"I bought Bitcoin last year, but the transaction speeds are surprisingly slow"*. Entity and Aspect sentiment analysis first identifies the subject or entity in a sentence, followed by analyzing sentiment related to that entity. For example, in the sentence *"The blockchain technology behind Bitcoin ensures transparency, but the volatility makes it a risky investment"*, the sentiment about Bitcoin's blockchain technology is positive. In contrast, the investment aspect is viewed negatively due to volatility.

Sentiment analysis plays a crucial role in financial markets by converting unstructured data into structured, time-series datasets. This allows investors to use sentiment analysis for trading or as an input feature in multivariate models for predicting financial market prices.

Historically, financial market participants have utilized Loughran and McDonald (2011a) positive and negative word lists (i.e., lexicon), which are adept at capturing financial jargon such as “bullish”, “bearish”, “long”, “short”, “yield”, etc. Loughran and McDonald also developed word lists encompassing themes like “uncertainty”, “litigiousness”, “constraining”, and “modals”, with each list capturing distinct attributes.

Other research, such as Agaian and Kolm (2017), has explored using traditional machine learning techniques like Naive Bayes or Support Vector Machines for financial sentiment analysis. Labeling remains challenging, but the authors obtained 500 labels from the website “Seeking Alpha”, where authors indicate their intentions of buying or selling a particular stock. Liu (2012) conducted a survey on Deep Learning for sentiment analysis, covering traditional approaches such as Convolutional Neural Networks, Long Short-Term Memory, and Attention Networks.

As discussed in Appendix B.1, Transformers represent an innovative advancement in sentiment analysis, utilizing transfer learning from large language models. A prime example of this is Howard and Ruder (2018) work on Universal Language Model Fine-tuning for Text Classification (ULMFIT), which predicted the sentiment of IMDB movie reviews. They first trained a large language model based on LSTMs on a general Wikipedia corpus, then applied domain adaptation for IMDB language, ultimately classifying binary positive or negative sentiments. Notably, even with a small sample size of 50 observations, the model achieved competitive performance, highlighting the efficacy of transfer learning. According to Ruder's NLP progress tracking site[1], ULMFIT remains one of the top models for IMDB sentiment prediction accuracy, with the leading models being variations of Bidirectional Encoder Representations from Transformers (BERT). The following section will introduce BERT models.

5.2 Introduction to Models Used for Few Shot Learning

This section provides an overview of the BERT, RoBERTa, and Naive Bayes Classifier benchmark models. We test the BERT model first but ultimately switch to the RoBERTa for social media data.

5.2.1 BERT Introduction

The BERT language model introduced by Devlin et al. (2018), which stands for Bidirectional Encoder Representations Transformers, at its time of release in 2018 achieved state-of-the-art results across eleven natural language processing tasks. Given the bidirectional nature of BERT, it aimed to provide deep training from both left and right on over 2,500M words from Wikipedia to estimate a language model. In contrast, GPT models are only pre-trained from left to right. From here, the next phase is to generate a fine-tuned language model using the foundation of the initial BERT language model trained on Wikipedia. In this thesis case,

[1] `https://github.com/sebastianruder/NLP-progress/blob/master/english/sentiment_analysis.md`

Reddit text is used to create a fine-tuned Reddit-based language model. Once these stages are complete, this information can be fed across various NLP tasks, including question-answering, summarization, sentiment, etc.

BERT's base model architecture includes the following parameters: L = 12, H = 768, A = 12, 110M params, twelve transformer blocks (encoder layers), 768 feed-forward networks, and 12 self-attention heads representing over 110m parameters. Self-attention is a new concept introduced within Transformers.

BERT models work both left-to-right and right-to-left, hence their Bidirectional nature. In contrast, GPT models work left-to-right, as the emphasis is on language prediction instead of classification.

In this thesis, Fine-Tuning implies using an underlying language model for classification as the final output, in contrast to Domain Adaptation, which takes the base-level language model but first generates a customized language model as per a custom data set, where the final step leads to a classification prediction task. Note that when we estimate BERT models, we explore how much domain adaptation can improve results when applied to the entire Bitcoin subreddit text corpus.

5.2.2 RoBERTa Introduction

The RoBERTa (Robustly Optimized BERT) was released from the Facebook AI team shortly after BERT by Ott et al. (2019). It was built on the BERT model's success but addressed some limitations while achieving breakthroughs on several NLP benchmarks. Specifically, the significant improvement RoBERTa resides in its training methodology.

The significant advantage of the RoBERTa model over BERT was that it was trained for extended periods. There are two main advantages of architecture setup with RoBERTa over BERT, including 1) dynamically masking tokens versus static masking, ensuring more variability, and 2) dropping the next sentence prediction in the loss function. Point 2) above becomes essential when dealing with social media (Reddit in this case) data, as there is typically only one sentence.

RoBERTa was also trained on slightly different datasets. It is hard to say officially how much of a different dataset mix contributed to the results as the authors could not access the entire BERT dataset for proprietary reasons. Specifically, RoBERTa included Bookcorpus plus English Wikipedia, CC-News, a common crawl of news sites, OpenWebText Reddit text data, and a common crawl of web stories. Note that common crawl is a systematic way of webscaping website text data.

The authors keep the model architecture the same as the BERT base (L = 12, H = 768, A = 12, 110M params) and reserve that for future research. The authors find that the RoBERTa model outperforms BERT across three primary NLP benchmark datasets, including SQUAD, GLUE, and RACE.

5.2.3 Naive Bayes Classifier (as Benchmark model)

We use the Naive Bayes Classifier as our benchmark model. It is a popular supervised learning technique within text classification, traditionally used for spam classification. The construction of this model applies the Bayes theorem with a "naive" assumption of conditional independence between input features and classification output (Zhang (2004)). Specifically, as per Dan Jurafsky from Stanford[2], a multinomial Naive Bayes can be defined as follows:

[2] https://web.stanford.edu/class/cs124/lec/naivebayes2021.pdf

$$c_{NB} = \arg\max_{c_j \in C}[logP(c_j) + \sum_{i \in positions} logP(x_i|c_j)] \tag{5.1}$$

where d is considered the document, $x_1..x_n$ are the features within a document, $C = c_1, c_2, ...c_J$ is a fixed set of classes and c_{NB} would be the learned classifier. Note that this framework is summing logs instead of the product of probabilities, which can be problematic if probabilities are close to zero. Then, maximum likelihood combined with a Laplace smoother can be used to obtain an estimate as per below:

$$\hat{P}(w_i|c) = \frac{count(w_i, c) + 1}{\sum_{w \in V}(count(w, c)) + 1} \tag{5.2}$$

Word frequencies can be summed up where V represents the entire text corpus.

One consequence of transforming the text features by encoding them into token counts is a loss of the words' ordering in terms of their representative context. However, for simple benchmark models with natural language processing, Naive Bayes acts as an excellent baseline to ensure that added complexity is beneficial.

5.3 Methodology for Sentiment Noise Reduction

The analysis of Bitcoin-related topics on Reddit, as shown in Table 4.1, reveals categories like "advertisement" and "technical advice" that are unrelated to Bitcoin's price. This thesis introduces a novel approach to eliminate these noisy categories. Initially, we employ a RoBERTa-based topic classification model across five categories: other, technical advice, advertisements, price, and economic musings. This step is the first in a three-phase process aimed at predicting Bitcoin's price, as illustrated in Figure 5.1.

The second phase estimates Bitcoin sentiment from Reddit data, using RoBERTa to classify sentiments into positive, neutral, and negative categories. Initially, we experimented with the BERT model. However, we ultimately chose RoBERTa, as it is better suited for the concise nature of social media data, focusing less on predicting subsequent sentences, unlike BERT. The third phase involves feeding a daily, reduced noise sentiment signal, smoothed by an Exponentially Weighted Moving Average, into a nonlinear, multivariate LightGBM model. This model predicts the direction of Bitcoin's price change 60 days ahead.

We augment the initial training sample with ChatGPT for both Reddit topic and sentiment analysis. This approach combines few-shot learning with Large Language Model-based training sample augmentation. The dataset's initial manual labeling ensures human oversight and precise topic specification. Once the initial topics are labeled, we use models like ChatGPT to expand the training set. A Naive Bayes Classifier is a benchmark model for subsequent sections' Reddit topic and sentiment classification tasks.

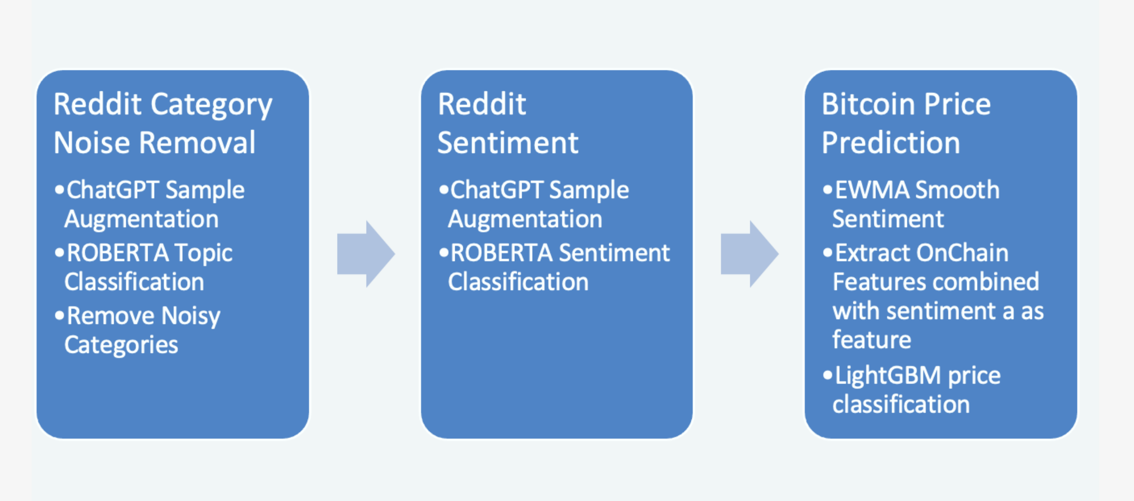
Reddit Category Noise Removal
•ChatGPT Sample Augmentation
•ROBERTA Topic Classification
•Remove Noisy Categories
Reddit Sentiment
•ChatGPT Sample Augmentation
•ROBERTA Sentiment Classification
Bitcoin Price Prediction
•EWMA Smooth Sentiment
•Extract OnChain Features combined with sentiment a as feature
•LightGBM price classification

FIGURE 5.1: Methodological Flow Chart

The upcoming section will showcase the outcomes of applying few-shot learning without sample augmentation, employing BERT as our benchmark. However, the methodology shifts to using RoBERTa in the following sections because its loss function is not trained on next-sentence prediction, making it more suitable for social media text, which often exhibits shorter content.

5.4 BERT Topic and Sentiment Results

As a baseline, we present the few-shot learning results for both BERT topic and sentiment classification models. We also demonstrate the power of leveraging the transfer learning from the BERT models. Further, we also utilize domain adaption on the Reddit text corpus.

Overall, the results seem compelling as per figure 5.2 for topic classification, where we see there is clear outperformance with the BERT fine-tuned vanilla and domain adaptation models obtaining F1 Micro scores of nearly 0.5 versus the Naive Bayes of 0.16 on the test set. Note that there is some marginal outperformance with domain adaptation versus without for the model. There is some imbalance with the class distribution, but the macro F1 test score also shows outperformance. Note that we have five class topic classifications, so achieving close to 0.5 with an F1 score is a competitive score.

In Figure 5.3, which pertains to sentiment classification, we observe a comparable level of outperformance, mirroring what we observed in the topic classification experiments. The BERT models achieve F1 scores of up to 0.43 on the Macro test set, whereas the Naive Bayes approach yields a score of 0.29. Notably, the sentiment model exhibits an even more significant class imbalance, encompassing

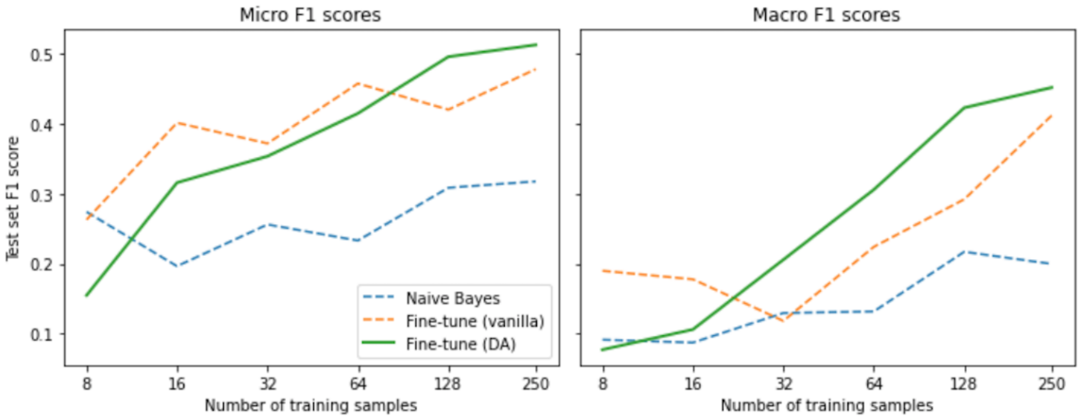

FIGURE 5.2: BERT Topic classification (5 topics) results as training samples grow

three classes: "neutral" "positive" and "negative" where a substantial proportion of observations fall into the "neutral" category. Consequently, the macro F1 score emerges as a more informative metric, assigning equal importance to each class.

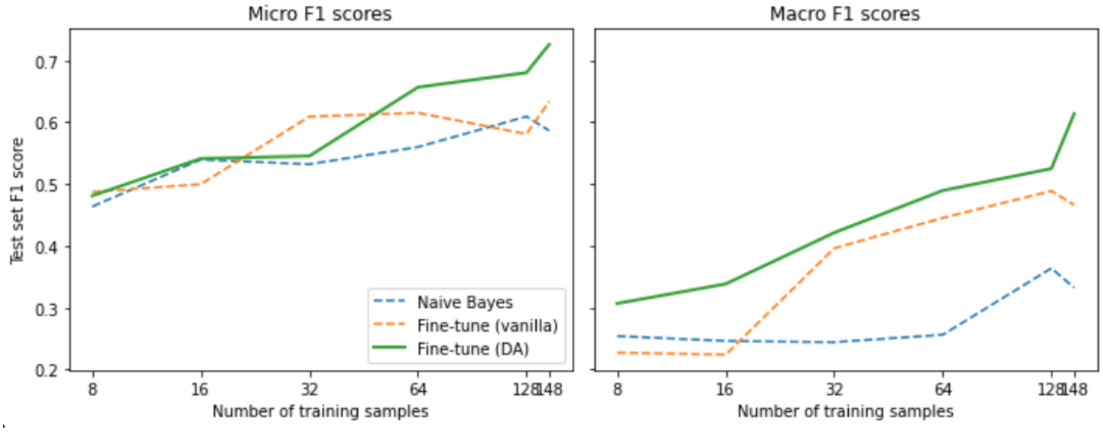

FIGURE 5.3: BERT Sentiment classification (3 classes) results as training samples grow

The following sections introduce our Large Language Model training sample augmentation methodology utilizing ChatGPT. Subsequently, we present a section that combines this augmentation technique with RoBERTa models to enhance Few-Shot Topic and Sentiment classification.

5.5 Sample Augmentation with Large Language Models

5.5.1 Introduction to ChatGPT

The popularity of ChatGPT has surged, surpassing even the Turing test for many applications. Notably, UBS, an investment bank, has confirmed that ChatGPT is the fastest-growing consumer application in history, reaching 100 million users in just two months after its launch[3]. ChatGPT (also known as GPT 3.5), was introduced by OpenAI, builds upon its predecessors, GPT models 1 through 3. Specifically, GPT1 was introduced by Radford et al. (2018), while GPT2 was discussed by Radford et al. (2019), and GPT3 was detailed by Brown et al. (2020). The primary architectural difference in ChatGPT that Schulman et al. (2023) introduced when compared to the other GPT predecessors is the incorporation of Reinforcement Learning from human feedback (RLHF), which informs whether language model outputs were hallucinations versus plausible responses.

All the GPT models, including ChatGPT, are built using the original Transformer architectures (see introduction to Transformers section in the Appendix B.1). The focus on advancements lies in optimizing the training procedures, increasing model size, and cleaning and curating the training data. Notably, the original GPT models contained 117 million parameters, rising to 175 billion parameters with GPT3 and the latest ChatGPT model. The dataset has expanded from several billion tokens to over 300 billion tokens, drawing from diverse data sources such as the

[3] https://www.reuters.com/technology/chatgpt-sets-record-fastest-growing-user-base-analyst-note-2023-02-01/

Common Crawl, Wikipedia, and books corpora. Recent Deepminds paper by Hoffmann et al. (2023) suggests that the GPT models may need more data relative to the parameter count. They discover this by training their Chinchilla language model with 4x of data and outperforming GPT3 and other language models. Additionally, they find that for every doubling in model size, the underlying data should be doubled for language models.

5.5.2 Sample Augmentation methodology

In the previous chapter 4, the Reddit data section introduced a relatively small training set comprising 500 observations for training, 250 for validation, and 250 for testing. However, the potential of Large Language Models (LLMs) can be harnessed to generate additional training samples. Dai et al. (2023), in their paper titled "AugGPT", demonstrate how augmenting the original training set with samples generated by ChatGPT can enhance the performance of a BERT model. Specifically, their study illustrates that the AugGPT method can achieve F1 score improvements ranging from 5-20 percent across three datasets (Amazon, Symptoms, PubMed20k) in BERT-based text classification tasks.

To maintain consistency with the AugGPT study, we utilize the same prompt they used to interact with the ChatGPT API. The prompt is defined as follows:

"Please rephrase the following sentence: text"

Where **text** represents the Reddit question being rephrased, for a specific example, see the original text below that received a manual topic label of "musings" and sentiment label "positive".

"rise of the decentralized exchange and it's perfect for bitcoin. bitshares' decentralized exchange is gaining some traction in China this week. This is very good for Bitcoin because it removes the regulatory hurdles associated with centralized exchanges so that individuals can serve as on-amps for fiat to crypto. i'm even more excited by the prospect that the bitshares exchange will for the first time in human history allow assets (like an ounce of gold or a bushel of corn) to have real price discovery by markets without the manipulation of middlemen. if this works, it will be a major milestone in the development of the crypto ecosystem and a huge step for humanity. since agricultural began centralizing our society 10,000 years ago, price discovery beyond one's village or tribe has been major tool for abuse of power. even today, when a farmer brings in his haul, he must depend on a network of scheming distributors to set the price - and thus determine his/her income. it would be nice to see that opportunity for abuse wiped off the face of the earth. i believe bitshares (and probably other crypto projects) are a step in that direction. http://coinmarketcap.com/currencies/bitshares/markets edit : contrarian arguments as to why this is not important or practical are, of course, welcome."

ChatGPT rephrased it as the following:

"the decentralized exchange is becoming more popular, particularly with bitshares' exchange gaining traction in china. this is beneficial for bitcoin as it eliminates regulatory obstacles associated with centralized exchanges, allowing individuals to act as fiat to crypto on-ramps. additionally, the bitshares exchange has the potential to enable assets such as gold or corn to have genuine price discovery without middlemen's manipulation, a significant milestone in the crypto ecosystem's development and a massive step for humanity. for the last ten thousand years, price discovery beyond one's village or tribe has been a significant tool for abuse of power.

however, crypto projects like bitshares could be a step towards eradicating this opportunity for abuse. while some may argue that this is not essential or feasible, opposing viewpoints are welcome."

From the example above, it is obvious how well ChatGPT summarises text data while maintaining consistency with the initial underlying text. We use the approach outlined above to generate additional training samples to improve the accuracy of the classification models. Future research could explore different variations of how to construct the prompt and to what extent this matters.

5.5.3 Exploratory Data Analysis

Utilizing the initial set of 500 training samples, we employed ChatGPT to generate an additional 12,869 training samples, as detailed in Table 5.1. Due to OpenAI's pricing model, which charges per token request, we constrained the size of the augmented training sample. Nevertheless, this expanded dataset increases our initial training set by 25. Once trained, the RoBERTa classification models can be applied to generate predictions for the entire corpus of 257,435 subreddit Bitcoin posts.

Categories	Count	Count %
other	4,474	34.8%
advice	3,949	30.7%
musings	1,973	15.3%
price	1,631	12.7%
advertisements	842	6.5%
total	12,869	

TABLE 5.1: ChatGPT sample generated training Reddit category classification within the "r/bitcoin" subreddit

Figure 5.4 shows the word count distribution of the initial training set, while Figure 5.5 presents the distribution for the Large Language Model (LLM) augmented training set. Both histograms display similar means, with the initial training set averaging 84 words and the ChatGPT-augmented version averaging 77 words. However, the initial training set exhibits higher volatility, with a standard deviation of 77 compared to 52 for the augmented set. A two-sample Kolmogorov-Smirnov test reveals a statistically significant difference between these two distributions. This difference is somewhat expected, as ChatGPT does not explicitly maintain the same output token length as the input. It's important to note that word length only sometimes directly correlates with the conveyed meaning of the text.

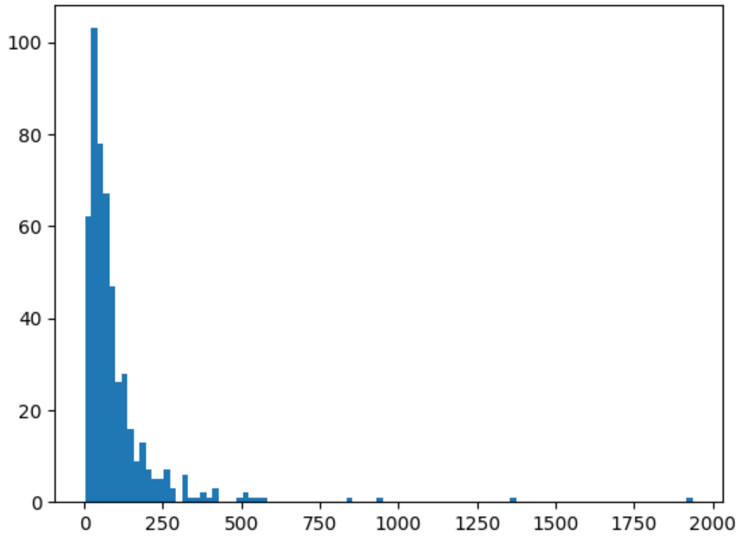

Figure 5.4: Word count histogram of the initial training

Furthermore, there is a strong correlation between the original training text and the ChatGPT-generated responses. Figure 5.6 illustrates the cosine similarity between the texts of the original training sample and the ChatGPT-generated responses. The average cosine similarity is around 0.5, with all positive values indicating a solid alignment. There are a few instances where the cosine similarity

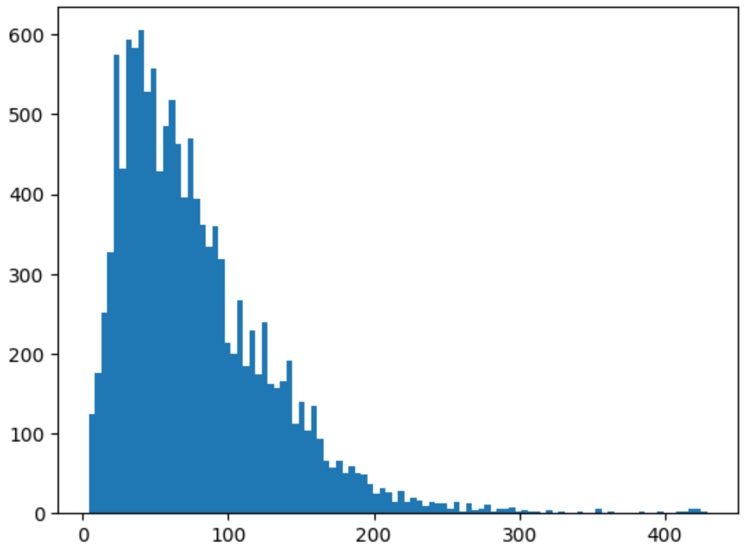

Figure 5.5: Word count histogram of the ChatGPT training generated data

scores zero, suggesting occasional irrelevant responses from ChatGPT. However, most generated content aligns well with the original training set.

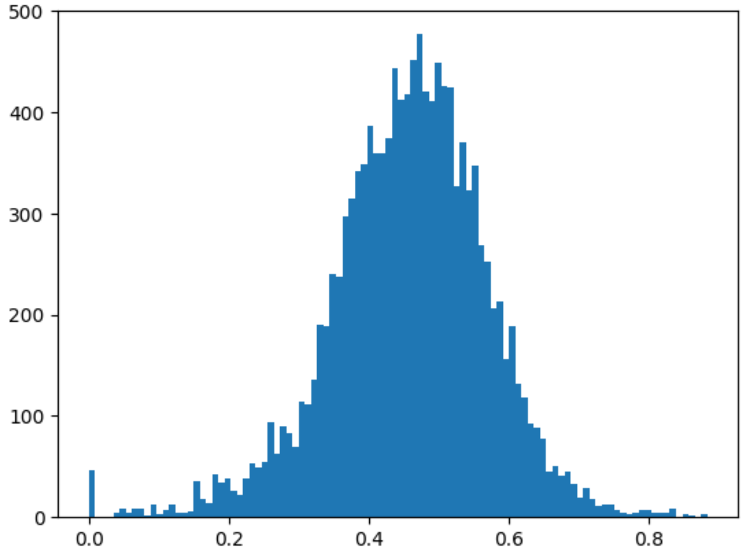

Figure 5.6: Histogram of the cosine similarity between ChatGPT generated response and original training sample

5.6 RoBERTa Topic and Sentiment Results with Sample Augmentation

The augmentation of the sample with Large Language Models (LLMs) enhances the performance for both Naive Bayes and RoBERTa models, evident in both categorical and sentiment-based models. For context, this section represents the first two parts of the methodology diagram described in section 5.3 with ChatGPT sample augmentation. The first box of the diagram is the 5-topic classification model to remove nonrelevant Reddit posts. The second box in that diagram is a 3-topic classification model to predict sentiment (positive, neutral, and negative) for a 3-labeled classification model.

In Figure 5.7, which displays the results for categorical topic classification, the advantage of augmenting the training sample with LLMs is notable. The RoBERTa model augmented with AugGPT achieves an F1 Macro score of 0.53 compared to 0.46 for the original RoBERTa model on the test set. This improvement is even more pronounced when comparing RoBERTa AugGPT against both versions of the Naive Bayes benchmark, with and without AugGPT augmentation. Given the class imbalance in the underlying categorical data, the macro F1 score is significant. The macro F1 score evaluates each class independently and assigns equal weight to each class, as opposed to the micro F1 score, which calculates a weighted average based on the sample sizes in each class.

Similarly, Figure 5.8 demonstrates comparable enhancements in sentiment analysis. The RoBERTa model augmented with AugGPT outperforms the original RoBERTa, achieving an F1 macro score of 0.5 compared to 0.43. Furthermore,

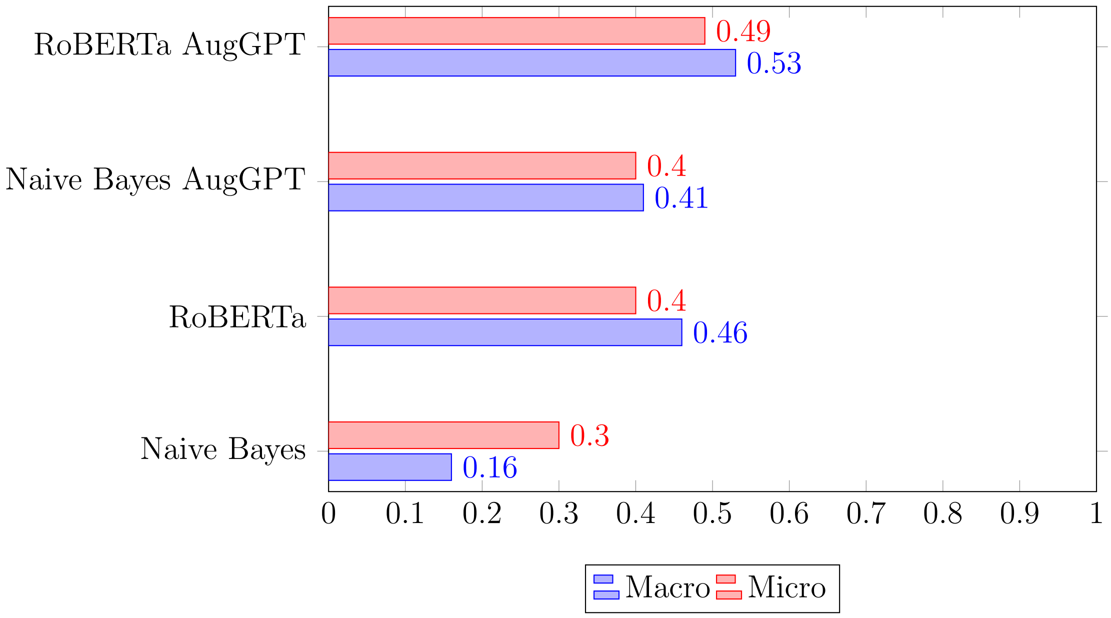

FIGURE 5.7: F1 Test scores for Topic Classification Models

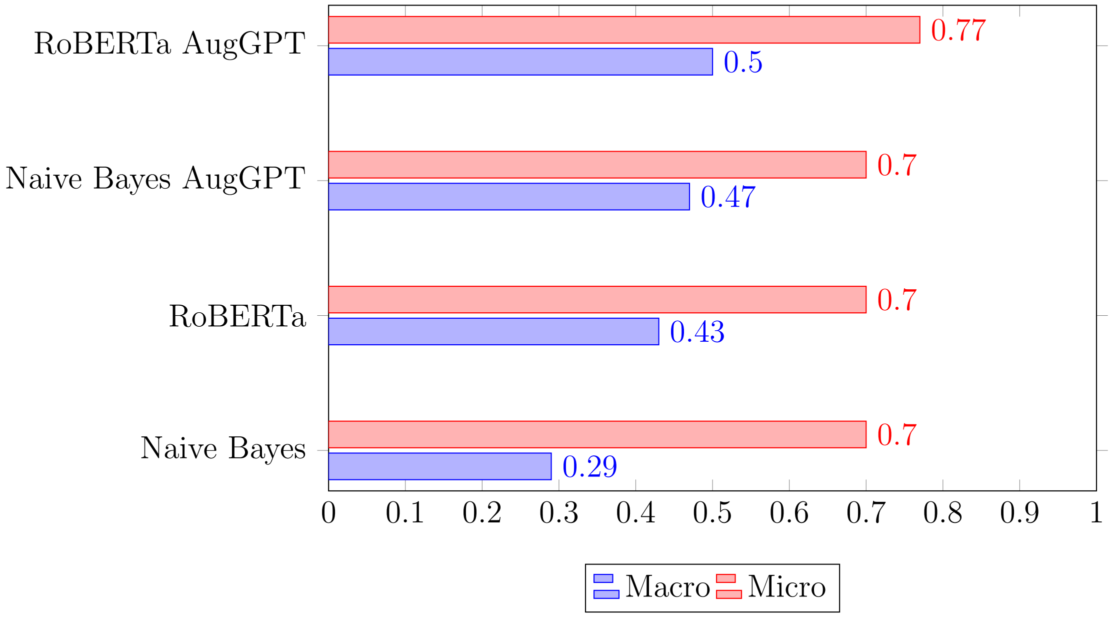

FIGURE 5.8: F1 Test scores for Sentiment Classification Models

RoBERTa AugGPT significantly surpasses the original Naive Bayes benchmarks, obtaining an F1 macro test score of 0.5 versus 0.29.

A comparative analysis of the sentiment and topic classification models reveals that integrating RoBERTa with LLM sample augmentation yields a significant performance enhancement, surpassing the traditional Naive Bayes approach without LLM sample augmentation. Notably, the empirical results demonstrate a substantial improvement of approximately 0.20 in the Macro accuracy F1 metric, underscoring the value of generative AI and state-of-the-art transformer technologies over traditional natural language processing techniques in addressing text classification problems.

5.7 ChatGPT versus RoBERTa

An important question arises regarding ChatGPT's performance compared to our RoBERTa models, specifically in the context of topic and sentiment classification on the test set. To explore this, we designed a prompt for ChatGPT to extract topic and sentiment information and then compared the results with those obtained from the RoBERTa AugGPT model.

5.7.1 ChatGPT prompt design

Our approach involved creating a specific prompt to facilitate the extraction of topic and sentiment classifications using ChatGPT. We also extracted tense information, with ChatGPT tasked to assign topics. It was observed that ChatGPT generally faced challenges in accurately assigning nuanced topic classifications. See the below prompt definition:

"Decide whether the subject of the below Reddits text for the Bitcoin subreddit is 'advertisement', 'technical advice', 'unrelated', 'other', 'price speculation' or 'philosophical/macro musings'. Please only use one of those six phrases for your

responses. After this, determine if the Reddit text below is positive, neutral, or negative, and adjoin this response to the first response using a comma. After this, then determine if the below Reddits text is in the future, present or past tense with only responding future, present or past and adjoin this response to the first two responses by using a comma. After this, then determine if the below Reddits text subject is with only one word and adjoin this response to the first three responses by using a comma."

Note that earlier experiments did try to have ChatGPT identify the topics without any suggested topic prompts as per above. Still, it struggled and cast too broad of a net for topic attribution or went too general with simple labels of "Bitcoin" despite explicit instructions saying not to mention anything about Bitcoin.

5.7.2 Results

Figure 5.9 presents a comparison between RoBERTa AugGPT and ChatGPT based on the previous prompt, focusing on topic and sentiment classification. In the topic classification on the test set, ChatGPT exhibits lower performance compared to RoBERTa AugGPT, with an F1 Macro score of 0.45 against RoBERTa's 0.49. However, ChatGPT demonstrates superior performance in sentiment classification, achieving a Macro F1 score of 0.57 compared to 0.5 for RoBERTa. This outcome is somewhat expected, considering ChatGPT's extensive training on hundreds of millions of tokens, which include a wide range of examples emphasizing overall sentiment tone. Nonetheless, ChatGPT faces challenges in specific topic classification within niche domains, such as Bitcoin-focused subreddits, where nuanced topics are less represented in large language models (LLMs). Future research might investigate the potential benefits of fine-tuning or adapting a large language

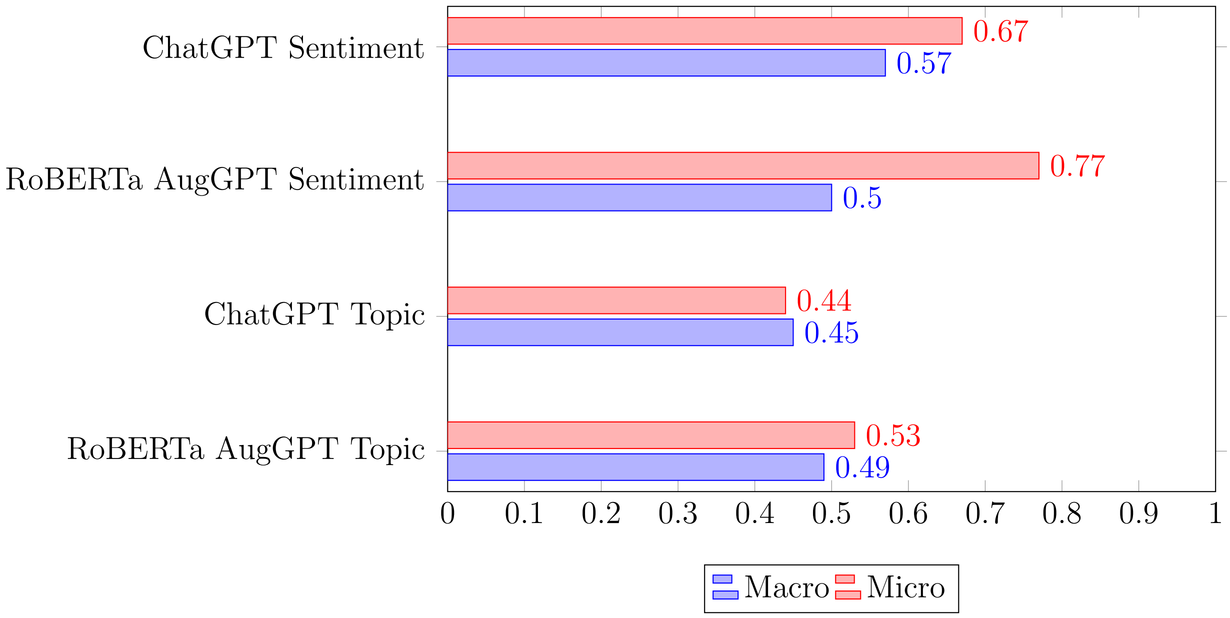

Figure 5.9: F1 Test scores comparing RoBERTa AugGPT versus ChatGPT

model for more specific social media topic classification. Or perhaps, In-Context LLM training may also be a worthwhile research area.

It is also worth noting that additional experiments conducted without guiding ChatGPT on specific topic labels resulted in poorer performance. The model often generalized topics too broadly; for instance, in multiple iterations, it incorrectly labeled texts as “Bitcoin” despite instructions to avoid such generic labeling. Since all data was sourced from the “r/bitcoin” subreddit, labeling everything as “Bitcoin” proved unhelpful.

Furthermore, practical considerations regarding cost are significant, especially as expenses increase proportionally with data size when utilizing OpenAI’s API. For this thesis, the total expenditure for API calls amounted to a few hundred dollars for approximately 15,000 observations. Extrapolating this to our entire Reddit dataset of 250,000 entries, the estimated cost would have been around $2,500. In contrast, using RoBERTa via Hugging Face incurred no financial costs.

5.8 Multivariate model for Bitcoin Prediction

This section consolidates the analysis to determine if filtering out irrelevant Reddit posts (the "noise") while using sentiment significantly enhances the prediction of Bitcoin's price. As per section 5.3, where the methodology is represented in a diagram, this section predicts if Bitcoin will be positive or negative (i.e., two labels) 60 days out in a multivariate classification model. Note that

We include various on-chain metrics to account for additional external factors influencing Bitcoin's value. The primary objective is to compare two models, one incorporating the Reddit sentiment with reduced noise and another with its exclusion, to assess their respective impacts on predictive accuracy. Given the complexities of financial markets, our focus shifts to a classification approach, as predicting binary outcomes often proves more manageable than forecasting continuous variables. Authors Wu et al. (2020) have a similar conclusion: classification models outperform level estimation models in terms of accuracy and profitability when it comes to predicting stock market trends and optimizing investment returns.

Considering the success of gradient boosting models in numerous data science competitions, particularly XGBoost and its faster derivative, LightGBM, as highlighted by Goldbloom (2016), we have chosen to leverage the LightGBM model for this thesis. LightGBM's efficiency in handling large datasets and its ability to capture complex nonlinear patterns make it an ideal choice for this research.

Subsequent sections will offer an overview of LightGBM models, detailing the data used, the model training process, the results obtained, and an evaluation through a simple trading strategy backtest.

5.8.1 LightGBM introduction

LightGBM is a gradient-boosting framework based on an ensemble of decision trees whose aim is its speed and efficiency for estimating the model. It helps with larger data sets, has many input features, and captures non-linearity. LightGBM is similar to the popular machine learning algorithm called XgBoost, where the primary difference between the two methods is with the construction of the trees. Specifically, LightGBM grows leaf-wise, i.e., from left to right, whereas XgBoost grows tree level-wise, row by row. Further, Ke et al. (2017), when introducing the LightGBM framework, states that the traditional approaches need to scan all the data instances to estimate information gain, which can be very time-consuming. Instead, they introduce two innovative ideas: Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB). For GOSS, they exclude data instances with small gradients and use the rest to estimate the information gain. With EFB, the approach bundles mutually exclusive features to reduce the feature redundancy in the data. Ultimately, the authors find that LightGBM achieves the same accuracy as classical gradient boosting approaches but at 20 times the speed.

For the GOSS algorithm 3, the accuracy of the algorithm would go down if the algorithm completely discards all data with small gradients, as data with small gradients implies the training error is small and well-trained (this would also impact the underlying data distribution). However, to avoid this problem, the GOSS algorithm keeps all large gradients and performs random sampling on data with small gradients. The algorithm sorts the absolute value of all gradients and selects the top z x 100% of the data. It then randomly samples b x 100% the rest of the data not considered in the large gradient bucket. Finally, GOSS scales up the randomly sampled data with small gradients by $\frac{1-a}{b}$ when computing information gain.

Algorithm 3 Gradient-based One-Side Sampling

```
Input: I: training data, d: iterations
Input: a: sampling ratio of large gradient data
Input: b: sampling ratio of small gradient data
Input: loss: loss function, L: weak learner
  $Models \leftarrow [], fact \leftarrow \frac{1-a}{b}$
  $topN \leftarrow a * len(I), randN \leftarrow b * len(I)$
  for i = 1 to d do do
      $preds \leftarrow models.predict(I)$
      $g \leftarrow loss(I, preds), w \leftarrow (1, 1, ...)$
      $sorted \leftarrow GetSortedIndices(abs(g))$
      $topSet \leftarrow sorted[1 : topN]$
      $randSet \leftarrow RandomPick(sorted[topN : len(I)], randN)$
      $usedSet \leftarrow topeSet + randSet$
      $w[randSet]x = fact$ ▷ Assign weight small fact to small gradient data
      $newModel \leftarrow L(I[usedSet], -g[usedSet], w[usedSet])$
      $models.append(newModel)$
  end for
```

For the EFB approach, the authors construct a graph with weight edges where the weights represent the total amount of conflicts between features. The EFP approach then sorts the features by the degrees in the graph. Finally, each feature gets assigned to an existing bundle with low conflict or creates a new bundle. To further improve the EFB, the authors propose a more efficient ordering strategy without building a graph. Specifically, they order by the count of nonzero values, similar to ordering by degrees, since more nonzero values can lead to more conflicts. They then create feature bundles to allow exclusive features to reside in different bins by adding offsets to the original values.

5.8.2 Gradient Boosting Decision Tree Model Setup

In this study, we employ the LightGBM model to examine whether the inclusion of noisy Reddit posts in the sentiment analysis feature skews predictions regarding Bitcoin's price movement over 60 days. We utilize a LightGBM binary classification model, drawing on the concept of Gradient Boosting Decision Trees introduced by Friedman (2001). This approach involves an output response y, defined as the sign of Bitcoin's 60-day return, and a set of input variables $X = (x_1, x_2, .., x_p)$. These input variables comprise various on-chain features along with the Reddit Smoothed Sentiment feature, with models running with and with reduced noise.

As detailed in the Glassnode data 4.3.2 section, the thesis incorporates these blockchain indicators as input features to counteract the influence of market dynamics on Bitcoin's price. Moreover, the Reddit sentiment feature is scrutinized to determine its contribution with and with reduced noise. Given daily sentiment fluctuations and potential weekday seasonality, we apply a 60-day exponential weighted moving average (EWM), as depicted in Figure 5.10.

We can see from Figure 5.11 on the left hand side that the daily aggregated Reddit sentiment is a stationary process as does not contain much memory whereas the right hand side after applying a 60-day EWMA converts it into a nonstationary process allowing for some memory. This is important to do before we feed it into a multivariate LightGBM model, as this matches the timescale of the Reddit Sentiment input feature when predicting the sign of Bitcoin 60 days out.

For testing purposes, the study adheres to LightGBM's default hyperparameters, maintaining a standard framework for evaluation. For example, a default learning rate of 0.1 is utilized. Specifically, the hypothesis tests the accuracy of two models:

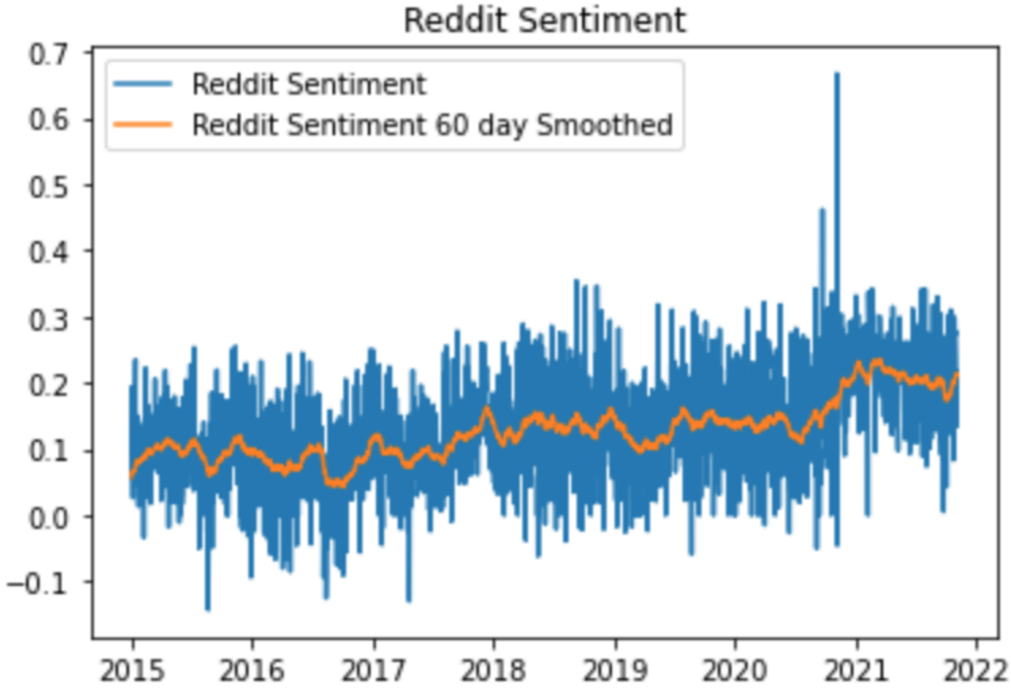

FIGURE 5.10: Reddit Sentiment with 60-day EWM smoother

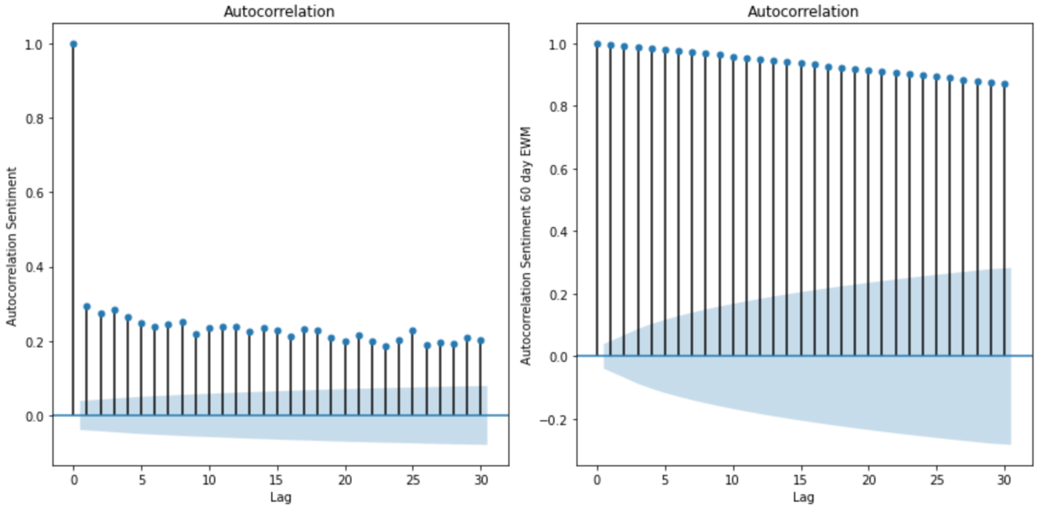

FIGURE 5.11: Autocorrelation function plots. The left-hand side is the daily Reddit Sentiment with no smoother. The right-hand side is the daily Reddit Sentiment after applying the 60-day EWMA

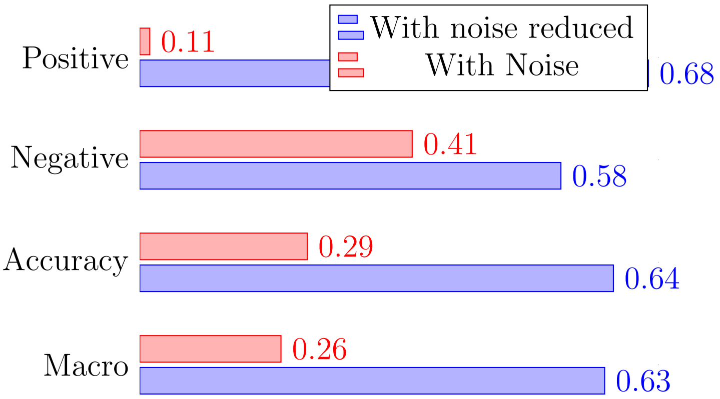

FIGURE 5.12: F1 Test scores predicting 60-day Bitcoin return using onchain and Reddit with and with noise reduced

one using Reddit Sentiment smoothed across all topic categories and the other employing Reddit Sentiment smoothed with noise filtered out.

5.8.3 With and with reduced noise results

Figure 5.12 shows the F1 test score outperforms with noise reduced across positive, negative, accuracy, and Macro metrics. Said differently when comparing the two different versions of the model when Reddit Sentiment is included with noise, i.e., all the topic categories, including advertisement, technical advice, other, which are hypothesized to have no bearing on the price of Bitcoin and Reddit Sentiment with noise reduced, the noise reduced version for Reddit Sentiment has better predictive capability.

In summary, these results validate the hypothesis that noise reduction improves predictive accuracy and offer broader lessons for developing sentiment models using social media data. They underscore the necessity of data quality and relevance in model construction, providing a valuable reference point for future research in this area.

5.8.4 Trading Strategy: with noise reduction results

To illustrate a straightforward trading strategy utilizing the LightGBM model, which incorporates the Reddit sentiment feature with noise reduction, we can establish a buy signal when the model predicts a greater than 50% probability of a positive 60-day return for Bitcoin. Conversely, a sell signal is generated when the predicted probability falls below 50%. This approach is demonstrated in Figure 5.13, which presents the cumulative returns of an out-of-sample back-test starting from 2021, compared to simply maintaining a long position in Bitcoin.

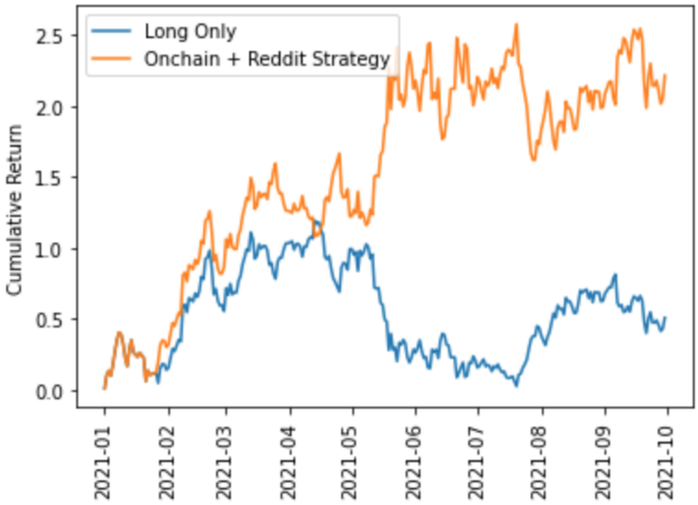

FIGURE 5.13: Out of sample backtest for Bitcoin, comparing long only versus a trading strategy of following the predictions of the LightGBM model using onchain and Reddit features

By filtering out noise and irrelevant data from social media, the model can more accurately gauge market sentiment, potentially leading to better informed trading decisions. This method of incorporating social media sentiment into financial trading should greatly interest financial market participants. Such insights are precious in the ever-evolving landscape of cryptocurrency trading, where traditional market analysis techniques may only sometimes capture the complete picture of investor sentiment and market trends.

5.9 Summary

This chapter presents a novel approach to enhancing sentiment analysis in financial markets, specifically on Bitcoin. By leveraging advanced encoders such as RoBERTa, the research systematically removes noise from Reddit data and integrates Large Language Models (LLMs) for augmenting training samples. Where in the end the development of a nonlinear multivariate LightGBM model that integrates a clean Reddit sentiment signal with various on-chain features, predicts price direction of Bitcoin 60 days in advance.

A novel contribution of this thesis is the development of a noise reduction methodology tailored for social media sentiment. This methodology involves using a RoBERTa-based topic classification model, refined through training sample augmentation with ChatGPT. The methodology merges three distinct models: RoBERTa for topic classification, RoBERTa for sentiment classification, and the LightGBM multivariate model. This trifecta aims to predict Bitcoin's price direction accurately.

This research fills a gap by augmented sentiment analysis, especially in filtering noisy social media datasets. Specifically, the research employs Few-Shot learning with LLM-based training sample augmentation, providing enhanced topic assignment compared to unsupervised methods like LDA.

The chapter begins with an introduction to Sentiment Analysis, laying the foundation for understanding the field. It then delves into various models, including BERT, RoBERTa, and the Naive Bayes Classifier, and outlines our methodology for isolating noisy topics. Subsequently, the chapter introduces sample augmentation using ChatGPT, followed by an analysis of combining RoBERTa with this method. A section benchmarking the augmented RoBERTa model against ChatGPT is also included. Overall, we show that utilizing generative AI and transformer technologies outperform traditional natural language processing techniques such as Naive Bayes by 0.2 Macro F1 Score performance gain for both topic and sentiment text classification models.

As another validation test, the chapter synthesizes the findings into a practical application using a noise-reduced sentiment signal versus the original sentiment signal, integrated into a multivariate LightGBM model that predicts the price direction of Bitcoin 60 days in advance. The findings are clear, with this evidence showing that using the noise-reduced sentiment signal results in a performance gain of 0.63 in Macro F1 Score versus 0.26.

This research is particularly relevant in an era where financial markets are increasingly influenced by social media sentiment. It provides a more accurate tool for sentiment analysis and a framework for incorporating social media data into financial market predictions. Integrating advanced NLP techniques and machine learning models like LightGBM demonstrates the potential for significant improvements in predicting market movements based on social sentiment. This approach is academically significant and has practical implications for investors and market analysts seeking to leverage social media data for better financial market prediction.

Chapter 6

Conclusion

This thesis stands out by pushing the boundaries of traditional natural language processing techniques such as lexicons, cosine similarity, and naive Bayes. Instead, it pioneers the use of cutting-edge technologies from deep learning and transformers to develop innovative methodologies within semantic textual differences and sentiment analysis, with a focus on noise reduction within the finance domain. The thesis leverages two primary data sets, including 10-Q filings for semantic textual differences and constructing reduced noise sentiment signals from social media data originating from Bitcoin's subreddit.

The first part of the thesis focuses on SEC 10-Q filing data for semantic textual comparisons between two quarterly filings. For this curated data set on S&P 500 individual stocks, we find that concatenating deep learning architectures and large language models outperform traditional natural language processing techniques such as cosine similarity in terms in terms of predicting excess returns as well as outperforming on accuracy metrics such as F1 score. As far as can be determined, this research has used Deep Learning and Large Language Models for semantic textual difference on SEC 10-Q filings. In addition, further contributions involve novel ways of using concatenating deep learning structures such as Max Embeddings and CNNs with underlying time series text data where each CNN or Max Embeddings represents the current and the previous quarterly filing where the ultimate classification is if the excess return 180 days out from the current filing is negative or positive. Regarding Large Language Models, the primary contribution of the thesis is developing the Summarize and Sentimentize approach by using Claude2 for constructing language summaries on what has changed from a legal, profit, and business risk perspective, where the final task is assigning a sentiment score.

As noted in this thesis, fundamentally comparing semantic textual differences will and has already started to change with these types of advances. Recently, a popular Python language interface, LangChain, has advocated using Large Language Models for comparing model-generated answers versus ideal response answers versus cosine similarity. For instance, you could have two responses to a question "is Apple's stock a good buy?" with the first one being "yes" and the second one being "Apple's stock is indeed a good buy given its current growth trajectory". If one used cosine similarity, it would generate a zero for how similar the two responses are; however, with a Large Language Model, one can reason that the two text passages are saying the same thing. The work done in this thesis, we believe, has fundamentally advanced the nature of comparing SEC filings for predicting S&P 500 stocks using the latest advances in deep learning and large language models in a new novel methodological manner.

The second half of this thesis uses social media data from Bitcoin's subReddit, constructing a clean sentiment signal by removing noisy topic categories. Combining this enhanced sentiment signal with other on-chain features in a nonlinear LightGBM model, the F1 score predicting Bitcoin's price direction 60 days out goes from 0.26 to 0.63 on the test set. This is the only research as far as can be determined that systematically removes noisy social media categories with the aim of an enhanced sentiment signal. In addition, further contributions of this thesis involve sample augmentation using ChatGPT on an initial manually curated topic attribution data set for Bitcoin's subreddit. This step is crucial as otherwise, one is left with unsupervised approaches for topic attribution, which either casts too broad of a net, too general, or nonrelevant. The other main contribution of the thesis is the overall methodology of using the Roberta model for estimating a topic classification model as a pre-step before the overall sentiment signal is computed.

After removing noisy categories, this research for enhanced sentiment on social media adds essential contributions to new approaches for conducting few-shot topic classification with RoBERTa, augmenting the training sample size with large language models. Typically, in the field of sentiment and financial markets, all topic categories are used to construct sentiment signals. This novel approach, as detailed above, has the potential to transform the way research is conducted with sentiment on social media. Further, the combination of RoBERTa few-shot learning with ChatGPT sample augmentation seems to be a powerful approach that can be applied beyond finance.

Future research can go in many directions, but one worthy of note would be to explore retraining LLMs from scratch without any look-ahead bias. For instance, if one were trying to use LLM to back-test (i.e., historically simulate) whether or not to purchase Apple's stock in April 2014, an LLM should only be trained on text data prior to March 2014 to prevent data snooping. This barrier would be a tremendous investment in training costs, but the pros and cons should be weighed. For this thesis, we believe this is not an issue as we are using LLMs in ways of objectively comparing either two 10-Q SEC financial filings or using it for sample augmentation for subreddits for Bitcoin, neither of which are instructing the model to look ahead to make predictions.

Other areas of future research could expand on what projects such as TimeGPT are pursuing where 100s of billions of rows of time series data are collected across financial, weather, and web data to enable forecasting. One can imagine a pre-trained model of this nature where specific fine-tuned applications would be decisive. There would be additional variability depending on what exogenous factors get fed in versus taking historical univariate time series patterns.

Appendix

Appendix A

Topic Extraction on SEC filings

This section represents work done at the very early stages of the thesis. It can be thought of as exploratory data analysis in terms of understanding the data more thoroughly. Additionally, there were some initial attempts at using topic differences between filings to uncover language changes from one quarter to the next; however, the initial results did not seem promising for this intent.

A.1 Traditional Topic Extraction

This section employs Latent Dirichlet Allocation (LDA) and Aspect Embeddings for exploratory topic extraction analysis on SEC filings. Utilizing the Aspect Embeddings methodology on SEC filings is considered a novel application as part of this thesis. The following sections present the LDA and Aspect Embeddings methodologies, followed by an application using Boeing's 10-Q document.

A.1.1 Methodologies

In Natural Language Processing, applying an unsupervised topic clustering algorithm is a common task to extract insights for text data. These models help uncover hidden themes within a document or classify documents. One of the most widely used unsupervised models for this task is Latent Dirichlet Allocation (LDA), introduced by Blei et al. (2012), which employs a generative statistical model. In LDA, observations are words within a collection of documents, where each document is a mixture of topics, and each word is associated with one of the word topics. While LDA performs reasonably well in overall clusters of topics, it often struggles to extract individual aspects related to those topics. Mimno et al. (2011) found that LDA models do not explicitly model word-to-word co-occurrences as the primary source of topic coherence. Still, they implicitly capture this through the generative process, where one fundamental assumption is that each word is conditionally independent. In the following sections, we will introduce both an LDA and Aspect Embeddings and show how this can generate more succinct topic clusters.

A.1.1.1 Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA), introduced by Blei et al. (2012), is a widely adopted unsupervised learning method for topic modeling. Topic modeling is the process of discovering hidden topics or themes within a collection of documents, allowing the identification of a collection of words associated with a specific theme. LDA is a generative probabilistic model applied to a text corpus. The fundamental concept underlying LDA is that documents can be characterized as random mixtures over latent topics, where each subject has a distribution of words. As described by the authors above, LDA is a generative process that can be defined as follows for each document w in a corpus D:

1. Choose $N \sim Poisson(e)$.
2. Choose $\theta \sim Dir(\alpha)$
3. For each of the N words w_n:
   (a) Choose topic $z_n \sim Multinominal(\theta)$
   (b) Choose word w_n from $p(w_n|z_n, \beta)$, a multinomial probability conditioned on the topic z_n

A word is defined as unit-based vectors from 1 to V, where it is assigned 1 when the word is present and zero otherwise across the vocabulary of the corpus. Where a document is a sequence of N words denoted by $\mathbf{w} = (w_1, ..., w_N)$ and a corpus is a collection of N documents denoted by $D = (\mathbf{w_1}, ..., \mathbf{w_N})$.

Then given the parameters α and β, the joint distribution of a topic mixture θ, set of N topics for $\mathbf{z}$ and set of N words $\mathbf{w}$ is defined by the joint distribution:

$$p(\theta, \mathbf{z}, \mathbf{w}|\beta, \alpha) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta) p(w_n|z_n, \beta)$$

Where the marginal probabilities and distributions of the Documents can be derived as per the authors, the final output of the LDA model generates a list of words per each thematic topic cluster that can be extracted.

A.1.1.2 Aspect Embeddings

Aspect extraction is a critical task in certain forms of sentiment analysis, where an algorithm identifies the topic (i.e., category) for a sentence to discern the associated opinion. He et al. (2017) developed an Unsupervised Neural Attention Model Aspect Extraction using a restaurant review dataset. For example, in the review "*The beef was tender and melted in my mouth*", the aspect term would be labeled "*beef*" and the second task would be to cluster this term with other similar terms consolidating into one overall aspect labeled *food*. Aspect Embeddings are valuable as they allow for extracting critical topics from a single text corpus.

Another significant advantage of Aspect Embeddings is their ability to capture specific sentiments associated with different aspects of the text. For example, in an Amazon review of a laptop where a user states "the screen is excellent but the keyboard doesn't work", the aspects are the "keyboard" and the "screen". When assigning a sentiment score, being specific about each topic cluster is advantageous.

In the Aspect Extraction model proposed by He et al. (2017), the process begins with a Word embeddings matrix, such as Word2Vec as per Mikolov et al. (2013), which already contains word mappings of the co-occurrences of words within the same context. The authors then utilize the Attention Mechanism, informed by Vaswani et al. (2017), to filter the word embeddings and construct what they term "Aspect embeddings." The process is analogous to autoencoders, conceptualized as a dimension reduction exercise, and referred to as Attention Based Aspect Extraction (ABAE). The ultimate objective is to learn the aspect embeddings matrix $T \in \mathbb{R}^{Kxd}$ where K is the number of aspects from a word embeddings matrix $E \in \mathbb{R}^{Vxd}$ where V is the vocabulary size. Figure A.1 illustrates the neural net approach of how word embeddings can be processed into Aspect embeddings.

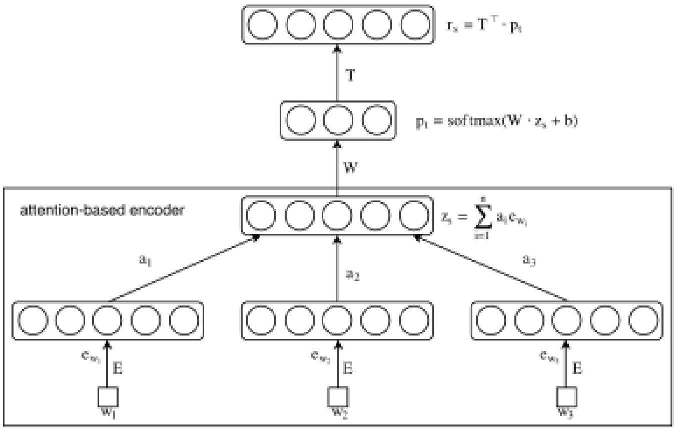

Figure A.1: Example of an ABAE structure

The basic approach first formulates a weighted sum z_s for each input sentence s where for each word w_i, a_i represents a positive weight that can be thought of as a positive probability the word is in the right group.

$$z_s = \sum_{i=1}^{n} a_i e_{w_i} \tag{A.1}$$

Where a_i is computed from the Attention model, which is a function of a word embedding e_{w_i}.

$$a_i = \frac{exp(d_i)}{\sum\limits_{j=1}^{n} exp(d_j)} \tag{A.2}$$

$$d_i = e_w^T \cdot M \cdot y_s \tag{A.3}$$

$$y_s = \frac{1}{n} \sum_{i=1}^{n} e_{w_i} \tag{A.4}$$

Where y_s is the average word embeddings, which aims to capture the global context, M is a mapping between global context and word embedding. The word transformation through M captures how closely it matches with the Aspect topics.

As described by He et al. (2017), the final step reconstructs the sentence embedding through a linear combination of the aspect embeddings T. This dimension reduction and reconstruction process transforms filtered sentences z_s into reconstructed sentences r_s across K embedded aspects.

A.1.2 Results

To assess and compare the effectiveness of the LDA and the Aspect Extraction models, Boeing 10-Q filings from Jan 1995 to Dec 2015 were processed. One crucial consideration in both models is determining the number of topics to pick. Existing LDA research on SEC filings by Brown et al. (2017) found that 13 topics generated the most intuitive clusters. While typically, for LDA, coherence plots can be used to determine the number of topics, a subject matter expert can decide the best number of topics based on intuition, which is typically a preferred option.

Tables A.1 and A.2 present the top five words for each of the 13 topic clusters per model. A clear distinction emerges in this example, highlighting that the LDA model struggles to distinguish between the topic clusters and suffers from word redundancy as words are repeated over the different topic clusters. In contrast, the ABAE model demonstrates clear, distinctive topic clusters. For instance, topic 2 is about order inventories, topic three about traditional financial results, topic 7 is about liquidity, and topic 13 is about pensions. The LDA model, on the other hand, lacks clear and distinctive topic themes.

An intriguing extension for utilizing topic extraction involves examining how topic clusters evolve to empirically test their potential for predicting excess returns. However, a drawback of Aspect Extraction is its inherent limitation compared to the LDA model. The LDA model allows for obtaining a probability distribution for each document. One way to bypass this is to use the ABAE model on a rolling basis when a sufficient sample size is available. However, initial tests with Aspect Extraction demonstrated suboptimal performance, likely due to the reduced sample size.

Topics	Word 1	Word 2	Word 3	Word 4	Word 5
1	aircraft	earnings	program	compared	commercial
2	due	three	earnings	program	aircraft
3	earnings	aircraft	due	operating	three
4	billion	systems	operating	backlog	earnings
5	aircraft	earnings	increased	program	commercial
6	revenues	program	due	programs	three
7	aircraft	program	due	earnings	commercial
8	three	program	due	earnings	aircraft
9	due	earnings	aircraft	operating	three
10	earnings	commercial	program	aircraft	compared
11	nine	iam	concession	oci	nonmonetary
12	earnings	due	program	three	aircraft
13	aircraft	commercial	program	earnings	increased

Table A.1: LDA model top five words

A.2 Similarity Topic Extraction using ChatGPT

This section introduces a novel method that extends beyond traditional topic modeling, leveraging the capabilities of the large language model ChatGPT for more nuanced topic extraction based on similarity when comparing two documents.

Topics	Word 1	Word 2	Word 3	Word 4	Word 5
1	breakeven	continuation	diluted	depends	negatively
2	backlog	unobligated	entitlement	canceled	fulfilled
3	charge	operations	recorded	loss	aircraft
4	pose	could	supplier	date	issue
5	ula	alliance	united	launch	venture
6	support	commercial	service	aviation	military
7	paper	borrowing	liquidity	operations	serve
8	statement	consolidated	condensed	see	incorporated
9	adding	learning	increasing	curve	coordinating
10	portfolio	investment	receivable	run	inability
11	preliminary	deteriorated	extending	extra	statutory
12	quantity	operations	unit	accounting	normal
13	insight	item	contentsother	pension	retirement

Table A.2: Aspect Embeddings top five words

This new methodology involves creating an algorithm for processing two documents to compute a similarity score based on reasoning, extracting topics where the documents are similar and least similar using ChatGPTs API.

A.2.1 Methodology

Large language models such as ChatGPT offer the potential to surpass traditional topic modeling by providing specific instructions embedded within the prompt. The approach extends beyond simple topic analysis, prompting the language model to extract the most similar and least similar words, offering additional contextual insights. The algorithm described below applies this approach for further context, instructing the language model to extract words that are the most and least similar 4.

Prompt instructions are defined as:

"Please only respond with one word representing a continuous score between zero and one indicating if the 10-Q language changed, zero indicating maximum change, and 1 indicating no change. Then please list the top five topics in which the filings are most similar, followed by the top five in which they are least similar. And only listing one word for each and everything separated by a comma."

Algorithm 4 Similarity Topic Extraction with ChatGPT

```
Input: pass in prompt instructions
for each data item $d_i$ in the dataset do
    Initialize with prompt "Please see these two below 10-Q filings: $d_i$ and $d_{i-1}$ "
    Pass prompt definition
    Send the prompt to ChatGPT and receive a response $r_i$
    Process $r_i$ to extract a similarity score and a list of the 5 top most similar and least similar words
end for
```

A.2.2 Results

For illustrative purposes and considering the cost associated with calling OpenAIs API, the analysis focuses on four quarters to show the power and capabilities of LLMs. Table A.3 presents the results from the above-introduced algorithm 4. We can see from the algorithm results that it can generate continuous scores and provide text-based topics, indicating the versatility of what an LLM can do.

The power of LLMs extends beyond traditional topic extraction methods, as they can identify which topics exhibit the highest and lowest similarity between two documents. The list of most similar words includes a range of generic terms such as "financial", "operations", and "expenses", which are commonly used words in each quarter. In contrast, the list of least similar words includes terms such as "legal", "liquidity", and "stock repurchase", which may need to be consistent or align with the typical language used in each quarter.

Furthermore, LLMs offer similarity scores when prompted, as demonstrated in the row labeled "ChatGPT" in table A.3. The actual Cosine Similarity is computed between the two 10-Q documents and presented in the "Cosine" row for comparative purposes. Unlike traditional metrics such as Cosine Similarity, LLMs can reason about the similarity between two passages of text instead of simply processing a metric that represents the percentage of word match. Notably, the example in table A.3 on 4/07 indicates a significant change in the language. During this period, Boeing struggled with employee strikes and customer concessions, aligning with the observed similarity score of 0.2, which reasonably reflects the shift in language during the company's challenging period.

similarity	10Q-10/06	10Q-4/07	10Q-7/07	10Q-10/07
ChatGPT	0.80	0.20	0.8	0.80
Cosine	0.90	0.81	0.96	0.96
similar1	financial	revenues	financial	financial
similar2	operations	expenses	operations	operations
similar3	revenue	net earnings	earnings	subsidiaries
similar4	expenses	assets	taxes	earnings
similar5	subsidiaries	liabilities	subsidiaries	taxes
leastsimilar1	acquisitions	tax positions	contingencies	contingencies
leastsimilar2	investments	stock repurche	investments	securities
leastsimilar3	stockholders	defined benefic	acquisitions	liquidity
leastsimilar4	pension	cust concessions	research	investments
leastsimilar5	taxes	non-employee comp	legal	legal

Table A.3: ChatGPT Topic Analysis

A.3 Summary

New applications using unsupervised NLP techniques, such as Aspect Embeddings, were employed for topic and theme extraction from the corpus and comparison with Latent Dirichlet Allocation (LDA). In an applied example using Boeing's 10-Q history, the Aspect Embedding model outperformed LDA in clearly distinguishing intuitive topics related to financials, operations, liquidity, and more.

Additionally, a new methodology leveraging Large Language Models was introduced to identify the similarity between two filings and extract the topics that were most and least similar. In an example comparing four Boeing 10-Q documents, the LLM similarity score demonstrated stronger intuition than the traditional cosine metric. The LLM identified expected consistent topics like financials and operations as most similar while surfacing useful outlier topics that changed, such as legal issues and liquidity.

Appendix B

Introduction to Transformers

This chapter provides an in-depth introduction to Transformers, covering their architecture, self-attention, multi-head attention, feed-forward neural network, encoder-decoder stacks, residual connections, layer normalization, and positional embeddings.

B.1 Background

The speed at which the development of Large Language Models is occurring is incredible. One of the significant breakthrough papers in Natural Language Processing occurred in 2017 by Vaswani et al. (2017) called "Attention is all you need" that paved the foundations for Transformer models. According to Google Scholar, the paper has been cited close to 100k times since 2017. Other research tools from Dimension AI put the citation number at around 540k across a broader range of sources, including arXiv and IEEE Access. Further, transformer-based models have significantly outperformed traditional natural language processing models and traditional deep learning models like Recurrent Neural Networks (RNN) across most top NLP benchmark datasets like GLUE and SQUAD, for example. For this section, the thesis draws inspiration from the Natural Language Processing with Transformers book by Tunstall et al. (2022) as found this to describe the Transformer anatomy the best along with Alammar (2019) who has some of the more popular blog posts on Transformers[1].

Transformer architecture reconceptualized how sequence text data is processed, where traditionally, using Recurrent Neural Networks (RNN) models was the convention. Specifically, unlike RNNs that process one word at a time, transformer-based models can leverage parallelization, which enables them to speed up estimation time and capture long-range dependencies more effectively.

Two of the more famous NLP models built utilizing the transformer architecture include Bidirectional Encoder Representations from Transformers (BERT) by Devlin et al. (2018) and Generative Pre-trained Transformer (GPT) by Radford et al. (2018). Variations of these models have beat every NLP benchmark across most tasks, including sentiment analysis, language models, named entity recognition, question answering, and text summary, to name the top ones.

B.2 Transformer Architectures

As discussed in Tunstall et al. (2022), there are hundreds of transformer models, but they broadly fall into three main architectures: encoder-only, decoder-only, and encoder-decoder.

[1] His blog posts have been recognized by top academic institutions and have incorporated them into their curriculum such as Stanford among others.

Encoder-only transformers like BERT, ROBERTA, and DistilBERT focus on encoding an input text sequence into a rich numerical representation. This representation is well-suited for NLP tasks like text classification, sentiment analysis, and named entity recognition. Encoder-only models process text bidirectionally, ingesting the full context in both directions. They encode the semantic relationships within and across sentences into vector representations.

In contrast, decoder-only transformers like GPT are autoregressive language models. They focus on predicting the next token in a sequence given the previous context, generating text token-by-token in a unidirectional manner. Different GPT variants have pushed the boundaries of how coherent a text sequence these decoder-only models can produce.

Encoder-decoder transformers like BART (Bidirectional and Auto-Regressive Transformers) combine encoder and decoder and excel at sequence-to-sequence tasks. The encoder ingests and encodes an input sequence, and the decoder generates an output sequence conditioned on that encoding. This makes them well-suited to language translation, summarization, and dialogue tasks.

For this thesis, a ROBERTA model is utilized for text classification, building on BERT's bidirectional encoding approach. Additionally, ChatGPT is used for data augmentation, given its impressive text generation capabilities as a decoder-only transformer.

Figure B.1 depicts an encoder-decoder architecture, which provides a helpful basis for explaining key components like multi-head self-attention, feed-forward networks, residual connections, and positional encodings that make transformers effective. Understanding the core capabilities of encoder-only, decoder-only, and encoder-decoder architectures enables better comprehension of how specific models like ROBERTA and ChatGPT build off these foundations.

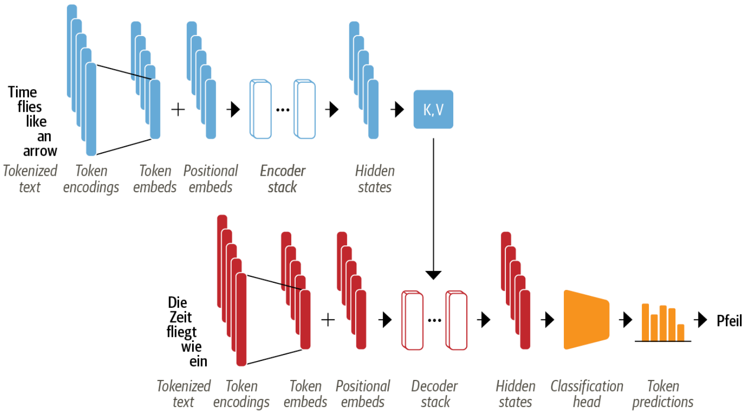

FIGURE B.1: Encoder-decoder transformer architecture with encoder on top and decoder on bottom. Inspired by "Attention is All you Need" paper with chart produced by NLP with transformers book.

B.3 Self-Attention

Self-attention is a core component of transformers, allowing the model to learn contextual relationships between all words in a sentence. As explained by Alammar (2019), self-attention computes interaction scores between each pair of words to determine their relatedness. For example, in the sentence "The animal didn't cross the street because it was too tired", self-attention can connect "it" back to "animal" based on their scored similarity.

Specifically, self-attention first projects the input word embeddings into Query, Key, and Value vectors using learned projection matrices, as illustrated in Figure B.2.

The Query and Key vectors are then dot-producted and scaled to compute an attention score reflecting word compatibility. These scores are normalized via softmax to create attention weights on the Values.

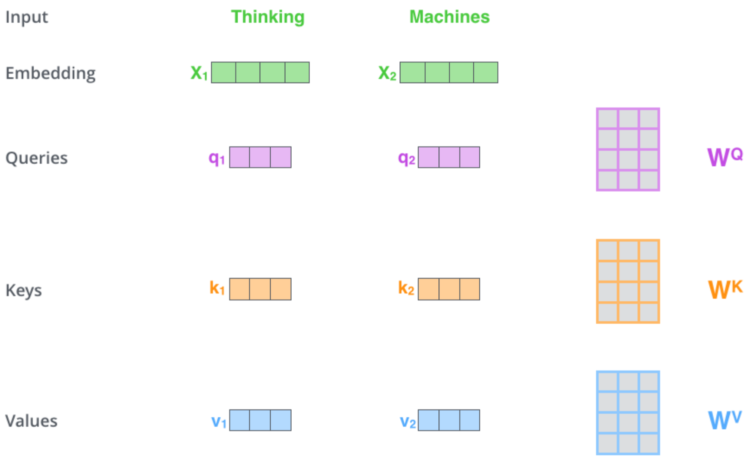

Figure B.2: Jay Alammar's illustration for the key, query, and value vectors are created.

Mathematically, as presented in Vaswani et al. Vaswani et al. (2017), the self-attention function is:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{B.1}$$

Where the d_k dimension scaling prevents dot product values from growing too large.

The weighted Value vectors are summed to produce the final contextualized representation for each word. This allows the model to build a contextual understanding of a word by attending to relevant words across the entire input sequence rather than only processing sequentially like RNNs.

B.4 Multi-head Attention

In addition to standard self-attention, transformers employ multi-head attention, which linearly projects each of the queries, keys, and values h times representing learned projected dimensions d_v, d_q and d_k. The multi-head attention representation allows the Transformer architecture to account for which position matters most.

Figure B.3 shows an example of multiple linear projections where each one is called attention head. The advantage of having numerous heads is that it can focus on several aspects simultaneously. For instance, if there were only one head, it would be isolated to find similar words. If there are multiple heads, one aspect can focus on semantic relationships, others on grammatical structure, another on local and global contexts, etc.

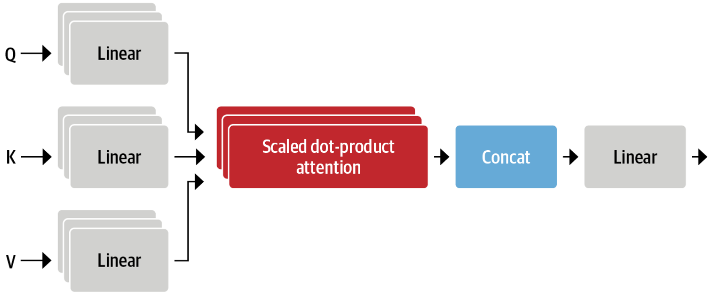

FIGURE B.3: Multi-head example from NLP with transformers book

B.4 shows a simple multi-head attention example where we have two sentences “Miners get new coins for processing transactions” and “Miners found new gold when they brought extra shovels” using a BERT model where sentences get separated by [SEP]. The first chart on the left shows all the weights between words where heavy-colored lines represent higher weights and lighter-colored ones have lower weights. The middle diagram shows clearly that the word “miners” is associated with coins and processing. In contrast, in the third chart on the right, the word “miners” is associated with gold and shovels. This demonstrates how multi-head attention can isolate contextual relationships within sentences.

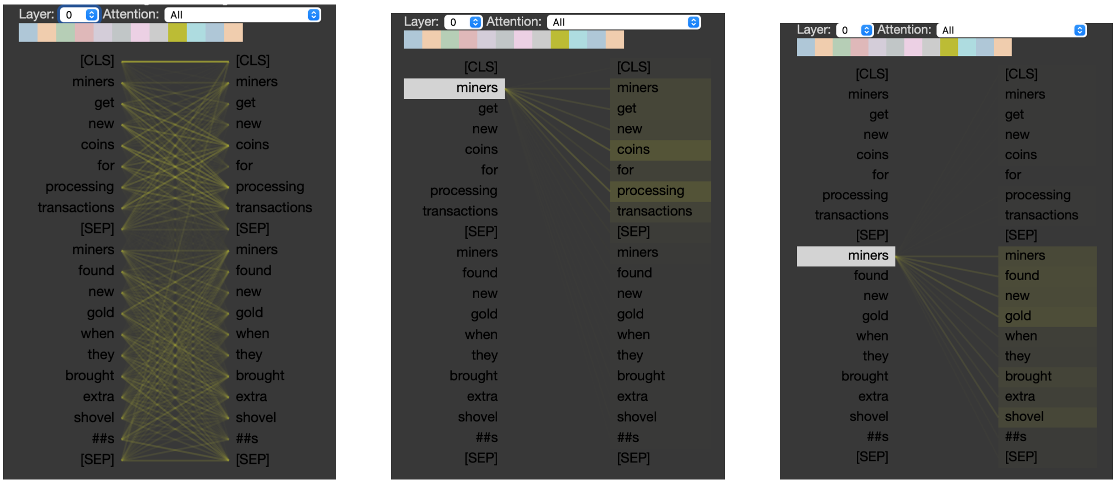

FIGURE B.4: Crypto "Mining" multi-head attention example using Bertviz package

By leveraging multiple heads, transformers can simultaneously consider different types of relevance and context when constructing representations of words and sentences.

B.5 Feed-Forward Neural Network

After self-attention processing, transformers employ a feed-forward neural network to introduce non-linearity and compute higher-level semantic relationships from the attention-encoded representations.

The feed-forward network is a primary type of neural network architecture where data can flow from input to output through hidden layers. This feed-forward network processes each input embedding separately, applying non-linear transformations using an activation function. Typical, the activation that gets used is GELU (Gaussian Error Linear Unit), which induces slight curvature compared to the sharp cut-off of ReLUs, as shown in Figure B.5.

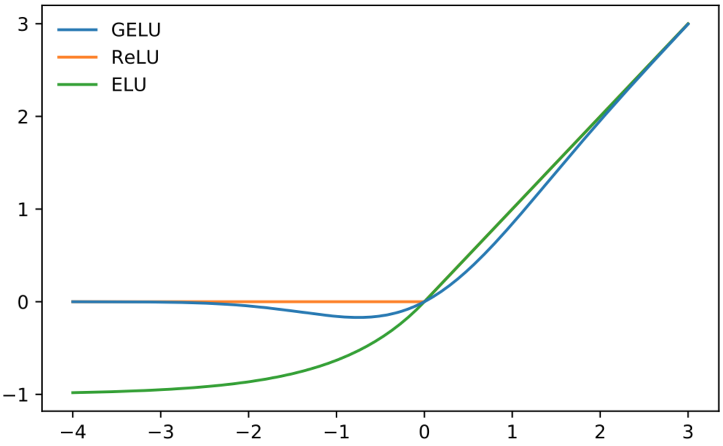

Figure B.5: GELU example from Gaussian Error Linear Units (GELUS) paper by Hendrycks and Gimpel

The authors define GELU as a cumulative distribution function from a Guassian with an error function:

$$GELU(X) = xP(X < x) = x\phi(x) = \frac{1}{2}[1 + erf(\frac{x}{\sqrt{2}})] \tag{B.2}$$

And GELU can be approximated as:

$$0.5x(1 + tanh[\sqrt{\frac{2}{\pi}}(x + 0.044715x^3)] \tag{B.3}$$

This feed-forward layer enriches the self-attention outputs, allowing the model to incorporate non-linear interactions and more complex semantics beyond attention-based processing. The feed-forward network provides vital depth and non-linearity to the transformer model. Combined with the self-attention outputs, it enables representing and manipulating linguistic meaning at higher levels of abstraction.

B.6 Encoder-decoder stacks

Figure B.6 brings multi-head attention and feed-forward neural networks together, illustrating an encoder stack. Each layer stacks independently, developing levels of abstraction representing different aspects of meaning. The decoder stack has a similar stack representation.

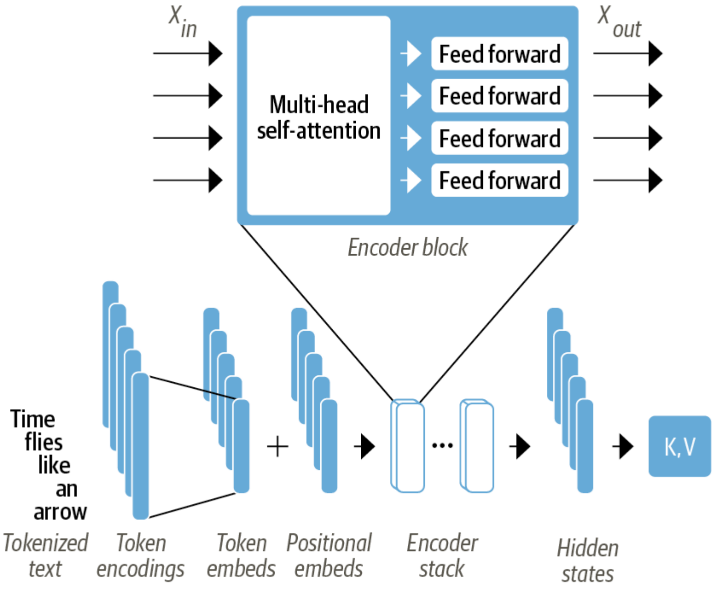

FIGURE B.6: Encoder zoom from NLP with transformers book

A less technical explanation by Tunstall et al. (2022) states that Attention instead of producing a single hidden state for the input sequence, the encoder outputs a hidden state at each step the decoder can access. The attention mechanism prioritizes which state to use through its calculation of weights. During each decoder time step, the decoder computes a weight for each encoder state. This allows the attention mechanisms to assign the context of a sentence by learning which parts of the input tokens matter most for predicting the next word in a language model.

B.7 Residual Connections and Layer Normalization

Two of the main features embedded within the Transformer architecture are residual connections (also known as skip connections) and layer normalization. The later process of each input is normalizing with zero mean and unity variance. Layer normalization is also essential to ensure the input data is scaled appropriately and that the model can learn comparable weights during the training. With that, layer normalization computes each mean and variance concerning each input feature. Residual connections help alleviate the vanishing gradient problem (where gradients can collapse to zero) by allowing shortcuts for information to flow between layers.

B.8 Positional Embeddings

In general, the Transformer model is autoregressive as seen in the original "Attention is All you Need" figure B.7 but differs from traditional recurrent neural network approaches in that it is less dependent on the previous time step of the input sequence. Said differently, Transformers allow modeling dependencies without regard to their input or output sequences as noted in Vaswani et al. (2017).

In the previous sections, we can almost characterize the transformer framework as a combination of intricate weighted combinations where the token position is not considered. Positional embeddings are a valuable trick for transformers to track token positions. The essential idea behind introducing this to the framework is augmenting the embedding by a position-based vector.

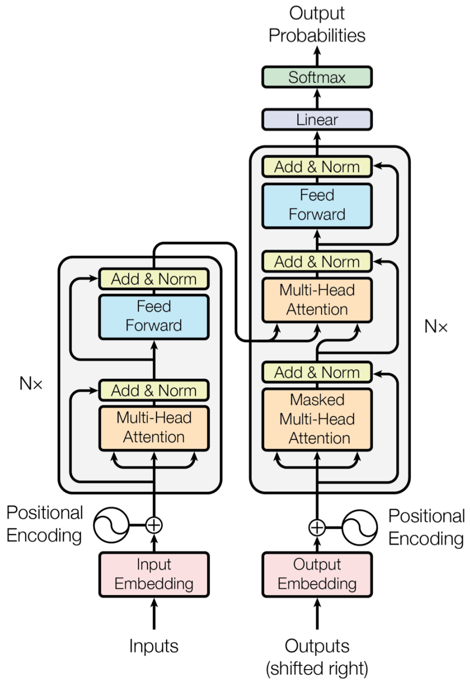

FIGURE B.7: The Transformer - model architecture from "All you need is Attention" paper

There are two options for how position embeddings can be defined per Tunstall et al. (2022). The first is Absolute positional embeddings, which are pre-defined and fixed, capturing global information about their positions' absolute value. Relative positional embeddings are computing by accounting for the surrounding tokens, with these embeddings tending to be more dynamic. One key difference between the two approaches is that absolute is more global, whereas relative is more locally based. Note that we did try experimenting with selecting absolute versus relative positional embeddings for our Roberta model, but it did not have a material impact on accuracy performance.

B.9 Summary

This chapter provides a comprehensive overview of transformer architectures and the key components that enable their effectiveness for natural language processing tasks. Transformers can be categorized into encoder-only, decoder-only, or encoder-decoder structures. Core elements like multi-head self-attention allow modeling contextual relationships between all words. The feed-forward network adds non-linearity and higher-order semantic modeling. Stacking attention and feed-forward blocks enable richer representations. Residual connections and layer normalization facilitate training deep models, and positional embeddings help track token order.

For the thesis in Part 2, a ROBERTA encoder model is used for text classification, leveraging bidirectional context encoding. We use ChatGPT to augment training data with its text-generation capabilities as a decoder-only transformer. As described in Part 1, we used Claude2, a decoder-only transformer, for text generation, introducing our Summarize and Sentimentize methodology.

Overall, transformers have become dominant in NLP by avoiding traditional deep learning architectures for more flexible attention-based processing. This provides superior handling of linguistic structure, semantics, and long-range dependencies in text. The components described equip transformers with the reasoning capacity to achieve state-of-the-art results across diverse NLP tasks. Their flexible architectures continue to enable breakthroughs in language understanding.

In summary, this chapter provides a technical deep dive into transformers, laying the groundwork for leveraging their capabilities in later chapters through models

like ROBERTA and ChatGPT. It reviews the architectures, attention mechanisms, and other elements that enable transformers with remarkable language abilities.

Bibliography

Agaian, S. and Kolm, P. (2017). Financial sentiment analysis using machine learning techniques. *International Journal Investment Management Financial Innovation*, 3:1–9.

Alammar, J. (2019). The illustrated transformer. http://jalammar.github.io/illustrated-transformer/.

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., Joseph, N., Kadavath, S., Kernion, J., Conerly, T., El-Showk, S., Elhage, N., Hatfield-Dodds, Z., Hernandez, D., Hume, T., Johnston, S., Kravec, S., Lovitt, L., Nanda, N., Olsson, C., Amodei, D., Brown, T., Clark, J., McCandlish, S., Olah, C., Mann, B., and Kaplan, J. (2022). A general language assistant as a laboratory for alignment. *Anthropic.*

Blei, D., Ng, A., and Jordan, M. (2012). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Bowman, S., Angeli, G., Potts, C., and Manning, C. (2015). A large annotated corpus for learning natural language inference. *In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics.*

Brown, N. C., Crowley, R. M., and Elliott, W. B. (2017). What are you saying? using topic to detect financial misreporting. *Journal of Accounting Research.*

Brown, S. and Tucker, J. W. (2011). Large-sample evidence on firms' year-over-year mda modifications. *Journal of Accounting Research*, 49:309–46.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems.*

Cao, S., Jiang, W., Yang, B., and Zhang, A. L. (2022). How to talk when a machine is listening?: Corporate disclosure in the age of ai. *Working paper at NATIONAL BUREAU OF ECONOMIC RESEARCH.*

Chalkiadakis, I., Zaremba, A., Peters, G. W., and Chantler, M. J. (2022). On-chain analytics for sentiment-driven statistical causality in cryptocurrencies. *Blockchain: Research and Applications.*

Cohen, L., Malloy, C., and Nguyen, Q. (2020). Lazy prices. *The Journal of Finance*, vol 75(3):1371–1415.

Cristescu, M. P., Maniu, I., Mara, D. A., Neris, R. A., and Culda, L. C. (2023). Analyzing the impact of financial news sentiments on stock prices—a wavelet correlation. *Mathematics*, 11(23):4830. https://doi.org/10.3390/math11234830.

Dai, H., Liu, Z., Liao, W., and et al. (2023). Auggpt: Leveraging chatgpt for text data augmentation. *arXiv preprint.*

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv - Computation and Language.*

Dyer, T., Lang, M., and Stice-Lawrence, L. (2017). The evolution of 10-k textual disclosure: Evidence from latent dirichlet allocation. *Journal of Accounting and Economics.*

Farmer, J. D. (2002). Market force, ecology and evolution. *Industrial and Corporate Change*, 11(5):895–953.

Friedman, J. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, pages 1189–1232.

Goldberg, Y. (2016). A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research.*

Goldbloom, A. (2016). What algorithms are most successful on kaggle. https://www.kaggle.com/code/antgoldbloom/what-algorithms-are-most-successful-on-kaggle.

Graves, A. (2011). Practical variational inference for neural networks. *NIPS.*

Hanley, K. W. and G., H. (2010). The information content of ipo prospectuses. *Review of Financial Studies*, 23:2821–64.

He, R., Sun Lee, W., Ng, H., , and Dahlmeier, D. (2017). An unsupervised neural attention model for aspect extraction. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 1.

Herremans, D. and Lowa, K. W. (2022). Forecasting bitcoin volatility spikes from whale transactions and cryptoquant data using synthesizer transformer models. Unpublished on arxiv.

Hoffmann, J., Borgeaud, S., Mensch, A., and et al. (2023). Training compute-optimal large language models. *arXiv preprint.*

Howard, J. and Ruder, S. (2018). Ulmfit: Universal language model fine-tuning for text classification. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).*

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *NEURIPS.*

Kim, G., Shin, D.-H., Choi, J. G., and Lim, S. (2022). A deep learning-based cryptocurrency price prediction model that uses on-chain data. *IEEE Access.*

Li, F. (2008). Annual report readability, current earnings, and earnings persistence. *Journal of Accounting and Economics*, 45:221–247.

Liu, B. (2012). Sentiment analysis and opinion mining. *Lectures on Human Language Technologies*, pages 1–167. https://doi.org/10.2200/s00416ed1v01y201204hlt016.

Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., and Liang, P. (2024). Lost in the middle: How language models use long contexts. *Working paper from Stanford.*

Loughran, T. and McDonald, B. (2011a). Textual analysis in accounting and finance: A survey. *The Journal of Accounting Research.*

Loughran, T. and McDonald, B. (2011b). When is a liability not a liability? textual analysis, dic- tionaries, and 10-ks. *Journal of Finance*, 66:35–65.

Merity, S. (2016). Keras snli baseline example.

Mikolov, T., Yih, W., and Zweig, G. (2013). Linguistic regularities in continuous space word representations. *In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.*

Mimno, D., Wallach, H., Talley, E., Leenders, M., and McCallum, A. (2011). Optimizing semantic coherence in topic models. *In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing.*

Nakamoto, S. (2009). Bitcoin: A peer-to-peer electronic cash system. www.bitcoin.org.

Narayanan, A., Bonneau, J., Felten, E., Miller, A., and Goldfeder, S. (2016). Bitcoin and cryptocurrency technologies, preface — the long road to bitcoin. *Princeton University Press*, pages 3–21.

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., and Auli, M. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv - Computation and Language.*

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., and et al. (2022). Training language models to follow instructions with human feedback. *OpenAI.*

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. *Empirical Methods in Natural Language Processing (EMNLP).*

Phillips, R. (2019). The predictive power of social media within cryptocurrency markets. PhD thesis at University College London.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training. *Open AI Blog.*

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2019). Better language models and their implications. *Open AI Blog.*

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv.*

Schulman, J., Zoph, B., Kim, C., Hilton, J., Menick, J., Weng, J., Ceron Uribe, J. F., and et al. (2023). Introducing chatgpt. *Open AI Blog.*

Sermanet, P. and LeCun, Y. (2011). Traffic sign recognition with multi-scale convolutional networks. *In Proceedings of the International Joint Conference on Neural Networks.*

Tunstall, L., von Werra, L., and Wolf, T. (2022). Natural language processing with transformers. *O'Reilly.*

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, , and Polosukhin, I. (2017). Attention is all you need. *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems.*

Wang, X. and Luo, J. (2021). Predicting gme stock price movement using sentiment from reddit r/wallstreetbets. *Proceedings of the Third Workshop on Financial Technology and Natural Language Processing.*

Wu, D., Wang, X., Su, J., Tang, B., and Wu, S. (2020). A labeling method for financial time series prediction based on trends. *Entropy (Basel)*, 10(22).

Yaga, D., Mell, P., Roby, N., and Scarfone, K. (2018). Blockchain technology overview. *National Institute of Standards and Technology*, pages 9–23.

Zhang, H. (2004). The optimality of naive bayes. *Proc. FLAIRS*.

Zhang, L., Wang, S., and Liu, B. (2018a). Deep learning for sentiment analysis : A survey. *arxiv*.

Zhang, L., Wang, S., and Liu, B. (2018b). Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews-Data Mining and Knowledge Discovery*.